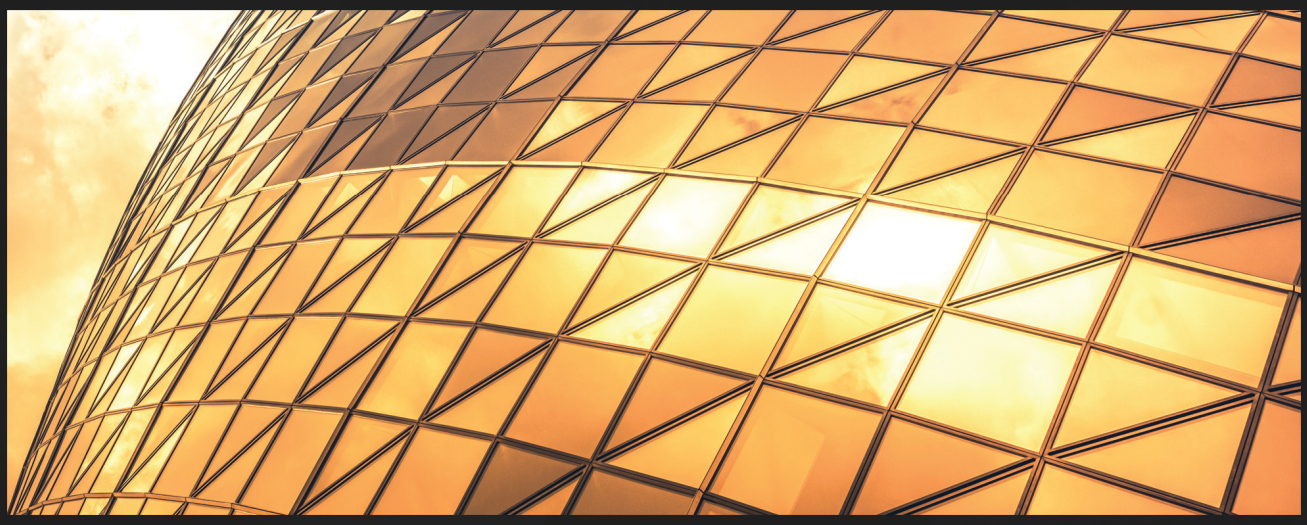


3RD EDITION

Digital Forensics with Kali Linux

Enhance your investigation skills by performing network and memory forensics with Kali Linux 2022.x

SHIVA V. N. PARASRAM

Digital Forensics with Kali Linux

Enhance your investigation skills by performing network and memory forensics with Kali Linux 2022.x

Shiva V. N. Parasram

BIRMINGHAM—MUMBAI

Digital Forensics with Kali Linux

Copyright © 2023 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Group Product Manager: Pavan Ramchandani
Publishing Product Manager: Prachi Sawant
Senior Content Development Editor: Adrija Mitra
Technical Editor: Arjun Varma
Copy Editor: Safis Editing
Project Coordinator: Sean Lobo
Proofreader: Safis Editing
Indexer: Manju Arasan
Production Designer: Shankar Kalbhor
Marketing Coordinator: Marylou De Mello

First published: December 2017
Second edition: April 2020
Third edition: April 2023

Production reference: 1160323

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-83763-515-3

www.packtpub.com

I dedicate this book to my father, Harry Goolcharran Parasram (1950–2021), an author, teacher, poet, artist, the most brilliant man I've ever known, and the most loving father a son could hope and pray for. The man who taught me the importance of being patient and kind and knowing when to take risks. The one who got me started with computers and technology. The man who taught me to care for my family and be a strong, intelligent, and loving man. Not a day goes by that I don't think of you. You're missed every day. Thank you, daddy. Love you endlessly.

Contributors

About the author

Shiva V. N. Parasram is a cybersecurity and risk consultant with over 19 years of experience and is the executive director of the **Computer Forensics and Security Institute** (**CFSI**), which specializes in pentesting, **Digital Forensics and Incident Response** (**DFIR**), and advanced security training with a global reach. As the only **Certified EC-Council Instructor** (**CEI**) in the Caribbean, he has trained thousands and is the founder of the CFSI CyberFence program. Shiva is also the author of three other books from Packt Publishing and has delivered workshops regionally and globally for ISACA, ISC2, universities, and security agencies. He is also a Security Risk Manager Consultant for PTRMS (Canada) positioned within a global financial institution, and a cybersecurity mentor at Springboard (US).

I'd like to thank the team at Packt (Shrilekha, Sean, Adrija, and Prachi) for their support; the technical reviewers, Alex Samm and Deodath Ganga; my guru, Pt. Persad; my parents, Harry and Indra; my wife, Savi; the loveable Bindi; and Dr. Mala, Dr. Nilash Ramnarine, and Dr. Sharad Mohip. I also have to thank all my friends who were there for me throughout my most trying times recently. Special thanks to the CFSI family also. I am truly blessed.

About the reviewers

Alex Samm has worked in the cybersecurity space for over 10 years, primarily focused on penetration testing and red teaming. He has conducted penetration tests for organizations in the financial sector, education, public utilities, oil and gas, and state entities. He has also executed incident response and digital forensics for financial institutions and other state entities.

Alex is currently employed at BDO B.V. as a consultant in their advisory services team and provides services that include penetration testing, ERP assessments, data analytics, IT risk assessments, and other digital services.

I'd like to thank my family for all the support they provide. They have encouraged my obsession with technology and driven me to learn more. Huge thanks to my friends that keep me grounded and remind me to take time to relax.

Deodath Ganga is an information security and networking professional with over 20 years' experience in information technology, networking, and cybersecurity. He is a senior security advisor and consultant who is positioned as an information security technology risk manager for a client in the global banking sector. He is also an experienced penetration tester, digital forensic investigator, and purple teamer, as well as a senior cybersecurity lecturer who teaches ethical hacking, digital forensic investigation, and cyber defense. Deodath is passionate about cyber safety and works as a senior cybersecurity awareness officer, educating people about the dangers of the cyber realm and ways to keep themselves safe.

Table of Contents

3

Installing Kali Linux 45

4

Additional Kali Installations and Post-Installation Tasks 81

5

Installing Wine in Kali Linux 99

Part 2: Digital Forensics and Incident Response Fundamentals and Best Practices

6

7

Part 3: Kali Linux Digital Forensics and Incident Response Tools

8

9

10

11

Part 4: Automated Digital Forensics and Incident Response Suites

12

13

Part 5: Network Forensic Analysis Tools

14

15

16

Preface

In this third edition of this book, you'll find that the theory and methodologies have remained mostly the same with updates on general technical information, best practices, and frameworks, as the procedures and documentation are standard throughout the field; however, you'll find that the technical chapters contain new labs using new examples. I've also included a few completely new chapters that go deeper into artifact analysis, automated data recovery, malware, and network analysis, showcasing several tools with practical exercises that even beginners will find easy to follow. We even utilize Wine, which will allow us to install very popular (**Digital Forensics and Incident Response**) (**DFIR**) tools built for the Windows platform (such as Autopsy 4) within Kali Linux. This book is quite useful for Red Teamers and penetration testers who wish to learn about or enhance their DFIR and Blue Teaming skillsets to become Purple Teamers by combining their penetration testing skills with the digital forensics and incident response skills that will be taught throughout this book.

Who this book is for

The third edition of this book was carefully structured to be easily understood by individuals at all levels, from beginners and digital forensics novices to incident response professionals alike, as the first six chapters serve to get you acquainted with the technologies used and also guide you through setting up Kali Linux, before delving into forensic analysis, data recovery, malware analysis, automated DFIR analysis, and network forensics investigations. Red teamers and penetration testers wanting to learn Blue Teaming skillsets to become Purple Teamers may also find the contents of this book very useful.

What this book covers

Chapter 1, *Red, Blue, and Purple Teaming Fundamentals*, informs you about the different types of cyber security teams to which penetration testers and forensic investigators belong, and the skillsets required.

Chapter 2, *Introduction to Digital Forensics*, introduces you to the world of digital forensics and forensic methodology, and also introduces you to various forensic operating systems.

Chapter 3, *Installing Kali Linux*, covers the various methods that can be used to install Kali Linux as a virtual machine or as a standalone operating system, which can also be run from a flash drive or SD card.

Chapter 4, *Additional Kali Installations and Post-Installation Tasks*, builds upon the Kali installation and guides you through performing additional installations and post-installation tasks such as enabling a root user and updating Kali Linux.

Chapter 5, *Installing Wine in Kali Linux*, shows the versatility of Linux systems, where you will learn how to install and use forensic tools designed to be used in the Windows platform, in a Kali Linux system using Wine.

Chapter 6, *Understanding File Systems and Storage Media*, dives into the realm of operating systems and the various formats for file storage, including secret hiding places not seen by the end user, or even the operating system. We also inspect data about data, known as metadata, and look at its volatility.

Chapter 7, *Incident Response, Data Acquisitions, and DFIR Frameworks*, asks what happens when an incident is reported or detected. Who are the first responders and what are the procedures for maintaining the integrity of the evidence? In this chapter, we look at best practices, procedures, and frameworks for data acquisition and evidence collection.

Chapter 8, *Evidence Acquisition Tools*, builds on the theory behind data acquisitions and best practices and teaches you to use industry-recognized tools such as DC3DD, DD, Guymager, FTK Imager, and RAM Capturer to perform data and image acquisition while preserving evidence integrity.

Chapter 9, *File Recovery and Data Carving Tools*, introduces the investigative side of digital forensics by using various tools such as Magic Rescue, Scalpel, `Bulk_Extractor`, `scrounge_ntfs`, and `recoverjpeg` to carve and recover data and artifacts from forensically acquired images and media.

Chapter 10, *Memory Forensics and Analysis with Volatility 3*, takes us into the analysis of memory artifacts and demonstrates the importance of preserving volatile evidence such as the contents of the RAM and the paging file.

Chapter 11, *Artifact, Malware, and Ransomware Analysis*, carries us much deeper into artifact analysis using p0f, swap_digger, and mimipenguin, and, thereafter, demonstrates how to perform malware and ransomware analysis using pdf-parser, `hybrid-analysis.com`, and Volatility.

Chapter 12, *Autopsy Forensic Browser*, showcases automated file recovery and analysis within Kali Linux using a single tool.

Chapter 13, *Performing a Full DFIR Analysis with the Autopsy 4 GUI*, dives much deeper into automated file carving, data recovery, and analysis using one of the most powerful and free forensic tools, which takes forensic abilities and investigations to a professional level, catering for all aspects of full digital forensics investigations, from hashing to reporting.

Chapter 14, *Network Discovery Tools*, showcases network scanning and reconnaissance tools such as netdiscover, nmap, and Shodan, which, although not specifically designed for use as forensic tools, are useful in providing additional information when performing incident response.

Chapter 15, *Packet Capture Analysis with Xplico*, gives an insightful use of automated packet analysis using one tool for investigating network and internet traffic.

Chapter 16, *Network Forensic Analysis Tools*, ends the book by demonstrating how to capture and analyze packets using a variety of tools and websites including Wireshark, NetworkMiner, `packettotal.com`, and `apackets.com`.

To get the most out of this book

Although we have tried our best to explain all concepts and technologies in this book, it may be beneficial if you have prior knowledge of downloading and installing software and are at least familiar with basic computer and networking concepts such as RAM, CPU, virtualization, and network ports.

Software/hardware covered in the book	Operating system requirements
Kali 2022.x and later	Minimum specs: A PC or laptop with 8 GB RAM, 250 GB free hard drive space, and a Ryzen 7 or i5 CPU Recommended specs: 16 GB RAM, 250 GB free hard drive space, and a Ryzen 7 or i7 CPU

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at `https://github.com/PacktPublishing/Digital-Forensics-with-Kali-Linux-Third-Edition`. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `https://packt.link/vLuYi`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Power on your Pi and Kali will boot. Again, the default username and password are both `kali` (in lowercase)."

Any command-line input or output is written as follows:

```
sudo apt update
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "You can view some of the forensics tools by clicking on **Applications** | **11-Forensics** on the main Kali menu."

> **Tips or important notes**
> Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customercare@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Share Your Thoughts

Once you've read *Digital Forensics with Kali Linux*, we'd love to hear your thoughts! Please click here to go straight to the Amazon review page for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below

`https://packt.link/free-ebook/9781837635153`

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly

Part 1: Blue and Purple Teaming Fundamentals

As we begin our journey into **Digital Forensics and Incident Response** (**DFIR**), it is important that we have a clear understanding of Blue and Purple Teaming, which is compared to Red Teaming, and also have a firm grasp on fundamental knowledge required to create a Blue and Purple Teaming lab environment. This section explains the terminology and looks at the skillsets required in becoming a Blue and Purple Teamer, and also demonstrates various methods of setting up a DFIR lab environment.

This part has the following chapters:

- *Chapter 1*, *Red, Blue, and Purple Teaming Fundamentals*
- *Chapter 2*, *Introduction to Digital Forensics*
- *Chapter 3*, *Installing Kali Linux*
- *Chapter 4*, *Additional Kali Installations and Post-Installation Tasks*
- *Chapter 5*, *Installing Wine in Kali Linux*

1

Red, Blue, and Purple Teaming Fundamentals

Welcome to the third edition of *Digital Forensics with Kali Linux*, and for those of you who may have purchased the previous editions, welcome back. I'd also like to sincerely thank you for once again choosing this exciting title. As with the second edition, this third edition has been updated with new tools, easy-to-follow labs, and a couple of new chapters. We have an exciting journey ahead of us, and I'm pleased to announce the inclusion of some major additions, including the installation of Wine, which will allow us to run Windows tools within Kali Linux and will be covered in its entirety in *Chapter 5, Installing Wine in Kali Linux*. *Chapter 10*, *Memory Forensics and Analysis with Volatility 3*, is also brand-new and shows how to perform RAM artifact analysis on newer operating systems. Another new chapter on using the Autopsy v4 **Graphical User Interface** (**GUI**) to perform full **Digital Forensics and Incident Response** (**DFIR**) analysis and investigations can be found in *Chapter 13*, *Performing a Full DFIR Analysis with the Autopsy 4 GUI*.

Besides these major additions, we will also look at some new topics, such as creating a portable Kali Linux box using Raspberry Pi 4 and learning about tools such as DD-rescue, scrounge-ntfs, Magic Rescue, PDF-Parser, Timeliner, netdiscover, and introduce **Shodan.io** and **apackets.com** for **Internet of Things** (**IoT**) discovery and packet analysis.

For this book, we take a very structured approach to digital forensics, as we would in forensic science. First, we will stroll into the world of digital forensics, its history, and some of the tools and operating systems used for forensics, and we will immediately introduce you to the concepts involved in evidence preservation.

With that said, we have a lot to cover and will start by learning about Kali and the various cybersecurity teams and the differences between red, blue, and purple teaming. For our returning and advanced readers who may have prior knowledge of Kali Linux and the respective teams, feel free to skim through the first two chapters and get straight into the practical aspects in *Chapter 3*, *Installing Kali Linux*, *Chapter 4*, *Additional Kali Installations and Post-Installation Tasks*, and *Chapter 5*, *Installing Wine in Kali Linux*, which detail the installations of Kali and Wine.

In this chapter we will cover the following key topics:

- What is Kali Linux?
- Understanding red teaming
- Understanding blue teaming
- Understanding purple teaming

Before we get started with these topics, the following is a sneak peek at how I got into the world of Kali Linux, as I feel some of you will be able to relate to my story!

How I got started with Kali Linux

Digital forensics has had my attention for well over 15 years. Ever since I was given my first PC (thanks, Mom and Dad), I've always wondered what happened when I deleted my files from my massively large 2 GB (**Gigabyte**) hard drive or moved my files to (and often hid them on) a less-than-inconspicuous 3.5-inch floppy diskette that maxed out at 1.44 MB (**Megabytes**) in capacity.

I soon learned that hard and floppy disk drives did not possess the digital immortality I so confidently believed in. Sadly, many files, documents, and priceless fine art created in Microsoft Paint by yours truly were lost to the digital afterlife, never to be retrieved again. Sigh. The world shall never know.

It wasn't until years later that I came across an article on file recovery and associated tools while browsing the magical **World Wide Web** (**WWW**) on my lightning-fast 42 Kbps dial-up internet connection (made possible by my very expensive USRobotics dial-up modem), which sang the tune of the technology gods every time I tried to connect to the realm of the internet. This process involved a stealthy ninja-like skill that would make even a black-ops team envious, as it involved doing so without my parents noticing, as this would prevent them from using the telephone line to make or receive phone calls (apologies, dear Mother, Father, and older teenage sister).

The previous article on data recovery wasn't anywhere near as detailed and fact-filled as the many great peer-reviewed papers, journals, and books on digital forensics widely available today. As a total novice (also referred to as a noob) in the field, I did learn a great deal about the basics of file systems, data and metadata, storage measurements, and the workings of various storage media. It was at this time that, even though I had read about the Linux operating system and its various distributions (or distros), I began to get an understanding of why Linux distros were popular for data recovery and forensics.

I managed to bravely download the Auditor and Slax Linux distributions, again on a dial-up connection. Just downloading these operating systems was quite a feat, which left me feeling highly accomplished as I did not have any clue as to how to install them, let alone actually use them. In those days, easy installation and GUIs were still under heavy development, as user-friendly, or in my case, user-unfriendly, as they were at the time (mostly due to my inexperience, lack of recommended hardware, and also lack of resources, such as online forums, blogs, and YouTube, which I did not yet know about).

As time passed, I researched many tools found on various platforms for Windows, Macintosh, and many Linux distributions. I found that many of the tools used in digital forensics could be installed on various Linux distributions or flavors, and many of these tools were well maintained, constantly being developed, and widely accepted by peers in the field. Kali Linux is a Linux distribution or flavor, but before we go any further, let me explain the concept of a Linux distribution or flavor. Consider your favorite beverage: this beverage can come in many flavors, some without sweeteners or sugar, in different colors, and even in various sizes. No matter the variations, it's still the basic ingredients that comprise the beverage at the core. In this way, too, we have Linux and then different types and varieties of Linux. Some more popular Linux distros and flavors include RedHat, CentOS, Ubuntu, Mint, KNOPPIX, and, of course, Kali Linux. More on Kali Linux will be discussed in *Chapter 3*, *Installing Kali Linux*.

With that said, let's move on to our next section as we get started with exploring the enchanting world of Kali Linux!

What is Kali Linux?

Kali Linux is a Debian-based operating system used globally by cyber security professionals, students, and IT enthusiasts. Debian is a flavor of Linux that is completely free, stable, constantly updated, supports many types of hardware, and is also used by popular operating systems such as Ubuntu and Zorin. Kali Linux is certainly not new to the cybersecurity field and even goes back to the mid-2000s, but it was known then as BackTrack, which was a combination of two platforms called Auditor Security and Whax. This merge happened in 2006, with subsequent versions of BackTrack being released up to 2011 when BackTrack 5, based on Ubuntu 10.04, was released.

In 2013, *Offensive Security* released the first version of Kali v1 (Moto), which was based on Debian 7, and then Kali v2 in 2015, which was based on Debian 8. Following this, Kali Linux Rolling was released in 2016, with the names of the distribution reflecting both the year of release and the major update of the quarterly period. For example, at the time of writing, I use Kali `2022.3` and `2022.4`, both based on recent versions of Debian. You can find more on the open source and free Debian Project at `https://www.debian.org/intro/about`.

As a cybersecurity professional, a **Chief Information Security Officer** (**CISO**), **penetration tester** (**pentester**), and subject matter expert in DFIR, I have used BackTrack and now Kali Linux for well over a decade since I first came across it when I started studying for the Certified Ethical Hacker exam in 2006. Since then, I've used a myriad of operating systems for pentesting and digital forensics, but my main tool of choice, particularly for pentesting, is Kali Linux. Although Kali Linux has focused less on DFIR and more on penetration testing, it makes it much easier for me to have both penetration testing and DFIR tools on one platform rather than have to switch between them.

For our readers who may have purchased the first and second editions of this book, I'd say you're certainly in for a treat as I've not only updated many labs and introduced new tools in this edition, but I've also included a chapter on installing Wine in Kali Linux. **Windows Emulator** (**Wine**) allows

you to run Windows applications in Kali Linux. Although it takes a bit of configuration, I've compiled a step-by-step guide on how to install Wine in *Chapter 5*, *Installing Wine in Kali Linux*.

Some of you may be wondering why we would install Wine instead of simply using a Windows machine. There are quite a few valid reasons actually. Firstly, cost is a major factor. Windows licenses aren't cheap if you're a student, in between jobs, changing careers, or live in a region where the exchange rate and forex are limiting factors in purchasing licensing. At the time of writing, the cost of a Windows 10 Professional license is $199.00, as listed on Microsoft's site at `https://www.microsoft.com/en-us/d/windows-10-pro/df77x4d43rkt?activetab=pivot:overviewtab`.

Although we will not be using commercial tools in this book, there are some amazing free DFIR tools that are available for Windows, such as **Belkasoft RAM Capturer**, **Autopsy 4 GUI**, and **NetworkMiner**, which we can now install within our open source Kali Linux environment instead of on a licensed Windows machine. These tools will be covered in detail in *Chapter 8*, *Evidence Acquisition Tools*, *Chapter 13*, *Performing a Full DFIR Analysis with the Autopsy 4 GUI*, and *Chapter 16*, *Network Forensic Analysis Tools*, respectively.

Another consideration is that Wine again saves us the hassle of having to switch between physical machines and can also save on resource utilization such as **Random Access Memory** (**RAM**), **Central Processing Unit** (**CPU**), **Hard Disk Drive** (**HDD**) space, and other resources when using virtual machines, which we will discuss more in detail in the next chapter.

Finally, we can install many other Windows applications in Kali Linux using tools, whether they be productivity tools or even tools for penetration testing, thus making our Kali Linux installation the perfect purple teaming operating system environment, which we will discuss later in this chapter.

Why is Kali Linux so popular?

Aside from being one of the oldest, InfoSec distros (distributions), Kali Linux has a very large support base, and you can find thousands of tutorials on installation, using built-in tools, and installing additional tools on YouTube, TikTok, and the internet at large, making it one of the more user-friendly platforms.

Kali Linux also comes with over 600 tools, all of which are nicely categorized in Kali's **Applications** menu. Many of the tools included in Kali can perform various cybersecurity tasks ranging from **Open Source Intelligence** (**OSINT**), scanning, vulnerability assessments, exploitation and penetration testing, office and productivity tools, and, of course, DFIR. The full listing of tools can be found at `https://www.kali.org/tools/all-tools/`.

The following screenshot gives a preview of the category listings in the Kali Linux menu.

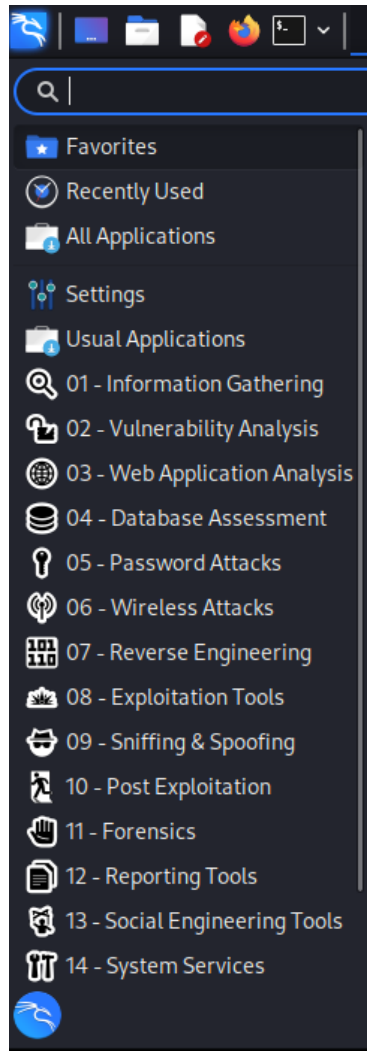


Figure 1.1 – Category listing in the Kali Linux menu

Kali Linux users also have the option to download and install (meta)packages manually rather than downloading a very large installation file. Kali Linux (meta)packages contain tools and dependencies that may be specific to an assessment or task, such as information gathering, vulnerability assessments, wireless hacking, and forensics. Alternatively, a user can download the **kali-linux-everything (meta) package**. We'll go into more detail about (meta)package installations in *Chapter 4*, *Additional Kali Installations and Post-Installation Tasks*, but if you'd like to know more about what (meta)packages exist, you can find the full listing at `https://www.kali.org/docs/general-use/metapackages/`.

Yet another reason why Kali Linux is so popular is that there are several versions available for a multitude of physical, virtual, mobile, and portable devices. Kali is available as a standalone operating system image and can also be installed virtually using their pre-built images for virtual platforms such as VMware and VirtualBox, which will be covered in detail in *Chapter 3*, *Installing Kali Linux*, and *Chapter 4*, *Additional Kali Installations and Post-Installation Tasks*. There are also versions of Kali for ARM devices, cloud instances, and even the ability to run Kali Linux in Windows 10 under the **Windows Subsystem for Linux** (**WSL**). On a personal note, I also use the mobile version of Kali Linux called Kali NetHunter on an old OnePlus phone and also on a Raspberry Pi 4, which, when connected to a power bank, serve as the ultimate portable security assessment toolkit. As far as installation on mobile phones goes, NetHunter (and even Kali Linux itself in some cases) can be installed on a variety of phones from Samsung, Nokia, OnePlus, Sony, Xiaomi, Google, or ZTE. We'll look at installing Kali Linux in VirtualBox and Raspberry Pi 4 in *Chapter 4*, *Additional Kali Installations and Post-Installation Tasks.*

The fact that Kali Linux offers all these features for free and can be easily upgraded with the addition of new tools just a couple of clicks and commands away makes it the perfect purple teaming solution. Let's take a look at red, blue, and purple teaming and the skillsets required for each team.

Understanding red teaming

Possibly the most commonly known team among users of Kali Linux, the red team is the name given to the collective of individuals responsible for handling the offensive side of security as it relates to OSINT, scanning, vulnerability assessments, and the penetration testing of resources, including but not limited to individuals, companies, host end users (desktops, laptops, mobiles), and network and critical infrastructure such as servers, routers, switches, firewalls, NAS, databases, WebApps, and portals. There are also systems such as IoT, **Operational Technology** (**OT**) devices, and **Industrial Control Systems** (**ICS**), which also require assessments by highly skilled red teamers.

Red teamers are generally thought of as highly skilled ethical hackers and penetration testers who, apart from having the skill sets to conduct the assessments listed previously, may also have the technical certifications that allow them to do so. Although certifications may not directly reflect the abilities of the individuals, they have been known to aid in obtaining jobs.

Some red teaming certifications include (but are not limited to):

- **Offensive Security Certified Professional** (**OSCP**): Developed by the creators of Kali Linux
- **Certified Ethical Hacker** (**CEH**): From the *EC-Council*
- **Practical Network Penetration Tester** (**PNPT**): Developed by TCM Security
- **Pentest+**: By CompTIA
- **SANS SEC**: Courses from the SANS Institute
- **e-Learn Junior Penetration Tester** (**eJPT**): Developed by *e-Learn Security* for beginners interested in becoming red teamers

Ultimately, all of this knowledge allows red teamers to conduct offensive attacks (with explicit permission) against companies to simulate internal and external threat actors and essentially hack systems and security mechanisms in the same manner in which malicious actors would compromise and exploit the attack surface of an individual, company, or valued asset.

Kali Linux generally contains all the tools required to perform almost all types of offensive security and red teaming assessments. On a personal note, Kali Linux is my go-to operating system of choice for penetration testing as most of the tools required for fingerprinting, reconnaissance, OSINT, vulnerability assessments, exploitation, and reporting are all readily available and preinstalled on the platform. I've been using Kali to conduct red team exercises for over 12 years and I don't see that changing anytime soon, as they've always maintained the OS and support for tools over the years.

Let's move on to blue teaming now.

Understanding blue teaming

Blue teamers are generally considered to be on the defensive side rather than the offensive, as previously written about red teamers. While red teamers focus on threat simulation and possible exploitation, blue teamers are the protectors of the realm.

Red and blue teamers are quite similar when considering that the main goal of each team is mainly to protect resources and understand the potential impact and risk associated with breaches and data leaks. The red team may focus on attack techniques, such as the cyber kill chain and penetration testing, whereas the blue team then focuses on ensuring that not only are mechanisms in place to protect against attacks but also that formal policies, procedures, and even frameworks are implemented to assure effective DFIR.

The work of a blue teamer covers far more than that of a red teamer, as blue teamers must analyze threats, understand their risk and impact, implement security and protective measures, understand forensics and incident response, and ensure that effective monitoring, response services, and measures are implemented. It also certainly helps if a blue teamer has the knowledge or experience of a red teamer, as this provides an additional depth of understanding of attack surfaces and threat landscapes.

Blue teamers must also be knowledgeable about a wide scope of technology and analytics. While it is not impossible for people new to IT to get into blue teaming and DFIR, it does require prior knowledge along the lines of a network and systems administrator and also of a security analyst and threat hunter. For example, understanding that systems must be updated and patched accordingly is more of a best practice. The blue teamer will understand why there is a need for patching and also understand that there is much more to be done when hardening devices to reduce attack surfaces while also taking into consideration the possibilities of zero-day exploits and even human weaknesses, which may easily facilitate a breach by a threat actor and then circumvent all technical measures implemented.

It is also not uncommon to see job posts asking that blue teamers be proficient in **Security Information and Event Management** (**SIEM**) tools, which provide real-time analysis, monitoring, and alerts that greatly aid in DFIR management and allow for a greater understanding of the level of protection required in maintaining a high-security posture rating when safeguarding data, systems, and assets.

Blue teamers must also accept that their responsibilities do not only apply to internal and external resources but will be extended when considering the threat landscape of the assets to be protected. The threat landscape can be devices, persons, data, and any information that may be useful to an attacker when planning an attack. This is where an in-depth understanding of OSINT comes in. Although previously mentioned as a red teaming skill set, this proves equally important to the blue teamer in being able to scout the internet, social media, and the dark web for any information that could either pose a threat or aid the threat actor in some way.

A good example would be to search the dark web for breach databases where the blue teamer (after taking all necessary precautions to protect themselves) browses the dark web in search of compromised emails or **Virtual Private Network** (**VPN**) credentials of the company they work for. The blue teamer may also use a site such as Shodan.io, which we will cover later on in this book, to find accessible devices from an external perspective, such as external access to firewalls, servers, and CCTV cameras. All of the preceding scenarios aid the blue teamer in developing what is known as a threat profile, which, while not directly focusing on internal and external assets, will still compile potential threats and even **Indicators of Compromise** (**IoC**) found externally.

A great free resource for learning OSINT is TCM Academy's free 4-hour course on YouTube, which can be found here `https://www.youtube.com/watch?v=qwA6MmbeGNo`.

Although many of the previously mentioned skills are learned via research and countless hours digging, looking at YouTube videos, and attending specialized courses. I've listed just a few certifications that may assist in furthering your studies and career in blue teaming and DFIR.

Some blue teaming certifications include (but are not limited to):

- **Computer Hacking Forensic Investigator** (**CHFI**) from EC-Council
- **Certified Cloud Security Engineer** (**CCSE**) from EC-Council
- **Certified Forensic Computer Examiner** (**CFEC**) from IACIS
- **GIAC Certified Forensics Examiner** (**GFCE**) from SANS

We will look at the tools required to be a DFIR investigator and analyst in more detail throughout this book. Although we won't be going into detail about commercial tools used, I will mention some that you may wish to look into at some point if heading into a career in DFIR or as a blue teamer, although the open source tools covered in this book are more than enough to get you started and conduct entire DFIR investigations as long as the best practices and procedures are followed.

It is also of paramount importance that DFIR investigators and analysts understand the importance of following best practices and procedures in evidence collection, acquisition, analysis, and documentation, as the integrity of the evidence and case could be easily compromised. Analysis of evidence and results in reports should also be repeatable, meaning that other DFIR investigators and analysts should be able to repeat the tests performed and produce the same results as you.

In this regard, blue teamers should have a detailed and well-documented plan of action along with knowledge of purpose-specific tools. There are many freely available and well-documented best practices and frameworks for blue teams, some of which we'll look at in the next chapter.

Let's briefly look at an overview of the tools you may be required to use in a DFIR investigation, which are all covered in this book. The following list gives a one-liner for a specific task and the tools used to achieve the task. Think of this as a blue team cheat sheet where open source tools are concerned. Feel free to also make a copy of this page to use as a reference sheet for your forensics and incident response fieldwork:

- Forensic operating systems for DFIR – our customized version of Kali Linux, CSI Linux, and CAINE
- Creating a live bootable USB with Kali Linux – Rufus and Etcher
- Creating a portable version of Kali Linux for Raspberry Pi – Imager (Pi Imager)
- Installing Windows tools in Kali – Wine
- Memory acquisition – FTK Imager and Belkasoft RAM Capturer
- Evidence and drive acquisition – DD, DC3DD, Guymager, and FTK Imager
- File recovery and data carving – Foremost, Magic Rescue, DD-Rescue, Scalpel, and Bulk_extractor
- PDF forensics – pdfparser
- NTFS drive recovery – scrounge-ntfs
- Memory/RAM analysis – Volatility 3
- Operating system identification – p0f
- Live Linux forensics – Linux Explorer
- Artifact discovery – swap_digger, mimipenguin, and pdgmail
- Browser-based forensic analysis tool – Autopsy Forensic Browser
- Complete forensic analysis tool – Autopsy 4
- Network discovery tools – netdiscover and nmap
- IoT search engine – Shodan.io
- Browser-based network packet capture analysis – Xplico
- Automated network packet capture analysis – Network Miner and PcapXray
- Online Pcap Analysis tools – `packettotal.com`, `apackets.com`

Next, let's have a look at purple teaming.

Understanding purple teaming

We can now have our cybersecurity moment of Zen as we get into purple teaming. The term **purple teaming** refers to the combination of skill sets in red and blue teaming. The color purple can also be achieved by mixing the colors red and blue, hence the name purple teaming. Looking back at all the skill sets and certifications mentioned in the red and blue teaming sections, it may seem like an impossible accomplishment; however, I guarantee you that there are many purple teamers out there who started as novices and ended up as professionals, myself included.

When I started my journey in cybersecurity in the early 2000s, I was far more interested in ethical hacking and pentesting (red teaming) at that point in time and spent many a night in front of my desktop reading, researching, and using the very limited tools available at that time. It was not until perhaps 2008 that I decided to get into DFIR and became very interested in the field of forensics, to the point where I started to teach the CHFI course alongside the CEH course.

Every time I thought to myself that I'd specialize in one, I'd come across a new tool that would point me in the direction of the other. Thankfully, this all worked out in my favor as I soon realized that red and blue teaming overlap in many aspects and also that there was never a point where I could say that what I had already learned was enough. My point here is that cybersecurity is such a dynamic field with so many paths that you can never know just enough. There is always some new exploit, an investigative tool, or an incident response procedure to learn, and it's up to you to decide whether you would like to specialize in one field or continue to learn and grow as I did and apply your knowledge when necessary.

Fast forward to today, and I'm the owner of the Computer Forensics and Security Institute, where I not only lead a purple team but I'm also the lead penetration tester as well as the lead forensic and incident response investigator. Again, it is very much possible to be well versed in both fields once you commit to it.

In this regard, I can comfortably state that Kali Linux is the perfect place to get started, as it offers the best tools for purple teaming. Let's have a sneak peek at some of the exploitation (red teaming tools) available to us, which are all preinstalled with any version of Kali.

This is just a snippet of the tools within the **Exploitation** menu of Kali; however, I use the **metasploit framework**, the **msf payload creator**, and the **social engineering toolkit (root)** religiously for red team assessments.

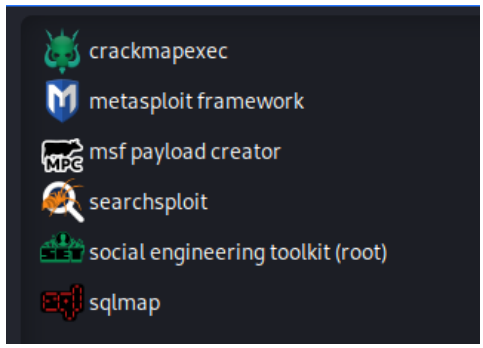


Figure 1.2 – Tools within the Exploitation menu

Now let's have a look at the **Forensic** menu in Kali Linux:

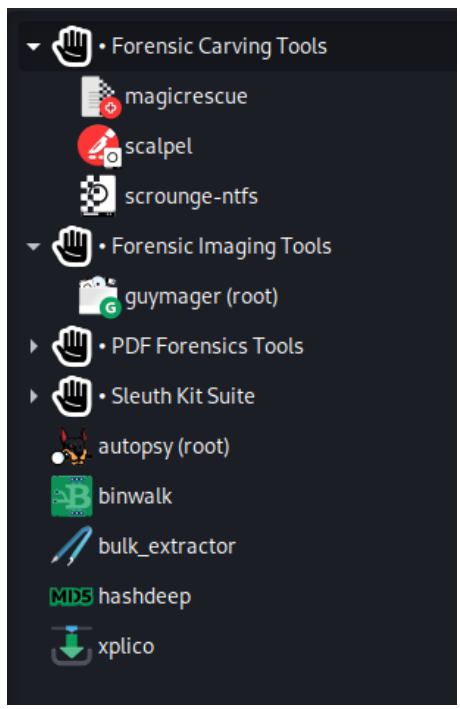


Figure 1.3 – Tools within the Forensics menu

Again, these are just some of the forensics tools, as the others can also be found by viewing the **All Applications** menu, which we will explore in *Chapter 3*, *Installing Kali Linux*. Kali Linux is one of the few user-friendly platforms that offers a variety of tools for purple teaming, and I look forward to showing you how to effectively use many of them in the coming chapters.

In *Chapter 3*, *Installing Kali Linux*, I'll show you, step by step, how to set up Kali Linux in a safe, virtual test environment where we can use our tools and download sample files for analysis. Although this virtual machine will be connected to the internet, we will use it in a sandboxed environment to ensure that it does not affect your production environment. In *Chapter 5*, *Installing Wine in Kali Linux*, I will also walk you through the process of installing Wine in Kali Linux to help build your ultimate blue and purple team arsenal of tools that will now combine the best open source Windows and Linux tools.

Now that we've looked at the differences between red, blue, and purple teaming, we will be moving on to understand digital forensics and also have a look at other forensic platforms and some commercial tools and quite importantly, gain some insight into forensic frameworks in *Chapter 2*, *Introduction to Digital Forensics*.

Summary

In this chapter, we were introduced to Kali Linux's Debian-based operating system and its usefulness in the world of cybersecurity. We also learned about the different teams in cybersecurity, such as red teams, comprised of individuals concerned with offensive security and ethical hacking, such as penetration testers, and blue teams, comprised of individuals concerned with defending networks and data, such as forensic investigators. We also learned that having both red and blue teaming skill sets and experience puts an individual into the highly skilled purple team, which suggests that the individual is versed in a wide range of tools for vulnerability assessments, penetration testing, and also incident response and digital forensics, many of which can be found in Kali Linux.

Next, we will dive a bit deeper into digital forensics, look at other forensic operating systems, and learn about forensic frameworks and commonly used open source and commercial tools. See you in the next chapter!

2
Introduction to Digital Forensics

This chapter introduces the various aspects of the science of digital forensics. It introduces you, particularly those of you who may be new to **Digital Forensics and Incident Response** (**DFIR**), to the basics, which we will build upon as we progress further into the book.

The topics we are going to cover in this chapter are:

- What is digital forensics?
- The need for blue and purple teams
- Digital forensics methodologies and frameworks
- Comparison of digital forensics **operating systems** (**OSs**)
- The need for multiple forensics tools in digital investigations
- Comparison of commercial versus open source forensic tools

What is digital forensics?

The first thing I'd like to cover in this chapter is an understanding of digital forensics and its proper practices and procedures. At some point, you may have come across several books, blogs, and even videos demonstrating various aspects of digital forensics and the different tools used. It is of great importance to understand that forensics itself is a science involving very well-documented best practices and methods to reveal whether something exists or does not.

Digital forensics involves the preservation, acquisition, documentation, analysis, and interpretation of evidence from various storage media types found. It is not limited to laptops, desktops, tablets, and mobile devices but also extends to data in transit, which is transmitted across public or private networks.

In most cases, digital forensics involves the discovery and/or recovery of data using various methods and tools available to the investigator. Digital forensics investigations include, but are not limited to, the following:

- **Data recovery**: Investigating and recovering data that may have been deleted, changed to different file extensions, and even hidden.
- **Identity theft**: Many fraudulent activities, ranging from stolen credit card usage to fake social media profiles, usually involve some sort of identity theft.
- **Malware and ransomware investigations**: To date, ransomware spread by Trojans and worms across networks and the internet are some of the biggest threats to companies, military organizations, and individuals. Malware can also be spread to and by mobile devices and smart devices.
- **Network and internet investigations**: Investigating **Denial-of-Service** (**DoS**) and **Distributed DoS** (**DDoS**) attacks and tracking down accessed devices, including printers and files.
- **Email investigations**: Investigating the source and IP (**internet protocol**) origins, attached content, and geolocation information can all be investigated.
- **Corporate espionage**: Many companies are moving away from print copies and toward cloud and traditional disk media. As such, a digital footprint is always left behind; should sensitive information be accessed or transmitted?
- **Child pornography investigations**: Sadly, the reality is that children are widely exploited on the internet and within the deep web. With the use of technology and highly-skilled forensic analysts, investigations can be carried out to bring down exploitation rings by analyzing internet traffic, browser history, payment transactions, email records, and images.

Next, we'll look at the need for blue and purple teams and the skillsets required for each.

The need for blue and purple teams

As discussed in *Chapter 1*, *Red, Blue, and Purple Teaming Fundamentals*, blue teams are more concerned with DFIR, and purple teamers are those individuals who can operate and understand both the offensive and defensive sides of cybersecurity.

In recent times, and especially during the Covid-19 pandemic lockdowns, you may have read or noticed that there was a substantial increase in cyberattacks. A major contributing factor to these attacks will have been the lack of awareness training by employees at all levels who may have fallen victim to phishing attacks. There were numerous phishing campaigns during the lockdown period, which tricked and social-engineered users into either opening malicious emails or divulging personal or corporate information. This was done by threat actors using email subjects relating to Covid-19 outbreaks, health updates, and even through requests for help from the compromised (hacked) email accounts of friends and family who may have appeared to be stranded in countries under lockdown and were asking for financial assistance to get back home.

In the last year or two, there has also been an alarming increase in ransomware attacks from notorious ransomware and **Advanced Persistence Threat** (**APT**) groups, with ransomware being readily available for purchase on the dark web. ALPHV, Darkside, Revil, and BlackCat are just some of the most notorious ransomware and APT groups whose names generally strike fear into the hearts of any cybersecurity professional. These groups are known to compromise individuals and companies and then request that a ransom be paid, ranging from thousands to millions of dollars. Not only is data highly encrypted and lost, but many of these groups now attack the victims using DDoS attacks and also release exfiltrated data belonging to the victim on dark web blogs. We'll delve a bit more into ransomware later on in the book.

Responding to an incident is one thing, as we (hopefully) may have a plan or formal procedures to adhere to, but if we don't have the right mechanisms in place to even detect the incident, then we may be far worse off than initially thought. This is where the need for blue and purple teamers arises. Hackers, malware developers, APTs, and the like do not take time off and certainly do not care about the financial state of companies and economies. Ransomware has, in the past, proven to be very profitable in some cases, and it must be considered that at any given day or time, malware developers are working on new methods of compromise.

It is imperative that each organization have a blue team, if not a purple team, or at least outsource the talent and functions of a team if deciding against having one in-house. There are many **Managed Security Services** (**MSS**) companies out there that can perform all necessary monitoring and alerting systems. However, a company with one or two well-trained blue and/or purple teamers will always have the advantage where preparedness and response are concerned, as in-house teams tend to have the advantage of not just knowing the technical details of the company but also having an understanding of the organizational culture and the level of (and sometimes lack of) awareness of the staff and general human resources, which more often than not, may prove to be the weakest link within the organization as far as cybersecurity is concerned.

Having blue teamers conduct periodic cybersecurity awareness training is just as important as patching critical infrastructure devices, such as firewalls, routers, switches, and servers. Cybersecurity awareness is now something that must be considered just as much as physical safety awareness, as one small mistake could potentially bring an entire conglomerate to its knees and cost the organization millions of dollars. It helps a great deal if employees are aware of the general cybersecurity threats to the organization and themselves, and more so if they are made aware of specific threats to their organization and sector and educated on specific threat instances that may have occurred in the organization. This way, the blue teamers can create a cybersecurity awareness culture, where persons are more alert and understand the risks and impact of individual actions as far as the security of the organization and its data and assets are concerned.

To summarize the above statements, blue and purple teams are responsible for several critical roles within the organization, including:

- Threat intelligence
- Threat hunting
- Incident response
- Monitoring and event management
- Log reviews
- Vulnerability assessments (red/purple team)
- Penetration testing (red/purple team)
- Threat simulation

In the next section, we'll look at digital forensics methodologies and frameworks, which we can use as best-practice guidelines when performing evidence acquisitions and forensic investigations.

Digital forensics methodologies and frameworks

Keeping in mind that forensics is a science, digital forensics requires that you follow appropriate best practices and procedures in an effort to produce the same results time and time again, providing proof of evidence, preservation, and integrity, which can be replicated if called upon to do so.

Although many individuals may not be performing digital forensics to be used as evidence in a court of law, it is best to practice DFIR in such a way as it can be accepted and presented in a court of law. The main purpose of adhering to best practices set by organizations specializing in digital forensics and incident response is to maintain the integrity of the evidence for the duration of the investigation. In the event that the investigator's work must be scrutinized and critiqued by another or an opposing party, the results found by the investigator must be able to be recreated, thereby proving the integrity of the investigation. The purpose of this is to ensure that your methods can be repeated and, if dissected or scrutinized, produce the same results time and again. The methodology used, including the procedures and findings of your investigation, should always allow for the maintenance of the data's integrity, regardless of what tools are used.

The best practices demonstrated in this book ensure that the original evidence is not tampered with, or in cases of investigating devices and data in a live or production environment, show well-documented proof that the necessary steps were taken during the investigation to avoid unnecessary tampering with the evidence, thereby preserving the integrity of the evidence. For those completely new to investigations, I recommend familiarizing yourself with some of the various practices and methodologies available and widely practiced by the professional community.

As such, there exist several guidelines and methodologies that you should adopt, or at least follow, to ensure that examinations and investigations are forensically sound.

The best-practices documents mentioned in this chapter are:

- The **Association of Chief Police Officers** (**ACPO**) *Good Practice Guide for Digital Evidence*
- The **Scientific Working Group on Digital Evidence** (**SWGDE**) Forensics Publications

Although written in 2012, the ACPO, now functioning as the **National Police Chiefs' Council** (**NPCO**), put forth a document in a PDF file called *Good Practice Guide for Digital Evidence*, which includes best practices when carrying out digital forensics investigations, particularly focusing on evidence acquisition. The ACPO *Good Practice Guide for Digital Evidence* was then adopted and adhered to by law enforcement agencies in England, Wales, and Northern Ireland and can be downloaded in its entirety at `https://www.npcc.police.uk/documents/crime/2014/Revised%20Good%20Practice%20Guide%20for%20Digital%20Evidence_Vers%205_Oct%202011_Website.pdf`.

Some useful topics provided by this guide include:

- Digital evidence locations
- Issues relating to seizure
- Capturing online evidence
- Data analysis and interpretation

Other useful and more recent document sets on best practices in digital forensics can be found on the SWGDE website. The SWGDE was founded in 1998 by the Federal Crime Laboratory Directors Group with major members and contributors, including the FBI, DEA, NASA, and the Department of Defense Computer Forensics Laboratory. Though this document details procedures and practices within a formal computer forensics laboratory setting, the practices can still be applied to non-laboratory investigations by those not currently in or with access to such an environment.

The previous editions of this book referred to the SWGDE documents published between 2006 and 2016. The older *SWGDE Best Practices* publication from 2006 is no longer on the SWGDW website, but it sheds light on many of the topics, including:

- Evidence collection and acquisition
- Investigating devices that are powered on and powered off
- Evidence handling
- Analysis and reporting

This document is publicly available at `https://www.oas.org/juridico/spanish/cyb_best_pract.pdf`.

There have since been several updates, and all newer living documents can now be found on the updated SWGDE site at `https://www.swgde.org/documents/published-by-committee/forensics`. There are over 35 publications available for free download, 17 of which were published between 2017 and 2022. I highly recommend downloading these documents as they are nothing short of magnificent for DFIR practitioners and are each approximately 10 pages or less.

The *SWGDE Best Practices* updated living documents, as they are referred to, contain excellent, well-researched, and modern publications for DFIR best practices on the following:

- *Vehicle Infotainment and Telematics Systems* (2022)
- *Drone Forensics* (2022)
- *Acquiring Online Content* (2022)
- *Digital Evidence Acquisition from Cloud Service Providers* (2020)
- *Mobile Evidence Collection and Preservation* (2020)
- *Digital Evidence Collection* (2018)
- *Computer Forensic Examination* (2018)
- *Computer Forensic Acquisition* (2018)

There are many other useful best practices publications.

Now that we are familiar with DFIR methodologies and best practices, let us now have a look at DFIR frameworks.

DFIR frameworks

The aforementioned best practice guides, although dated, can still be used in conjunction with one or more frameworks. Whatever the task, the goals should remain the same, with the integrity of evidence always preserved throughout the entire DFIR process of evidence collection, examination, analysis, and reporting.

There are several DFIR frameworks available, which, if applied and implemented by organizations of all sizes, can aid in the business continuity and disaster recovery processes as any worthy DFIR plan should. Here is a list of some DFIR frameworks that will be discussed in more detail in *Chapter 7*, *Incident Response, Data Acquisition, and DFIR Frameworks*. It may be a good idea to download these documents and keep them together as reference materials, which you should become familiar with as a DFIR practitioner (feel free to proceed to the next section and come back to these later):

- *ISO/IEC 27037:2021 Information technology — Security techniques — Guidelines for identification, collection, acquisition, and preservation of digital evidence*: `https://www.iso.org/obp/ui/#iso:std:iso-iec:27037:ed-1:v1:en`

- *NIST Special Publication (SP) 800-86 Guide to Integrating Forensic Techniques into Incident Response*: `https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-86.pdf`
- *NIST Special Publication (SP) 800-61 Revision 2 Computer Security Incident Handling Guide*: `https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf`
- *D4I - Digital forensics framework for reviewing and investigating cyber*: `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=929147`
- *NISTIR 8428 Digital Forensics and Incident Response (DFIR) Framework for Operational Technology (OT)* (June 2022): `https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8428.pdf`

Let's now look at some forensic OSs and the differences between them in the following section.

Comparison of digital forensics operating systems

Just as there are several commercial tools available, there exist many open source tools available to investigators, amateurs, and professionals alike. Many of these tools are Linux based and can be found on several freely available forensic distributions, as we will discuss in this section.

The main question that usually arises when choosing tools is based on commercial versus open source. Whether you are using commercial tools or open source tools, the result should be the same, with the preservation and integrity of the original evidence being the main priority.

> **Important note**
>
> Budget is always an issue, and some commercial tools (as robust, accurate, and user-friendly as they might be) can cost thousands of dollars.

The open source tools are free to use under various open source licenses and should not be counted out just because they are not backed by enterprise developers and researchers.

Many open source tools are widely reviewed by the forensic community and may be open to more scrutiny, as they are available to the public and are built in non-proprietary code.

Though the focus of this book is on the forensic tools found in Kali Linux, which we will begin looking at toward the end of this section and onward, here are some of the more popular open source forensic distributions, or distros, available.

Each of the distros mentioned in the following sections is freely available at many locations, but for security reasons, we will provide the direct link from their homepages. The OSs featured in this section are not listed in any particular order and do not reflect any ratings, reviews, or even the author's personal preference.

The forensic platforms we will be discussing in this section are:

- DEFT Linux 8
- **Computer Aided INvestigative Environment** (**CAINE**) 12.4 Sidereal and CAINE 11 Wormhole
- CSI Linux 2022.1
- Kali Linux 2022.3

In case you're wondering, I use all of the platforms mentioned previously as live systems and virtual machines, and I also have dedicated machines for CSI Linux and Kali. Here's a snip of my **VirtualBox Manager** listing:

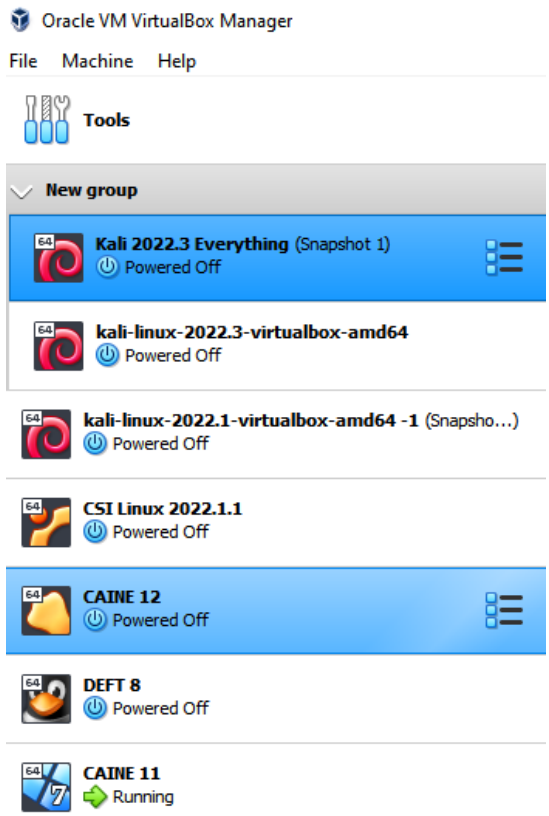


Figure 2.1: Listing of virtual machines in VirtualBox Manager

Digital evidence and forensics toolkit Linux

Digital Evidence and Forensics Toolkit (**DEFT**) Linux comes in a full version and a lighter version called **DEFT Zero**. For forensic purposes, you may wish to download the full version, as the Zero version does not support mobile forensics and password-cracking features:

- Download page for DEFT Linux 8: `https://archive.org/download/deft-8.2/deft-8.2.iso`
- Download page for DEFT Linux Z (2018-2): `https://sourceforge.net/projects/archiveos/files/d/deft/deftZ-2018-2.iso/download`
  - Based on: Ubuntu Desktop
  - Distribution type: Forensics and incident response

Like the other distros mentioned in this list, DEFT, as shown in the following screenshot, is also a fully capable **live response** forensic tool that can be used on the go in situations where shutting down the machine is not possible and also allows for on-the-fly analysis of RAM and the swap file. Although a bit outdated, I personally still use DEFT 8.2 specifically for packet analysis with Xplico:

Figure 2.2: DEFT Linux 8 boot menu

When booting from the DEFT Linux DVD, bootable flash, or other media, you are presented with various options, including the options to install DEFT Linux to the hard disk, use it as a live-response tool, or as an OS, by selecting the **DEFT** Linux 8 live option, as shown here:

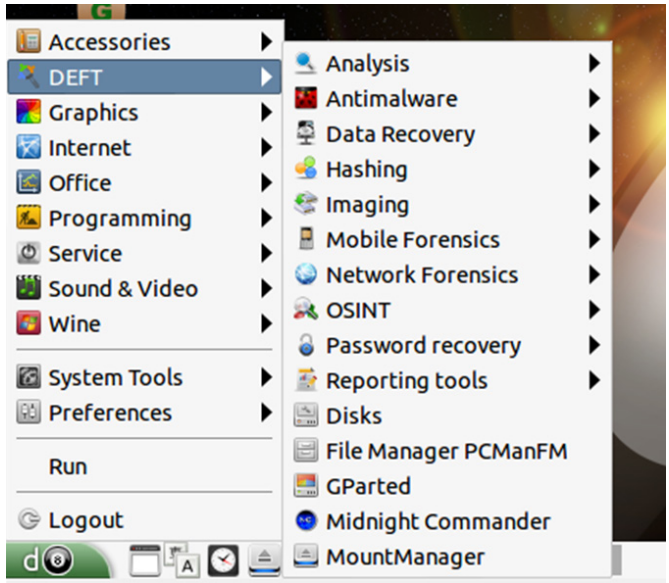


Figure 2.3: DEFT Linux menu

In the previous screenshot, it can be seen that there are several forensic categories in DEFT Linux 8, such as **Antimalware**, **Data Recovery**, **Hashing**, **Imaging**, **Mobile Forensics**, **Network Forensics**, **Password recovery**, and **Reporting tools**. Within each category exist several tools created by various developers, giving the investigator quite a variety from which to choose.

DEFT contains many useful tools, and for comparative purposes, let's look at the tools available in the **Analysis** menu of DEFT:

Figure 2.4: Forensic analysis tools in DEFT Linux

Computer Aided INvestigative Environment (CAINE)

CAINE is a live-response bootable CD/DVD with options for booting in safe mode, text mode, as a live system, or in RAM, as shown as follows. It should be noted that the latest version, CAINE 12.4 (Sidereal), is not installable and is strictly a live-system OS only. If you wish to install CAINE, you can do so with CAINE 11 (Wormhole), although they can both be used as live systems for real-time acquisition and forensics, as seen in the following screens.

Figure 2.5 shows a screenshot of the CAINE 12.4 (Sidereal) boot screen menu, which allows you to choose between the different boot options in CAINE 12.4:

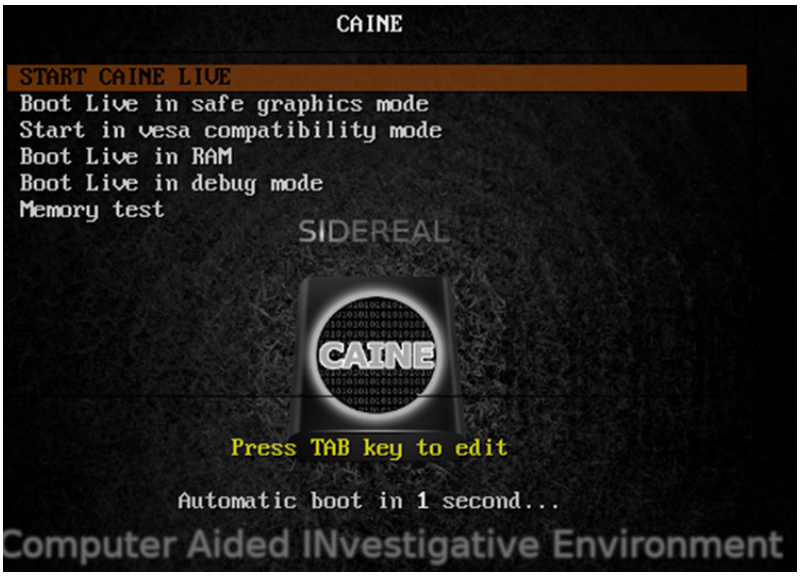


Figure 2.5: The Caine 12.4 boot menu

Figure 2.6 shows the CAINE 11 boot screen, which is almost identical to CAINE 12, as seen in *Figure 2.5*:

Figure 2.6: The Caine 11 boot menu

The following is a summary of CAINE details:

- Both CAINE 11 and 12 can be downloaded at `http://www.caine-live.net/`
- Based on: GNU Linux
- Distribution type: Forensics and incident response

After selecting your boot option, one of the most noticeable features of CAINE is the easy way to find the write-blocker feature, seen and labeled as an **UnBlock** icon, as shown in the following screenshot:

Figure 2.7: CAINE 12 desktop interface

Activating this feature prevents the writing of data by the CAINE OS to the evidence machine or drive. This feature is incredibly useful if you do not have access to a hardware write-blocker, which we will look at in *Chapter 7, Incident Response, Data Acquisitions, and DFIR Frameworks*.

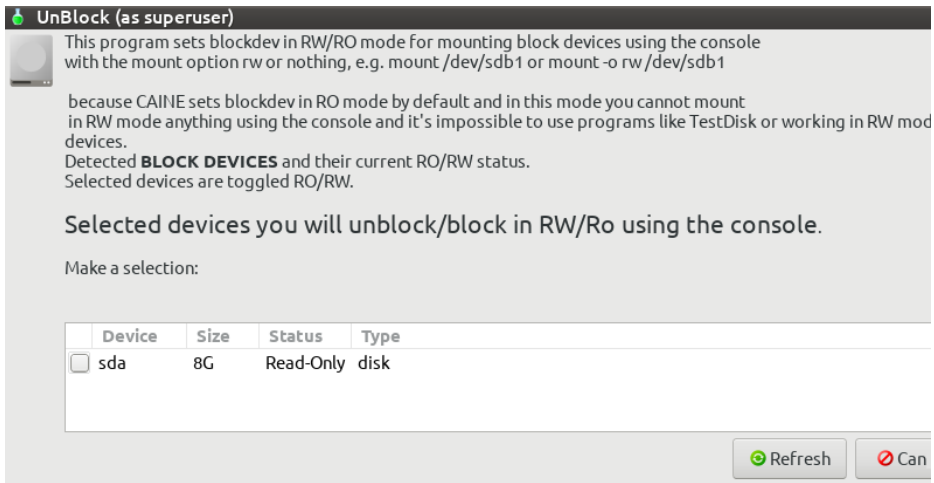


Figure 2.8: CAINE write-blocker interface

Forensic tools is the first menu listed in CAINE. Like DEFT Linux, there are several categories in the menu, as seen in the following screenshot, with several of the more popular tools used in open source forensics. Besides the categories, there are direct links to some of the more well-known tools, such as **Guymager** and **Autopsy**, which will both be covered in detail in later chapters:

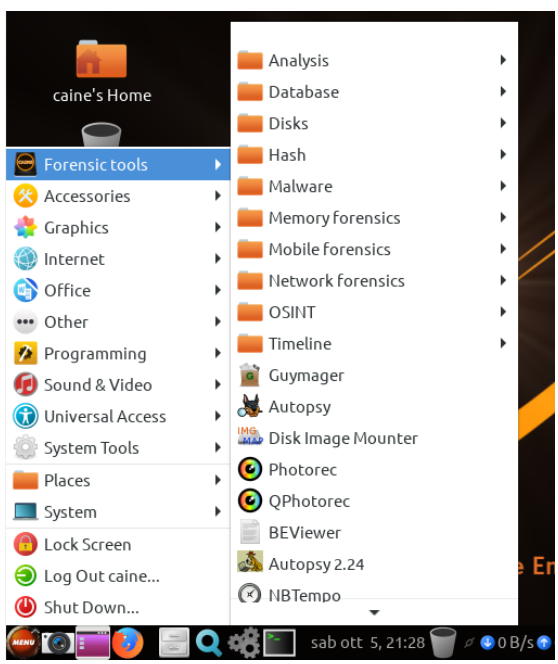


Figure 2.9: Forensic tools available in CAINE 12

Let's have a look at the tools within the **Analysis** menu in CAINE. There are far more **Graphical User Interface** (**GUI**) DFIR tools in CAINE than in DEFT, as previously seen:

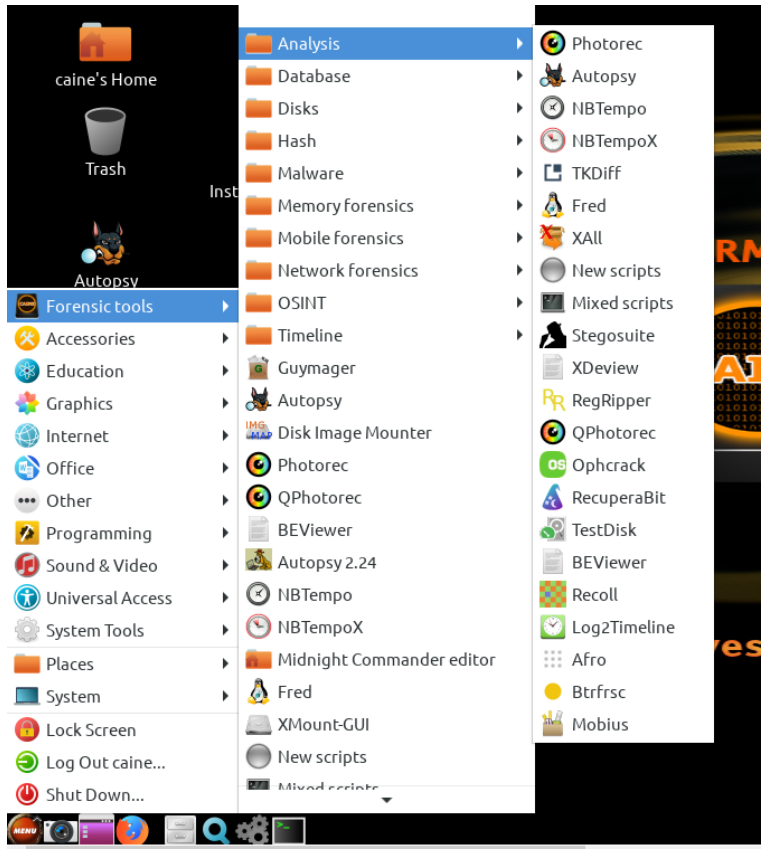


Figure 2.10: Analysis tools available in CAINE 12

We also have quite a variety of tools available in CAINE's **Memory forensics** menu:

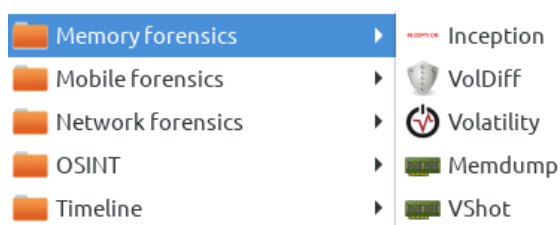


Figure 2.11 – Memory forensics tools available in CAINE 12

For a full list of the features and packages included in CAINE at the time of publishing, please visit `http://www.caine-live.net/page11/page11.html`.

The latest version of CAINE 12.4 Sidereal can be downloaded at `https://deb.parrot.sh/direct/parrot/iso/caine/caine12.4.iso` in ISO format and is approximately 3.9 GB in size.

As mentioned before, if you wish to use the installable CAINE 11 Wormhole, it can be downloaded at `https://deb.parrot.sh/direct/parrot/iso/caine/caine12.4.iso`.

For installation on a USB thumb drive, please ensure that the drive capacity is no less than 8 GB; however, a 16 GB drive is preferred. Automated creation of a bootable CAINE drive can be done using the Rufus tool, as shown in *Chapter 3*, *Installing Kali Linux*.

CSI Linux

CSI Linux is relatively new to the scene and was developed by visionary *Jeremy Martin* and his amazing team, who have put together an incredibly robust forensic/blue teaming platform, which I often use just as much as Kali Linux. CSI Linux 2022.1 comes on virtual appliance installations for both VirtualBox and Kali Linux and can also be downloaded as a bootable image for live DFIR. There is also a plan to release a CSI Linux **Security Information and Event Management** (**SIEM**) in the near future.

CSI Linux 2022.1 can be downloaded directly from `https://csilinux.com/download`.

> **Helpful hint**
>
> If using CSI Linux, both the username and the password are `csi`.

Figure 2.12: CSI Linux login screen

Much like DEFT, CAINE, and Kali, CSI Linux has a very user-friendly desktop interface:

Figure 2.13: CSI Linux desktop interface

Clicking on the CSI Linux menu button reveals an extensive list of categories, including the **OSINT and Online Investigations**, **Dark Web**, and **Threat Intelligence** categories, which is where CSI Linux really shines, in my opinion, as other platforms lack these pre-installed tools:

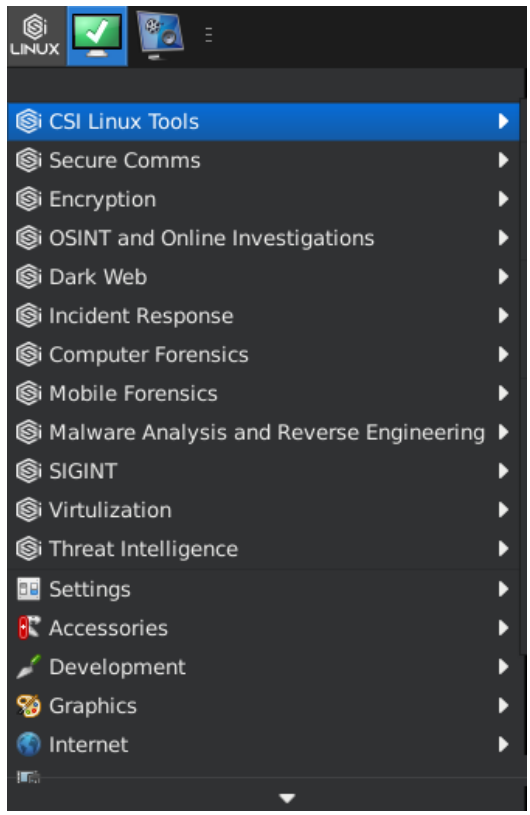


Figure 2.14: CSI Linux Forensic tools

Again for comparative purposes, let's have a look at the **OSINT and Online Investigations**, **Dark Web**, and **Computer Forensics** menus.

The **OSINT and Online Investigations** menu has several sub-menus with very useful tools for social media, email, phone, and social-engineering tools, all of which I frequently use:

Figure 2.15: CSILinux OSINT and Online Investigations tools

The **Dark Web** investigation tools are also very popular in purple teaming as they are used by pen testers and DFIR practitioners alike:

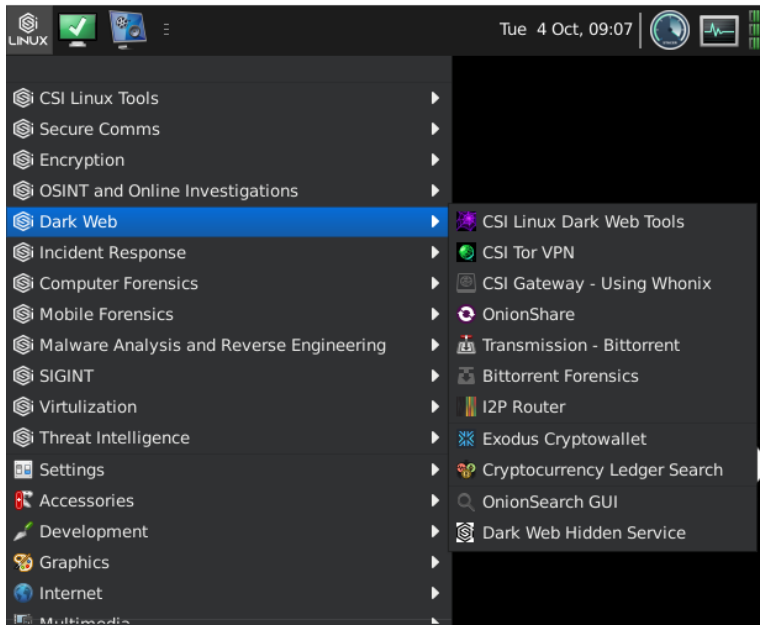


Figure 2.16: CSI Linux Dark Web investigation tools

The **Computer Forensics** and **Incident Response** menus are populated with an extensive list of updated and modern GUI and **Command Line Interface** (**CLI**) tools for conducting full investigations:

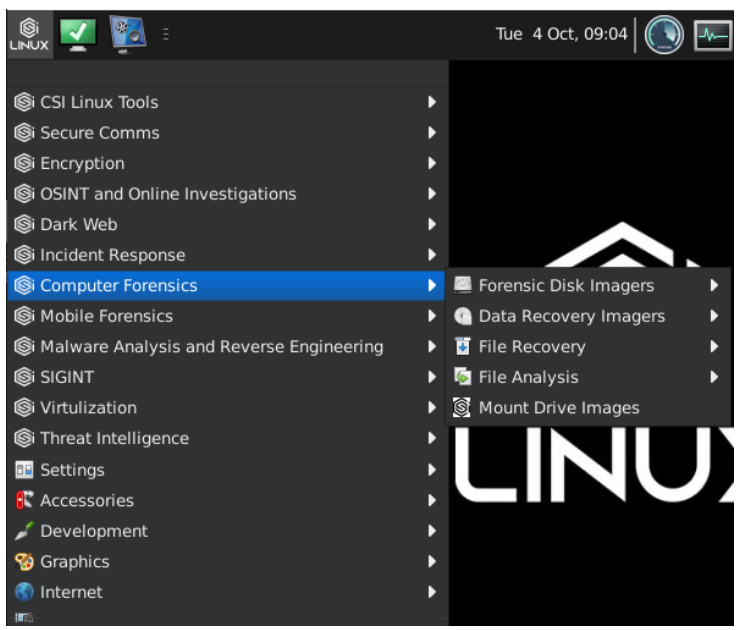


Figure 2.17: CSI Linux Computer Forensics tool menu

CSI Linux also has an impressive category listing for **Incident Response** tools, as seen in *Figure 2.18*:

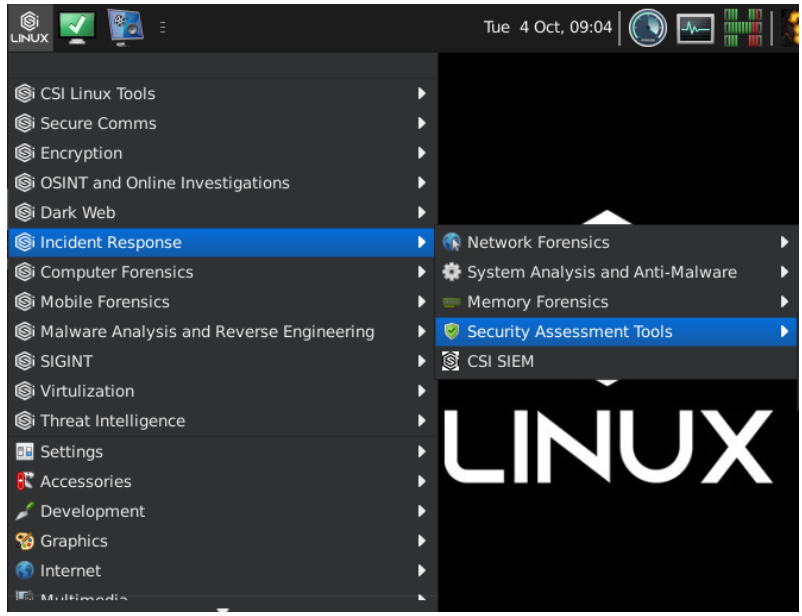


Figure 2.18: The CSI Linux Incident Response menu

I strongly suggest downloading and becoming acquainted with CSI Linux. Many of the tools we will be learning to use in this book are also installed in CSI Linux, thereby lessening the learning curve.

Kali Linux

Finally, we get to this lovely gem, Kali Linux, fully discussed in detail from its installation to advanced forensics usage in the next chapter and throughout this book:

- Homepage: `https://www.kali.org/`
- Based on: Debian
- Distribution type: Penetration testing, forensics, and anti-forensics

Kali Linux was created as a penetration testing or pen-testing distro under the name BackTrack, which then evolved into Kali Linux in 2015, as mentioned in *Chapter 1*, *Red, Blue, and Purple Teaming Fundamentals*. This powerful tool is the definite tool of choice for penetration testers and security enthusiasts worldwide. As a **Certified EC-Council Instructor** (**CEI**) for the **Certified Ethical Hacker** (**CEH**) course, this OS is usually the star of the class due to its many impressive bundled security programs, ranging from scanning and reconnaissance tools to advanced exploitation tools and reporting tools.

Like the aforementioned tools, Kali Linux can be used as a live response forensic tool, as it contains many of the tools required for full investigations. Kali, however, can also be used as a complete OS, as it can be fully installed on a hard disk or flash drive and contains several tools for productivity and entertainment. It comes with many of the required drivers for successful use of hardware, graphics, and networking, and also runs smoothly on both 32-bit and 64-bit systems with minimal resources; it can also be installed on certain mobile devices, such as **Nexus** and **OnePlus**, and other phones and tablets.

Adding to its versatility, upon booting from a live CD/DVD or flash drive, the investigator has several options to choose from, including **Live (forensic mode)**, which leaves the evidence drive intact and does not tamper with it by also disabling any auto-mounting of flash drives and other storage media, maintaining the integrity of the original evidence throughout the investigation.

Again, Kali Linux is the ultimate purple teaming platform of choice, as it comes with pen testing and forensic tools all nicely bundled into one fantastic package.

When booting to Kali Linux from a DVD or flash drive, the user is first presented with options for a live environment and installation. Choosing the third option from the list carries us into **Live (forensic mode)**, as seen in the following screenshot:

Figure 2.19: The Kali Linux boot menu

Once the Kali **Live (forensic mode)** option has booted, the investigator is presented with the exact same home screen as can be seen if using any of the GUIs in Kali, as shown in the following screenshot:

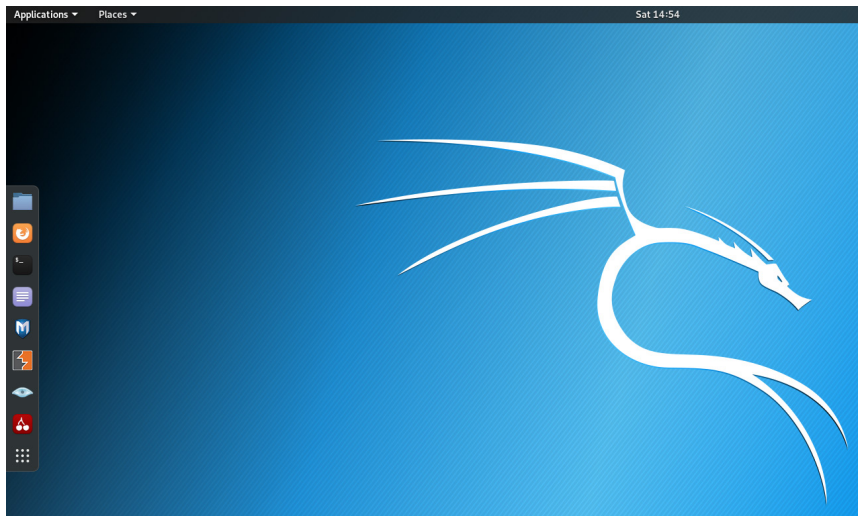


Figure 2.20: The Kali Linux desktop interface

Of course, Kali can also be installed as a full OS on a virtual machine or directly onto the hard drive, which will be covered in *Chapter 3*, *Installing Kali Linux*. Here's a screen of my desktop with some Windows forensic's tools, such as FTK Imager and the Autopsy GUI, installed using Wine:

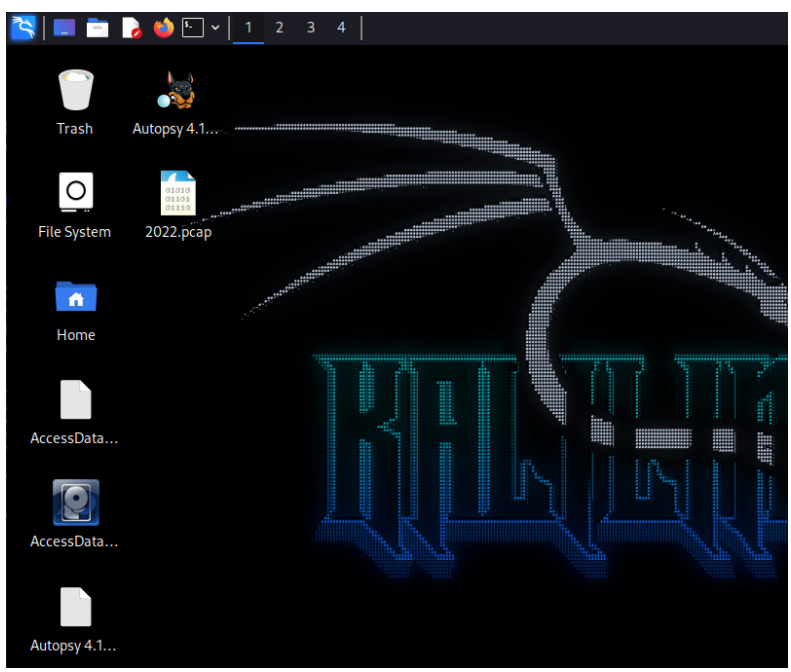


Figure 2.21: The Kali Linux customized desktop

The Kali menu can be found at the top left corner by clicking on **Applications**. This brings the user to the menu listing, which shows the forensics category lower down, as **11 - Forensics**. The following screenshot gives an idea of some of the **Forensics** tools available in Kali that we'll be using later on in the book:

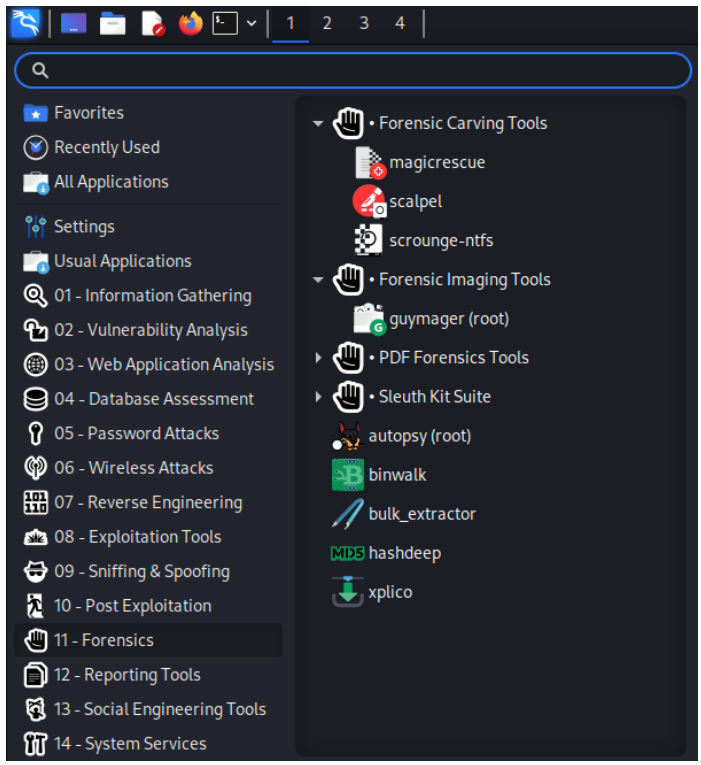


Figure 2.22 – The Kali Linux Forensics tools menu

It should be noted that the tools listed are not the only tools available in Kali. Several other tools can be brought up via the terminal, as we'll see in later chapters.

It's also noteworthy that, when it is in forensic mode, not only does Kali not tamper with the original evidence drive, but it also does not write data to the swap file, where important data that was recently accessed and stored in memory may reside.

For a full list of the features and packages included in the Kali Linux OS at the time of publishing, please visit `https://www.kali.org/tools/`.

Kali can operate as a live response forensic tool but can also be used as a full OS, just like Windows, macOS, and Android, as it contains several built-in tools for productivity and everyday use. The fact that Kali can be installed on a hard disk means that several other tools can be downloaded and updated regularly, giving continuous access to all IT security and forensic tools. This allows you to save progress as you use the tools and not have to worry too much about restarting your machine should you decide to use it as a full OS.

Using these open source OSs, such as Kali, gives us a range of tools to choose from and work with. Many tools exist for performing the same tasks within each category in the distros. This is good because our findings should be able to be replicated using different tools. This is especially useful in instances where the investigator's work may be critiqued, and the integrity of the case and evidence questioned and scrutinized; using multiple tools correctly will yield consistent results. Let's now look at why we may need to use more than one tool in investigations in the next section.

The need for multiple forensics tools in digital investigations

Preservation of evidence is of the utmost importance. Using commercial and open source tools correctly will yield results; however, for forensically sound results, it is sometimes best if more than one tool can be used to produce the same results.

Another reason to use multiple tools may simply be cost-effectiveness. Some of us may have a large budget to work with, while others may have a limited one or none at all. Commercial tools can be costly, especially due to research and development, testing, advertising, and other factors. Open source tools, while tested by the community, may not have the available resources and funding as with commercial tools.

So then, how do we know which tools to choose?

Digital forensics is often quite time-consuming, which is one of the reasons you may wish to work with multiple forensic copies of the evidence. This way, you can use different tools simultaneously in an effort to speed up the investigation. While fast tools may be a good thing, we should also question the reliability and accuracy of the tools.

The **National Institute of Standards and Technology** (**NIST**) has developed a **Computer Forensics Tool Testing** (**CFTT**) program that tests digital forensic tools and makes all findings available to the public. Several tools are chosen based on their specific abilities and placed into testing categories, such as disk imaging, carving, and file recovery. Each category has a formal test plan and strategy for testing, along with a validation report, again available to the public.

More on the CFTT program can be found at `https://www.cftt.nist.gov/disk_imaging.htm`. Testing and validation reports on many of the tools covered in this book can be found at `https://www.dhs.gov/science-and-technology/nist-cftt-reports`.

To re-enforce the importance of using multiple tools in maintaining the integrity of your investigations and findings, multiple tools will be demonstrated from *Chapter 7*, *Incident Response, Data Acquisitions, and DFIR Frameworks*, onward within this book.

Thus far, we've covered open source tools. In the next section, I will discuss some popular commercial forensic tools that you may come across if you are, or intend to be, a DFIR professional.

Commercial forensics tools

Even though we'll be working with open source tools, I'd like to mention some of the more popular commercial tools that you may be interested in using or adding to your arsenal as a forensic professional.

Belkasoft Evidence Centre X

Belkasoft **Evidence Center** (**EC**) X is an automated incident response and forensic tool that is capable of analyzing acquired images of memory dumps, virtual machines, cloud, mobile backups, and physical and logical drives.

Belkasoft EC X is also capable of searching for, recovering, and analyzing the following types of artifacts:

- Office documents
- Browser activity and information
- Email
- Social media activity
- Mobile applications
- Messenger applications (WhatsApp and Facebook Messenger)
- Cloud acquisitions

More information on Belkasoft products can be found at `https://belkasoft.com/`.

Belkasoft also has a free acquisition tool and Ram Capturer tool available along with a trial version of their EC, available at `https://belkasoft.com/get`.

Exterro Forensic Toolkit (FTK)

Exterro **Forensic Toolkit** (**FTK**) has been around for some time and is used professionally by forensics investigators and law enforcement agencies worldwide. FTK also integrates with Belkasoft for a better experience. Some features of FTK include:

- Fast processing with multi-core support using four engines
- Ability to process large amounts of data
- Indexing of data to allow faster and easier searching and analysis

- Password cracking and file decryption
- Automated analysis
- Ability to perform customized data carving
- Advanced data recovery

More information on FTK can be found at `https://accessdata.com/products-services/forensic-toolkit-ftk`.

The trial version of FTK can be downloaded at `https://go.exterro.com/exterro-software-demo`. Exterro also has an image acquisition tool that is free to download and use and is available at `https://go.exterro.com/l/43312/2022-08-23/f7rylq`.

OpenText EnCase Forensic

Previously known as Guidance EnCase Forensic and now known as OpenText EnCase Forensic, this tool has also been at the forefront for many years and has also been used internationally by professionals and law enforcement agents alike for almost two decades. Much like FTK, EnCase comes with several solutions for incident response, e-discovery, and endpoint and mobile forensics.

Apart from being a full digital forensics solution and suite, some of the other features of EnCase include the following:

- Acquiring images from over 25 different types of mobile devices, including phones, tablets, and even GPS devices.
- Support for Microsoft Office 365.
- Evidence decryption using Check Point's Full Disk Encryption.
- Deep forensic and triage analysis. More information on EnCase can be found at `https://www.guidancesoftware.com/encase-forensic`.

Now that we've covered some of the more popular commercial forensics tools, I'd like to briefly talk about anti-forensics, which you can research further in your own time.

Anti-forensics – threats to digital forensics

As much as we would like the tasks involved in digital forensics to be as easy as possible, we do encounter situations that make investigations, and life as a forensics investigator, not-so-simple and sometimes stressful. People wishing to hide information and cover their tracks, and those who have malicious intent or participate in cyber crimes, often employ various methods to try to foil the attempts of forensic investigators with the hope of hampering or halting investigations.

In recent times, we've seen several major digital breaches online, especially from 2011 onward. Many of these attacks allegedly came from or were claimed to be the work of infamous hacker groups, such as LulzSec, Anonymous, Lizard Squad, and many others, including individuals and hacktivists (people that hack for a specific cause or reason and are less concerned about doing time in prison). Some of these hacks and attacks not only brought down several major networks and agencies but also cost millions in damage, directly and indirectly; as a result, the loss of public confidence in the companies contributed to further increases in damages.

These daring, creative, and public attacks saw the emergence of many other new groups that learned from the mistakes of past breaches by Anonymous and others. Both social media and underground communication channels soon became the easiest forms of communication between like-minded hackers and hacktivists. With the internet and World Wide Web becoming easily accessible, this also saw competition not only between IPs but also private companies and corporations, which led to the creation of free wireless hotspots on almost every street with businesses, small or large.

The result of having internet access at just about every coffee shop enabled anyone with a smartphone, tablet, laptop, or other devices to acquire almost unauthenticated access to the internet. This gave them access to hacker sites and portals, along with the ability to download tools, upload malware, send infected emails, or even carry out attacks.

We should also consider the impact of encryption techniques on our investigations, which we will look at next.

Encryption

Adding to this scenario is the availability of more user-friendly tools to aid in the masking of **Publicly Identifiable Information** (**PII**), or any information that would aid in the discovery of unveiling suspects involved in cyber-crimes during forensic investigations. Tools used for the encryption of data and anonymity, such as masking of IP addresses, are readily available to anyone, most of which are increasingly more and more user-friendly.

It should also be noted that many Wi-Fi hotspots themselves can be quite dangerous, as these can be easily set up to intercept personal data, such as login and password information, together with PII (such as social security numbers, date of birth info, and phone numbers) from any user that may connect to the Wi-Fi and enter such information.

The process of encryption provides confidentiality between communication parties and uses technology in very much the same way we use locks and keys to safeguard our personal and private belongings. For a lock to open, there must be a specific matching key. So too, in the digital world, data is encrypted or locked using an encryption algorithm and must use the same key to decrypt or unlock the data. There also exists another scenario where one key may be used to encrypt or lock the data and another used to decrypt the data. Two very popular encryption tools are **TrueCrypt** and **VeraCrypt**.

These two encryption tools use very high encryption methods that keep data very confidential. The main barrier to forensics may be acquiring the decryption key to decrypt or unlock access to the data.

> **Important note**
>
> TrueCrypt and VeraCrypt not only encrypt files but also encrypt folders, partitions, and entire drives!

Added to the use of encryption tools, such as those mentioned here, we should also consider the use of anonymity tools, such as those mentioned in the next section.

Online and offline anonymity

Encryption, in particular, can make investigations rather difficult, but there is also the concept of anonymity, which adds to the complexity of maintaining the accuracy of the true sources found in investigations. Like encryption, there exist several free and open source tools for all OS platforms, such as Windows, MacOS, Linux, and Android, which attempt to (most often successfully) mask the hiding of someone's digital footprint. This digital footprint usually identifies a device by its IP and **Media Access Control** (**MAC**) address. Without going into the network aspect of things, these two digital addresses can be compared to a person's full name and home address, respectively.

Even though a person's IP address can change according to their private network (home and work) and public network (internet) access, the MAC address remains the same.

However, various tools are also freely available to spoof or fake your IP and MAC addresses for the purpose of privacy and anonymity. Adding to that, users can use a system of routing their data through online servers and devices to make tracing the sources of the sent data quite difficult. This system is referred to as **proxy chaining** and does keep some of the user's identity hidden.

A good example of this would be **Tor Browser**; it uses onion routing and several proxies worldwide to route or pass the data along from proxy to proxy, making the tracing of the source very difficult but not impossible. You can think of proxy chains as a relay race, but instead of having four people, one passing the baton to the next, the data is passed between hundreds of proxy devices worldwide.

Now you should have a basic understanding of anti-forensics and also understand the use of encryption and anonymity tools to hide or obscure information and data that may complicate DFIR investigations. Let's summarize the topics we just covered and then move on to the next chapter.

Summary

We saw that digital forensics is still a relatively new field, although forensic science has been around for a very long time, as far back as the early 1900s. Although digital forensics may have only been on the scene since the early 2000s, as a science, we have certain best practices, procedures, standards, and frameworks, such as those created by the ACPO, SWGDE, and NIST, to adhere to. These maintain the accuracy and the integrity of both the findings and the actual evidence when carrying out investigations, whether as an amateur or professional digital forensic investigator.

Some of the commercial tools mentioned were Belkasoft EC X, FTK, and EnCase Forensics. Many of the open source tools available are made for Linux-based distributions and can be downloaded individually, but many are readily available within certain forensic and security OSs or distributions. Some of these distros are DEFT Linux, CAINE, CSI Linux, and of course, Kali Linux; all of these are freely available for download at the links provided.

I hope this introduction to digital forensics was informative and fun for you. Now that we have a foundation of forensics, let's go deeper into Kali Linux, as we learn how to download, install, and update Kali in the next chapter. See you on the next page!

3
Installing Kali Linux

Join me as we get started with the practical aspects of Kali Linux. Some of you may already be familiar with the installation process and perhaps even some of the advanced features, such as partitioning and networking. For beginners and those of you who are new to Kali Linux, we encourage you to pay particular attention to this chapter as we begin from the absolute basics of downloading Kali Linux, working our way up to a successful installation.

The topics that we are going to cover in this chapter are the following:

- Installing Kali Linux on portable storage media for live DFIR
- Downloading and installing Kali Linux as a standalone operating system
- Installing Kali Linux in VirtualBox

Technical requirements

As previously mentioned, Kali Linux has been around for quite some time. Known previously as BackTrack, with releases from versions one to five, Kali Linux was first seen in 2015 and released as Kali 1.0. From 2016 onward, Kali was then named according to the year. So, for instance, at the time of writing this book, the version used is Kali 2022.3, released in August 2022.

For those running older versions of Kali Linux or purchasing this book at a later date when new versions of Kali Linux, such as 2023 and later, may be available, you can easily update your instance of Kali Linux by using the `sudo apt-get update` command, demonstrated later in the *Updating Kali* section.

Downloading Kali Linux

For safety and security reasons, it is always best to download Kali Linux directly from the website of its creators, **Offensive Security**. The main reason for this is that the downloads of Kali on other pages could possibly be fake or, worse, infected with malware, such as Trojans, rootkits, and even ransomware. Offensive Security has also included hashes of all versions of Kali downloads on their site, allowing users to compare the hash of their downloaded version of Kali with what was generated and posted by Offensive Security on their website (`https://www.kali.org`). Once there, you can click on the **Downloads** link or go directly to the Kali Linux **Downloads** page by visiting `https://www.kali.org/get-kali/`.

Once on the **Downloads** page, four main options are available, and you are prompted to choose your version of Kali Linux for the platform of your choice:

- **Installer Images**: These ISO images can be downloaded and installed directly on the hard drive to be used as a main operating system in the same way that you would install Windows or another operating system. ISO files (or ISO images, as they are commonly called) are exact copies of data used specifically when duplicating data.
- **Virtual Machines**: These images are used on virtual platforms, such as VirtualBox and VMware, and are pre-configured to run straight out of the box.
- **ARM**: These images are specifically for low-powered devices and **Single Board Computers** (**SBC**) such as Raspberry Pi, Pinebook, and Gateworks devices.
- **Mobile**: This portable version of Kali Linux is called Kali NetHunter and can now be installed on many more devices since its inception, including OnePlus, ZTE, Nexus, Samsung, Sony, Nokia, and even the TicWatch Pro smartwatch. Kali NetHunter Lite can also be installed on ARM64 and ARMhf devices.

  The following screenshot shows the main download menu with various platform options.

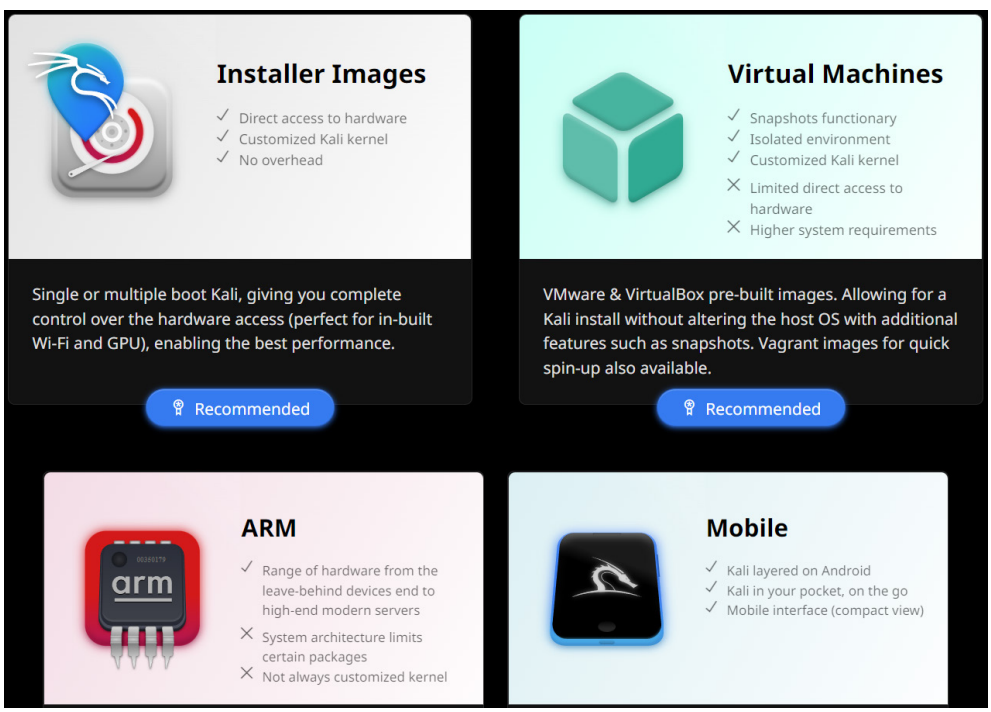


Figure 3.1: Available versions of Kali Linux

Once you click on a menu item, it will take you to the downloads section for that specific platform. For example, if you click on the **Virtual Machines** option in the menu, you will be taken to the download section for both 64-bit and 32-bit versions of Kali Linux for **VMware** and **VirtualBox**.

> **Important note**
>
> 32-bit operating systems are limited to utilizing only 4 GB of RAM. Should you have a system with more than 4 GB of RAM, you may wish to download the 64-bit version of Kali Linux.

In the following screenshot, the **64-bit** version is selected.

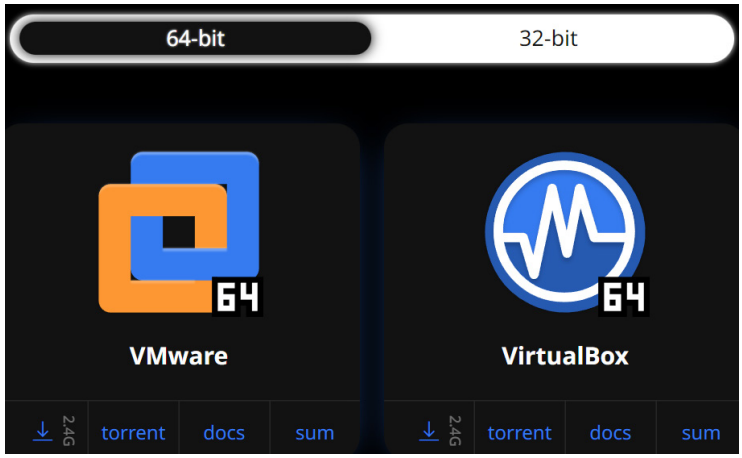


Figure 3.2: Available 32-bit and 64-bit Kali Linux versions for VirtualBox and VMware

You can choose either the **VMware** or **VirtualBox** platforms. You can download Kali directly within the browser by clicking on the download arrow below the icons, which indicates that the file is **2.4G** in size. You can also download it using the Torrent software of your choice, such as Tixati or BitTorrent, by clicking on the **torrent** option. Torrent software must be downloaded separately. The **docs** option directs us to the documentation pages for each platform, and the **sum** option provides the `SHA256sum` output of the original file, which was uploaded by the creators.

> **Important note**
>
> The `sha256sum` command is used in Linux to generate a checksum or digital output representing the existing data, which can then be used to compare against the checksum of the downloaded copy to ensure that no data or bits were changed or tampered with.

Now that we're familiar with the versions of Kali Linux available to us, let's look at the software required for installing the various instances of Kali Linux.

Downloading the required tools and images

To be able to follow along with the installations of Kali Linux on various platforms in this chapter, you will be required to download the following versions of Kali as well as some additional tools, which I've listed here:

- **Kali Linux 64-bit Installer Image** (this can be installed as a standalone operating system or as a **Virtual Machine** (**VM**)): `https://cdimage.kali.org/kali-2022.3/kali-linux-2022.3-installer-amd64.iso`
- **Kali Linux 64-bit for VirtualBox**: `https://kali.download/virtual-images/kali-2022.3/kali-linux-2022.3-virtualbox-amd64.7z`
- **Kali Everything torrent**: `https://cdimage.kali.org/kali-2022.3/kali-linux-2022.3-installer-everything-amd64.iso.torrent`
- **Tixati torrent client software**: (to download the Kali Everything ISO image): `https://www.tixati.com/download/windows64.html`
- **7-Zip tool** (to unzip and decompress `.7z` files): `https://www.7-zip.org/download.html`
- **Rufus 3.2** (a tool for creating a bootable drive): `https://github.com/pbatard/rufus/releases/download/v3.20/rufus-3.20.exe`
- **VirtualBox and the VirtualBox Oracle VM extension pack**: `https://www.virtualbox.org/wiki/Downloads`
- **Raspberry Pi Imager** (for downloading and installing Kali on Raspberry Pi): `https://www.raspberrypi.com/software/`. When visiting the Imager downloads page, be sure to choose the correct platform. I'll be using Imager on a Windows machine, which can be downloaded at `https://downloads.raspberrypi.org/imager/imager_latest.exe`.

Downloading the Kali Linux Everything torrent

There is a variation of the Kali Linux Installer called Kali Everything. The Kali Everything image is significantly larger than the other installer versions as it contains all tools available for Kali Linux. Although these tools are available for manual download and installation in the form of repositories (often referred to as repos), it can be much simpler to download the Everything version and explore the tools if you are unfamiliar with installing Kali tools and repos. You can download the Kali Everything ISO image by performing the following steps:

1. Kali Everything can be installed directly to the hard drive as a standalone **operating system** (**OS**) and also as a VM within VirtualBox if you prefer this option instead. We'll cover both instances in the coming sections. The Kali Everything torrent can be downloaded by visiting `https://kali.download/base-images/kali-2022.3/kali-linux-2022.3-installer-everything-amd64.iso.torrent`.

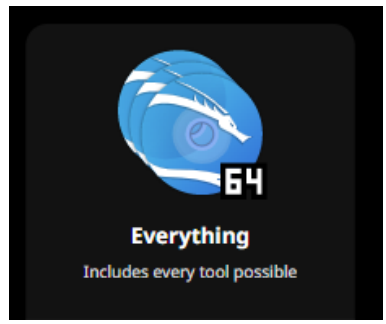


Figure 3.3: Image of the Kali Everything download button

2. You can then open that link with the torrent application of your choice. I used the **Tixati** software for Windows, which I downloaded from `https://www.tixati.com/download/` and then installed on my Windows 10 laptop. After clicking on the link, the Kali Everything torrent file opens in the Tixati client. I've chosen to save the Kali Everything file to my `Downloads` folder, as seen in the following screenshot. After choosing your download location, click on the **Start** button.

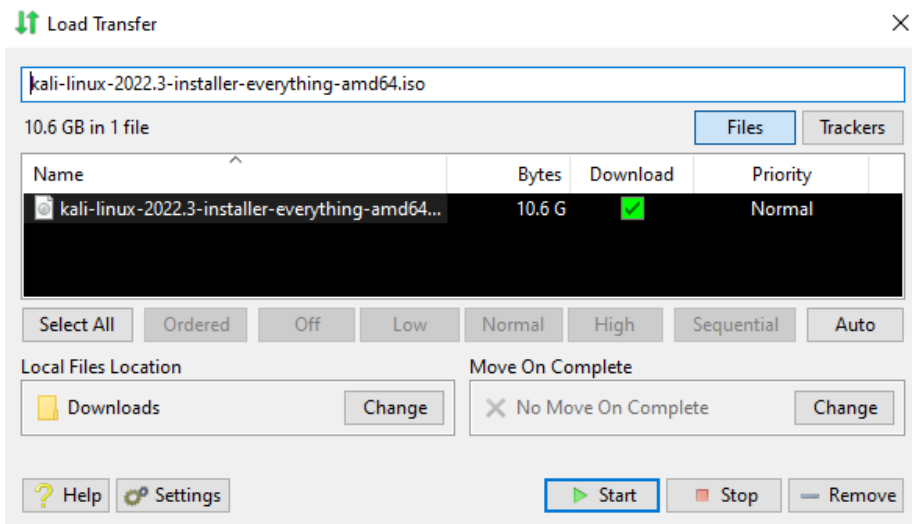


Figure 3.4: The Tixati torrent interface

3. The Kali Everything file download will begin in Tixati and will be saved to the selected folder with the `kali-linux-2022.3-installer-everything-amd64.iso` filename. We will be using this ISO file in the coming sections.

   You should now have at least two versions of Kali Linux downloaded to your system, as I do in my `Downloads` folder.

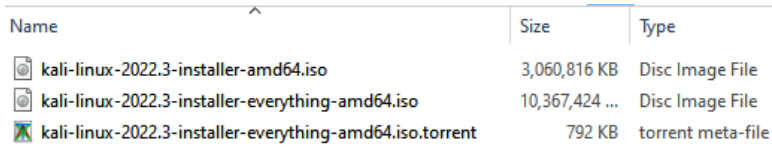


Figure 3.5: Downloaded Kali images in the Downloads folder

Let's now look at creating bootable versions of these two instances of Kali Linux on flash drives or any other storage media type using Rufus 3.2.

Installing Kali Linux on portable storage media for live DFIR

As mentioned previously, Kali Linux can be installed on a variety of platforms and can be installed either as a standalone OS or as a VM. We'll first look at creating a bootable Kali Linux flash drive using the Kali Installer ISO file we downloaded. If you did not download ISO files, you can do so now by clicking on the links and following the instructions in the previous section.

This `.iso` file, or image file as it is sometimes called, can be used with Rufus 3.20 to create bootable drives from which to install Kali Linux onto a PC or laptop. It can also be installed and used within VirtualBox as a VM, which we will get into using the Kali Everything ISO file later in this chapter. Let's first look at how to create a bootable flash drive using the Kali Linux Installer ISO and then learn how to install Kali as a standalone OS.

Creating a bootable flash drive or SD card with Kali Linux can be very useful as it eliminates the need to carry around your personal laptop or desktop with Kali Linux on it as Kali can be run from the portable storage media device itself once inserted into and booted from a desktop or laptop.

For this lab, (*we will refer to practicals as labs from time to time*), I'll be using the Rufus 3.20 tool on a Windows 10 laptop to create the bootable flash drive using **Kali Installer ISO**, which we downloaded. If you haven't downloaded Rufus 3.20, please do so now by clicking on `https://github.com/pbatard/rufus/releases/download/v3.20/rufus-3.20.exe`.

Rufus is very user-friendly and has a simple interface which is one of the reasons why I've chosen it to create our bootable drive. Once Rufus has been downloaded, locate the downloaded file and double-click on it to run Rufus. At this point, you should have the flash drive you wish to use as your Kali Linux bootable drive inserted into your computer or laptop. In this instance, I'm using a 32 GB flash drive that I've labeled as CFSI for easy recognition.

Have a look at the following screenshot and observe the following items:

- **Device**: This is your target device. I've selected my 32 GB drive.
- **Boot selection**: We will be creating our bootable drive using the Kali Installer and Kali Everything `.iso` files that we downloaded.
- **Partition scheme**: **MBR** (older devices) or **GPT** (newer devices). I've selected **MBR** as I'll be creating this bootable Kali drive to run on older systems. If you are only using this bootable drive on newer laptops or desktops, you may want to choose **GPT**. You can also create two bootable drives, one with an MBR partition scheme and another with GPT if you have available drives.
- **File system**: Many flash drives and SD cards are formatted as FAT32 by default and work without issues, so we'll leave it selected as **FAT32 (Default)**.

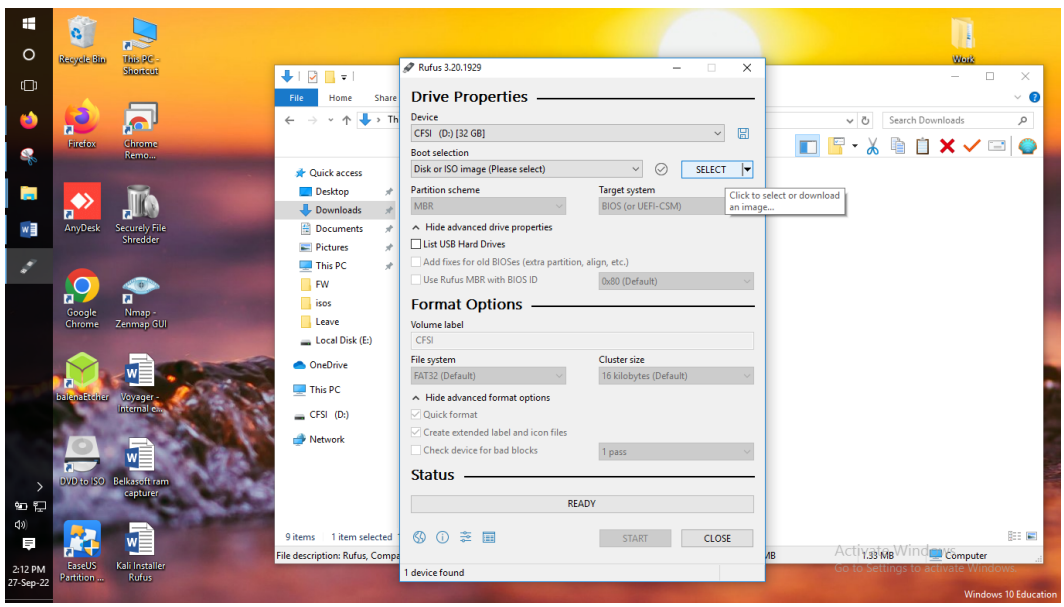


Figure 3.6: The Rufus interface

Now follow these steps for installing Kali Linux on portable storage media for live DFIR:

1. In the preceding screenshot, you will notice that there is a screen tip to the right that reads, **Click to select or download an image**. Click on the **SELECT** drop-down menu and browse to the location where you downloaded and saved the Kali Installer (`kali-linux-2022.3-installer-amd64.iso`) file.

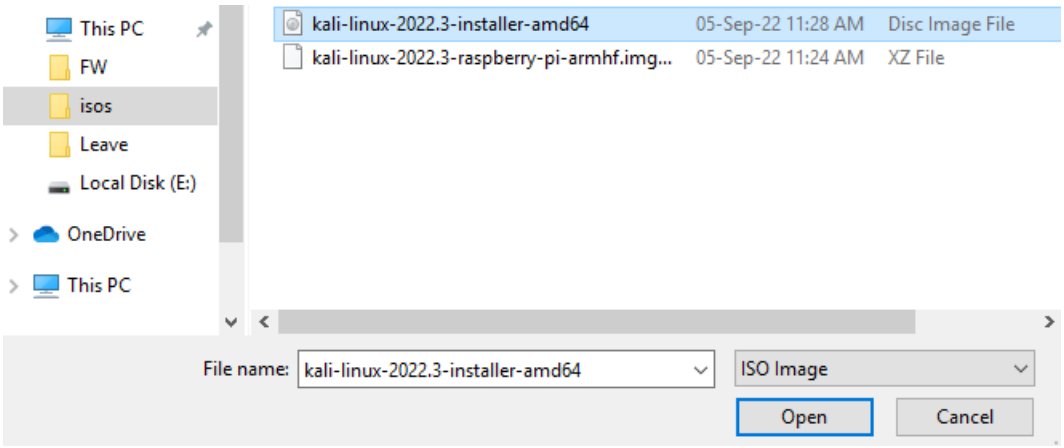


Figure 3.7: Selecting the Kali ISO image in Rufus

2. I've saved the `kali-linux-2022.3-installer-amd64.iso` file to my `isos` folder within the `Downloads` folder. Click on the `kali-linux-2022.3-installer-amd64.iso` file to select it, and then click on the **Open** button.

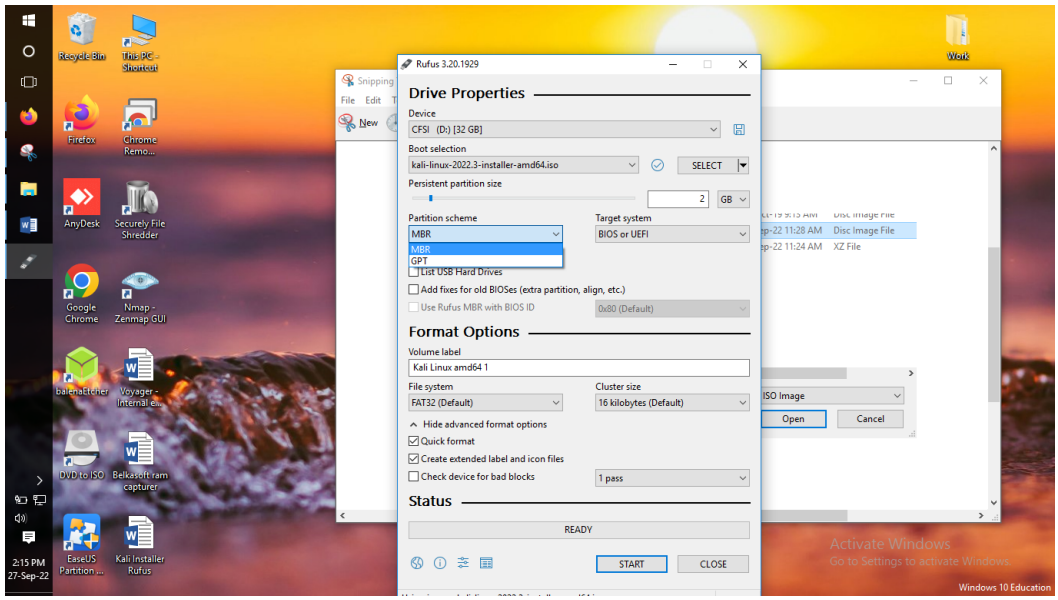


Figure 3.8: Choosing the Partition scheme in Rufus

As seen in the preceding screenshot, you should now notice `kali-linux-2022.3-installer-amd64.iso` listed under the **Boot selection** section. I've also set **Persistent partition size** to `2` **GB**, which creates a space to act as storage in the event I need to download files or install tools within the bootable installation of Kali.

3. At this point, you may again choose between the **MBR** or **GPT** options in the **Partition scheme** field. I've also changed the **Volume label** option to read `Kali Linux amd 64 1`. The reason for using the number `1` at the end of the volume label is that I've used this drive with the **MBR** partition scheme, and I've created another bootable drive using the **GPT** partition scheme, which is labeled `Kali Linux amd 64 2`.
4. Once you have configured all the options as you would like them and ensured that all settings are correct, click on the **START** button.
5. You may be prompted to download two additional files named `ldlinux.sys` and `ldlinux.bss`. Click on the **Yes** button to allow Rufus to do so automatically. Your computer must be connected to the internet for Rufus to download these files.

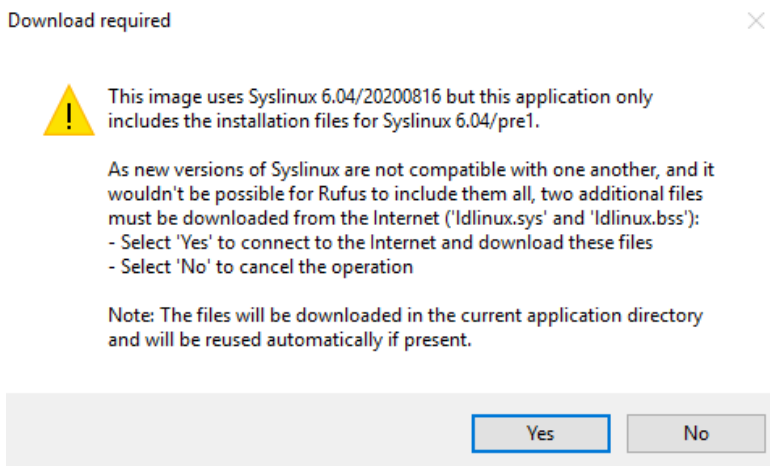


Figure 3.9: The Syslinux download requirement confirmation screen

6. Rufus will then ask for confirmation and warns that all data on the removable storage device will be destroyed. If you are certain that you have selected the correct drive, click the **OK** button to proceed.

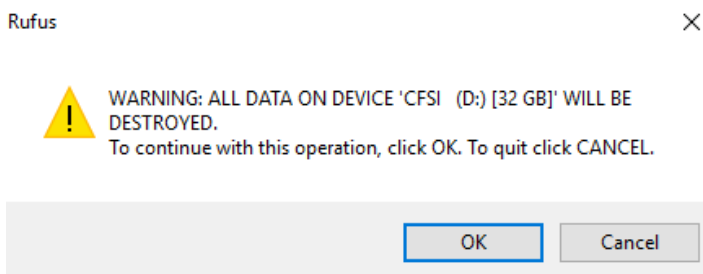


Figure 3.10: The operation confirmation screen

Rufus will then begin formatting the drive, as seen in the following screenshot.

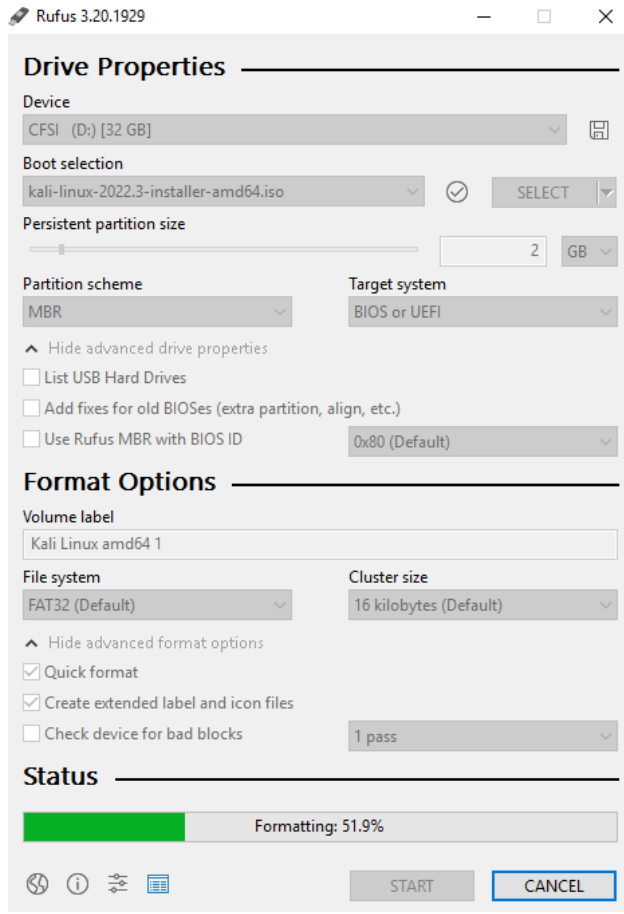


Figure 3.11: Rufus formatting operation status

Once formatting is complete, Rufus will then copy the ISO file to the drive.

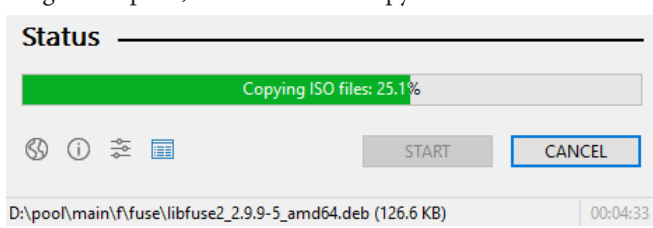


Figure 3.12: Rufus file copying status

7. When the process is complete, the **Status** bar will appear green with the word **READY** in the middle of the bar. You can now click on the **CLOSE** button, as seen in the following screenshot.

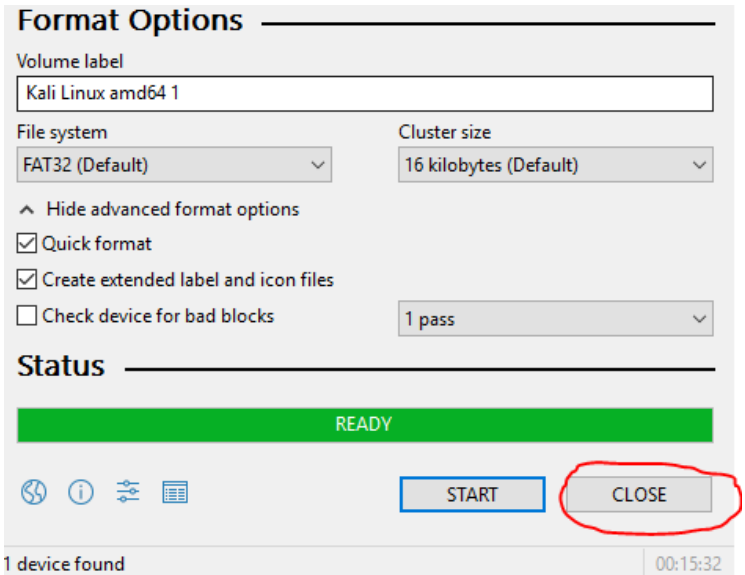


Figure 3.13: Bootable drive creation completed

8. You can now remove your bootable Kali Linux drive and place it into a computer or laptop on which you wish to run Kali as a live OS or install it on the hard drive.

   Running Kali as a live OS means that it does not install on the hard drive and runs using the device's **Random Access Memory** (**RAM**). This way, we can preserve the original OS and use Kali and all the tools within for **Digital Forensics and Incident Response** (**DFIR**) while leaving the current installed OS intact.

9. To do so, you insert the bootable Kali Linux drive into the desktop or laptop and reboot the system. You may have to press the *F12* key upon reboot to choose the flash drive as the bootable OS. This tells the **Basic Input Output System** (**BIOS**) to boot from the removable storage media and not the hard drive, thus allowing us to run Kali Linux within RAM as a live OS.

Now that we have learned how to install Kali Linux on a bootable flash drive, we will learn how to install Kali as a standalone OS on a desktop or laptop device using the bootable flash drive.

Installing Kali as a standalone operating system

In this section, we will be installing Kali Linux onto our desktop or laptop as a standalone operating system. For this exercise, we will be using the bootable Kali Installer flash drive, which we just created using Rufus 3.2. If you've ever booted a desktop or laptop from a flash drive or any removable storage media, you may already be familiar with the process. For anyone new to the process, let's take a step-

by-step look at booting into Kali Linux. The installation and configuration of Kali Linux are the same (if using the Kali Linux Installer version, which we are using) when installing Kali onto a physical machine and a VM. So, we'll look at the installation itself in the following section to keep things consistent:

1. Once your bootable Kali Linux Installer flash drive has been created, insert the drive into your desktop or laptop and power on the device. If your device is already on, restart the device with the flash drive inserted in the USB port.
2. As soon as the device powers on, press the *F12* key on the keyboard for **Advanced Boot Options**. This key and setting may vary on different machines, and you may have to refer to the device manual if *F12* does not carry you to the boot menu.
3. Select the USB or removable drive as the main boot device and press the *Enter* key.

   You should now be presented with the Kali boot menu as seen in the photo I took of my monitor.

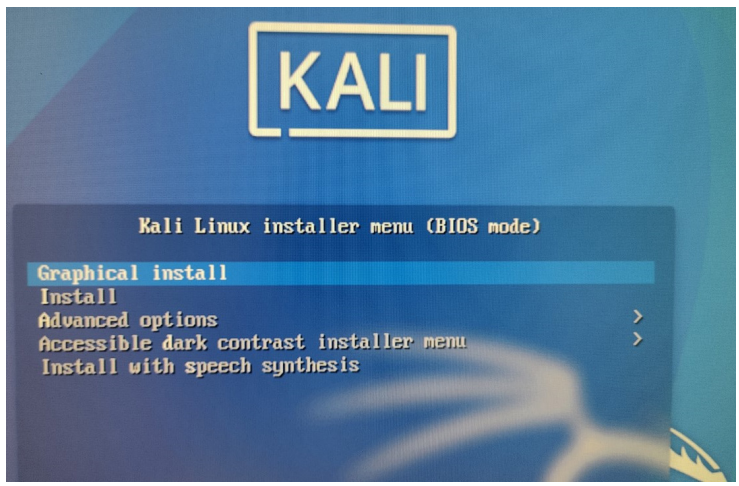


Figure 3.14: Picture of the Kali boot screen on a desktop pc

Once here, you can refer to the following *Installing Kali Linux in VirtualBox* section to continue the installation and configuration process. Let's pause here in our standalone installation and learn how to install Kali Linux as a VM in VirtualBox. Once we've installed Kali in VirtualBox, we'll then resume installation and configuration, which can be applied to both standalone and virtual systems.

Installing Kali in VirtualBox

The creators of Kali Linux (Offensive Security) have been kind enough to simplify our lives by creating a pre-configured version of Kali for VirtualBox and VMware, which requires minimal configuration. These pre-configured images and even the installer images, including Kali Everything, can all be installed on virtual platforms and run no differently than they would on physical machines. The advantage of using Kali Linux as a VM is that you do not have to purchase any new or additional hardware. You can also delete and reinstall the VMs very easily and even create a snapshot of the OS

once you've configured exactly as you would like it, which you can boot into at any time if you've broken your Kali machine and need to revert to a saved state.

For those of you who are installing Kali as a standalone OS, you can continue the installation after booting from your bootable Kali flash drive in *Step 3* in the *Installing Kali as a standalone operating system* section, as the steps for installation of the Kali Linux Installer on both physical and VMs are exactly the same.

Before we begin, please ensure that you have already downloaded the latest stable version of VirtualBox and the VirtualBox additions. If you haven't, you can do so now by visiting `https://www.virtualbox.org/wiki/Downloads`. Please download and install the appropriate VirtualBox platform packages and the VirtualBox Oracle Extension Pack. I'm using a Windows 10 host, so I'll be downloading and VirtualBox platform for Windows by clicking on **Windows hosts**.

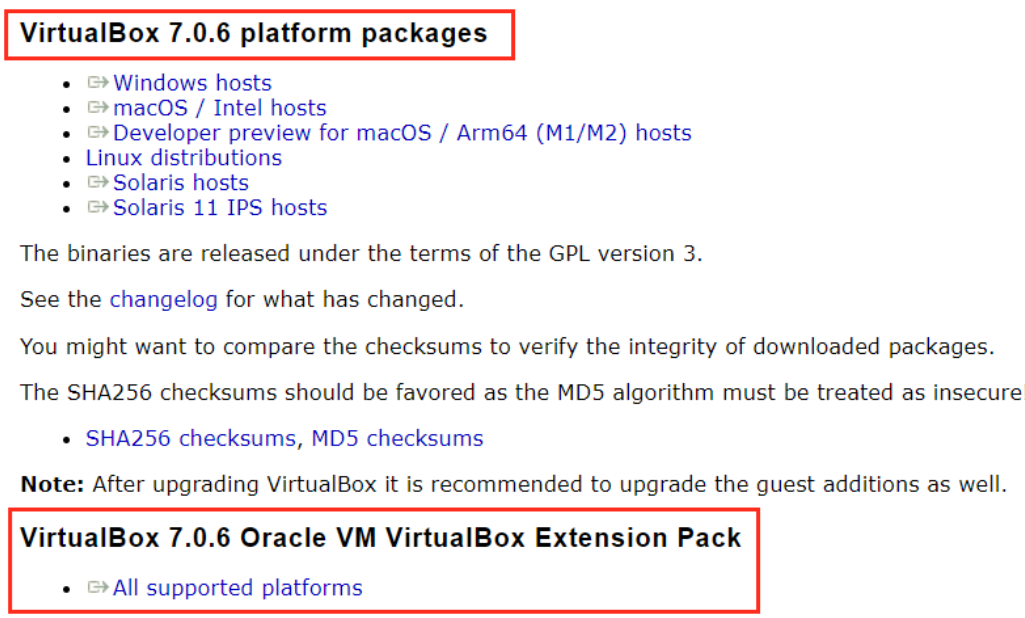


Figure 3.15: The VirtualBox website downloads page

Preparing the Kali Linux VM

Once VirtualBox has been downloaded, it can be installed and then configured to run Kali Linux and many other OSs, depending on the amount of RAM available:

1. When setting up a new guest OS or guest VM, we first click on **New** and then fill in the following details:

    - **Name**: `Kali Linux 2022.3 Everything` (or name of your choice)

Note

For this installation, I am using the large Kali Linux Everything ISO version, as it comes with every tool installed. You may also use the Kali Installer ISO, as the installation is exactly the same for both versions.

- **Machine Folder**: Default location for the VM
- **Type**: **Linux**
- **Version**: **Debian (64-bit)**

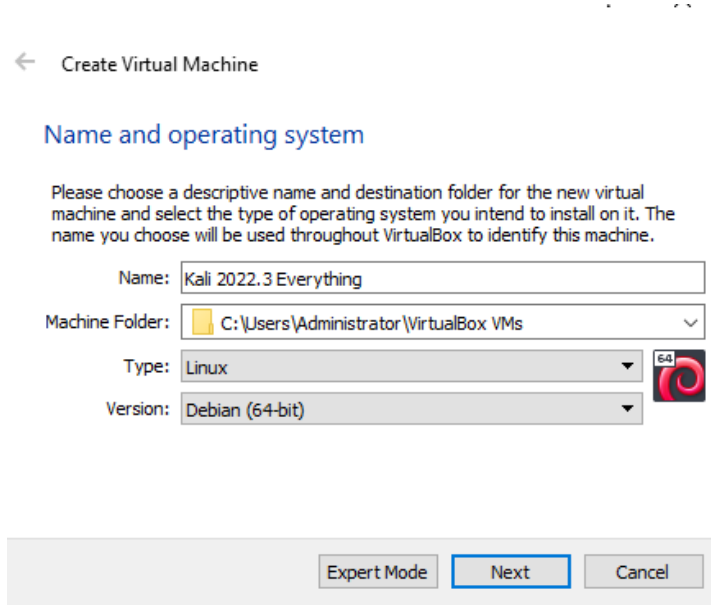


Figure 3.16: VirtualBox name and OS details

2. We then click **Next** and proceed to allocate RAM in the **Memory size** slider prompt. On my screen, you'll notice that I've selected 10 GB of RAM for the VM. I've chosen this amount as I have 32 GB RAM in total. It should be noted that when choosing the amount of RAM for our VMs, we do not exceed more than half the total amount of RAM or exceed an amount that shows as pink in the slider window, as doing so may cause our host and virtual systems to become unstable or crash. I do recommend at least 4 GB of RAM for smooth functionality when using the forensic tools.

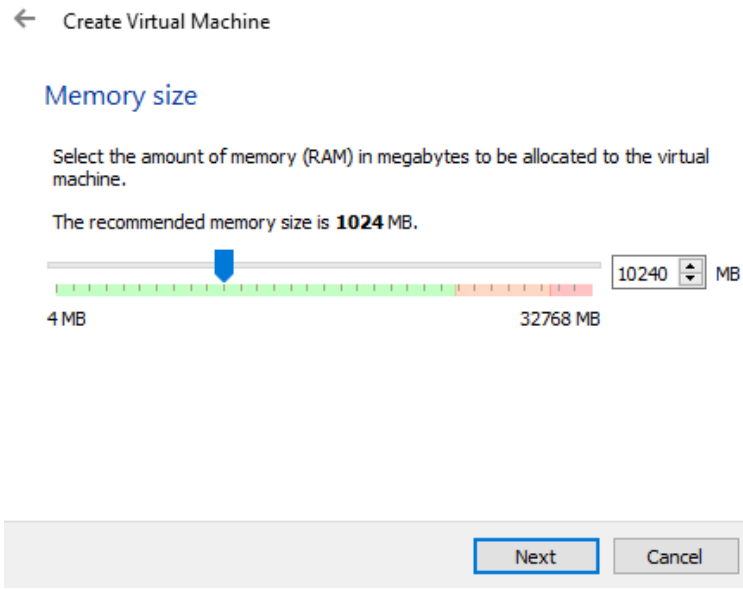


Figure 3.17: VirtualBox memory allocation

3. Next, we create the VM by adding a virtual hard disk. I recommend starting with a new virtual hard disk, which is the second option in the selection. Click on **Create** to proceed:

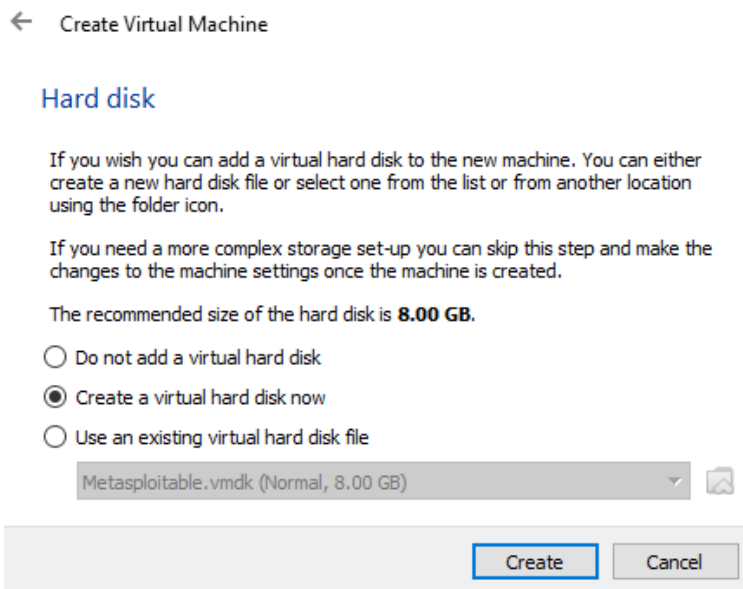


Figure 3.18: VirtualBox hard disk creation

4. Then choose **VDI (VirtualBox Disk Image)** as **Hard disk file type** and click **Next**:

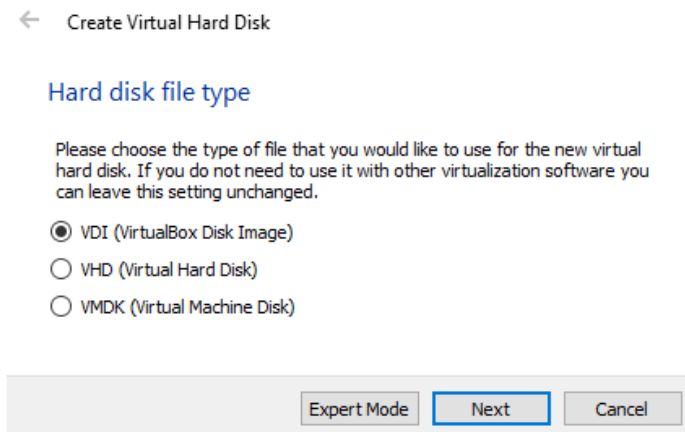


Figure 3.19: VirtualBox hard disk file type selection

5. Once the VDI has been selected, choose the **Dynamically allocated** option to allow the virtual hard disk to be expanded if the need arises:

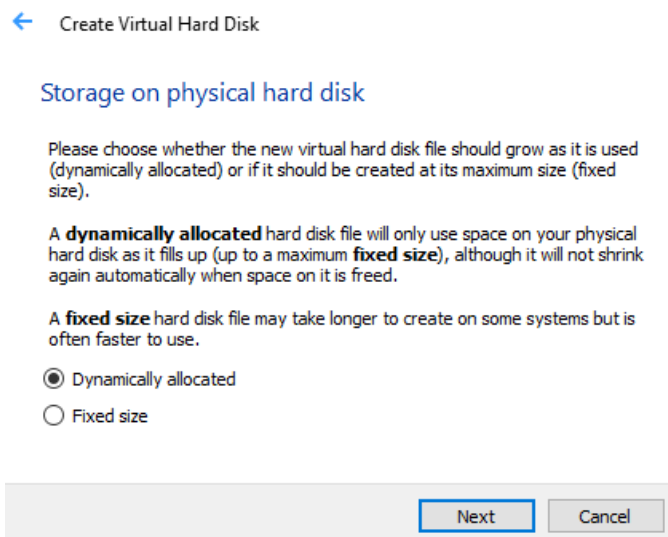


Figure 3.20: VirtualBox storage allocation

6. For the next step, we select the file location and the size of the virtual hard disk chosen. The recommended size for the Kali Linux VDI is 8 GB, but I've assigned an ample `32.00` GB. I've also chosen the location for the VDI as `C:\Users\Administrator\VirtualBox VMs\Kali 2022.3 Everything\Kali 2022.3 Everything`.

7. Once the location is selected, click on **Create** to complete the creation of the virtual hard disk.

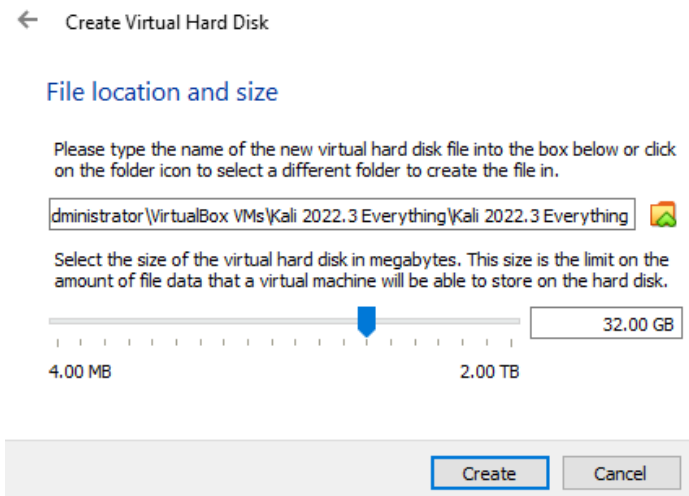


Figure 3.21: VirtualBox storage allocation size

Now that we have set up the Kali Linux ISO image in VirtualBox, we can begin the installation process, which will perform in the next section.

Installing Kali Linux on the virtual machine

Once the virtual hard disk has been prepared and completed by following the steps in the previous subsection, we can then begin the Kali Linux installation process. In the **Oracle VM VirtualBox Manager** window, which is the main OS management window for VirtualBox, we can see that the VM has been prepared, and we can now install Kali Linux:

1. To the right of the screen, you can see the resources assigned, such as **Name** and the **Operating System** type in the **General** section and the amount of RAM assigned in the **System** section. Other settings, such as the **Video RAM** (**VRAM**), **Network**, and **Display** settings, can also be accessed within this section.

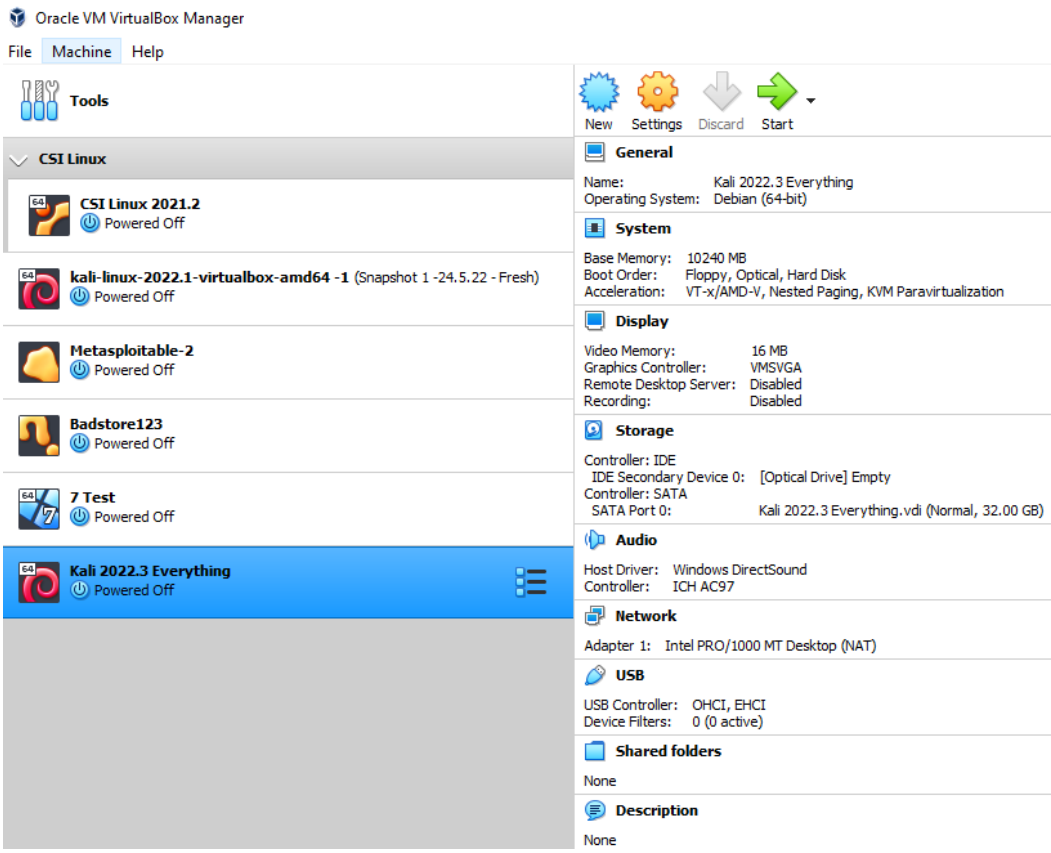


Figure 3.22: VirtualBox Manager

2. To begin our Kali Linux installation, click on the **Kali 2022.3 Everything** entry to the left and then click on the green **Start** arrow in the top right corner of the **Select start-up disk** section, as seen in the following screenshot.

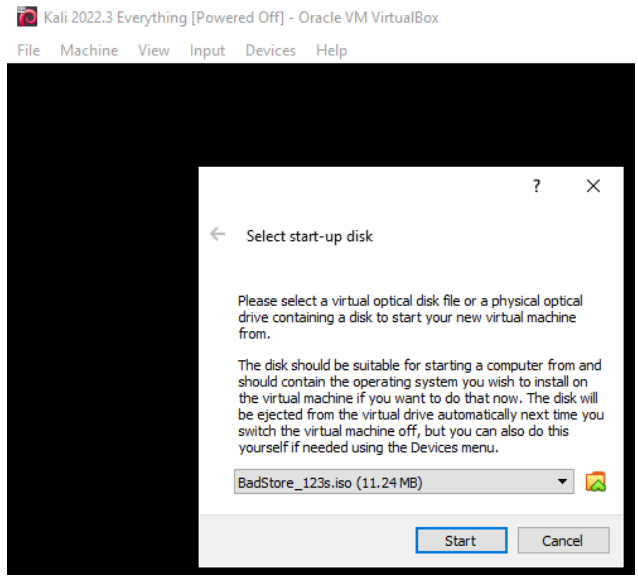


Figure 3.23: Start-up disk selection

3. In the next step, locate the Kali Linux Everything ISO image that we downloaded from the Offensive Security website. Click on the browse folder icon, which carries us to **Optical Disk Selector**, and then click on the **Add** icon at the top-left of the box.

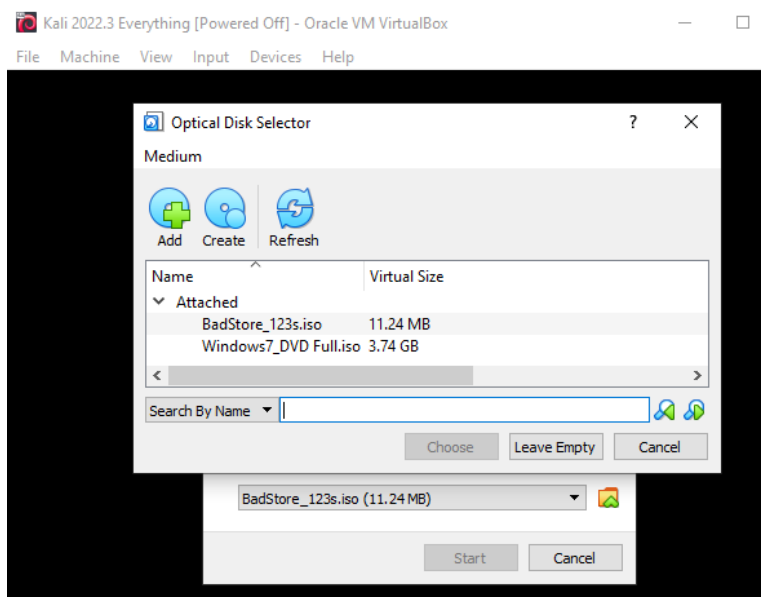


Figure 3.24: Optical disk selection fields

4. Navigate to the `Kali-linux-2022.3-installer-everything-amd64.iso` file you previously downloaded and click on it and then click on the **Open** button to see the ISO listed as an entry in **Optical Disk Selector**.

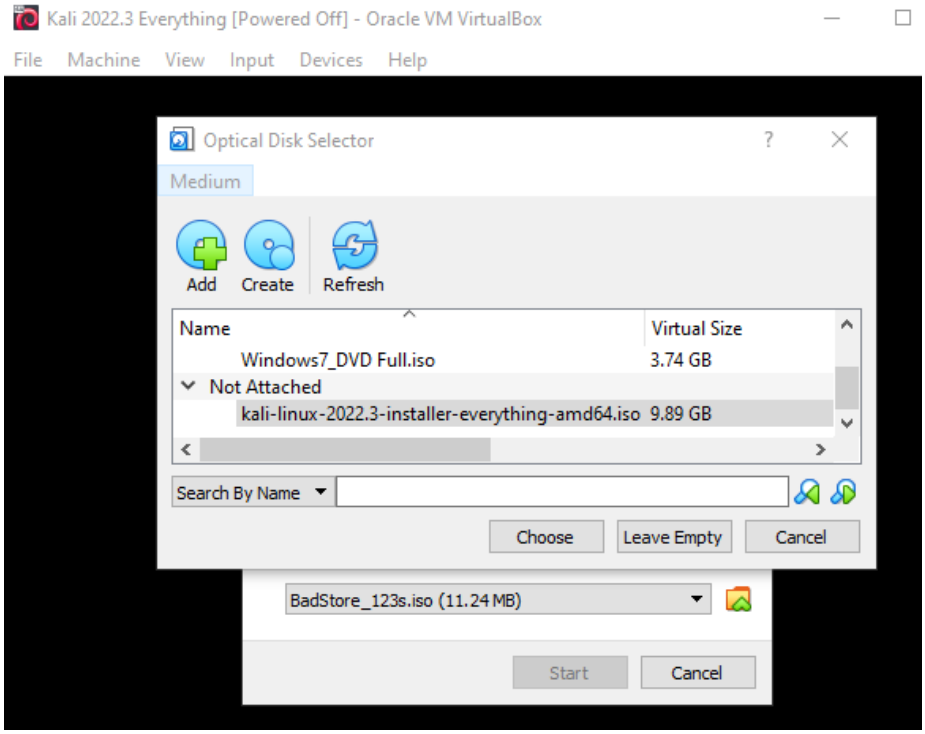


Figure 3.25: Selecting the Kali ISO

5. Click on the **Choose** button to continue and then click on the **Start** button in the **Select start-up disk** box once the `Kali-linux-2022.3-installer-everything-amd64.iso` file is selected, as seen in the following screenshot.

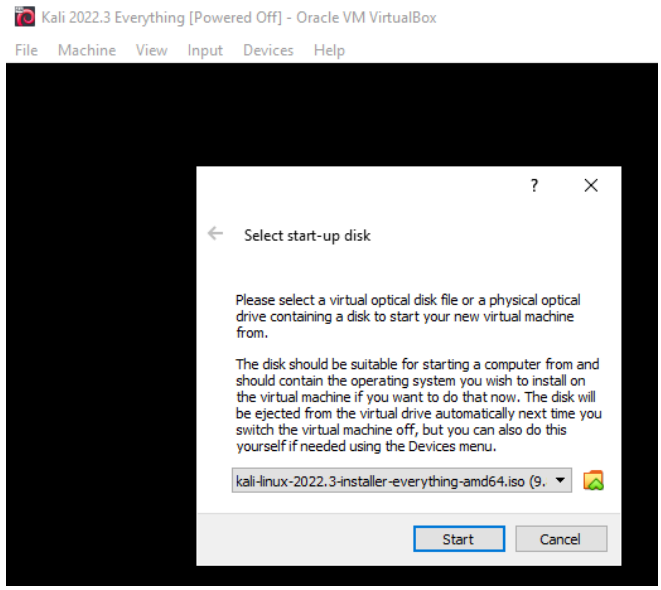


Figure 3.26: Selecting Kali ISO as the start up image

You should now see the Kali Linux Installer screen and menu options that you also saw in the previous section: *Installing Kali Linux as a standalone OS*.

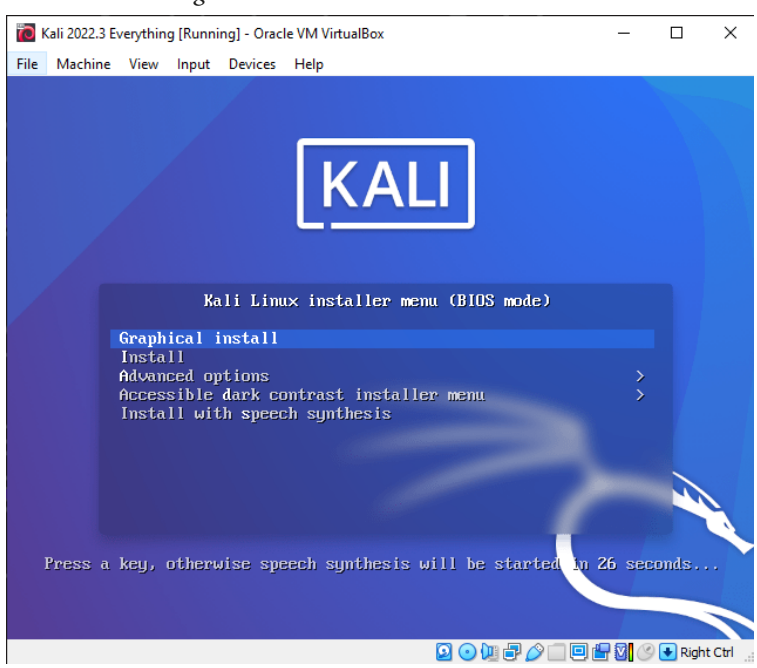


Figure 3.27: Kali boot menu in VirtualBox

> **Important note**
>
> For those of you installing Kali Linux as a standalone OS, you can also follow the steps in the *Installing and configuring Kali Linux* section, as they are exactly the same for both physical and virtual installations.

We're all set with Kali Linux configured and ready to be installed on both our standalone and virtual installations. Let's learn how to install and configure Kali in the next section.

Installing and configuring Kali Linux as a virtual machine or as a standalone OS

Now that we're at the Kali Linux installation screen, we can follow the steps below, which will be identical whether installing and configuring Kali as a VM or as a standalone OS:

1. To begin the installation, select the **Graphical install** option and press the *Enter* key. Again, whether installing Kali Linux as a standalone or VM, you will be presented with the same options as seen in the following screenshot.

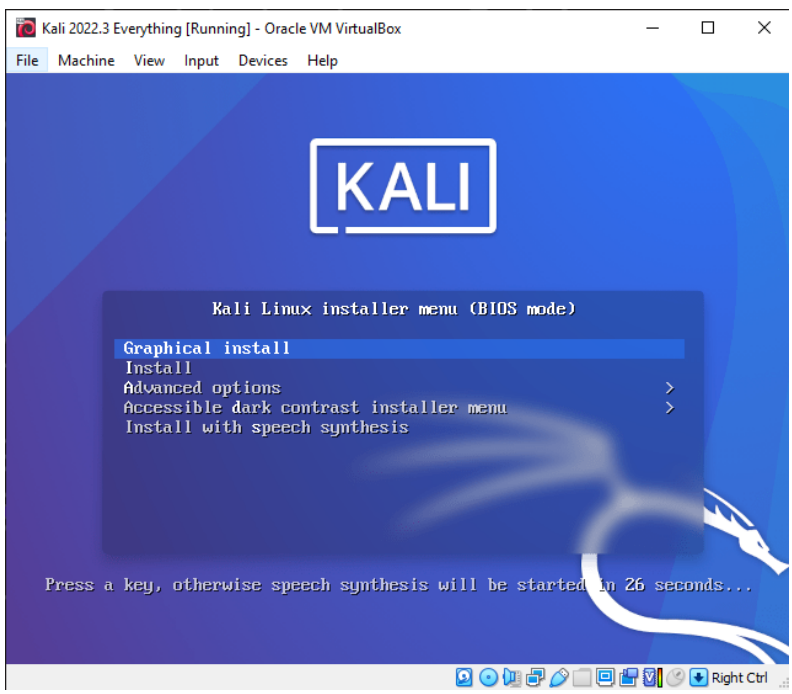


Figure 3.28: Graphical install option in the Kali boot menu

2. Next, select your language and click on **Continue**.

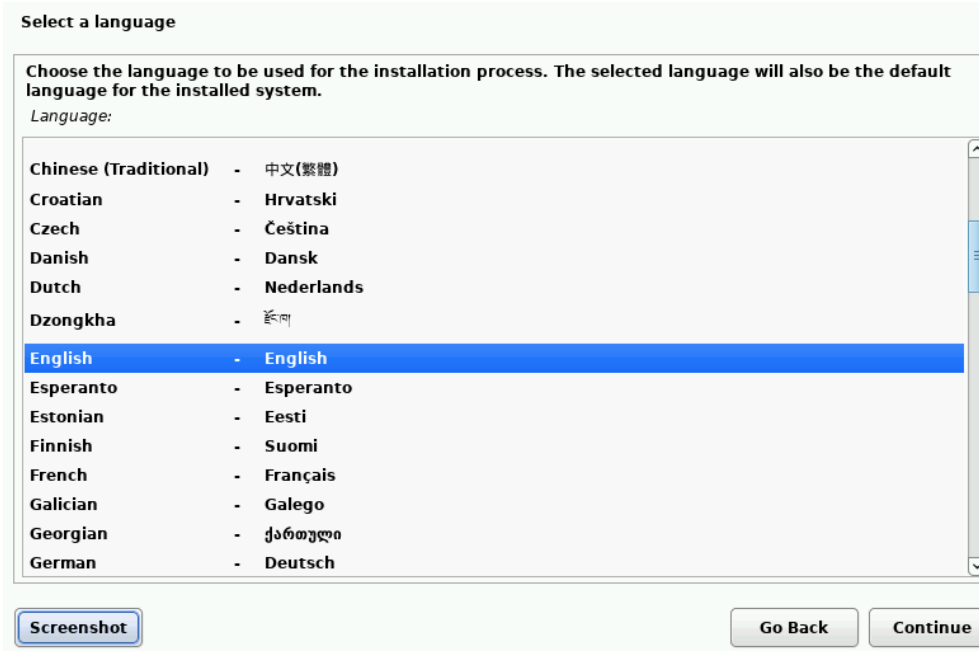


Figure 3.29: Kali language selection menu

3. Select your county or region and click on **Continue**.

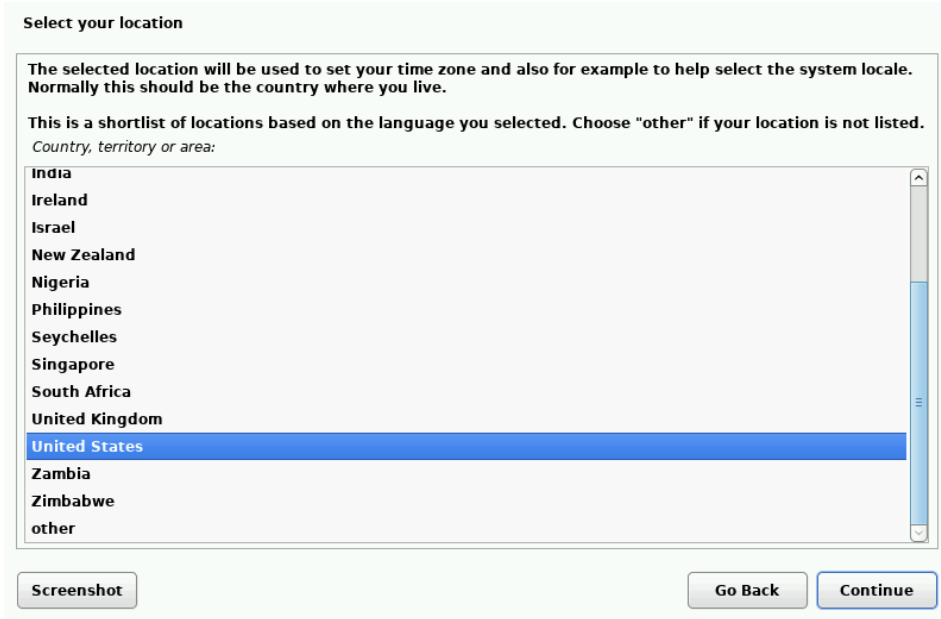


Figure 3.30: Country selection menu

4. Choose your keyboard configuration and click on **Continue**.

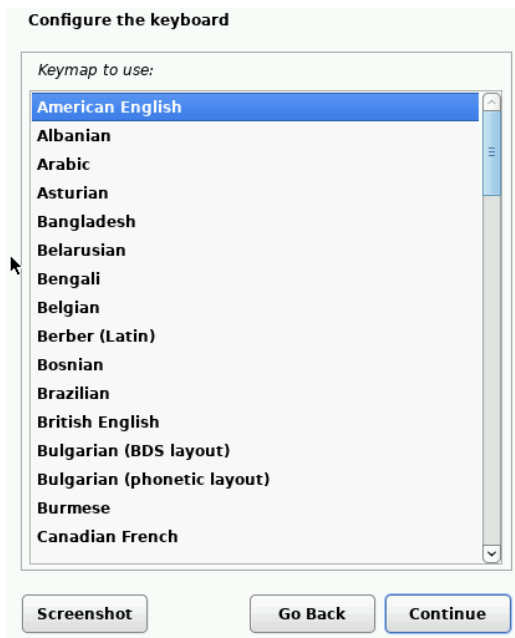


Figure 3.31: Language selection menu

5. Give your system a hostname or leave the default name, `kali`.

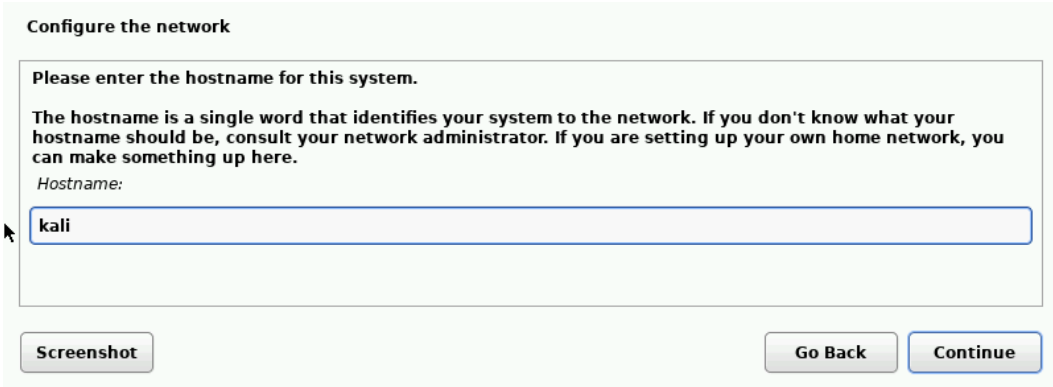


Figure 3.32: Assigning a hostname for the Kali system

6. If the machine will be joining a domain, you can enter it here, or leave it blank if not joining a domain, and click **Continue**.

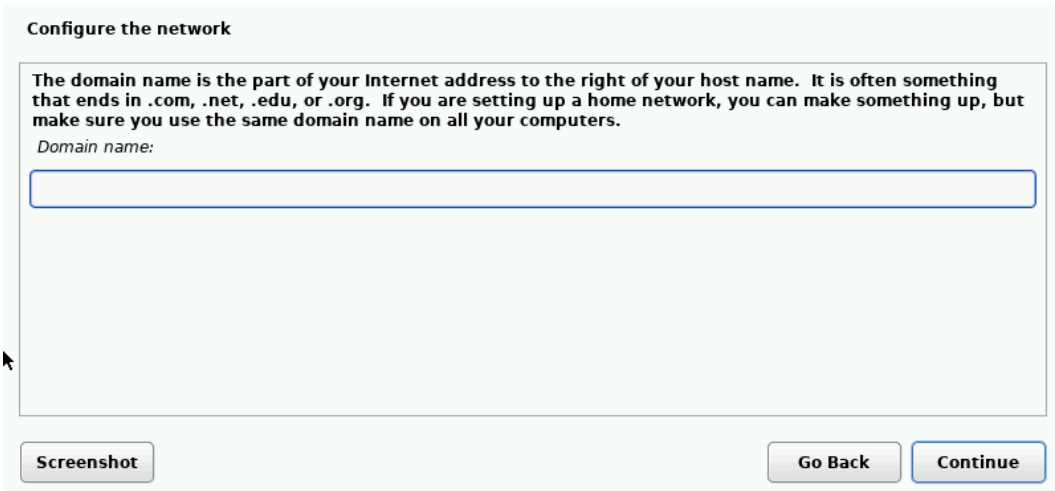


Figure 3.33: Configuring the network

7. Next up, we'll create a non-root user account that does not have root or administrator privileges. Please be sure to use lowercase letters for the name.

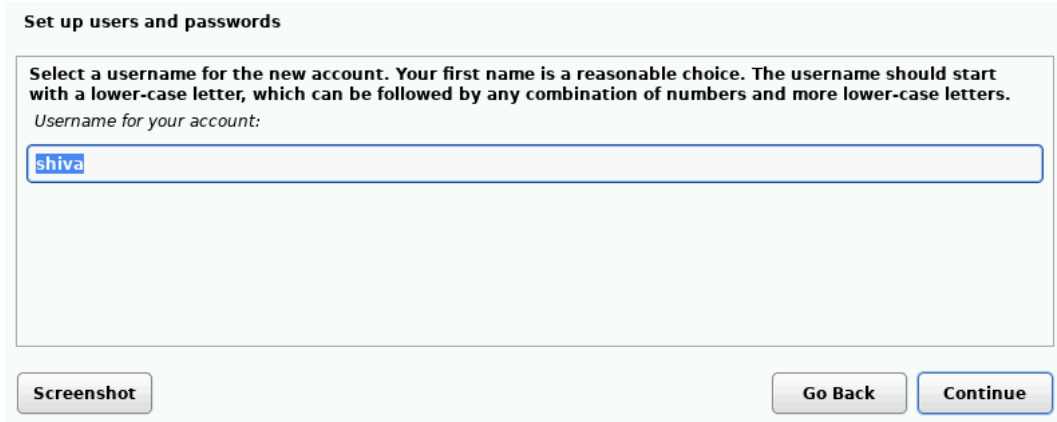


Figure 3.34: Username creation

8. Next, create a secure and complex password, then click on **Continue**.

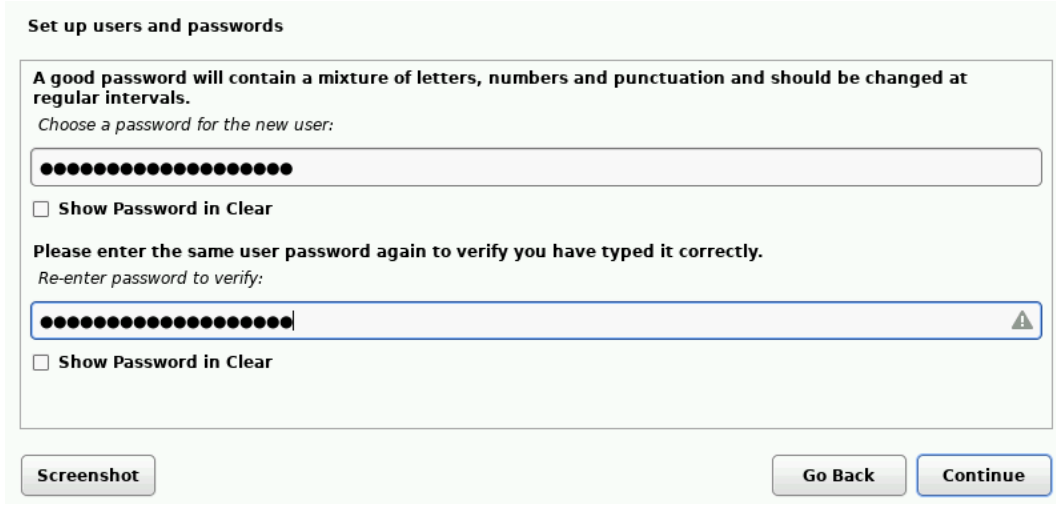


Figure 3.35: Password creation

9. Choose your timezone and click on **Continue**.

Figure 3.36: Timezone configuration

10. Now that our personalized settings have been configured, let's prepare the hard disk for installation. For simple installation, I recommend using the entire hard disk for Kali, whether virtual or physical. To do this, select the **Guided – use entire disk** option and click **Continue**.

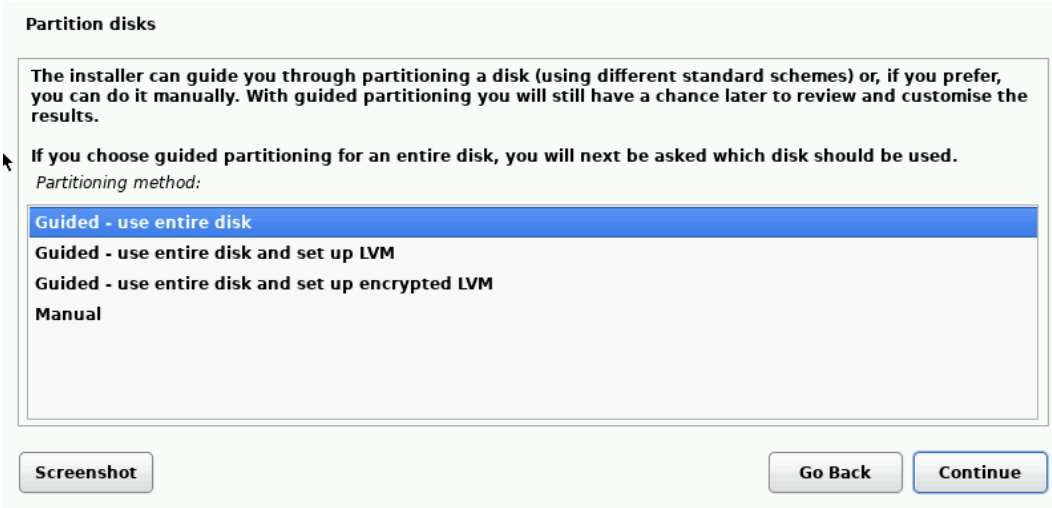


Figure 3.37: Disk partitioning options

11. Choose the disk for installation, which in my case is a 34.4 GB virtual hard disk, and click on **Continue**.

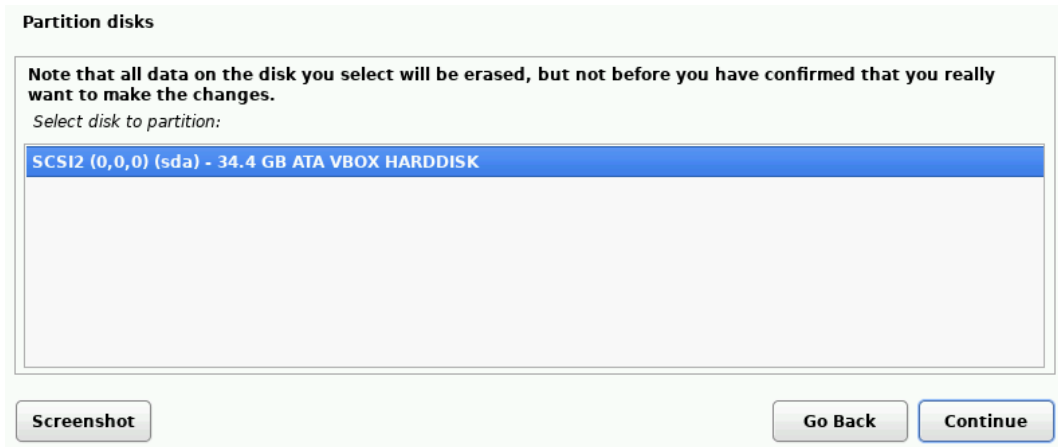


Figure 3.38: Disk partition selection

12. To keep things simple, I recommend leaving all files in one partition. Select the **All files in one partition (recommended for new users)** option and click on **Continue**.

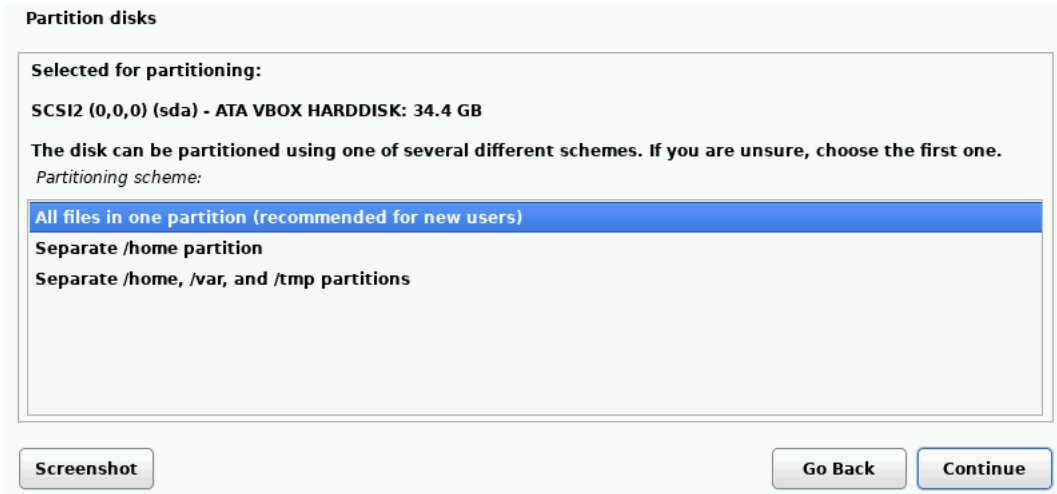


Figure 3.39: File partitioning

13. Review your disk partition settings and select the **Finish partitioning and write changes to disk** option and click on **Continue**.

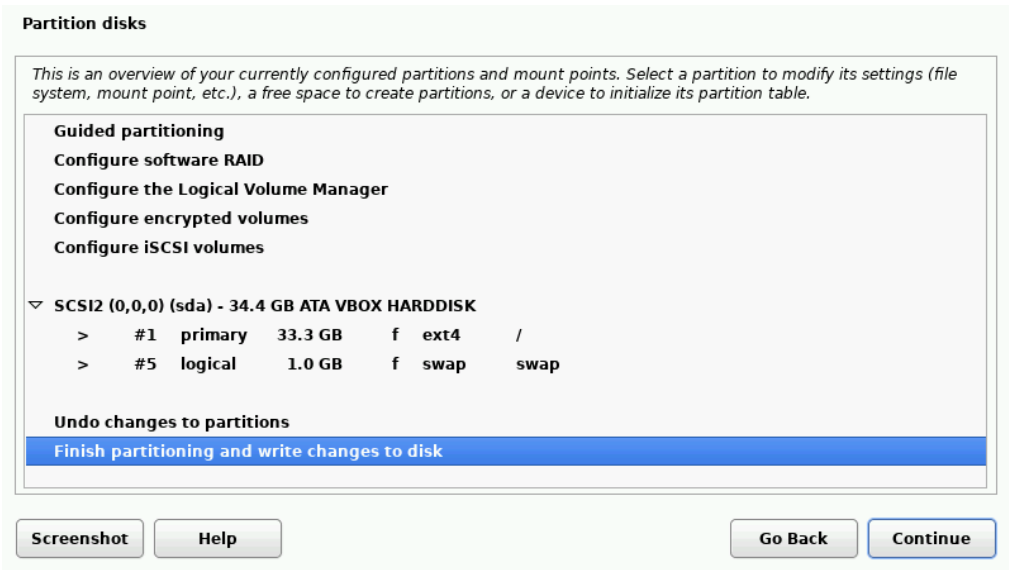


Figure 3.40: Writing changes to the disk

14. To confirm the partition and formatting settings and begin writing the changes to the disk, select the **Yes** button and click on **Continue**.

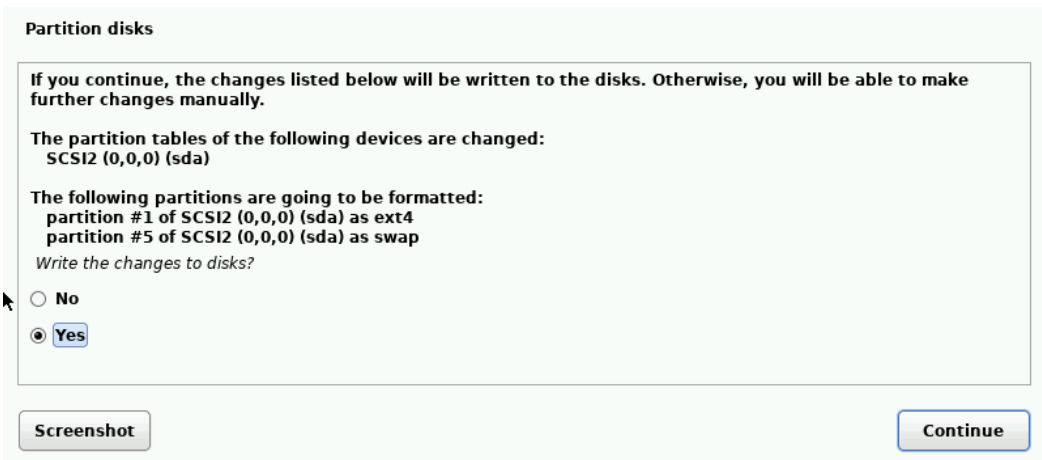


Figure 3.41: Finishing the partition process

Partitioning and formatting will begin, and the installation process will now continue with the extraction of required packages.

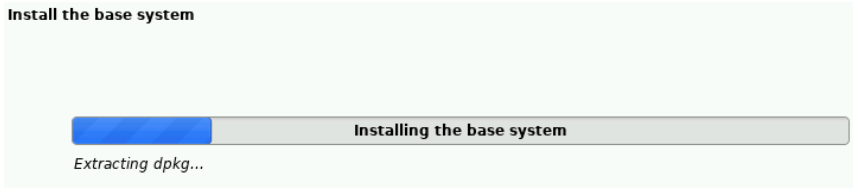


Figure 3.42: Base system installation process

15. We can now select various software packages and repositories to be automatically added to Kali. I've chosen to have all tools and supporting software and packages in my installation and have selected all checkboxes except for the **GNOME** and **KDE Plasma** options, as I prefer the typical Kali Linux interface with all possible tools.

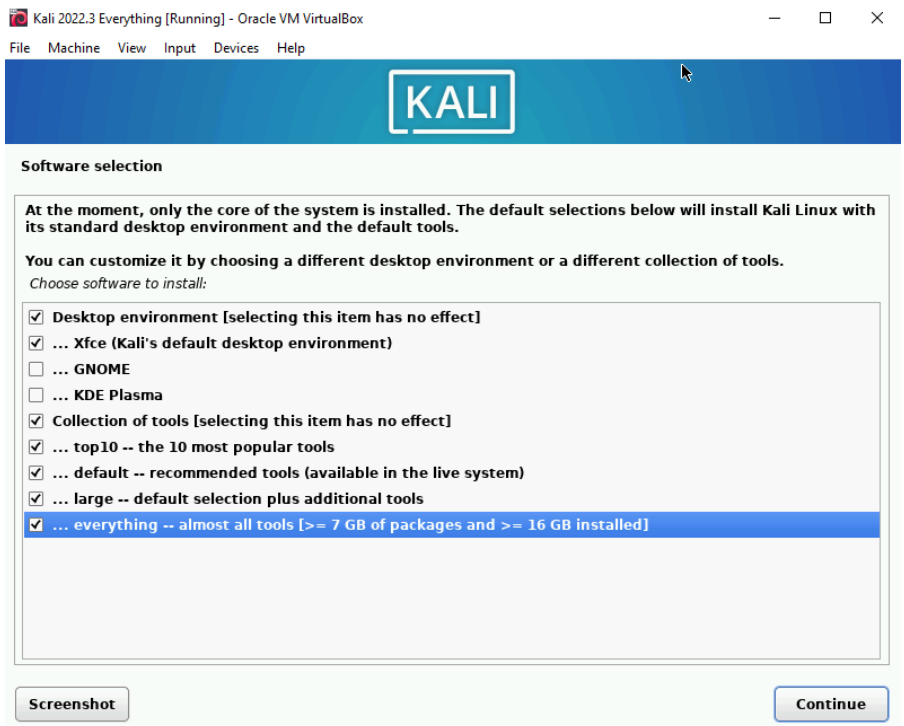


Figure 3.43: Additional software selection

16. Once you have selected your choice of software, click on **Continue** to continue with the installation. This process will take some time. In the meantime, feel free to browse the large selection of other brilliant digital forensics titles at `https://subscription.packtpub.com/search?category=Security&concept=Forensics&_ga=2.265355399.920003428.1665758595-170131085.1665758595`.

Figure 3.44: Software installation status

17. Next, you can choose a display manager or leave the default **gdm3** option if you are unfamiliar with the settings.

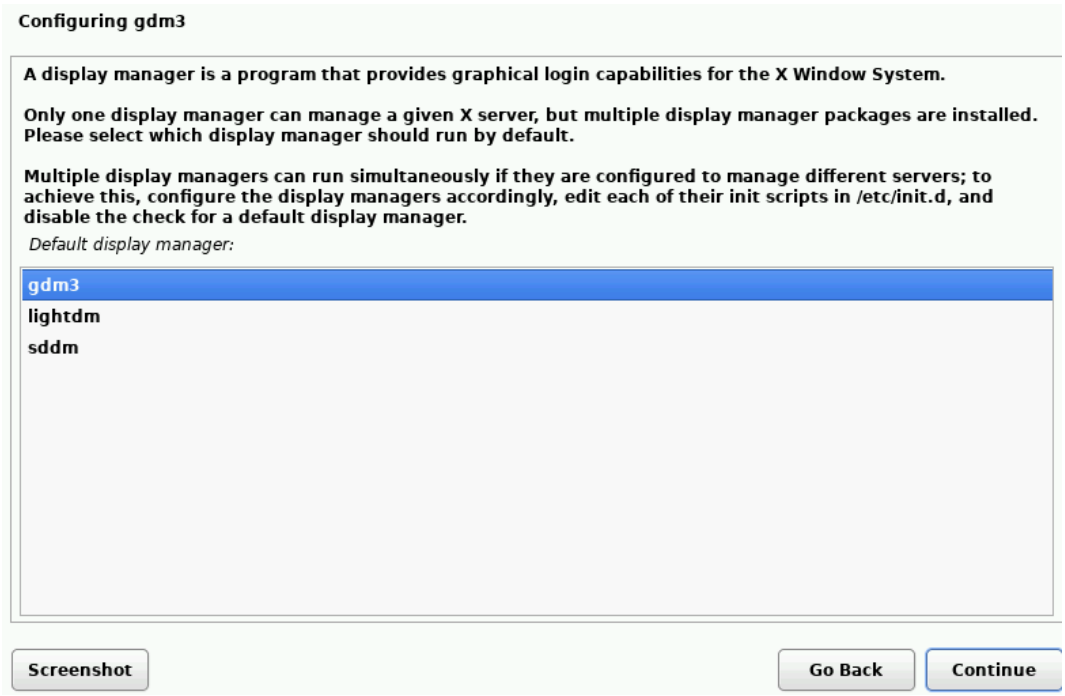


Figure 3.45: Graphical display manager selection

The installation will continue as indicated by the status bar.

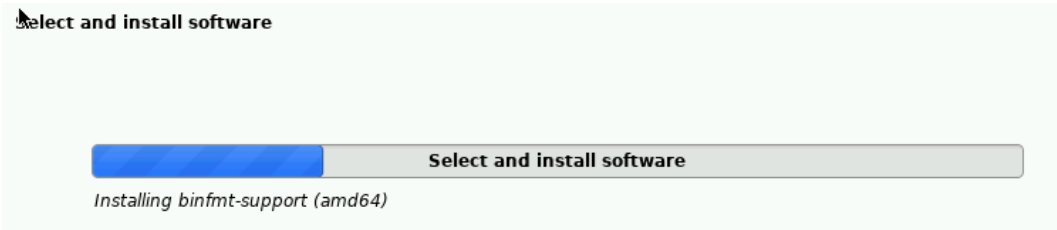


Figure 3.46: Software installation progress bar

18. Just a few more steps, and we'll be done. We now move on to the GRUB boot loader, which will allow Kali to boot as your primary OS on your primary hard drive. Select **Yes** and click on **Continue**.

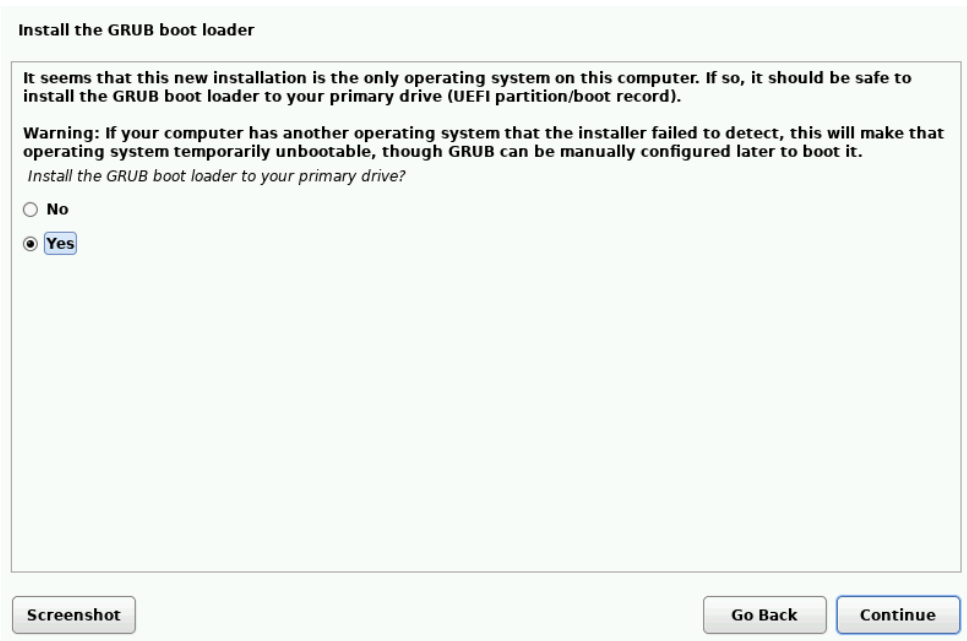


Figure 3.47: GRUB boot loader selection

19. For our final configuration, we need to make our system bootable by installing the boot loader on the disk. Select the disk with Kali installed (if not already selected by default) and click on **Continue**.

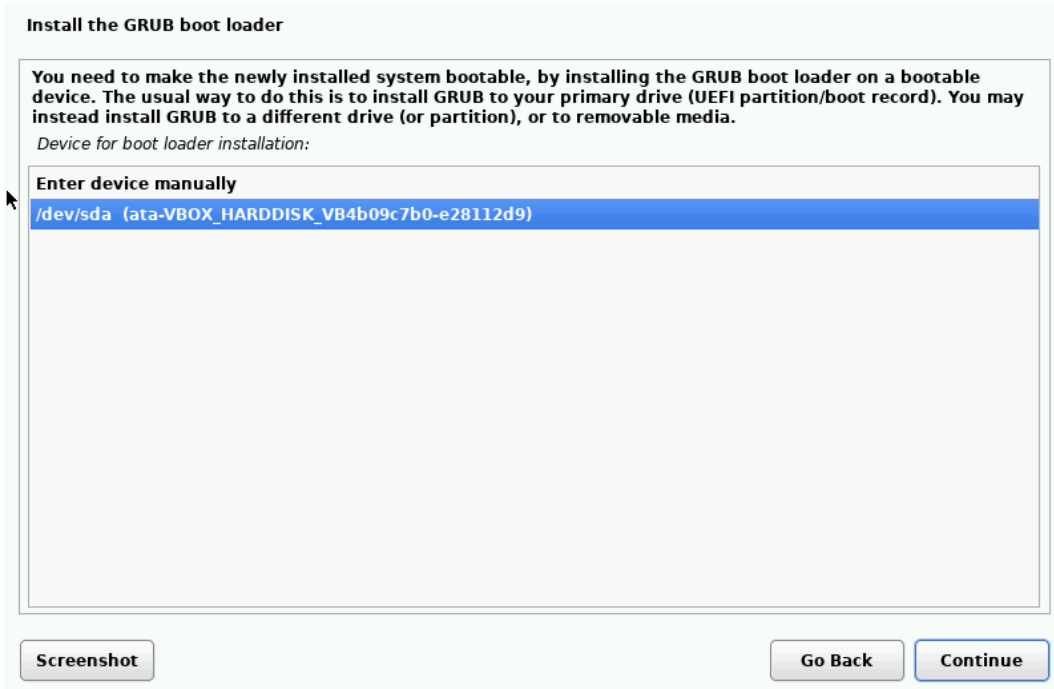


Figure 3.48: GRUB boot loader installation

20. Our installation will soon be complete.

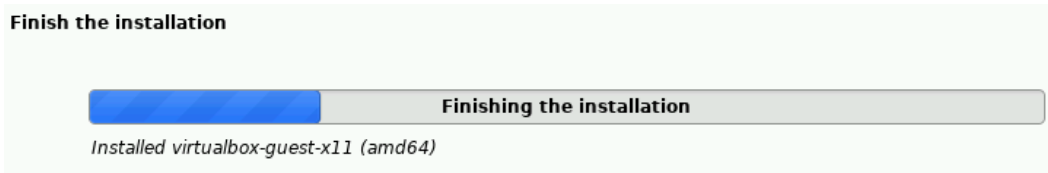


Figure 3.49: Kali installation process status

21. When prompted, click on **Continue** to complete the installation and reboot the system.

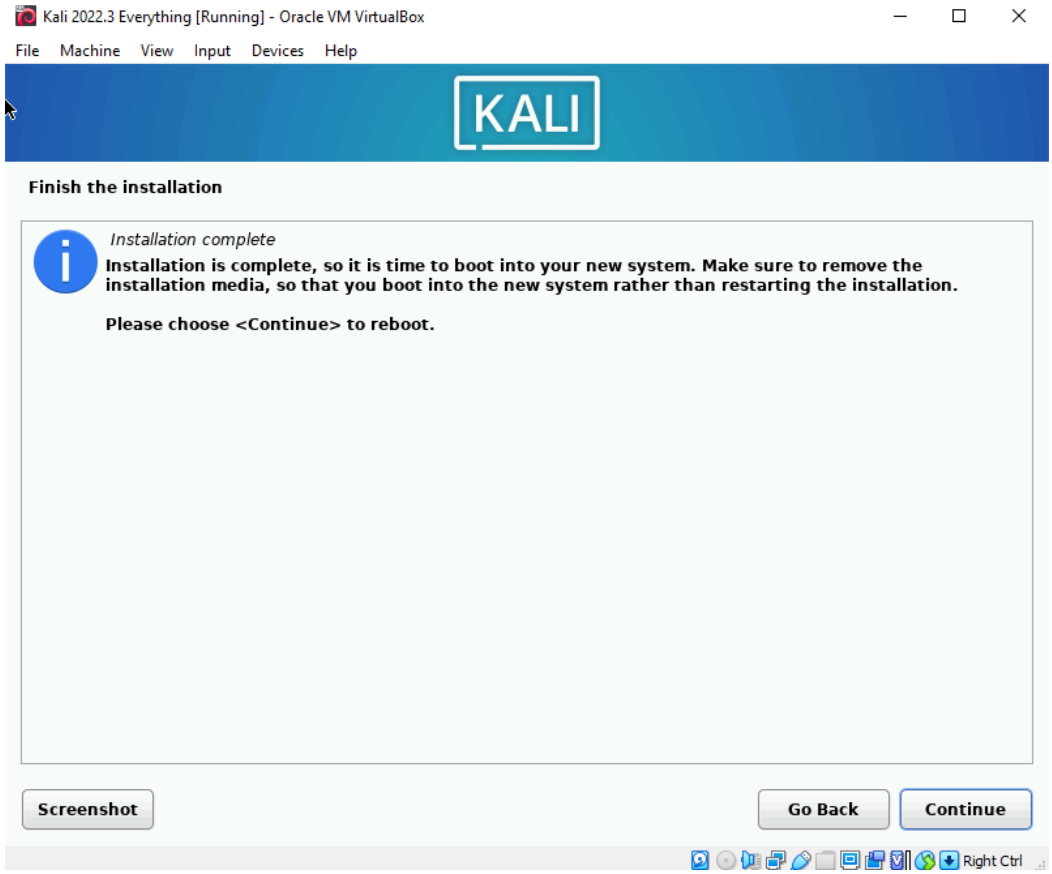


Figure 3.50: Kali installation completion screen

22. Enter your username and password.

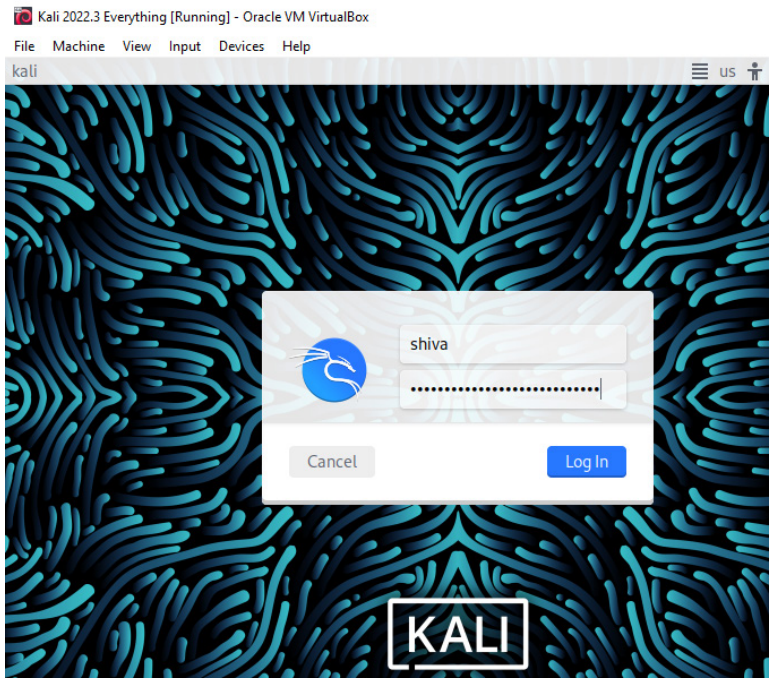


Figure 3.51: Kali Linux login screen

We've now completed the process of installing and configuring Kali Linux on both standalone and VMs. There are still other methods available for installing Kali Linux, which we will learn about in the next chapter. I'd like to show you another easier way to install a pre-configured version of kali Linux in VirtualBox.

Summary

We certainly learned quite a bit in this chapter. Feel free to wipe or delete your installations and try doing them again to become more acquainted with the processes for the different versions if need be.

In this chapter, we learned about the different versions of Kali Linux and the various platforms on which they can be installed and looked at the different methods for downloading different versions of Kali Linux. We then dived into the technical aspect by first learning how to create a bootable Kali Linux flash drive using Rufus utilizing the ISO image of Kali Linux, which we downloaded at the start of this chapter. We then learned how to install Kali on a physical device as a standalone OS and learned how to install Kali within VirtualBox as a VM.

In our next chapter, we will look at another method for installing Kali as a VM and learn how to install Kali on Raspberry Pi 4, followed by some post-installation tasks. See you in the next chapter.

4

Additional Kali Installations and Post-Installation Tasks

In our last chapter, we learned how to install Kali Linux as a standalone operating system and how to install Kali in VirtualBox. This chapter continues by looking at two other Kali Linux installations that can be considered much simpler and faster installations, but I'll let you be the judge of that. We will then perform some common post-installation tasks to ensure that we have a fully updated and functional version of Kali Linux for our **Digital forensics and incident response** (**DFIR**) investigations, regardless of the platform chosen for the installation.

The topics that we are going to cover in this chapter are as follows:

- Installing a pre-configured version of Kali Linux in VirtualBox
- Installing Kali Linux on Raspberry Pi 4
- Updating Kali Linux
- Enabling the root user account
- Adding the Kali Linux forensics repository to the installation

Installing a pre-configured version of Kali Linux in VirtualBox

Kali can also be installed in VirtualBox using a much simpler method by using the pre-configured versions of Kali, built specifically for VirtualBox:

1. If you haven't already downloaded the Kali VirtualBox image in *Chapter 3*, *Installing Kali Linux*, you can do so at `https://kali.download/virtual-images/kali-2022.3/kali-linux-2022.3-vmware-amd64.7z`.

2. You must extract the image with 7Zip, which can be downloaded for Windows at `https://www.7-zip.org/a/7z2201-x64.exe`.

   Once extracted, you should see the same files shown in the following screenshot.

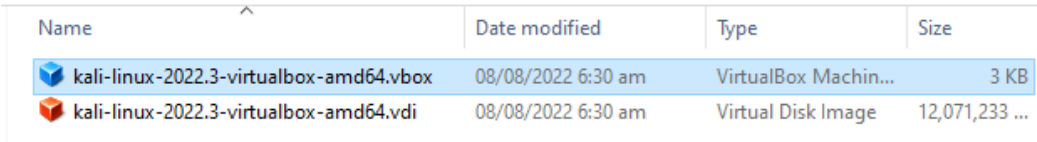


Figure 4.1 – The Downloads folder showing the pre-configured Kali images for VirtualBox

3. Double-click on the `.vbox` file, which is 3 KB in size, and it should immediately open within VirtualBox.

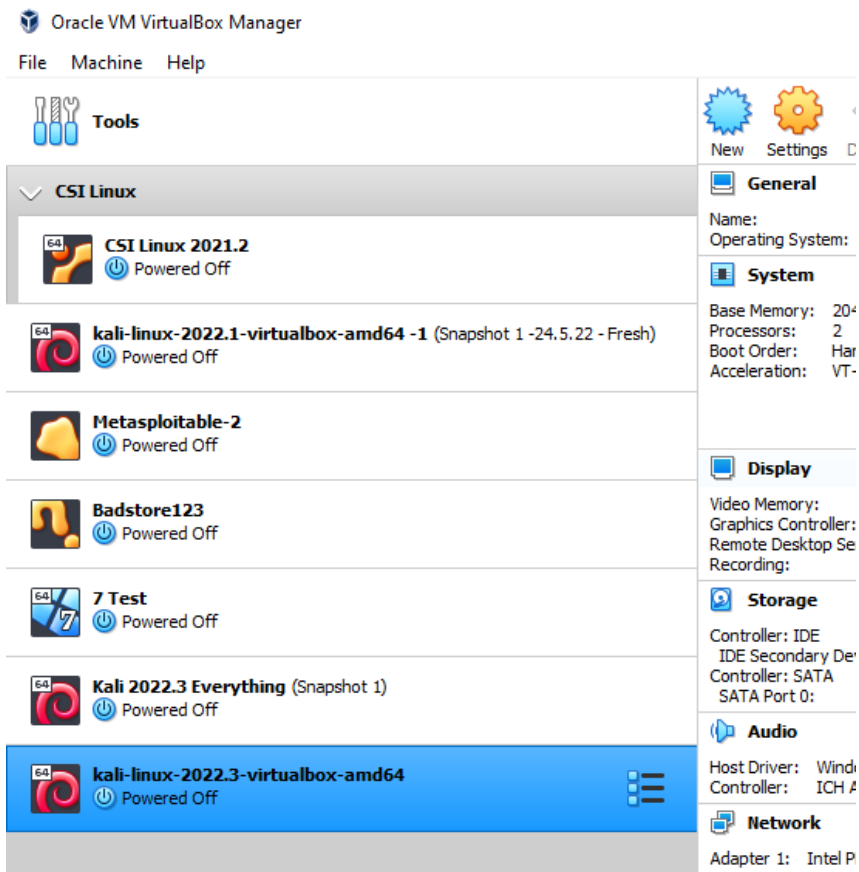


Figure 4.2 – VirtualBox Manager

4. Click on the **Settings** icon at the top of the **VirtualBox Manager** window.

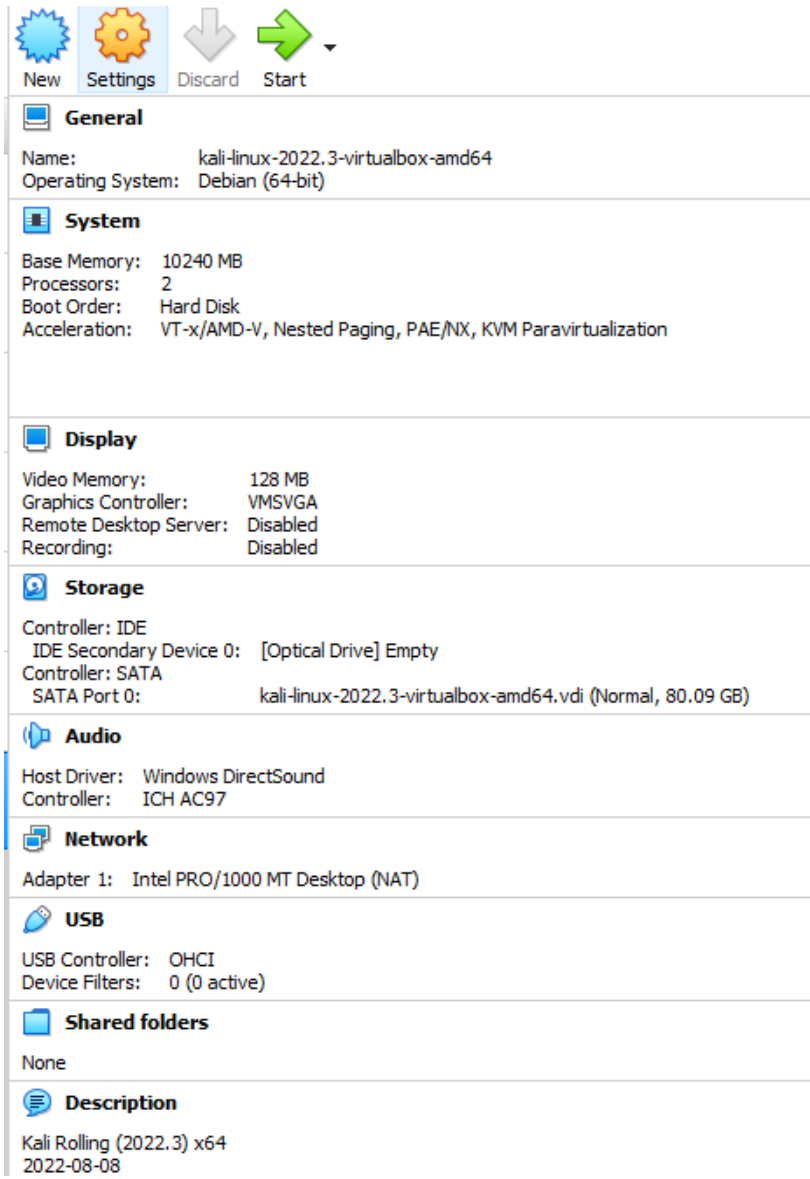


Figure 4.3 – The pre-configured settings for the Kali virtual machine

5. Click on **System** and use the slider to allocate the **Base Memory** size. When done, click **OK**.

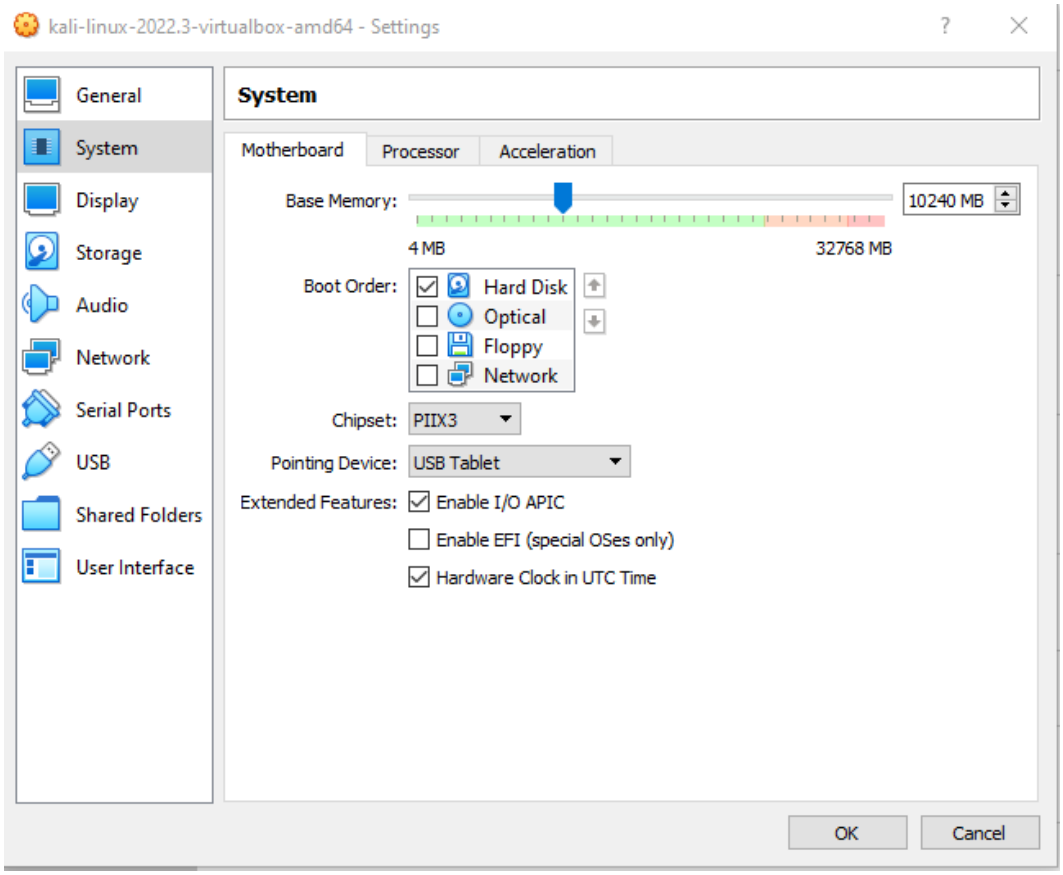


Figure 4.4 – Memory assignment in VirtualBox

6. To start the virtual machine, click on the green **Start** button at the top of the Virtual Manager.

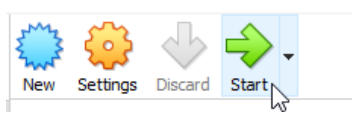


Figure 4.5 – The VirtualBox Start button

7. Use the following credentials to log in (all letters are lowercase):
   - Username: `kali`
   - Password: `kali`

We've now covered three methods for installing Kali Linux. I'm sure you found this one much easier than the others. Let's move on to another installation type on a portable Raspberry Pi 4.

Installing Kali Linux on Raspberry Pi4

Raspberry Pi 4, also known as the Pi 4, is a low-powered, small, portable, and powerful device that can be used for personal, enterprise, and even **Operational Technology** (**OT**) purposes, as mentioned earlier in this chapter. Having Kali on a Pi 4 can be very useful, as it is very powerful and comes with USB 3.0 ports, an RJ-45 network port, dual mini-HDMI ports, Wi-Fi, Bluetooth, and up to 8 GB of RAM, and it can run Kali off a bootable microSD card in just a few steps. You can learn more about the Pi 4 at `https://www.raspberrypi.com/products/raspberry-pi-4-model-b/`.

Follow these steps to install Kali on a Pi 4 (or even the older Pi 2, 3, and 400):

1. To install Kali on a Pi 4 (or even the older Pi 2, 3, and 400), we'll need the Pi Imager software, which can be downloaded at `https://www.raspberrypi.com/software/` and is available for Windows, Mac, and Ubuntu (x86) devices. Pi Imager will allow us to install the Kali Linux ARM version on a microSD card, which will run on our Pi 4. I also recommend using a 32 GB or larger capacity microSD card to ensure that you have enough space for upgrades and downloading additional tools in Kali.
2. Once you have downloaded Pi Imager, insert your microSD card into your laptop or desktop, either directly or via a card reader, and launch Pi Imager.

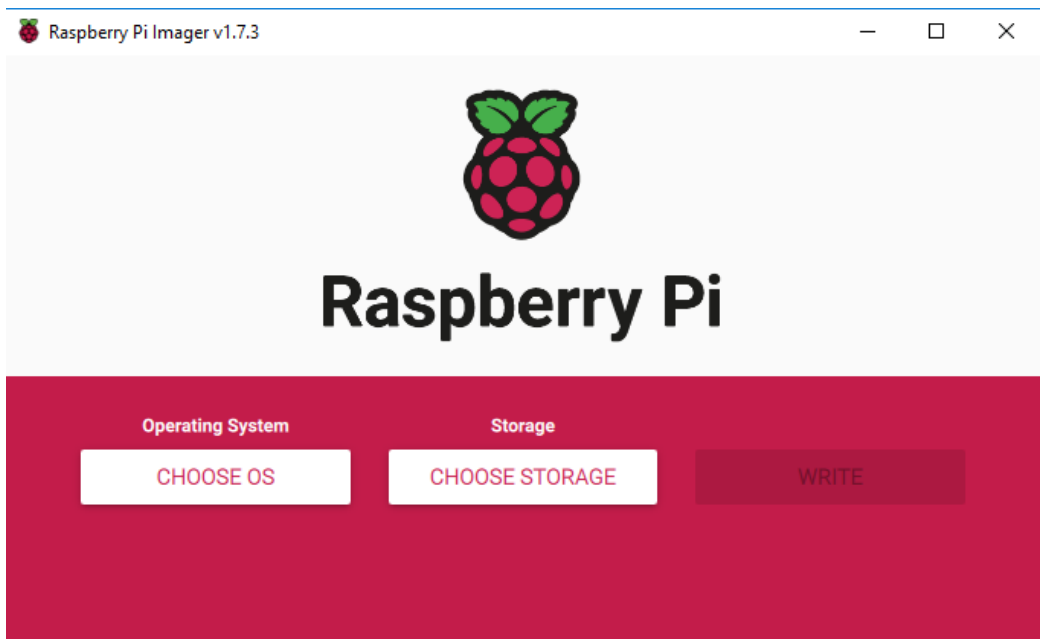


Figure 4.6 – The Raspberry Pi Imager interface

3. Click on **CHOOSE STORAGE** and select your microSD card.

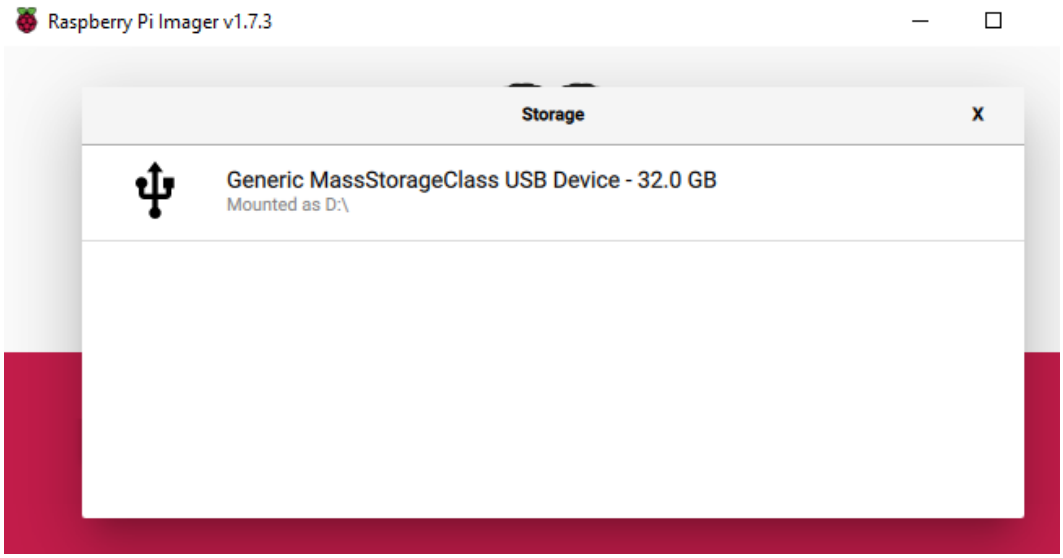


Figure 4.7 – Pi Imager storage selection

4. Next, click the **CHOOSE OS** button and click on the **Other specific-purpose OS** option.

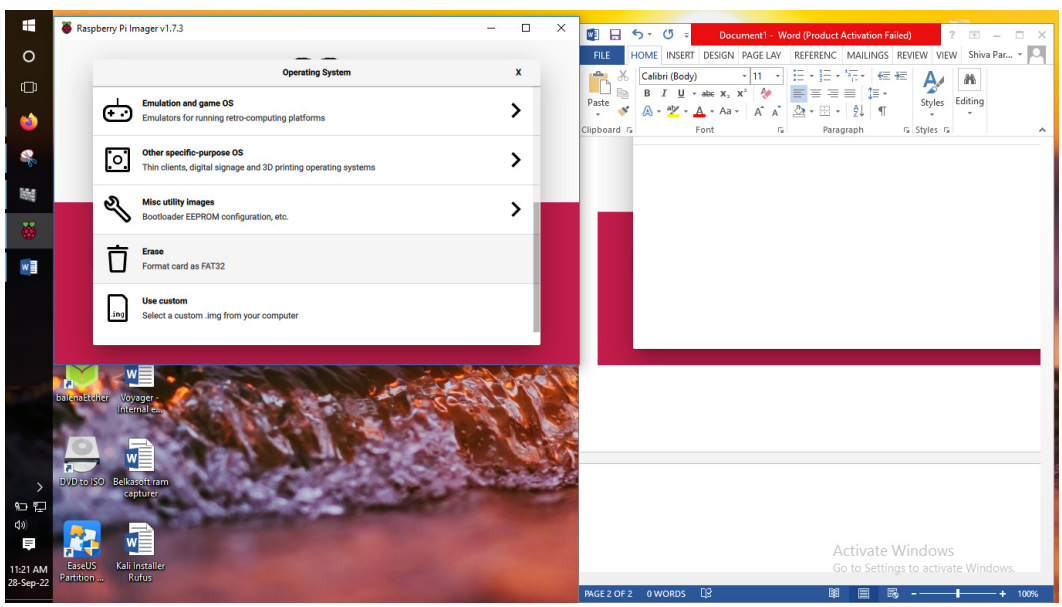


Figure 4.7 – Operating system selection

5. You should now see an option for **Kali Linux**, as shown in the following screenshot.

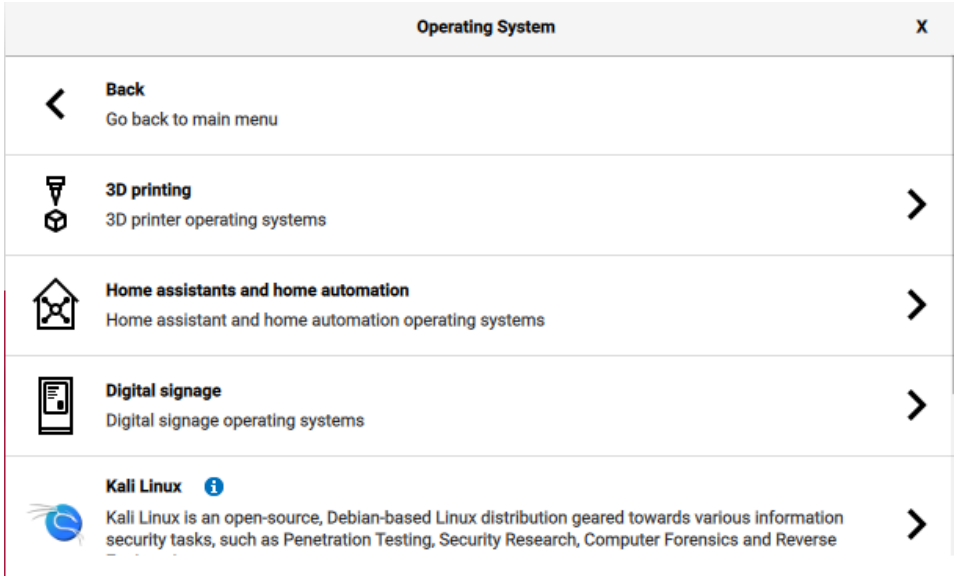


Figure 4.8 – Selecting the Kali Linux option

6. You will now be presented with the versions of Kali available for different Raspberry Pi devices. You can select your version depending on the model you have access to.

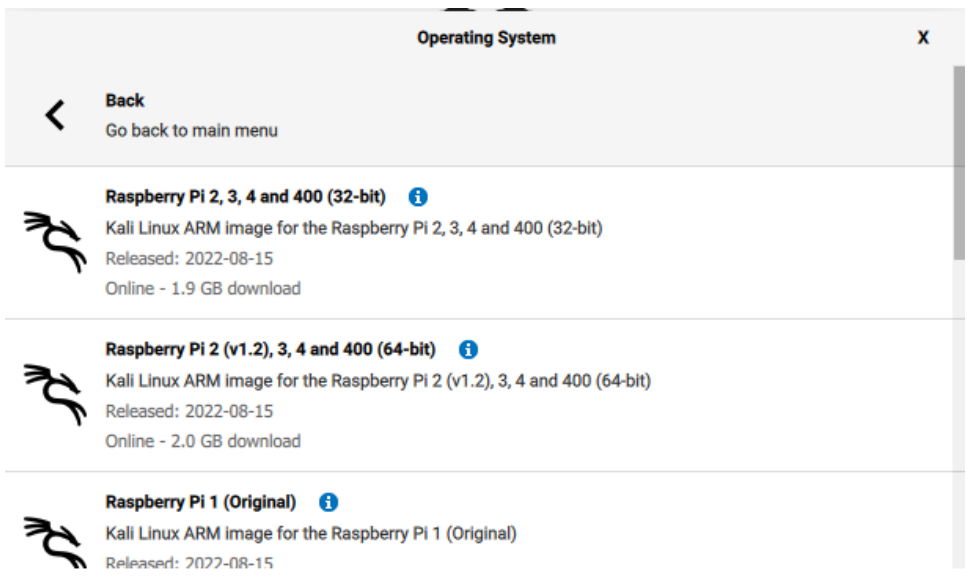


Figure 4.9 – The available Kali Linux versions

7. Once selected, you will be returned to the main interface. Click on the **WRITE** option to install Kali onto the microSD card.

Figure 4.10 – The final Pi Imager configurations

8. You will then be prompted to continue. Click **YES** to proceed.

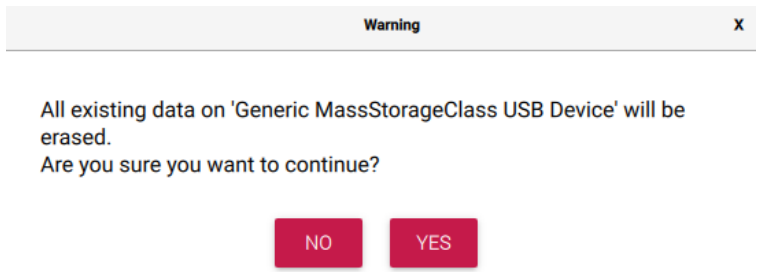


Figure 4.11 – Data erasure confirmation

The writing process will now begin.

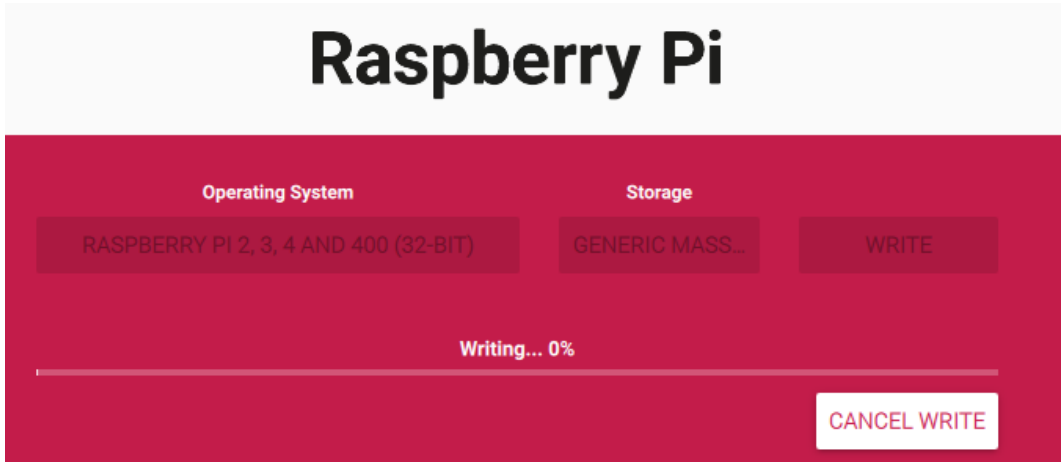


Figure 4.12 – Writing Kali to the SD card

9. Once the process is completed, remove the microSD card from your computer or card reader and gently insert it into your Raspberry Pi device, which should be powered off to avoid any damage to the card. Power on your Pi, and Kali will boot. Again, the default username and password are both `kali` (in lowercase).

This concludes the various types of installations of Kali Linux on a variety of platforms. Before we use Kali, we should always update our tools, which I'll demonstrate in the next section.

Updating Kali

Now that we've looked at the many possible ways of installing Kali Linux on various platforms, let's have a look at some important post-installation tasks once Kali Linux systems are up and running:

1. To view version installation details of our Kali systems, let's run the following command:

```
cat /etc/os-release
```

In the following screenshot, we can see the command output shows the names, versions, code name, and more details for verification purposes.

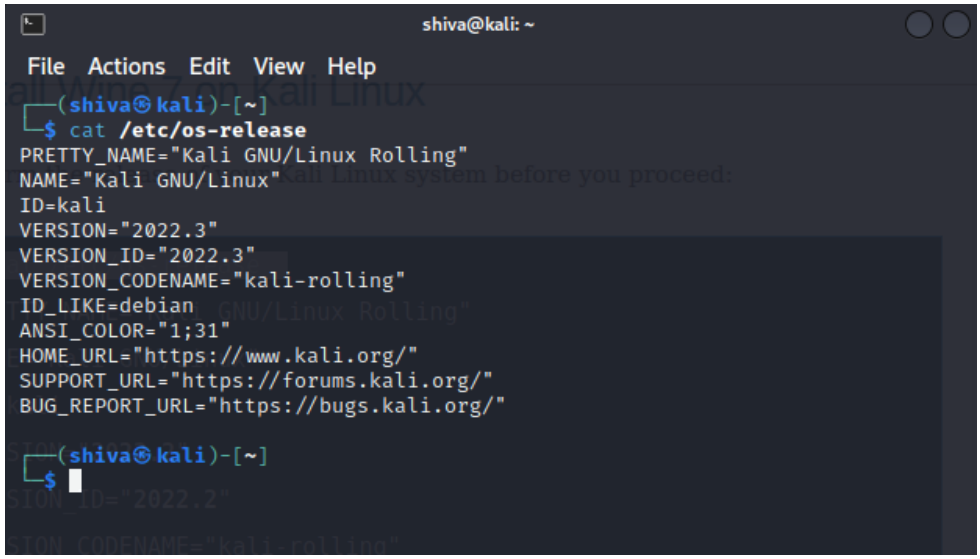


Figure 4.13 – The cat command output

2. We should also always update our Kali systems after installation to ensure that we have the current version of tools and supporting software.

   To update Kali Linux, type and run the following command:

   ```
   sudo apt update
   ```

   In the following screenshot, some updates were installed, and the last line states that there are 674 packages that can be upgraded:

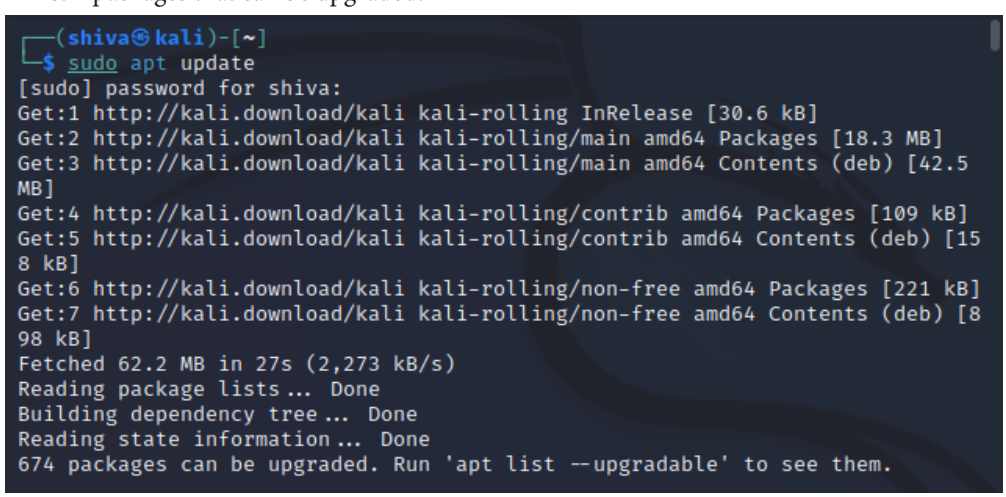


Figure 4.14 – Updating Kali Linux

3. Type and run the following command to see the list of upgrades:

```
apt list -upgradable
```

In the following screenshot, we can see the command being executed. It may take some time to upgrade all components.

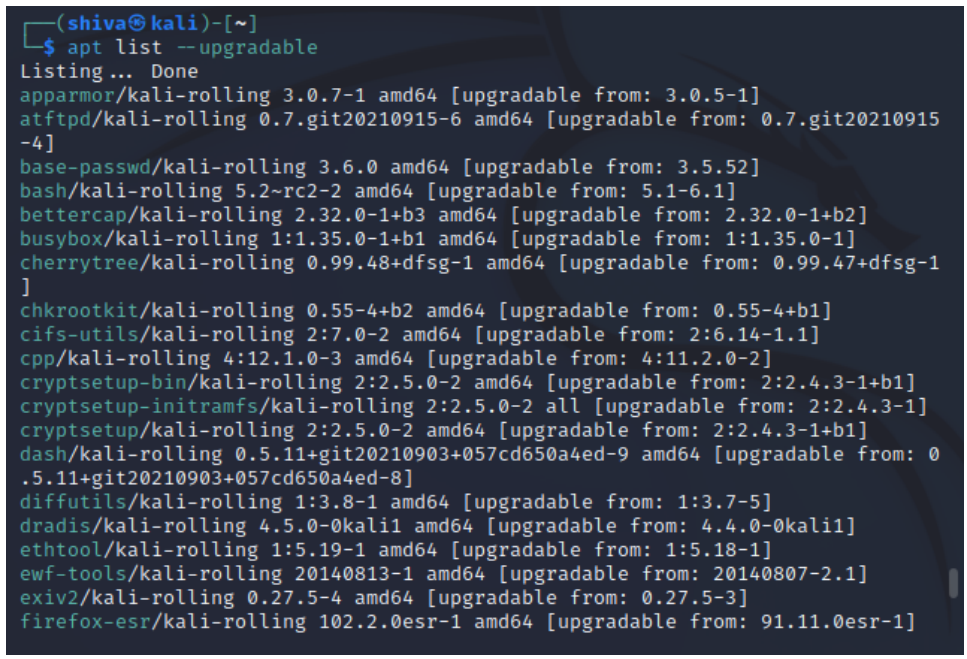


Figure 4.15 – The apt list –upgradable command output

4. Type and run the following command to allow Kali to upgrade all the preceding software and packages without prompting for confirmation:

```
sudo apt upgrade -y
```

The following screenshot shows the execution and output of the command.

```
┌──(shiva㉿kali)-[~]
└─$ sudo apt upgrade -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done
The following packages were automatically installed and are no longer require
d:
  libhttp-server-simple-perl liblerc3 libwacom-bin python3-dataclasses-json
  python3-limiter python3-mypy-extensions python3-responses python3-spyse
  python3-token-bucket python3-typing-inspect python3.9 python3.9-minimal
  ruby2.7-dev ruby2.7-doc xprobe
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  certipy-ad colly faraday-agent-dispatcher google-nexus-tools
  libdbus-1-dev liblerc4 libpcap-dev libpcap0.8-dev libwebsockets17
  mongo-tools python3-cryptography37 python3-json-pointer python3-munkres
  python3-rfc3987 python3-webcolors python3-zapv2 ruby-websocket
The following packages have been kept back:
  apparmor atftpd base-passwd bash bettercap busybox cherrytree chkrootkit
  cifs-utils cpp cryptsetup cryptsetup-bin cryptsetup-initramfs dash
  diffutils dradis ethtool ewf-tools exiv2 firefox-esr g++ gcc gcc-12-base
  gir1.2-javascriptcoregtk-4.0 gir1.2-nm-1.0 gir1.2-vte-2.91
```

Figure 4.16 – Upgrading Kali Linux

Note that this upgrade process can take some time to complete.

5. We can now reboot Kali to complete the entire process by typing the following command:

```
sudo reboot
```

The following screenshot shows the execution and output of the command.

```
┌──(shiva㉿kali)-[~]
└─$ sudo reboot
```

Figure 4.17 – Rebooting Kali Linux

After updating and upgrading your Kali Linux installations, we can now move on to other tasks. Next, we will learn how to enable the root user account.

Enabling the root user account in Kali

Now that all our updates and upgrades have been completed, let's look at enabling the root user account in Kali Linux. You will have noticed by now that some versions of Kali, when installed, allow access with the default username and password of Kali. This was done as a security feature to not allow unintentional changes to configurations or tools. If we wish to execute commands or perform various tasks using root or administrator privileges instead, we must type `sudo` before the respective commands.

Sudo is an abbreviation for **superuser do**, which allows us to run commands and tools as a superuser with the highest privileges, known as root privileges. It can sometimes be a bit of a nuisance to have to type this before commands, but it becomes routine after a while.

However, if you are new to Kali and Linux-based operating systems, I do recommend using the default profile and typing `sudo` when necessary, but the choice is ultimately yours.

Enabling the root account is fairly simple and can be done in just a couple of steps:

1. Firstly, boot your Kali machine if it is not running and log in using the username and password you have created, or using the default credentials provided – username: `kali`, and password: `kali`.

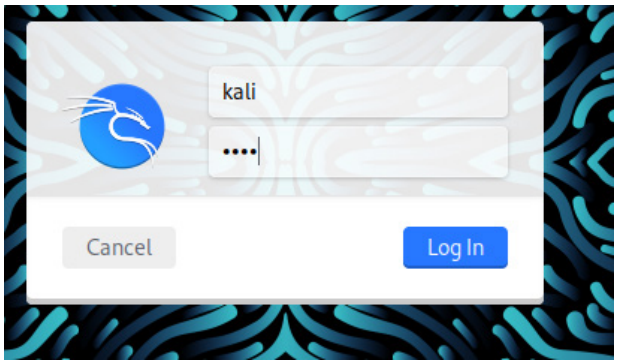


Figure 4.18 – The Kali Linux login screen

2. Click on the Terminal icon at the top to open a new Terminal.

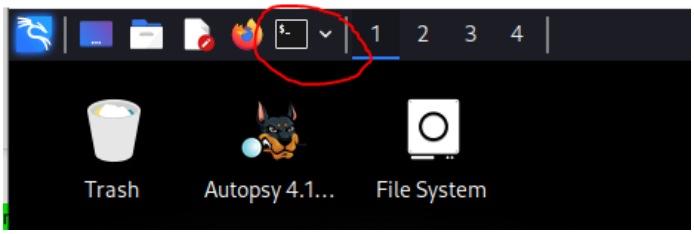


Figure 4.19 – The Kali Linux desktop with the Terminal shortcut circled

3. In the new Terminal, note that the prompt shows `(kali@kali)`. Type the following command to access the superuser account and press *Enter*. You will also have to type your password thereafter – `sudo su`.

The following screenshot shows the execution and output of the command.

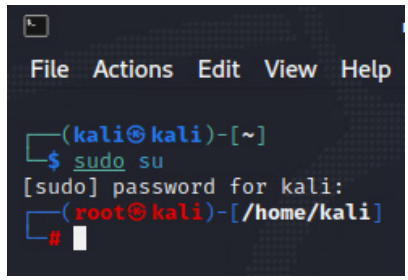


Figure 4.20 – The sudo su output

> **Note**
> Passwords are not displayed on screen when typed in Kali Linux.

In the preceding screenshot, note that the prompt has now changed to **(root@Kali)**. Let us now change the password for the root account by typing the following command.

4. You will then be prompted to type and retype the new password – `passwd root`.

The following screenshot shows the execution and output of the command.

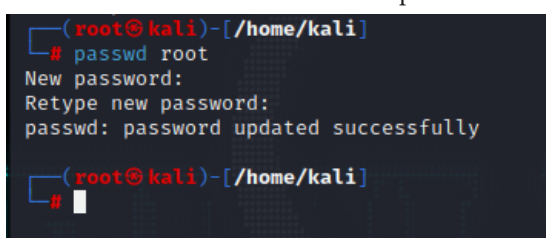


Figure 4.21 – Creating a password for the root account

5. Success! We can now log out as the current user and log back in, or just switch to the root user by first clicking on the power button icon at the extreme top-right of the screen and then either choosing **Log Out** or **Switch User**, as seen here.

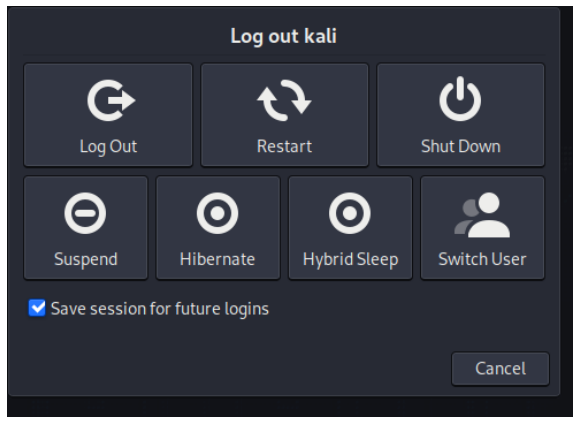


Figure 4.22 – The Kali Linux power and user options

Log in with the username `root` and the password you just created for the account.

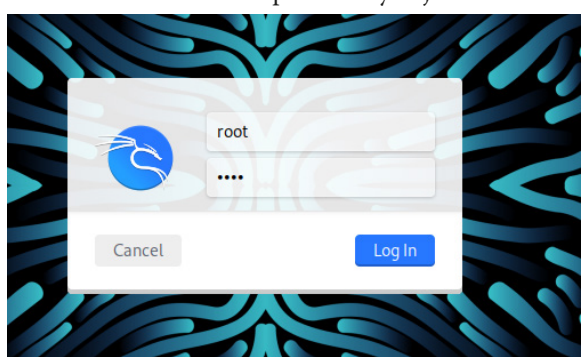


Figure 4.23 – The Kali Linux login screen

6. Click on the Terminal account again, and you will notice that the prompt is now **(root@kali)**, as seen here.

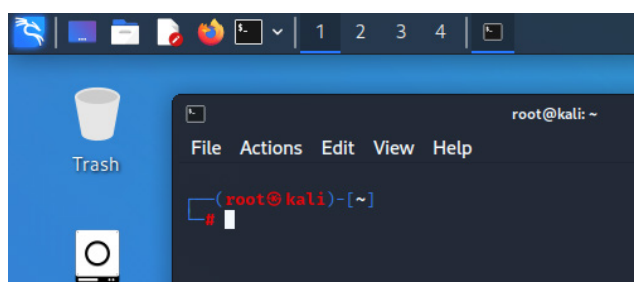


Figure 4.24 – The Terminal showing the root user account

Now that we have a root user account enabled on Kali, our last task in this chapter will be to add additional forensics tools using a forensics metapackage.

Adding the Kali Linux forensics metapackage

Kali metapackages are collections of tools that can be downloaded as a full package. Depending on the version of Kali you installed, it may not come with all the tools you may have expected or viewed on the tools listing page at `https://www.kali.org/tools/all-tools/`, which lists all the tools available in Kali Linux. For our DFIR purposes, for example, a listing of all forensics tools can be viewed at `https://www.kali.org/tools/kali-meta/#kali-tools-forensics`, where you can click on the drop-down arrow in the **Dependencies** section to view a listing of all tools.

All of the preceding tools are contained in the `kali-tools-forensics` metapackage and can be installed by typing the following command:

```
sudo apt install kali-tools-forensics
```

The following screenshot shows the execution and output of the command.

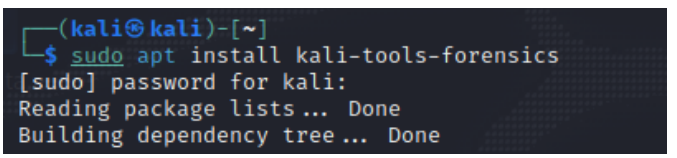


Figure 4.25 – Installing the forensics metapackage

> **Note**
>
> You should update your Kali Linux before installing any metapackage as we previously did, using the `sudo apt update` command.

This will install all listed forensics tools in Kali Linux, as shown in the previous screenshot. You can view some of the forensics tools by clicking on **Applications** | **11-Forensics** on the main Kali menu.

Summary

In this chapter, we learned that we can also install Kali Linux in a much simpler and faster manner by using the pre-configured version of Kali, specifically designed for VirtualBox. Just a few clicks and tweaks and we were done. We also learned to install Kali on a Raspberry Pi device, which can be very useful where portability is concerned, by simply using the Pi Imager tool to install Kali onto a microSD card.

We then looked at some very important post-installation tasks that should be done after any installation. We learned how to find the version of Kali Linux and then update and even upgrade all software and packages in Kali. Finally, in this chapter, we looked at enabling the root account to avoid having to use the `sudo` command when performing actions that require superuser privileges. We also learned how to install all forensics tools by adding the `kali-tools-forensics` metapackage to our installation.

Now that we have installed our various versions of Kali and updated our systems, we will move on to a very useful tool called Wine, which allows us to run Windows tools on our Kali Linux system. Pretty cool stuff if you ask me. See you in the next chapter!

5

Installing Wine in Kali Linux

In this chapter, we will learn how to extend the functionality of our Kali Linux environment by adding the capability to install Windows tools within Kali using Wine.

We will look at the following topics in this chapter:

- What Wine is and the advantages of using it in Kali Linux
- Installing Wine
- Testing our Wine installation

Let's begin by looking at Wine and its advantages.

What Wine is and the advantages of using it in Kali Linux

Wine is commonly, but mistakenly, referred to as a **Windows Emulator**, but as stated on the official website, it was initially known as **Wine Is Not an Emulator**, hence the acronym. Wine allows non-Windows users to run certain Windows applications on **Portable Operating System Interface** (**POSIX**)-compliant systems, including macOS and versions of Linux and BSD, including Kali Linux.

For our **Digital Forensics and Incident Response** (**DFIR**) purposes, this is incredibly useful, as we can now use many industry-standard tools, specifically made for Windows on our Kali Linux systems, whether using them as standalone operating systems or as virtual machines. This also addresses the issue of having to purchase a license for Windows, as we do not have to use a device running Windows. These cost savings are also extended when considering that you can now use less hardware if running all DFIR tools within your Kali Linux box.

By installing Wine in our Kali Linux installations, we can now have an advantage not only from a DFIR perspective but also from a purple teaming one, as we can also install many red teaming tools for Windows in our Kali machines. Drawing from personal experience, I've always used two laptops for DFIR exercises and investigations, with one running Linux-based operating systems such as Kali Linux, CSI Linux, and Ubuntu, and another running Windows 10. Some very important Windows applications that I use for almost every DFIR exercise and investigation are Belkasoft RAM capturer,

FTK Imager, and Autopsy 4 **GUI** (**Graphical User Interface**). These tools are highly used throughout the community, as they are very powerful and fairly simple to use. With Wine installed on my Kali Linux and other Linux-based systems, I can now perform all tasks using one machine, which saves on time, effort, and cost.

In the sections ahead within this chapter, we will be installing Wine manually using the Terminal and then installing the Google Chrome browser as a test to ensure that our Wine installation is a successful one. In later chapters, we will also install Belkasoft RAM Capturer, FTK Imager, and Autopsy 4 using Wine for DFIR purposes within Kali Linux. More information about Wine can be found on the official site at `https://www.winehq.org/`.

Installing Wine

Installing Wine requires that we run a few commands in precise order. Although I am sure that many of the commands used will be new to many of you, I've outlined the exact steps in the following sequence:

1. Be sure that you are connected to the internet before proceeding.

   Let's first download Wine by opening a new Terminal and typing the following command, followed by pressing the *Enter* key:

   ```
   sudo dpkg --add-architecture i386
   ```

   In the following screenshot, we added the 32-bit architecture:

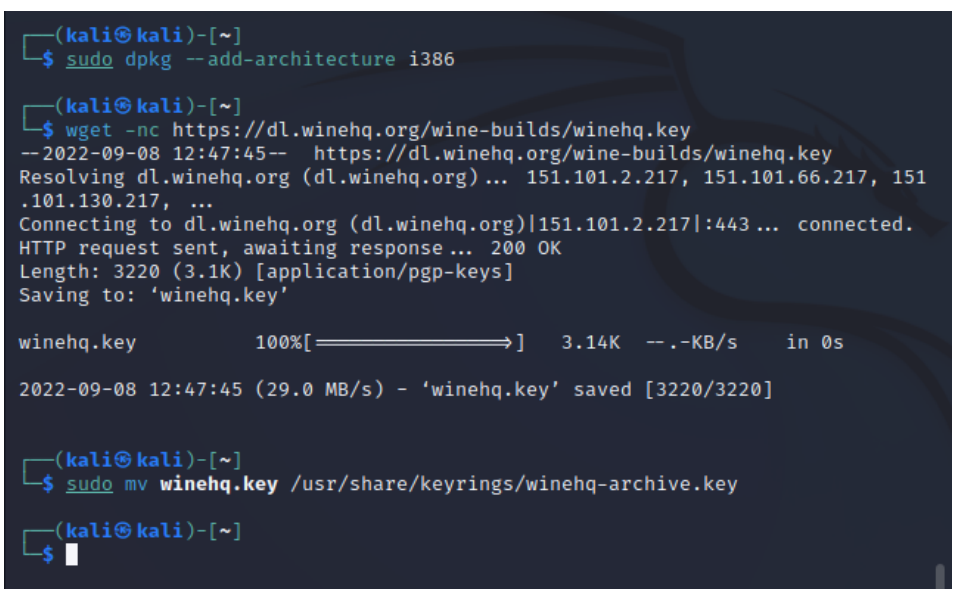


Figure 5.1: Running the sudo dpkg –add-architecture i386 command

2. Next, we need to download the `wine` key by typing the following command:

```
wget nc https://dl.winehq.org/wine-builds/winehq.key
```

The following screenshot shows the output of the preceding command:

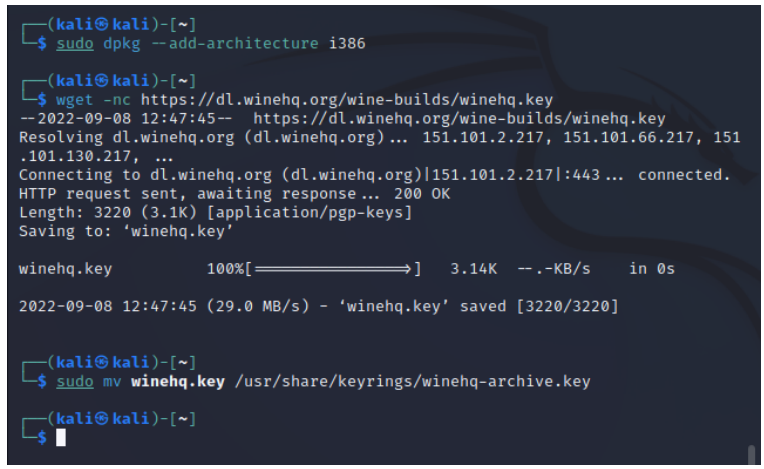


Figure 5.2: Installing Wine in the Terminal

3. Once our key has been downloaded, we can then add it to `/usr/share/keyrings/winehq-archive.key` by typing the following command:

```
sudo mv winehq.key /usr/share/keyrings/winehq-archive.key
```

The following screenshot shows the output of the preceding command:

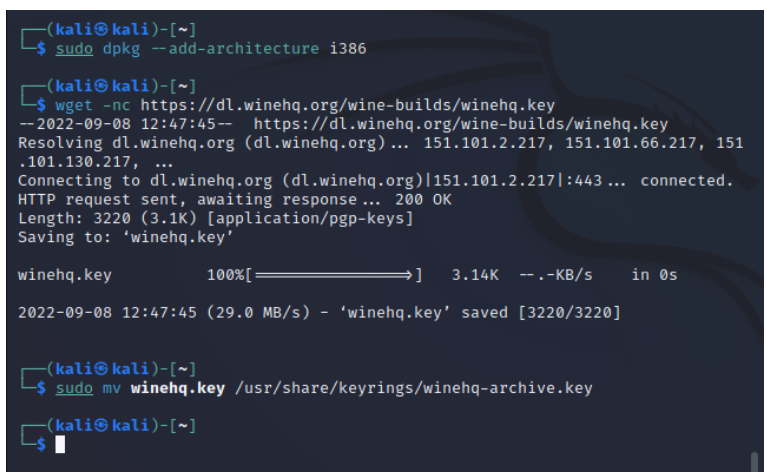


Figure 5.3: Adding the keyring

4. Next, we download the Wine `sources` file by typing the following command:

```
wget -nc https://dl.winehq.org/wine-builds/debian/dists/
bullseye/winehq-bullseye.sources
```

The following screenshot shows the output of the preceding command:

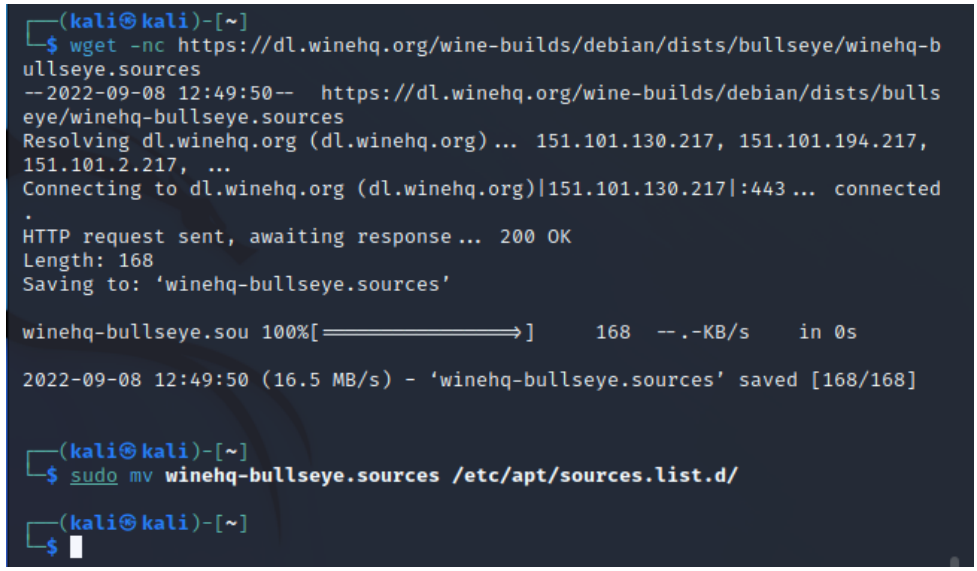


Figure 5.4: Downloading the required Wine sources

5. Once our `sources` file has been downloaded and saved successfully, we can add it to our sources list by typing the following command:

```
sudo mv winehq-bullseye.sources /etc/apt/sources.list.d/
```

The following screenshot shows the output of the preceding command:

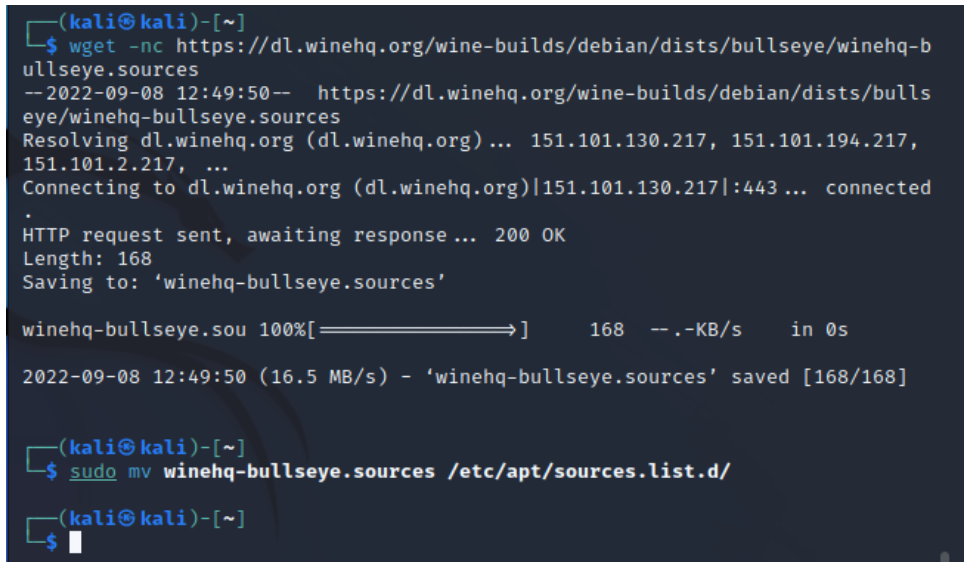


Figure 5.5: Adding the sources file to the sources list

6. We will now add the Debian repository for a smooth installation by typing the following:

```
echo "deb http://ftp.us.debian.org/debian bullseye main "
| sudo tee -a /etc/apt/sources.list
```

The following screenshot shows the output of the preceding command:

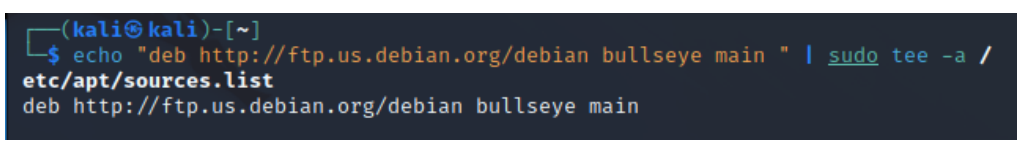


Figure 5.6: Adding the Debian repository

7. Once your Terminal looks like the preceding screenshot, you can check for any newer updates by typing the `sudo apt update` command. As seen in the following screenshot, there are quite a few updates to my installation:

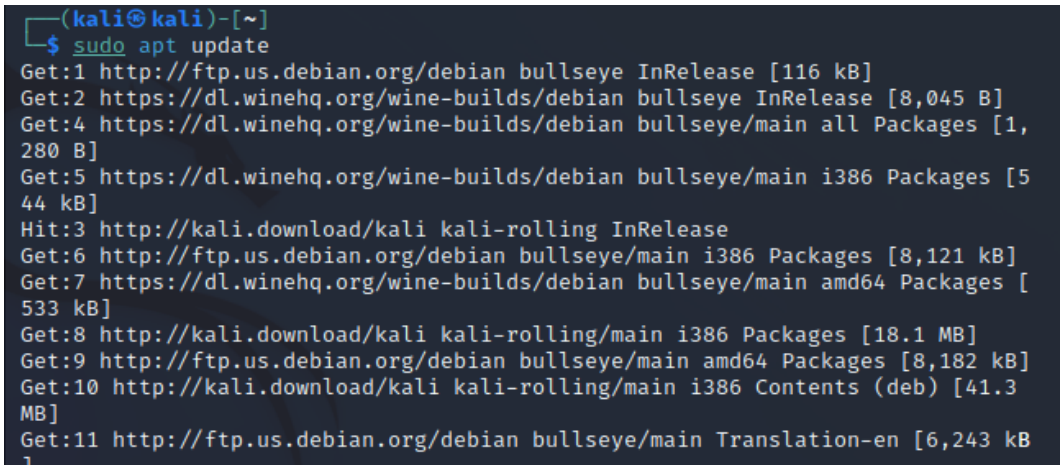


Figure 5.7: Updating Kali

8. Once all updates have been completed, we can now download the latest stable version of Wine by using the following command:

```
sudo apt install --install-recommends winehq-stable
```

The following screenshot shows the output of the preceding command:

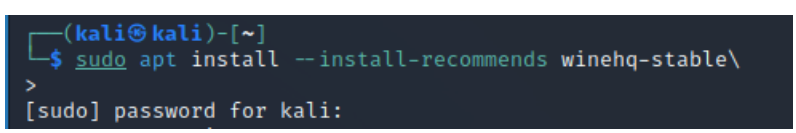


Figure 5.8: Installing the latest stable version of Wine

> **Note**
>
> You will be prompted to enter your password and press *Y* for Yes to continue the installation when prompted. This may take a couple of minutes to install as the file is large.

9. Once all goes well and all the necessary files and packages are downloaded and installed, we can verify the installation and version of Wine installed by running the following command:

```
wine --version
```

The following screenshot shows the output of the preceding command:

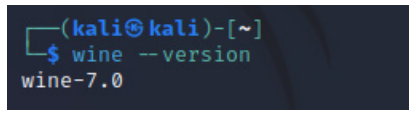


Figure 5.9: Verifying the version of Wine installed

Configuring our Wine installation

Our final step in the Terminal before we move on to the graphical installation and configuration of Wine is to run the `winecfg` command:

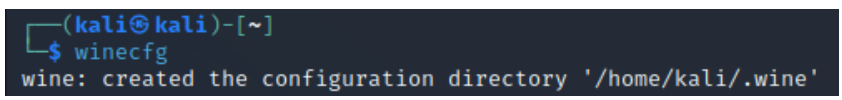


Figure 5.10: Running the winecfg command to start Wine

Once all is as it should be, the **Wine Mono Installer** box appears and asks us to install the Microsoft .NET Framework, which is required for Windows applications to run correctly, and then follow these steps to configure and complete the installation:

1. Click on the **Install** button.

Figure 5.11: Installing Wine Mono

2. Wine Mono will then begin downloading and installing.

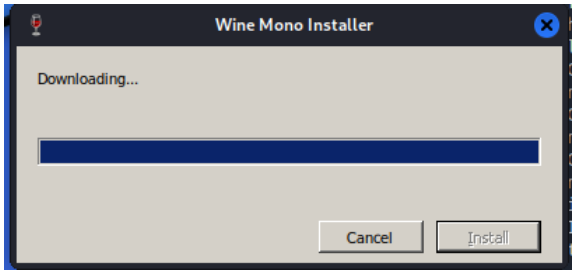


Figure 5.12: Wine Mono installation status bar

3. In the **Wine configuration** box, you can also select the version of Microsoft Windows you would like to mimic on your Kali Linux box.

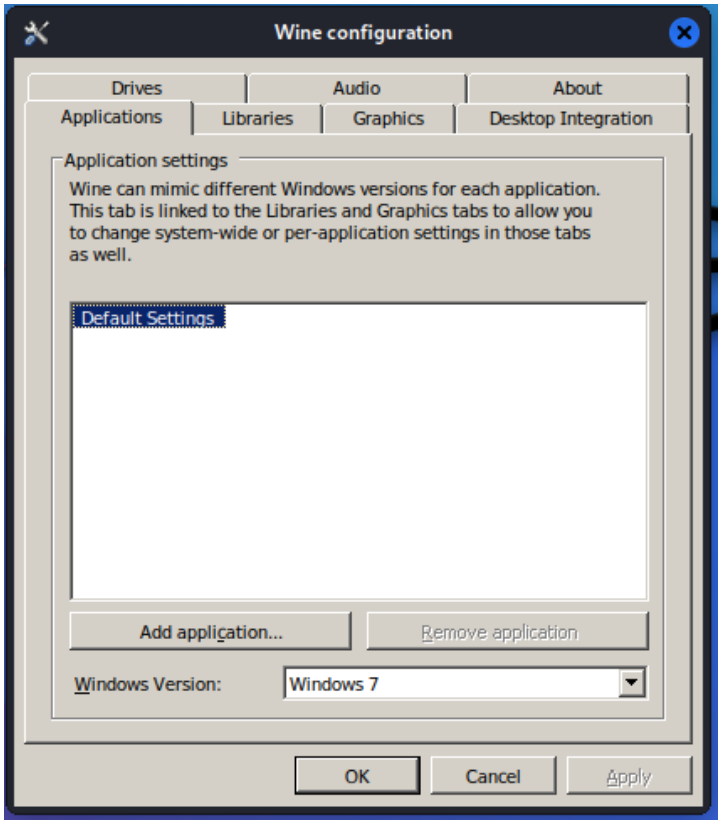


Figure 5.13: Configuring Wine

4. In this instance, we can see that **Windows 7** is selected and presented by default. To change this setting to another version of Windows, click the drop-down arrow to the right of the Windows version and select your operating system of choice, as seen here:

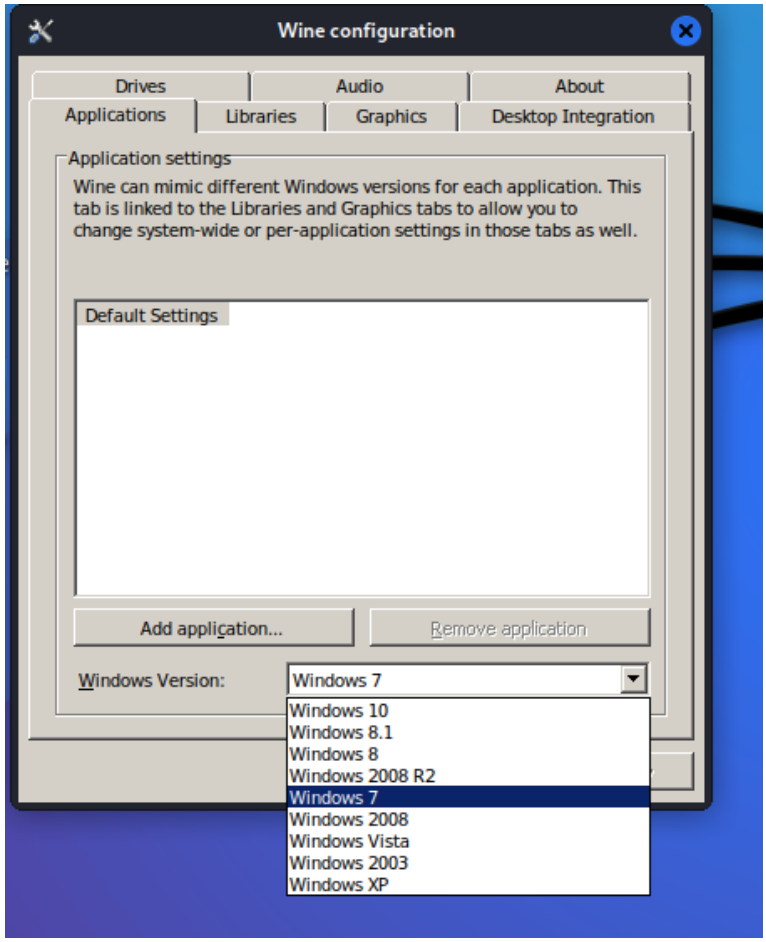


Figure 5.14: Windows OS selection

I've selected **Windows 10** as my operating system as the Windows DFIR applications that we will be installing all run on Windows 10.

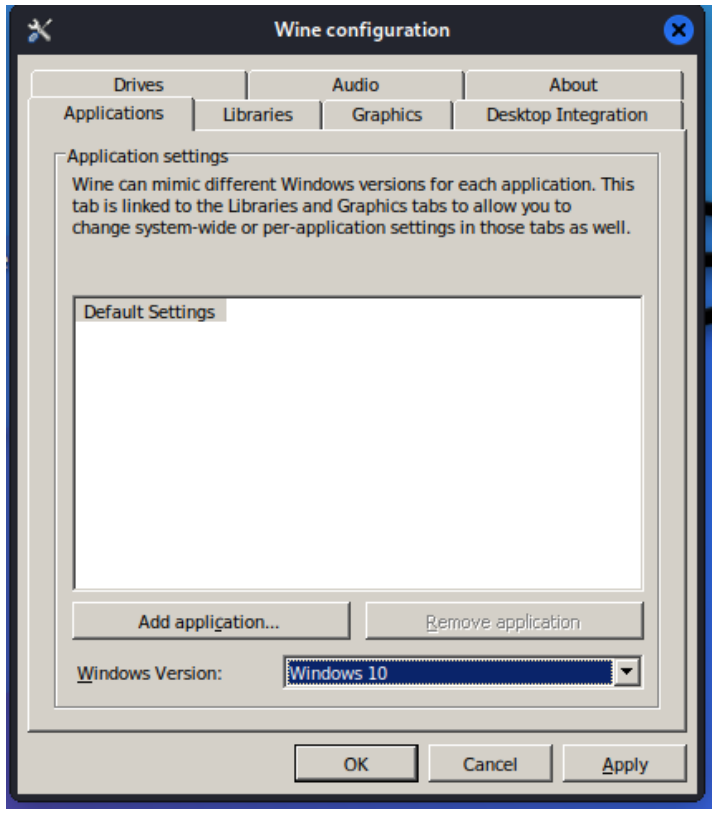


Figure 5.15: Selecting Windows 10

5. Click on **Apply** to continue, then enter your details for **Owner** and **Organization** in the **About** tab, as seen next, and click on **Apply** and **OK** to complete the Wine configuration.

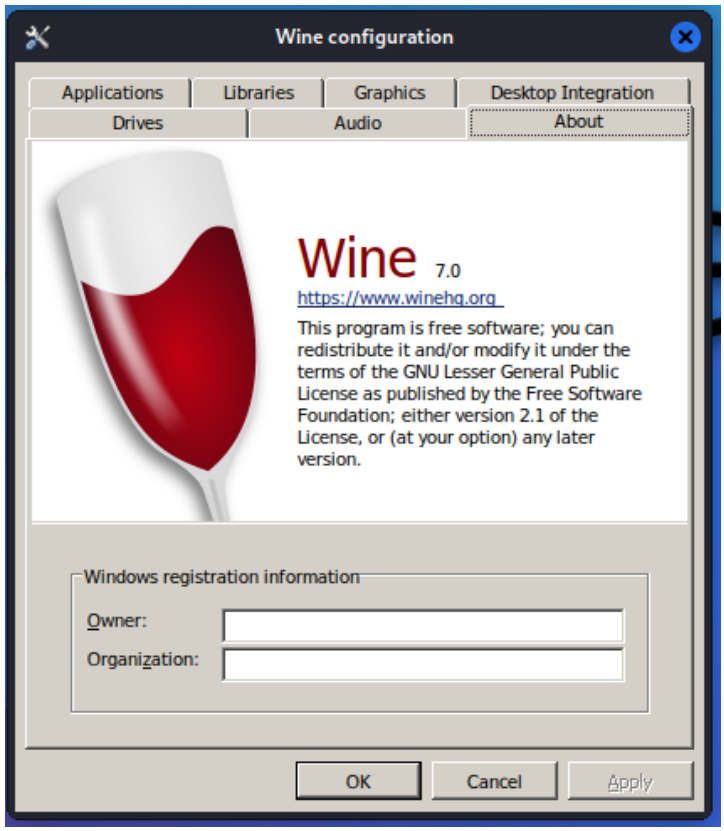


Figure 5.16: Completion of Wine configuration

Now that we've completed all the steps to install Wine in this section, we will need to install a Windows program as a test, which we will now do in the next section.

Testing our Wine installation

Good job on getting this far. To test our installation, let's try installing a file recovery tool called Recuva in our Kali Linux system using Wine. Recuva is a popular tool that comes in both a free and paid version. Let's get started with its download and installation:

1. You can download the free version at `https://www.ccleaner.com/recuva/download`, which allows users to only perform advanced file recovery. At the time of this writing, the file can also be downloaded directly at `https://www.ccleaner.com/recuva/download/standard`.

2. Whichever download method you choose, please be sure to click on the **FREE RECUVA** option.

   The filename will appear as `rcsetup153.exe`, which indicates that it is a Windows executable file, as seen in the following screenshot:

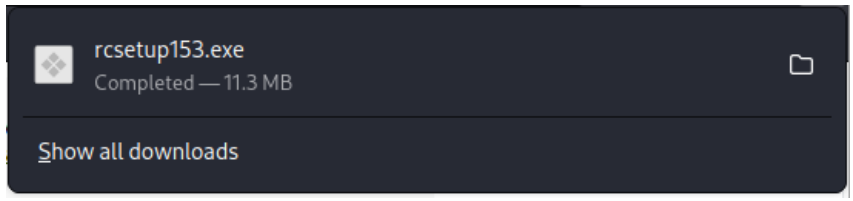


Figure 5.17: The Recuva installation file download

3. Click on the folder icon to the right of the file download or navigate to the `Downloads` folder, where we should find the downloaded Recuva file (`rcsetup153.3xe`).

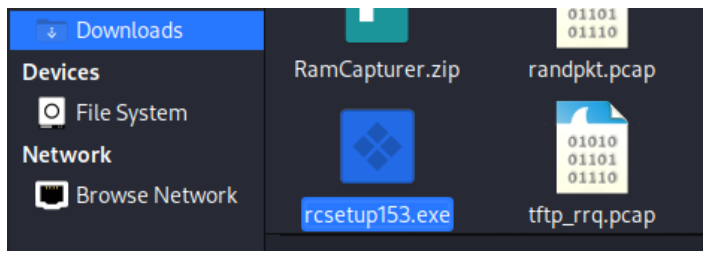


Figure 5.18: Recuva installation file in the Downloads folder

4. Installing Recuva from here is very simple. Double-click on the `rcsetup153.exe` file and it will open the installer, exactly as it would on a Windows machine, as seen in the next figure.
5. Alternatively, you may also right-click on the file and select **Open With Wine Windows Program Loader**.

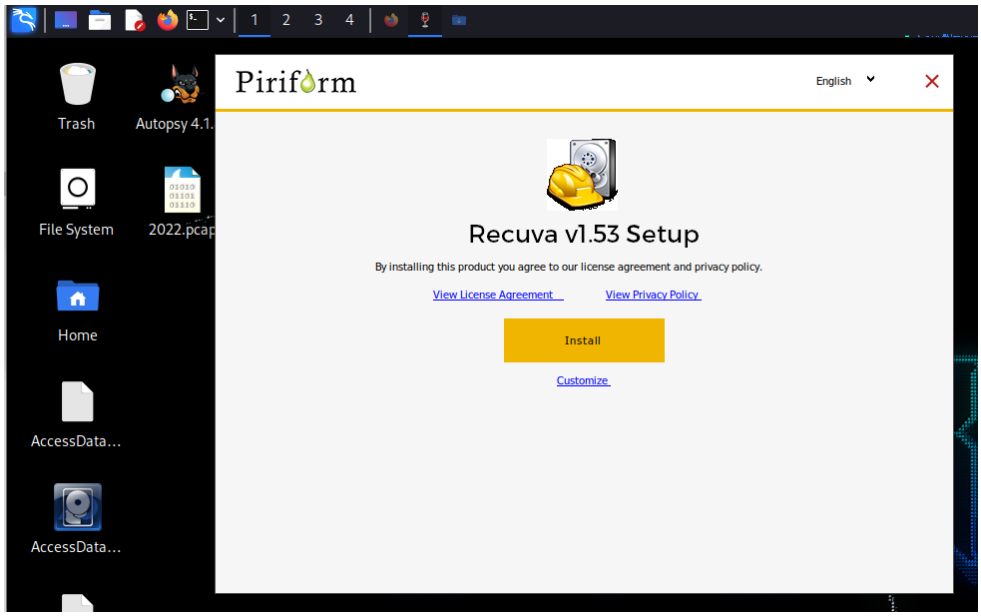


Figure 5.19: The Recuva installation interface

6. Click on **Install** to continue and install Recuva in Kali Linux.

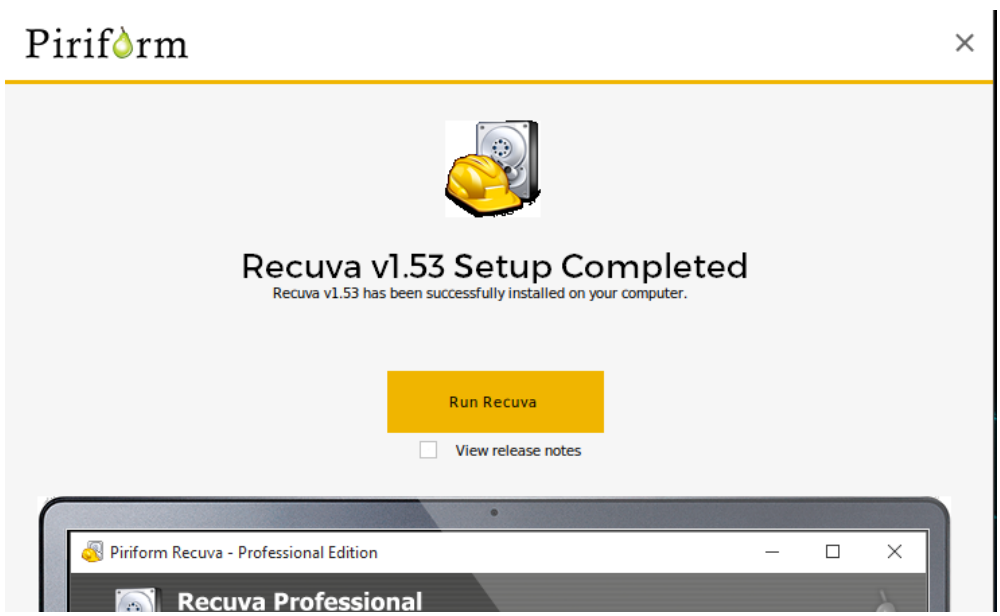


Figure 5.20: The Recuva installation completion screen

7. Once installed, you can uncheck the **View release notes** box and then click on the **Run Recuva** button to start Recuva.

   This completes our installation, and you should now also see the Recuva icon on your Kali Linux desktop, similar to the following screenshot.

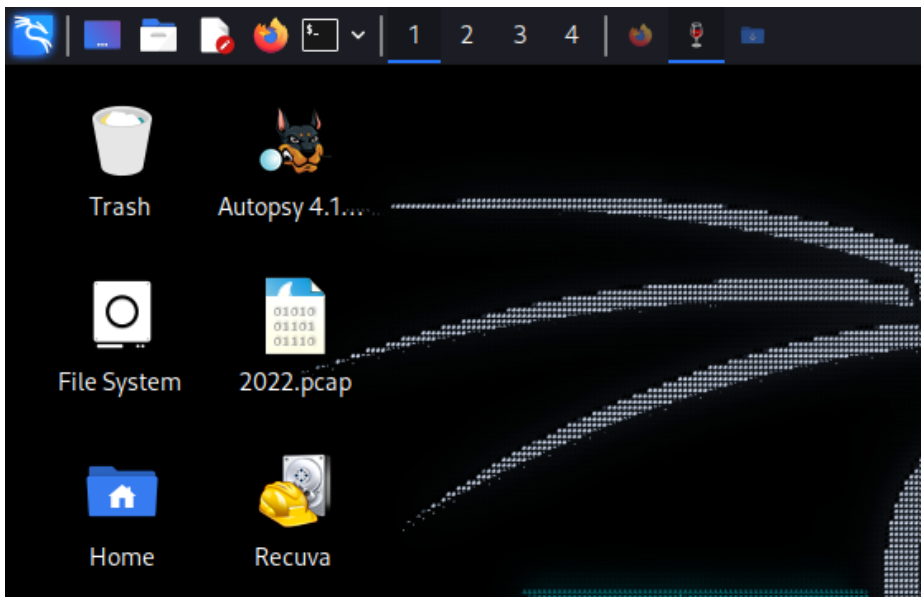


Figure 5.21: Recuva desktop shortcut

8. Click on the **Next** button to close off the installation screen. Notice in the following screenshot that there is a wine glass in the top-left corner of the screen, indicating that Recuva is being installed using Wine.
9. Although we will look at using Recuva in a later chapter, you can run Recuva as a final test to ensure that the program and Wine were both successfully installed.
10. Once you see the following screenshot, it means that both Wine and Recuva were installed successfully:

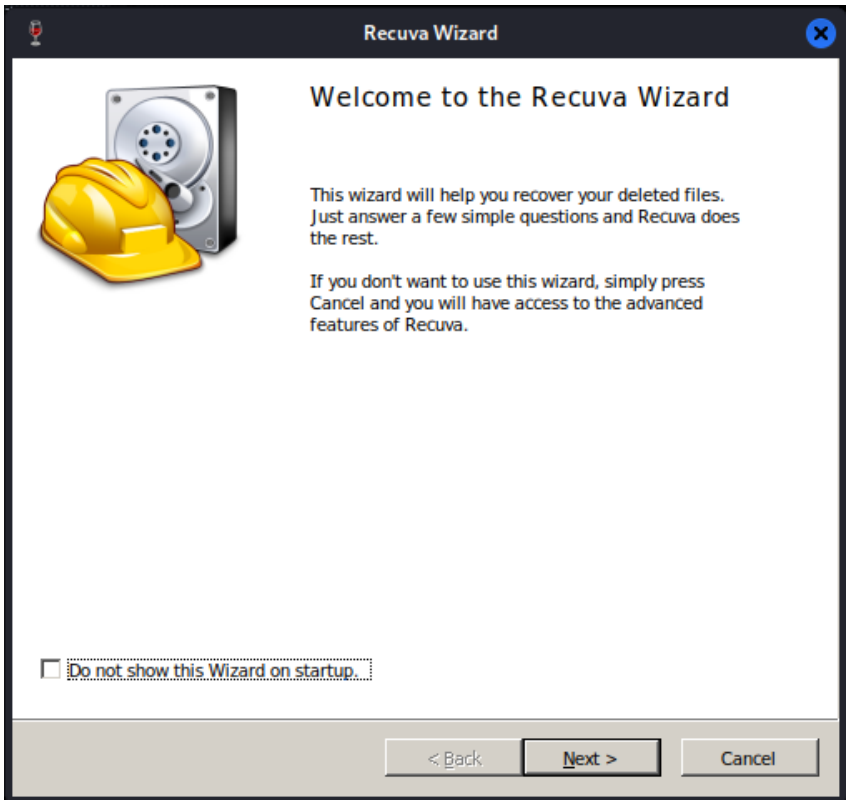


Figure 5.22: The Recuva scan configuration interface

At this point, you can click on the **Cancel** button to cancel running Recuva, as we will configure and run Recuva later on in *Chapter 9*, *File Recovery and Data Carving Tools*, where we will look at **File Recovery** and **Data Carving** tools.

We can now view our Wine installation and configuration as a success, as we were able to install and run the Recuva tool, which was built for Windows, within Kali Linux.

Summary

In this chapter, we learned about Wine and some great advantages of using the software for our DFIR purposes. We also learned how to install the latest stable version of Wine via the Terminal in Kali Linux, which allows us to run Windows applications in our Kali Linux system, along with the Microsoft .NET Framework, which is required by Windows to run programs and applications. We also looked at how to choose between Windows operating systems that Wine mimics, as some programs may only run on newer versions of Windows. Finally, we tested our Wine installation by downloading the Recuva file-recovery tool, which was built specifically for Windows, and then installed Recuva on our Kali system.

Before we continue any further on our exciting journey with Recuva, we'll next look at some fundamental knowledge on understanding file systems, which will cover the basics and background knowledge required for file recovery. See you in the next chapter!

Part 2: Digital Forensics and Incident Response Fundamentals and Best Practices

In this section, you will learn about the **Digital Forensics and Incident Response** (**DFIR**) processes and best practices for conducting investigations inclusive of the Order of Volatility, Chain of Custody, Evidence Acquisition, and other incident procedures, to be able to conduct your own investigations in accordance with international best practices.

This part has the following chapters:

- *Chapter 6*, *Understanding File Systems and Storage*
- *Chapter 7*, *Incident Response, Data Acquisitions, and DFIR Frameworks*

6

Understanding File Systems and Storage

It takes a lot more than just technical know-how to be a digital forensic investigator. There's a lot of research, processes, and analytics that also go into the case itself. Consider a scenario where you need to build a house. Sure, we need wood, nails, cement, metal, glass, and all other raw materials, but we also require skilled laborers and contractors to construct the structure and piece it together. Apart from the materials, tools, and resources, we will have also done our research to ensure that we understand what is needed for this to be a successful project.

For instance, we will have had to obtain permits to build, perform soil analysis, consider the weather, and then choose to specify types of materials based on the weather, location, soil type, and so on. It goes without saying that there must be an understanding of fundamental concepts in the field to efficiently carry out the task. In the same way, we need to understand the file systems, **operating systems** (**OSs**), data types, and locations, as well as a thorough understanding of methods and procedures for preserving data, storage media, and general evidence.

In this chapter, we will learn about the following topics:

- The history of storage media
- File systems and operating systems
- Data types and states
- Volatile and non-volatile data and the order of volatility
- The importance of RAM, the paging file, and the cache in DFIR

History and types of storage media

The aim of any investigation is to prove whether something previously existed, currently exists, or took place. In laptops, desktops, mobile devices, and smart devices, data must be stored somewhere, even if it's just temporarily. Most of us may be familiar with **hard disk drives** (**HDDs**) within laptops, desktops, mobile devices, and so on, but we also need to focus on removable and portable storage devices regardless of age, as they may still be used. These include DVDs, portable drives, thumb or flash drives, **secure digital** (**SD**) and microSD cards, older media such as CDs and floppies, and many other media types.

We should also consider that many portable storage devices, such as flash drives, come in many interesting shapes and sizes as novelty items and may not take the usual shape of the ordinary rectangular-shaped drive. Another issue to consider is that many of these storage media devices have changed in size over the years and may be smaller, usually because of evolving technology.

IBM and the history of storage media

There can never be a story, journal, book, or even discussion on the history of hard drives and storage media without mentioning three letters: IBM. We're all familiar with this well-known tech giant, but we might not all be familiar with some of its great achievements.

International Business Machines (**IBM**) has been around for quite some time. Known as the **Computing-Tabulating-Recording** (**CTR**) Company back in the early 1900s, IBM is better known for building the very first HDD, the first PC, along with its servers, desktops, and laptops.

Between 1956 and 1957, IBM made major inroads with the development and release of the 305 **Random Access Method of Accounting and Control** (**RAMAC**), which utilized the first disk storage technology. This revolutionary technology weighed in at approximately one ton and was roughly 16 square feet in size. The disk space capacity of this behemoth; however, was only 5 MB (megabytes, yes, I said megabytes) in size.

Although 5 MB by today's standard is roughly the size of a high-definition photo taken with a mobile device, all things considered, this really was a monumental achievement for its time. Before IBM's invention, data was stored on punch cards that could amount to as many as millions of cards just to hold a few megabytes.

A major issue faced back then with the introduction of this digital storage was the size of the device. Transportation by plane and truck may not have been an option for many; the space to store this would also have been an issue.

As technology progressed, IBM announced a much more portable computer in 1975, released as the IBM 5100 Portable Computer. In the 1980s, specifically 1981, we saw the birth of the IBM Personal Computer. Weighing in at much less than its predecessor, this portable computer also had a much more affordable price tag of between $8,000 and $20,000.

It wasn't until 1981 when IBM released the first personal computer, that the portability of computers became an actual reality. With a price tag of $1,565, owners were afforded a keyboard and mouse, with options for a monitor, printer, and floppy drives. Apart from the floppy drives, this was the standard for today's personal computers.

Along with this newer and more portable technology, there were also improvements in data storage media over the years, which saw advancements from magnetic tape storage to floppy disks and diskettes, CDs, DVDs, Blu-ray Discs, and, of course, mechanical and **solid-state drives** (**SSDs**).

Removable storage media

Continuing with our topic of storage media, I'd first like to start by discussing removable storage media, as it played a role just as important as that of fixed storage media in today's world.

Magnetic tape drives

Introduced by IBM in the 1950s, **magnetic tape** was an easy and very fast way to store data at a speed equal to its processing time. The IBM 726 magnetic tape reader and recorder was one of the first devices to offer this storage, with a capacity or tape density of 100 bits per linear inch of tape. An inch of tape should give an indicator of the size of the tape, which was wound on a large wheel, similar to an old film roll movie tape.

With magnetic tape media, data is written across the width of the magnetic-coated plastic strip in frames separated by gaps consisting of blocks. Magnetic tape is still very much used today, and like many other storage media types, it has significantly decreased in size while increasing capacity and speed.

To give an idea of how far magnetic tape storage has come as of 2017, IBM has developed newer tape storage media with a tape density of 200 **gigabits per second** (**Gps**) per inch on a single cartridge, which can record up to 333 **gigabytes** (**GB**) of data. These cartridges (for older folks like myself) are the size of a cassette tape or (for the younger ones) not much smaller than the average smartphone, which fits in your hand.

Floppy disks

The floppy disk, introduced yet again by IBM, was first seen along with its floppy disk drive in 1971. Although mainframe computers back then already had hard drives and magnetic tape storage media, there was the need for a simple and cheaper means of saving and passing on software and instructions to the mainframes, previously done using the much slower punch cards.

At the core of the floppy disk was a small magnetic disk which, although far more portable than magnetic tape storage and HDDs at the time, stored much less than other media we've mentioned.

The evolution of the floppy disk

This is how floppy disk evolved in size and capacity over the years:

- **Size: 8-inch**:
  - Year introduced: 1971
  - Maximum capacity: 80 **kilobytes** (**KB**)
- **Size: 5.25-inch**:
  - Year introduced: 1976
  - Maximum capacity: 360 KB
- **Size: 3.5-inch**:
  - Year introduced: 1984
  - Maximum capacity: 1.2 **megabytes** (**MB**)

> **Note**
> In 1986, the capacity of the floppy was increased to 1.44 MB, which remained as such until it was discontinued by Sony (the last remaining manufacturer of the floppy) in 2011.

Optical storage media

Optical storage media is so called because of how data is written to the various media types involving the use of different types of lasers on the surface of the disk itself.

Although it may be somewhat difficult to distinguish various optical disks in the event that there are no default labels on them, they do have slightly varying differences in color and hue due to the size of lasers used in writing data to them.

Compact disks

Compact disks (**CDs**) are made of pits and land noticeable as bumps on the underside of the disk, coated with a thin layer of aluminum that results in a reflective surface. Data is written in concentric circles further split up into sectors of 512 bytes, each known as tracks on the CD from the inside to the outside (or edge) of the disk:

- Diameter: 120 **millimeters** (**mm**)
- Type of laser used to write data: 780 **nanometers** (**nm**) infrared laser
- Maximum capacity of a CD: 650–700 MB

There are two broad categories of CDs:

- **Compact Disk – Read-Only Memory (CD-ROM)**: This disk comes with data on it in the form of programs, games, music, and so on, and can only be read from a **Compact Disk Recordable** (**CD-R**). Data can be written to this disk, but only once.
- **Compact Disk – ReWritable (CD-RW)**: Data can be written to this disk many times.

Digital versatile discs

Digital versatile discs (**DVDs**), although the same size in diameter, can store much more data than CDs:

- **Diameter**: 120 mm (same as a CD)
- **Type of laser used to write data**: 650 nm red laser
- **Maximum capacity of a DVD**: 4.7 GB and 15.9 GB (dual-layer DVD)

The various types of DVDs are as follows:

- **DVD – Read-Only Memory (DVD-ROM)**: This DVD comes with data already written to it, much like a CD-ROM.
- **DVD – Recordable (DVD-R)**: Data can be written once to the DVD.
- **DVD + Recordable (DVD+R)**: Data can be written once to the DVD. DVD+Rs utilize more advanced error detection and management technology.
- **DVD – ReWritable (DVD-RW)**: Data can be written to the DVD several times.
- **DVD – Recordable Dual Layer (DVD-R DL)**: This type of DVD contains dual layers resulting in higher storage capacities between 7.95 GB on a DVD-9 disk and 15.9 GB on a DVD-18 disk.
- **DVD – Recordable Dual Layer (DVD+R DL)**: Same as the DVD- R DL, but has been argued to be the more efficient format, resulting in fewer errors.
- **DVD – Random-Access Memory (DVD-RAM)**: Mainly used in video recording equipment due to its resiliency (lasting up to two decades) and the ability to rewrite data onto it. This disk is more expensive than other DVD formats and is also not compatible with many common DVD drives and players.

Blu-ray Disc

The current standard for removable disk media, the Blu-ray Disc gets its name from the color of the laser used to read from and write to the disk. Due to the high-capacity storage of Blu-ray Discs, **high-definition** (**HD**) content can easily be stored on Blu-ray Discs without loss of quality:

- **Diameter**: 120 mm (same as a CD and DVD)
- **Type of laser used to write data**: 405 nm blue laser
- **Maximum capacity of a DVD**: 25 GB and 50 GB (dual-layer Blu-ray)
- **Data Transfer**: up to 72 **megabits per second** (**Mbps**)

Both capacities of Blu-ray Discs also come in rewriteable and recordable formats:

- **BD-RE**: Blu-ray Disc Rewritable
- **BD-RE DL**: Blu-ray Disc Rewritable Dual-layer (50 GB)
- **BD-R**: Blu-ray Disc Recordable
- **BD-R DL**: Blu-ray Disc Recordable Dual-layer (50 GB)

Flash storage media

Flash memory is so named because the data is written to and erased using electrical charges. You may have perhaps heard someone say that they've had to flash their mobile device. This is quite similar to erasing flash storage media on smartphones and smart devices, except devices with OSs such as Android and iOS require a much more extensive procedure for flashing and reinstalling their OSs. However, the end result is very much the same in that the memory and storage areas are reset or wiped.

Flash storage chips come in two types, known as **NOT AND** (**NAND**) and NOR flash memory, and are responsible for high-speed and high-capacity storage of data on flash storage media. They are newer types of **electrically erasable programmable read-only memory** (**EEPROM**) chips that can wipe blocks of data or the entire drive, rather than just one byte at a time, as with the slower EEPROM. This type of flash memory chip is non-volatile, meaning that the data is still stored on the chip even after power to the chip is lost. Data is erased when specific instructions are sent to the chip in the form of electrical signals via a method known as in-circuit writing, which alters the data accordingly.

The following photo shows one of my old 1 GB (left) flash drives and a newer 32 GB flash drive with its NAND chips exposed (right). Notice that even though the chips are similar in size, the one on the right has 32 times the data capacity, as they were manufactured over a decade apart:

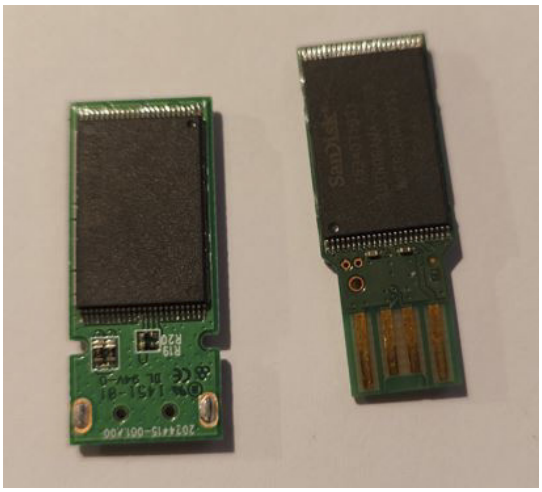


Figure 6.1 – NAND chips on flash drive boards

Flash media storage has so far become the ultimate in portability, with many types ranging from the size of your thumb to the size of the nail on your little finger. The lifespan of flash storage all depends on the usage, as they all have an average read-write usage, sometimes displayed on the packaging of the device. The read-write speeds are also some of the fastest at this point, which is why HDDs have moved away from the traditional mechanical disk mechanism to a solid-state one. SSDs will be discussed later in this chapter.

> **Note**
>
> With flash storage media capacities ranging from 2 GB to 1 **terabyte** (**TB**) and more particularly on SD, microSD, and flash drives, these can now act as very fast removable drives with OSs installed on them and can even be partitioned using various tools. So yes, indeed, Kali Linux most certainly can be installed onto a flash drive, SD, or microSD card (and be made bootable) with as little as 8 GB of storage space.

USB flash drives

The **Universal Serial Bus** (**USB**) port or interface, released in 1995, has become the standard for all devices, replacing older devices that would have connected to specific parallel ports on a computer. It's quite common to see almost any device or peripherals connected to a computer via a USB connection, including mice, keyboards, flash drives, printers, scanners, cameras, mobile devices, and just about every other device.

This table shows the evolution of the USB port:

USB version	Released year	Data transfer speed
USB 1.0 and 1.1	1995	12 Mbps
USB 2.0	2000	480 Mbps
USB 3.0	2008	5 Gbps
USB 3.1	2013	10 Gbps

USB flash drives come in all shapes and sizes today, from the standard rectangular shape to every shape imaginable. USB flash drives use NAND EEPROM chips to store their data, and today are available in various versions that define the read/write speeds of the flash drive.

The following picture shows various flash drives ranging from oldest to newest, left to right. The first three drives are all USB 2.1; however, the first two are 8 GB flash drives, and the third, which is significantly smaller, is a 32 GB flash drive. The fourth drive (Corsair) is a 64 GB USB 3.1 drive:

Figure 6.2 – Flash drive sizes

Important note

I should give special mention to the elephant in the room here, the novelty flash drive, which can easily pass as a keychain or toy, and may actually pose a threat to organizations that do not allow employees to bring to work or leave with flash drives due to the sensitive nature of the data within the organization.

Flash memory cards

Like flash drives, flash memory cards (or memory cards, as they are fondly referred to) also use NAND flash memory, which, as we previously learned, is non-volatile, solid-state memory. However, unlike USB flash drives, these cards do not come with a USB interface and must be used with either an adapter or a memory card reader.

Over the years and decades even, we've had several formats of memory cards grace our desktops, laptops, mobiles, and other devices, including cameras, MP3 players, and even toys. Although I'll only cover some of the more popular cards of today, it is important that you are at least familiar with memory cards and are also able to identify them.

The following are flash memory card types:

- **Memory Stick PRO Duo** (**MSPD**) (a proprietary card developed by Sony)
- SD
- MicroSD
- **Compact Flash** (**CF**)
- **MultiMedia Card** (**MMC**)
- **xD-Picture** (**xD**)
- **Smart Media** (**SM**)

Of the aforementioned types, I've opted to show three from my collection in the following photo. The card at the top is a Samsung SD card. SD cards typically have a sliding lock to the side, used to prevent data from being overwritten. To the middle and bottom, I have two Sony Memory Stick PRO Duo cards, which may be rare at this point in time but are fully functional:

Figure 6.3 – SD card and PRO Duo card comparison

I'd like to do a brief comparison of the PRO Duo cards, the SD card, and the microSD card shown in *Figure 6.4*. Developed at least a decade apart, the older PRO Duo cards are larger, with a capacity of 2 GB and 4 GB, respectively. Although not seen on the SD card in *Figure 6.3*, its capacity is 1 GB, again showing that with the evolution of technology, smaller chips are able to hold far more data than their predecessors.

To drive this point home, let's look at the photo in *Figure 6.4* (although a bit blurry due to the zoom from my camera) to get a close-up of the two microSD cards. The microSD card to the left has a capacity of 64 GB and is specified as a Class 10 U1 card. The Samsung microSD card to the right is a whopping 512 GB, Class 10, U3 card, which is as small as a fingernail! Still, microSD cards are being developed with even larger capacities of 1 TB and larger:

Figure 6.4 – MicroSD card comparison

The various classes of microSD cards identify their read/write speeds and suggested uses. I do suggest getting a Class 10 microSD card if purchasing one, as the C10 is much faster than the other classes (2, 4, and 6) and supports constant HD and even 4K video recording.

Let's have a look at the four Class speed ratings:

Class (C)	Minimum Write Speed (MBps)
C2	2
C4	4
C6	6
C10	10

There is also an **Ultra-High Speed** (**UHS**) class, recognizable by the U symbol on microSD cards.

UHS (U) Class	Minimum Write Speed (MBps)
U1	10
U3	30

To cater to large files, especially for 4K and 8K video footage, there is also a Video Speed Class rating:

Video Speed Class	Minimum Write Speed (MBps)
V6	6
V10	10
V30	30
V60	60
V90	90

As mentioned previously, flash memory cards require card readers, which connect to laptops, desktops, and other media players using USB ports. The photo in *Figure 6.5* shows one of my many card readers, which supports CF, Memory Stick PRO Duo, SD, and even the SM cards, and the newer card reader in *Figure 6.6* is an SD and microSD card reader with a USB and Type-C slot:

Figure 6.5 – SD card, CF, and PRO Duo card reader

Figure 6.6 – USB 3.0 and Type C SD card and microSD card reader

Hard disk drives

Now that we've had a good look at non-volatile storage, including tape storage and flash storage, let's go a bit deeper into the world of HDDs, which serve as fixed storage media. I'll try to keep things simple and short by focusing mainly on the knowledge necessary for forensics investigators.

HDD technology has certainly come a long way from the monstrous storage devices first seen in IBM mainframes and is now more compact, faster, and more affordable, with capacities in the terabytes.

Although the newer solid-state drives use the same type of memory found in flash memory devices, they are still a bit costly when compared to mechanical drives. This perhaps may be one of the contributing factors when wondering why older mechanical drive technology is still being used. Mechanical drives consist of moving parts, including platters, an actuator arm, and a very powerful magnet. Although it is very common to still find these mechanical HDDs in today's laptops and hard drives, they are much slower than the newer solid-state drives, which have no moving parts and look very similar to the chipset of a USB flash drive.

In your forensics investigations and adventures, you may come across or be presented with older HDDs that can have different interfaces and use different cable technologies to connect to motherboards. Let's have a look, shall we?

Integrated Drive Electronics HDDs

Many of the first PCs in the mid-1980s were outfitted with hard drives that used **Parallel Advanced Technology Attachment** (**PATA**) and **Integrated Drive Electronics** (**IDE**) technology. As with all older devices back then, parallel transmission (sending multiple bits simultaneously but slowly) was the order of the day, allowing for very limited throughput. An easy way to identify older IDE drives is to simply have a look at the interface where the data and power cables connect to the drive.

These older drives, as seen in the following photo, have four pins for power, which connect to a Molex connector, separated by eight pins used to set the device as a master or slave device, and then 40 pins for the IDE data cable, which transmits the data to the motherboard:

Figure 6.7 – 3.5-inch IDE hard drive

In 1994, advancements in technology led to the release of **Enhanced Integrated Drive Electronics** (**EIDE**), which saw an increase in the number of pins for the data cable from 40 to 80, also increasing the transmission speeds from 4 Mbps to a possible 133 Mbps.

IDE/EIDE was still, however, limited to a maximum of four IDE/EIDE drives per computer, as the jumper pins on the drive only allowed for two primary and two secondary drives, set in a master and slave configuration. Consideration also had to be given to the fact that CD-ROM and RW devices and DVD-ROM and RW devices were also using IDE/EIDE technology at that time.

Serial Advanced Technology Attachment HDDs

In 2002, Seagate released an HDD technology called **Serial Advanced Technology Attachment** (**SATA**), which used serial transmission instead of a slower parallel transmission. Whereas PATA drives speeds of 33/66/133 Mbps, SATA drives boasted speeds of 150/300/600 Mbps. This meant that the lowest SATA transmission speed of 150 Mbps was faster than the highest PATA speed of 133 Mbps.

The connector interfaces of the SATA drives were also different, but it was common at the time to see SATA drives with connectors for both SATA and PATA power cables for backward compatibility.

SATA data cables are much thinner than PATA cables, as they only contain seven wires connecting to seven pins. SATA devices use one cable per drive, unlike PATA devices, which connect two drives on one IDE/EIDE cable connected in a master/slave configuration.

SATA continues to be standard today for drive technology for both desktops and laptops and has had several revisions, as listed here. Speeds listed are in MBps and not Mbps:

- **SATA 1**: 150 MBps
- **SATA 2**: 300 MBps
- **SATA 3**: 600 MBps

The following photo shows two 2.5-inch laptop drives. The drive to the left is a much older IDE drive, and the one to the right is a modern SATA drive:

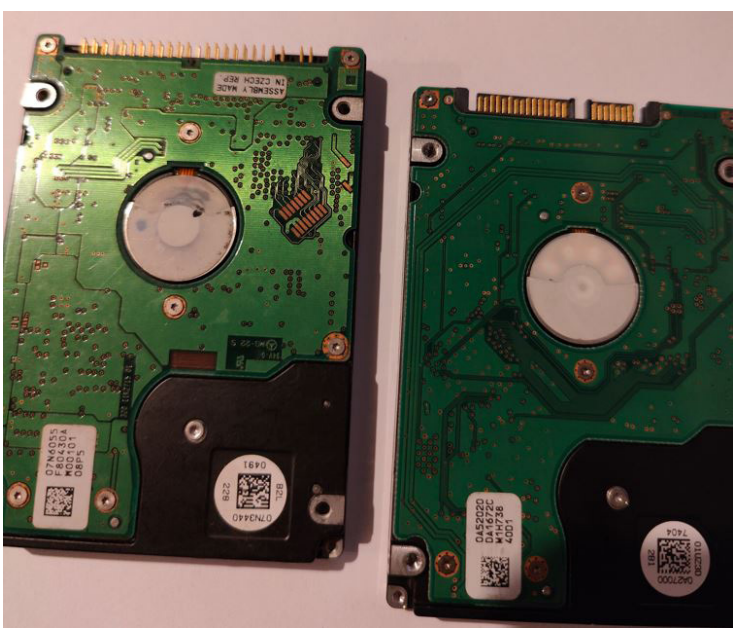


Figure 6.8 – 2.5-inch IDE (left) and SATA

The following photo shows two SATA laptop 2.5-inch drives. The one to the left is damaged and has been opened for us to see the circular platter in the middle with the actuator arm at the top, slightly positioned over the platter. At the end of the actuator arm is a read/write head, which reads and writes data to the platter.

The drive to the right in the photo is a hybrid drive or a **Solid-State Hybrid Drive** (**SSHD**). This is a mechanical drive like the one to the left, but it also has flash memory in it to allow for faster access to the data on the platters:

Figure 6.9 – Platters and actuator arm, which are housed in mechanical hard drives

Solid-state drives

As briefly mentioned, SSDs are non-volatile storage media and use NAND flash memory in an array to hold data. SSDs have been around for quite some time; however, mainstream use has been greatly hampered by the high cost of the drive. Samsung first released a 32 GB SSD with a PATA interface in 1996, followed by SanDisk's 32 GB SSD, but with a SATA interface.

Although SSD drives use flash memory, the materials used are more high-end than that found in flash drives, which makes them highly preferable for use as a hard drive, but again, this contributes to the very high cost.

Some advantages come from the fact that there are no moving parts in an SSD. No moving parts make the SSD more durable in the event of a fall or swift foot to the PC tower, as there are no platters or actuator arms to be scratched or struck. Also, the faster read/write speeds and access times greatly reduce the time taken for the device to boot or start and even give an enhanced experience when using resource-intensive software and games.

As far as digital forensics go, SSDs are still a relatively new technology that will be constantly improved upon for some time to come. It's important to remember that you are not dealing with a mechanical drive and to remember that data on an SSD, much like a flash drive or memory card, can be lost or wiped within minutes or even seconds. Although traditional tools can be used to image and recover data from SSDs, I strongly suggest researching any SSD drive before performing any forensic activities to get a better understanding of its workings and complexities, such as de-chipping and wear-leveling algorithms. De-chipping is a method of reading data from memory chips and wear-leveling involves techniques used to preserve or extend the lifespan of an SSD.

Here's a photo of a 250 GB SSD:

Figure 6.10 – SSD

Take note of the pin layout interface for the SSD connector (left side), which connects to a **Peripheral Component Interconnect Express** (**PCIe**) interface on the board instead of the usual SATA connectors. This connector in the previous photo is an M.2 **Non-Volatile Memory express** (**NVMe**) SSD connector type, but there are other types as well. When performing forensic acquisitions on SSDs that may require the use of a USB adapter, be sure to know which connector you are working with.

Types of SSD interface types include the following:

- M.2 NVMe (up to 32 Gb/s bandwidth)
- M.2 SATA (up to 6 Gb/s bandwidth)
- mSATA (up to 6 Gb/s bandwidth)

We're now up to speed on different types of HDDs and their differences. Let's move on to file systems and OSs that can be installed on the drive types we've just read about.

File systems and operating systems

Now that we've covered the physical aspects of drives, let's get logical! Any and every type of storage media needs to be formatted with a particular file system. The file system chosen will also determine which OS can be installed on the medium, along with file and partition sizes.

A simple way to think of this is to imagine a blank sheet of paper as any type of new or wiped storage media. We can put several types of information on this piece of paper, but we'll first want to organize or prepare the sheet of paper in a way that makes our data easy to understand, access, and even store. We can choose to write on it, from left to right, in sentences and paragraphs in English, or we can perhaps create tables using rows and columns. We can even use printed slides to display our data or use images, graphs, and flowcharts. Additionally, we can format your storage media to best suit the data that will be stored and used.

File systems ensure that the data is organized in such a way that it can be easily recognized and indexed. Consider the storage space within a filing cabinet with multiple compartments. Some may be used specifically for storing files in alphabetical order, others in chronological order, and some compartments for stationery supplies, miscellaneous, and even random items. Although all used for storing different items, they can all be labeled and easily recognized and organized in such a way that the contents of each compartment can be easily accessed or even removed.

To install any OS on a hard drive or removable storage media, the device must first be formatted and prepared for the OS by choosing the appropriate file system. Windows, macOS, Android, Kali, and so on, all have file systems that organize the storage medium so that the OS can be successfully installed.

Some of the more popular OSs and their file systems are included in the following sections.

Microsoft Windows

Here are some brief features of the Microsoft Windows OS:

- File system: **New Technology File System** (**NTFS**)
- Supported versions: Windows 11, 8, 7, Vista, XP, 2000, NT, and Windows Server 2022, 2019, 2016, 2012, and 2008
- Maximum volume size: 8 **petabytes** (**PB**) on newer versions of Microsoft Server (2019 and newer) and Windows 10 version 1709 and newer. Older versions support up to 256 TB
- Maximum supported file size: 256 TB (depending on cluster size)
- NTFS features: Compression, **Encrypted File System** (**EFS**), disk quotas

Info

Older versions of Microsoft Windows supported the **File Allocation Table** (**FAT**) file system by default. Newer versions of Windows also support FAT and FAT32, but with drive size limitations (8 TB) and file size limitations (4 GB). **Extensible File Allocation Table** (**exFAT**) was created to remove the limitations of FAT32 but may not be as widely supported as FAT32.

Macintosh (macOS)

Here are some brief features of the **Macintosh OS** (**macOS**):

- File system: **Hierarchical File System** (**HFS+**)
- Supported versions: macOS up until version 10
- Maximum volume size: 2 TB
- Maximum supported file size: 2 GB

Apple has since advanced to a newer 64-bit files system called **Apple File System** (**APFS**), which replaces HFS+ and is optimized specifically for SSDs and iOS devices.

Here are some brief features of the APFS:

- Supported versions: macOS 10.13 (and newer) and iOS 10.3 and newer
- Maximum supported file size: 8 **exbibytes** (**EiB**)

Linux

Here are some brief features of the Linux File System:

- File system: **Fourth Extended File System** (**ext4**). Several file systems are available for Linux, but I recommend this one if you are uncertain as to which should be used.
- Supported versions: Red Hat, Kali, Ubuntu, and so on.
- Maximum volume size: 1 EiB.
- Maximum supported file size: 16 **tebibytes** (**TiB**).

Important note

Many open source OS distros are based on Linux, including Kali Linux and Android, and so they use the ext2/ext3/ext4 filesystems. They are also able to use the FAT32 filesystem. FAT32 can be used across any platform, including older versions of Windows, Macintosh, and Linux, and is supported by almost any device with a USB port.

For the purpose of **Digital Forensics and Incident Response** (**DFIR**) it is important to know about the file systems and their differences that we read about. Once a file system is chosen (either by the drive manufacturer or user), data can then be stored on the media. Let's now look at different data types and states in the next section.

Data types and states

Firstly, there's **data in transit**, also called **data in motion**. These simply describe data on the move, perhaps traversing across the network between devices or even between storage media, actively moving between locations.

Then there's **data in use**. Data in this state is currently being accessed by a user or processed by a CPU. When data is accessed or used, it's pulled from the hard drive and temporarily stored in RAM, which is much faster than the hard drive (particularly mechanical drives), and stored there for as long as the user accesses it and there is power to the device.

When data is not in motion, transit, or in use, it is described as **data at rest**. In this state, the data rests or resides on non-volatile media, such as hard drives, optical media, flash drives, and memory cards.

Now that we know about the formal terms for describing data states, we will learn about metadata and slack space next.

Metadata

Metadata is simply data about data. Take an item such as a laptop stored in a warehouse, for example. Somewhere in the warehouse (and possibly in other locations such as the cloud), there may be several pieces of information that can be referred to as data about the laptop, or even laptop metadata, such as the following:

- Location of the laptop within the warehouse
- Laptop brand and model
- Manufacture date
- Warranty dates and information
- Hardware and software specs
- Color and size

Slack space

Clusters are the smallest amount of disk space or allocation units on storage media that store data. When formatting drives, we need to define the size of these allocation units, or we can use the default cluster size of 4 KB. This is where slack space comes in.

Slack space is the empty and unused space within clusters that contain data but are not filled with data. To fully understand this, we first need to understand the default cluster sizes specified by OSs. A drive formatted using NTFS (for Windows) has a default cluster size of 4 KB. Let's say that you've saved a text file to your disk with a file size of 3 KB. This means that you still have 1 KB of unused or slack space within that cluster.

Slack space is of particular interest to a forensic investigator as data can be easily hidden in slack space. Luckily, we have several tools available, such as the Sleuth Kit and Autopsy, within Kali Linux to help investigate slack space and find hidden files.

Be sure to familiarize yourself with the terms metadata and slack space, as they will constantly be referred to throughout your investigations. Feel free to also make notes on these and other terms as we progress throughout the chapter for easy reference. Let's now move on to get an understanding of the volatile and non-volatile nature of different types of memory and also look at the order of volatility in DFIR investigations.

Volatile and non-volatile data and the order of volatility

In this section, we look at why data is lost when power to the volatile memory is lost.

Data can exist if the media it is stored on can store the data. Hard drives (mechanical and solid-state), flash drives, and memory cards are all **non-volatile storage** media. Although SSDs offer and continue to make drastic improvements in data access times, RAM (typically referred to only as memory) thus far remains the faster type of memory inside devices.

RAM, however, is **volatile memory**. Unlike non-volatile memory found in hard drives and flash drives, data stored in RAM is kept there temporarily, only for as long as there is an electrical current being provided to the chips. There are two types of RAM that we need to be aware of: **Static RAM** (**SRAM**) and **Dynamic RAM** (**DRAM**).

SRAM is superior to DRAM but is far costlier than DRAM because of the extensive materials used in building the chips. SRAM is also physically much larger than DRAM. SRAM can be found in the CPU cache (L1 or Level 1) and on some chips on the motherboard (L2/L3), although in very small sizes (KB) due to the cost and physical size.

Although DRAM is slower, it is much cheaper and remains one of the reasons for its use as the main memory in devices. What makes RAM volatile is its components, such as transistors and capacitors. Some of you may already be familiar with this topic from certification courses such as A+, but for the benefit of all of you, allow me to go into a bit more detail.

DRAM uses capacitors, which store electrical charges temporarily as part of a refresh circuit. The chips need to be constantly refreshed to hold the data while being accessed. However, between refreshes, a wait state is created, which makes DRAM slower when compared to SRAM as it uses transistors instead of capacitors, which do not have wait states.

Over the decades, there have been many types of DRAM or memory sticks in slightly varying sizes and increased pins with which to contact the motherboard. Some of the RAM types in order of age are as follows:

- **Extended Data Output RAM (EDO RAM)**: One of the earlier types of DRAM.
- **Synchronous Dynamic RAM (SDRAM)**: This was the first form of RAM to synchronize itself with the CPU clock speed. It has a maximum transfer rate of 133 **megatransfers** or **millions of transfers per second** (**MT/s**). It is labeled PC100, PC133, and PC166.
- **DDR-SDRAM/DDR 1 (Double Data Rate - SDRAM)**: Effectively doubled the transfer rate of SD RAM. Maximum transfer rate of 400 MT/s.
- **DDR2**: Maximum transfer rate of 800 MT/s.
- **DDR3**: Consumes up to a third less power than DDR2. Maximum transfer rate of 1600 MT/s.
- **DDR4**: Maximum transfer rate of 4400 MT/s (**DDR4-4400**).

In today's laptops and desktops, one might mainly come across DDR3 and DDR4, but it may not be uncommon to run into a legacy machine with DDR2 or (miraculously) DDR1. The following photo shows different RAM **Dual Inline Memory Modules** (**DIMM**) types from top to bottom, SDRAM (top), DDR1, DDR2, and lastly DDR3:

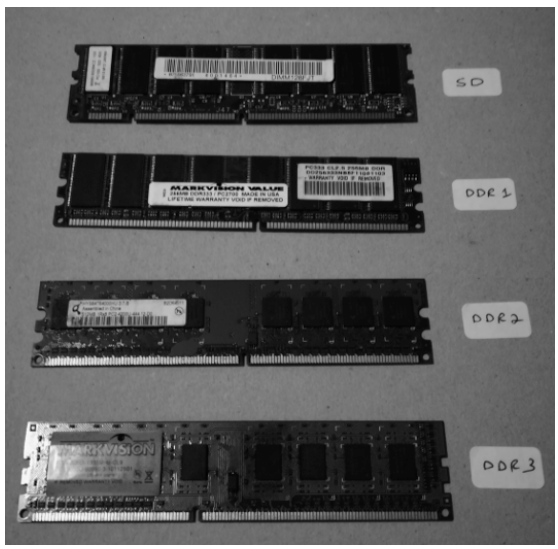


Figure 6.11 – Types of RAM

Important note

Laptops also use DDR RAM but are available in a more compact size called **Small Outline DIMM** (**SODIMM**) modules.

Now that we've looked at the evolution of RAM and its various types of form factors, let's look at the importance of RAM and learn about the uses of the paging file and RAM cache in the next section.

The importance of RAM, the paging file, and cache in DFIR

OSs can use a portion of the hard disk as an extension of RAM. This is referred to as virtual memory and is usually a good idea if a computer or laptop has limited RAM. Although the hard drive is much slower than the RAM, the swap file or paging file on the disk can store files and programs that are being accessed less, leaving the RAM available to store data that is frequently accessed. This process involves the OS swapping pages of data less frequently used and moving data to the dedicated paging file area on the hard drive.

In forensics investigations, the paging file is very important to us. Although not as volatile as RAM itself due to being stored on the hard disk, it is a hidden file in Windows called `pagefile.sys`, and should always be inspected using tools of your choice, as this file may reveal passwords for encrypted areas, information from sites visited, documents opened, logged-in users, printed items, and so on.

Data on mechanical drives is stored in a fragmented manner; however, the advantage of the paging or swap file is that the data can be stored in a contiguous manner, one piece after the next, allowing for faster access times.

It is recommended that the size of the paging file is set to 1.5 times the amount of memory and be stored on a separate drive if possible, not just a separate partition.

Note on Pagefile.sys

`Pagefile.sys` can be found in the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management` Windows registry path.

I hope the importance of the paging file and cache was well received, as acquiring and analysis of the paging file in any DFIR investigation is crucial. Feel free to review the chapter and make notes on all the terms covered before proceeding to the next chapter.

Summary

In this chapter, we took the time to cover some of the basics of non-volatile storage media, which stores data even after there is no power supplied to the medium. Non-volatile media includes different types of HDDs, such as mechanical and solid-state PATA and SATA drives, flash drives, and memory cards.

Newer storage media devices, including SSDs, use a special type of flash memory called NAND flash to store data. This flash memory is far faster and more durable than traditional mechanical drives, as the devices contain no moving parts; however, they are still quite costly for now.

We also looked at various file systems associated with various OSs and saw that the smallest allocation of data is called a cluster, which can reside in slack space. Slack space is unused space within a cluster in which data can be hidden. Data itself has different states and can be at rest, in motion, or in use. Regardless of the state of the data, there always resides some information about the data itself, called metadata.

Any data accessed by the user or OS is temporarily stored in volatile memory or RAM. Although data can be stored for lengthy periods on non-volatile memory, it is lost when electrical charges to volatile memory (RAM) are also lost. An area of the hard disk called the paging file can act as virtual RAM, allowing the computer to think it has more RAM than actually installed.

I do encourage you to do more research and expand your knowledge on these topics, allowing you to gain more understanding of what was covered.

Let's now move on to the next chapter, where we'll learn about investigative procedures and best practices for incident response, such as acquiring volatile data and procedures for working with and analyzing live machines.

7

Incident Response, Data Acquisitions, and DFIR Frameworks

Sometimes, it's difficult to ascertain exactly what qualifies as evidence, especially at the very start of an investigation when all the facts on what occurred may not have yet been collected or stated. As in any investigation, we should be aware of and follow the guidelines, practices, and procedures for acquiring evidence in such a way that it is not tampered with or, in a worst-case scenario, lost.

At the scene of a crime, let's say a shooting, there are specific items that may immediately qualify as evidence. The physical evidence is easily collected, put into evidence bags, labeled, and then shipped off to the labs and secure storage areas for safekeeping. This evidence may include spent bullet casings, perhaps a gun, fingerprints, and blood samples. Let's not forget witness statements and **Closed Circuit Television** (**CCTV**) footage too. It's also of interest to consider the individuals from law enforcement agencies that would be at the scene and the order in which they may have arrived. Seems simple enough.

However, when a breach or crime involving a computer or smart device is reported, collecting the evidence is sometimes not as simple, as there are many factors to consider before labeling any items as evidence.

For example, if a desktop was involved in the act, do we take the tower alone, or do we also seize the monitor, keyboard, mouse, and speakers? What about the other peripherals such as printers and scanners? Are there any additional fixed or removable storage media at the scene, and do we also seize them?

This chapter answers all these questions and provides guidelines and best practices for incident response, evidence acquisition, and other topics, including the following:

- Evidence acquisition procedures
- Incident response and first responders

- Evidence collection and documentation
- Live versus post-mortem acquisition
- The **Chain of Custody** (**CoC**)
- The importance of write blockers
- Data imaging and maintaining evidence integrity
- Data acquisition best practices, guidelines, and DFIR frameworks

Evidence acquisition procedures

As we covered in the last chapter, data can be stored on both fixed and removable storage media. However, data can easily be deleted or completely lost depending on a multitude of factors that must be considered if we are to ensure the preservation of data. It might even be argued that there are more threats to digital storage than paper-based storage. The following lists detail some comparisons of threats to both.

Threats to paper-based storage include the following:

- Water
- Fire and humidity bugs
- Age
- Natural disasters—floods, earthquakes, tornadoes, hurricanes, and more
- Human-error and negligence

Threats to data on storage media include the following:

- Human-error and negligence
- Magnetism and electromagnetic fields
- Water and condensation
- Heat dust
- Impact Voltage
- Static electricity
- Natural disasters—floods, earthquakes, tornadoes, hurricanes, and more

So, when exactly does data become evidence? Specific data might have a value that is relative to an investigation when considering the events that transpired. This should be considered when performing DFIR as the scope might then extend beyond just devices and storage media, as we will discuss further in this chapter.

Let's move on to incident response and the technical roles assumed by first responders.

Incident response and first responders

Preserving evidence does not begin only at the acquisition of data, but as early on as the physical viewing of the suspect device. There should be some structured response to the suspected crime or breach in the same manner as a crime reported to the police. In the same way, a person makes a call to the emergency services who then dispatch the police, fire services, ambulance personnel, and other first responders who might then escalate the issue to the FBI or other agencies. There should also be a similar chain of command when dealing with reports that require digital investigation.

When a breach or crime is discovered or suspected, there should be a dedicated first responder who is alerted and called to the scene. Usually, this person has some knowledge or understanding of the workings of devices, networks, and even of the IT infrastructure in the organization if applicable.

First responder personnel can include the following:

- HelpDesk and IT technicians
- Systems administrators
- Network administrators
- Security administrators
- IT managers and directors

While the individuals in the preceding roles might not be trained or skilled in digital forensics or digital investigations, they will be responsible for securing the scene and ensuring that the data, peripherals, equipment, and storage are not used, tampered with, removed, or compromised by unauthorized individuals.

The duties of first responders include the following:

- Being the first to respond to the scene (as the name suggests) and making an initial assessment
- Documenting the scene and room fully in a circular fashion using the center of the room as a focal point
- Securing the scene by not allowing unauthorized users or suspects, access to the devices or area, and especially to volatile data and storage media
- Preserving and packaging evidence for transportation, ensuring the use of the CoC forms that allow for tracing the exchange of evidence between persons

Now, let's look at examples of evidence, its collection, and documentation.

Evidence collection and documentation

The documentation of the scene should also be performed by the first responders to aid in investigations. The documentation of the scene should include photographs, video, voice recording, and manual documentation of the following:

- The room that the device is located in (desk, ceiling, entrance/exit, windows, lighting, electrical outlets, and data drops)
- State of the device (on, off, power light blinking)
- Screen contents, if the device is on (operating system, running programs, date and time, wired and/or wireless network connectivity)
- Books, notes, or pieces of paper
- Connected and disconnected cables
- Peripherals such as printers, scanners, and external drives in close proximity (whether connected or not)

Once the scene has been secured and documented by the first responders, the forensic investigator should also be alerted and called in to further examine and assess the situation.

If the first responder has been trained in evidence collection and preservation, they can also begin the process of acquiring what can be considered physical evidence.

Examples of physical evidence include the following:

- Desktops and computer towers
- Laptops
- Tablets
- Smartphones
- IoT devices
- General smart devices
- Fixed and removable storage media:
    - Hard drives
    - Optical media
    - Tape storage
    - Memory cards
    - Card readers

- Adapters
- Docking stations
- Printers
- Scanners
- Mobile phones
- iPods

  Media and MP3 players and other devices that might have been used in carrying out the breach. Routers and switches might also contain evidence such as connectivity logs, source, and destination addresses, and even visited websites.

- Cables and chargers

Here's a budget but portable and very well-organized screwdriver kit that I keep in my DFIR first responder toolkit. It has all the attachments for opening desktops, laptops, and tablets, and also for removing and even opening removable storage media such as hard disk drives, optical drives, and even floppy drives if encountered:

Figure 7.1 – Toolkit with various tips for opening devices and drives

Physical acquisition tools

Apart from your toolkit such as the one shown earlier, what other tools would we need for the forensic acquisition and extraction of digital evidence? Remember when we covered the different types of storage media back in *Chapter 6*, *Understanding File Systems and Storage*? We saw that many of them had their own connectors of various sizes.

Here's a list of just some of the equipment required when performing evidence acquisitions:

- Write blockers (these can be hardware- and/or software-based)
- Card readers (preferably USB 3)
- Various adapters (USB to SATA and EIDE, USB to various types of USB, and VGA to HDMI)
- Device cables such as power, SATA, EIDE, HDMI, and VGA
- Networking cables straight-through, crossover, and console

The following photograph shows a collection of various USB adapters, all costing under $10, and they are all available on Amazon:

Figure 7.2 – Various USB adapters

For laptop drives, I also use a **Serial Advanced Technology Attachment** (**SATA**) to USB 3.0 adapter, as shown in the following photograph. This device is also self-powered and comes with a removable power adapter so that it can be used to power full-sized hard drives, DVDs, and Blu-ray drives with SATA connectors:

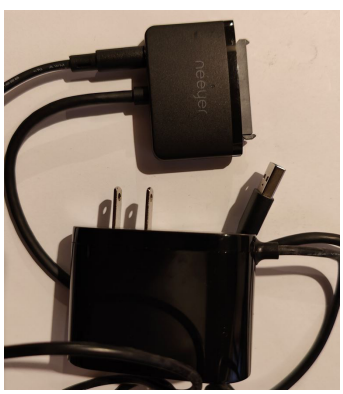


Figure 7.3 – USB 3.0 to SATA self-powered adapter for both 2.5-inch and 3.5-inch drives

For connecting to routers and switches, console cables and serial to USB cables can also be included in your kit, like the ones shown here:

Figure 7.4 – Serial to JJ-45 (left) and Serial to USB (right) cables

All mobile devices including phones and tablets can also connect to laptops and desktops via USB ports. The following photograph shows an OTG cable, which can connect a USB device to phones or tablets with OTG capabilities:

Figure 7.5 – OTG (on-the-go) cable

Here is a list of guidelines for physical collection and preservation:

- Label all cables and connectors
- Use labeled evidence collection bags as needed
- Special stronghold bags might have to be used when storing devices with wireless and radio capabilities, preventing communication with other devices
- Store sensitive equipment such as hard drives and flash drives in anti-static bags and protective casings
- Label containers used for storage during transportation
- Maintain the CoC forms when passing evidence from one person/handler to another (discussed later in this chapter)

Now, let's move on to understanding the order of volatility when collecting evidence and also comparing live versus post-mortem acquisitions.

Live versus post-mortem acquisition

In this section, we'll look at the different procedures for live and post-mortem evidence acquisition. However, before we begin, first, we must understand the **order of volatility**.

Order of volatility

When collecting evidence, we should keep in mind the volatility of data. As mentioned earlier in this chapter, data can be easily lost or destroyed. As such, when collecting data, a well-documented and common best practice is to collect evidence in the order of most volatile to the least volatile if possible.

The **Scientific Working Group on Digital Evidence** (**SWGDE**) capture live systems document and lists the order of volatility from most to least volatile and crucial, as follows:

- RAM
- Running processes
- Active network connections
- System settings
- Storage media

Powered-on versus powered-off device acquisition

When investigating devices that are powered on and powered off, special consideration must be given to the volatility of data. Booting, rebooting, or shutting down a device can cause data to be written to the hard drive or even lost within RAM and the paging file.

Powered-on devices

When investigating a powered-on device, the following precautions should be taken:

- Move the mouse or glide your fingers across the touchpad if you suspect the device might be in a *sleep* state. Do not click on the buttons as this might open or close programs and processes.
- Photograph and record the screen and all visible programs, data, time, and desktop items.
- Unplug the power cord on desktops and remove the battery, if possible, on portables.

It is of utmost importance that data stored in RAM and paging files be collected with as little modification to the data as possible. More on this will be covered in *Chapter 8*, *Evidence Acquisition Tools*, using imaging tools such as FTK Imager and Belkasoft RAM Capturer.

There are quite a few reasons for imaging and acquiring RAM. As mentioned in the previous chapter, data that might have been encrypted by the user could be stored in an unencrypted state of RAM. Logged-in users, opened programs, accessed files, and running processes can all be extracted and analyzed if the RAM and paging files are analyzed.

However, if the device is switched off or rebooted, this data and evidence can easily be lost.

For powered-on portable and powered-on devices, the battery should be removed, if possible. However, some devices might not have a removable battery. In these cases, the power button should be held down for 30 to 40 seconds, which forces the device to power off.

Powered-off devices

Powered-off devices should never be turned on unless done so by the forensic investigator. Special steps must be taken to ensure that existing data is not erased and that new data is not written.

Often, devices can seem as if they are off, but they can be in a sleep or hibernate state. As a simple test, the mouse can be moved and monitors (if any) can be switched on to determine whether they are in fact in either of those states. Even if they are in an off state, you should still photograph the screen and ports.

When investigating portable and mobile devices in an already off state, it is suggested that the battery is removed (if possible) and placed in an evidence bag to ensure that there will be no accidental way to turn the device on once unplugged. According to the

NIST.SP.800-101r1—Guidelines on Mobile Forensics, it should be noted that removing the battery can alter contents in volatile memory, even when in an off state.

Now that we understand the importance of taking note of the state of the device from which evidence must be acquired, let's look at the CoC form, which adds another layer of evidence integrity.

The CoC

The **CoC** is a form that legally ensures the integrity of evidence as it is exchanged between individuals, and so it also provides a level of accountability as personal identification is required when completing the forms. This form gives an exact log and account of the transportation and exchange between parties, from a collection at the scene to a presentation in a court.

Some of the typical fields on the CoC form are listed as follows:

- Case number
- Victim and suspect names
- Date and time seized:
  - Location when seized
  - Item number
  - Description of item
  - Signatures and IDs of individuals releasing and receiving the evidence
- Authorization for disposal
- Witness to the destruction of evidence
- Release (of evidence) to the lawful owner

The sample CoC form can be downloaded directly from the **National Institute of Standards and Technology** (**NIST**):

`https://www.nist.gov/document/sample-chain-custody-formdocx`

I urge you to download the preceding form as you might need it when working on DFIR investigations. Now, let's look at the importance of write blockers when acquiring data to preserve the integrity of the evidence.

The importance of write blockers

Once our physical evidence has been properly documented and collected, we can begin the forensic acquisition of digital evidence. I'll mention this a couple of times to drive the point home, but the original evidence should only be used to create forensic copies or images, which will be discussed later in this chapter and again in other chapters.

Examination and analysis should only be performed on forensic copies/images of the data and *not* on the original data or evidence, as working on original evidence can, and usually will, modify the contents of the medium. For instance, booting a seized laptop into its native operating system will allow data to be written to the hard drive and may also erase and modify contents contained in the RAM and paging file.

To prevent this from happening, the use of a write blocker must be employed. Write blockers, as the name suggests, prevent data from being written to the evidence media. Write blockers can be found in both hardware and software types. If a hardware write blocker is not available, software versions are readily available as standalone features in forensic operating systems including C.A.I.N.E, as mentioned in *Chapter 2*, *Introduction to Digital Forensics*, and also as a part of some commercial and open-source tools such as EnCase and Autopsy.

Again, it is of high importance that a write blocker is used in investigations to protect and preserve the original evidence from being modified. The following photograph shows a cheap and efficient portable SATA and IDE adapter with write-blocking switches, used in drive acquisition and recovery:

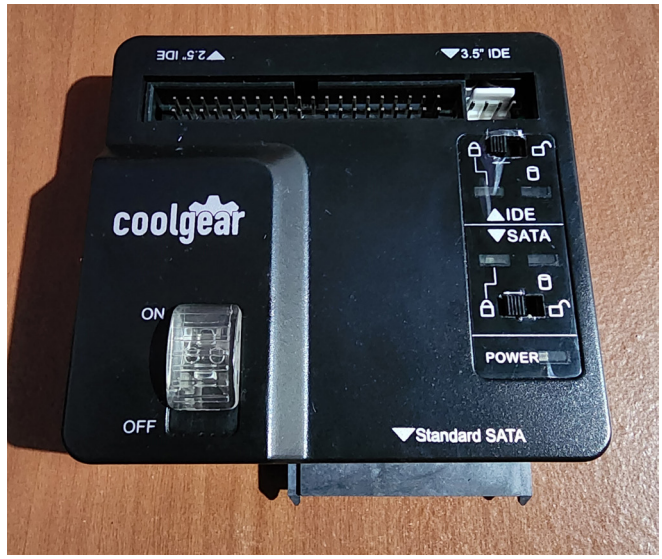


Figure 7.6 – USB to SATA and IDE write-blocker device

Now that we are familiar with the purpose and importance of write blockers, let's learn about data imaging and evidence integrity algorithms.

Data imaging and maintaining evidence integrity

Imaging refers to the exact copying of data either as a file, folder, partition, or entire storage media or drive. When doing a regular copy of files and folders, not all files may be copied based on their attributes being set to the system or even hidden. To prevent files from being left out, we perform a special type of copy where every bit is copied or imaged exactly as it is on the current medium as if taking a picture or snapshot of the data.

Creating a copy of each bit of data exactly as is, is referred to as a **physical or forensic image**. Performing a **bitstream copy** ensures the integrity of the copy. To further prove this, hashing of the original evidence and the physical image is calculated and compared (which we will delve into shortly). If the forensic copy is off by even one bit of data, the hash values created by the respective algorithms will be quite different.

> **Tip**
>
> The original evidence should only be handled by qualified and authorized professionals and should also only be used to create forensically sound physical images. The original evidence should otherwise never be used as this compromises evidence integrity and the investigation.

Message Digest (MD5) hash

Hash values are produced by specific algorithms and are used to verify the integrity of the evidence by proving that the data was not modified. Hash values can be thought of as digital fingerprints in that they are unique and play a major role in the identification of evidence and physical images.

One such algorithm, although older and now mainly used to verify the authenticity of downloads from the internet, is the **Message Digest** (**MD5**) cryptographic hashing algorithm, which produces a 128-bit hexadecimal output value.

For a working example, let's open a browser and head over to `http://passwordsgenerator.net/md5-hash-generator/`.

This site creates hashes of words and sentences as strings. For this example, I've entered the `Digital Forensics with Kali Linux` string without the parentheses. The MD5 value automatically calculated was displayed as **7E9506C4D9DD85220FB3DF671F09DA35**, as shown in the following screenshot:

Figure 7.7 – MD5 hash generator results

By removing `K` from `Kali`, from the same string that now reads `Digital Forensics with ali Linux`, the MD5 now reads **7A4C7AA85B114E91F247779D6A0B3022**, as shown in the following screenshot:

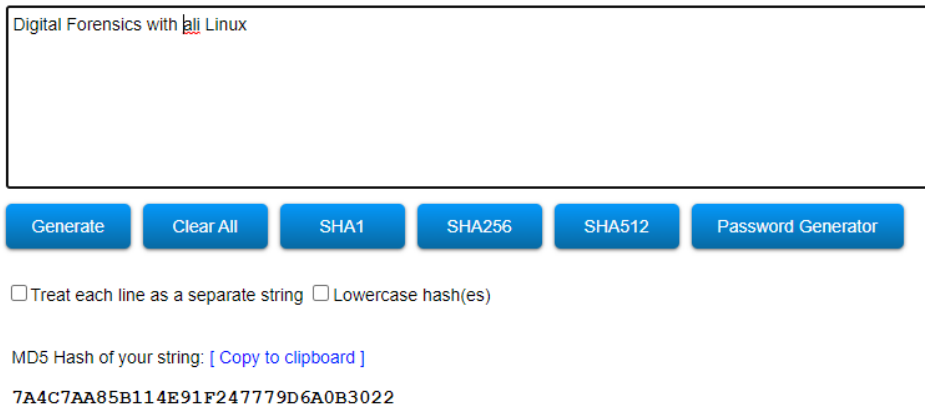


Figure 7.8 – Edited MD5 generator results

As a quick comparison, we can see that just removing `K` from `Kali` yields a noticeably different result:

- `Digital Forensics with Kali Linux`:

  7E9506C4D9DD85220FB3DF671F09DA35

- `Digital Forensics with ali Linux`:

  7A4C7AA85B114E91F247779D6A0B3022

I encourage you to try it yourself and perhaps add a comma or period to the string to further compare hash values.

Secure Hashing Algorithm (SHA)

Another cryptographic hash algorithm commonly used in forensics and used in the next chapter is **Secure Hashing Algorithm-1** (**SHA1**). SHA1 is more secure than MD5 as it produces a 160-bit output instead of a 128-bit output as with MD5. Due to known collision attacks against both MD5 and SHA-1, the safer and more robust option for hashing is now **SHA-2**.

SHA-2 is a group of hashes and not just one, as with SHA-1, with the most common bit-length being SHA-256, which produces a 256-bit output. Alternate bit-length algorithms of SHA-2 are SHA-224, SHA-384, and SHA-512.

The stronger the cryptographic algorithm used, the less chance of it being attacked or compromised. This means that the integrity of the evidence and physical images created remain intact, which will prove useful in forensic cases and expert testimony.

More on creating hashes will be demonstrated in *Chapter 8*, *Evidence Acquisition Tools*.

Now, let's move on to best practices and frameworks for data acquisition and DFIR.

Data acquisition best practices and DFIR frameworks

So far, we've covered a general overview of the DFIR procedures when collecting and preserving evidence. There are several official documents that I highly recommend you read and become familiar with, as they all give good details and guidelines on the documentation of the scene, evidence collection, and data acquisition.

The SWGDE has several best practice guidelines on forensic acquisition, evidence collection, forensic examination, and more. These very useful documents should be downloaded and kept as part of your DFIR playbook as they are concise and summarize all the necessary steps, which can act as a checklist for DFIR investigations. All documents can be found in the SWGDE's Forensic Publications section at `https://www.swgde.org/documents/published-by-committee/forensics`, but for the purposes of this chapter, I recommend, at the very least, downloading and reading the following two best practices guidelines on evidence collection and forensic acquisitions.

The first paper details best practices when collecting digital evidence and the second details best practices when performing data acquisitions from computers:

- 2018-07-11 SWGDE Best Practices for Digital Evidence Collection: `https://drive.google.com/file/d/1ScBeRvYikHvu6qtE_Lj3JtbOl94a5FDr/view?usp=sharing`
- 2018-04-25 SWGDE Best Practices for Computer Forensic Acquisitions: `https://drive.google.com/file/d/1ScBeRvYikHvu6qtE_Lj3JtbOl94a5FDr/view?usp=sharing`

The *SWGDE Capture of Live Systems document, version 2.0*, was released in September 2014. Although quite short and not as detailed, it is still applicable to forensic investigations involving live (powered-on) systems. This document provides guidelines when investigating live systems, including the order of volatility, memory forensics including paging file forensics, and live physical and filesystem acquisition.

This document is only six pages long and can be downloaded from here: `https://drive.google.com/file/d/10MJ9EJAkj6i6Xqpf3IcFb4dDPNUZ2ymH/view`

DFIR frameworks

The practice guides mentioned earlier can be used in conjunction with one or more frameworks. Whatever the task, the goals should remain the same with the integrity of evidence always preserved throughout the entire DFIR processes of evidence collection, examination, analysis, and reporting.

Although not free, there is an ISO/IEC publication called the *ISO/IEC 27037:2021 Information technology — Security techniques — Guidelines for identification, collection, acquisition, and preservation of digital evidence*, which serves as an excellent source of information for best practices on DFIR. A preview of the document can be found at `https://www.iso.org/obp/ui/#iso:std:iso-iec:27037:ed-1:v1:en`.

The **National Institute of Science and Technology** (**NIST**) also has an older publication called the *NIST Special Publication (SP) 800-86 Guide to Integrating Forensic Techniques into Incident Response*, which contains 121 pages of very useful information about various types of technologies and data identification, acquisition, and examination. The full document can be freely downloaded at `https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-86.pdf`.

Another useful publication is the *NIST Special Publication (SP) 800-61 Revision 2 Computer Security Incident Handling Guide*, which specifically focuses on incidents and the appropriate responses to them. This document can be viewed and downloaded in its entirety at `https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf`.

Additionally, the *NIST Guidelines on Mobile Device Forensics* is another very useful document that applies specifically to mobile devices. Revision one of this document, released in 2014, goes into detail about the aspects of mobile forensics investigations. Its content includes the following:

- Mobile and cellular characteristics
- Evaluation and documentation of the scene
- Device isolation and packaging
- Device and memory acquisition
- Examination, analysis, and reporting

The full document can be downloaded from:

`http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-101r1.pdf`

In relation to investigating cyber-attacks, reference should be made to the D4I – Digital forensics framework for reviewing and investigating cyber-attacks as a proposed framework. It also references the Cyber Kill Chain and its associated phases with cyber-attacks. You can find the 8-page document at:

`https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=929147`

Over the past two decades, there have been significant changes in technology and IT. Added to what we traditionally think of as Information Technology, there is now **Operational Technology** (**OT**). OT mainly deals with devices and machinery traditionally found in the manufacturing and utility sectors, which may be logically managed by IT processes. A good example is **SCADA** (short for **Supervisory Control and Data Acquisition**) systems, which are commonly used by gas and utility companies to monitor operational equipment. Disruption to these services could affect millions of persons and businesses, and formal incident response procedures should be implemented to ensure business continuity and disaster recovery. NIST has recently published the *NISTIR 8428 Digital Forensics and Incident Response (DFIR) Framework for Operational Technology (OT) (June 2022)* as a framework to address these issues, and it covers a very detailed OT DFIR process on asset identification, remote data collection, process monitoring, field analysis, and more. The full document is available for download at:

`https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8428.pdf`.

I hope this section was informative, and I encourage you to download and keep these documents within a folder or perhaps even print and bind them to have as reference materials or an entire DFIR playbook.

Summary

If there was only one thing that I'd like you to take away from this chapter, it would be to remember that the original evidence, particularly hard drives, storage media, and RAM images, should only be used to create forensically-sound bitstream copies. The original evidence is never to be worked on.

To recap, when a breach is reported, there should be an established first responder who, as per protocol, performs the tasks of documenting and securing the scene as well as collecting and preserving the evidence. The first responder should have a toolkit with various tools and items for the acquisition of evidence, and when handing over the evidence to other parties, ensure that the CoC is maintained.

Additionally, we looked at the various procedures and best practices when investigating devices that are powered on and powered off, and we discussed the importance of using a write blocker to prevent the original evidence from being tampered with and then using a hashing tool for integrity verification purposes. Finally, I've left you with some very useful DFIR frameworks, which, when combined with the SWGDE guidelines, make for an impressive DFIR playbook.

You've come this far, and I know it must have been a bit of an information overload, but now we can get to the practical section of this book where we can begin our investigation using digital forensics tools in Kali Linux. Let's go!

Part 3: Kali Linux Digital Forensics and Incident Response Tools

Here's the fun part. From this point on, we'll have a hands-on approach with practical labs. In this section, we will first learn how to use various tools for the forensic acquisition of memory (RAM) and storage media. We will then move on to file recovery and data carving tools, and finally, delve into the analysis of memory dumps, and perform different types of artifact analysis.

This part has the following chapters:

- *Chapter 8*, *Evidence Acquisition Tools*
- *Chapter 9*, *File Recovery and Data Carving Tools*
- *Chapter 10*, *Memory Forensics and Analysis with Volatility 3*
- *Chapter 11*, *Artifact, Malware, and Ransomware Analysis*

8

Evidence Acquisition Tools

In the previous chapter, we learned that documentation and proper DFIR procedures are key in any investigation. These ensure the integrity of the investigation by providing proof of data authenticity and preservation of the original evidence and documentation, which can be used to achieve the same exact results if the usage of tools and methods is repeated.

In this chapter, we will focus on and demonstrate forensically sound techniques for the acquisition of data by creating bitstream copies of evidence inclusive of data hashes, and also perform evidence acquisition of the drives, RAM, and paging files using various tools.

This is the first technical step in DFIR investigations, so it is very important to familiarize yourself with the tools and processes for evidence acquisition that are covered in this chapter. Upon completion of this chapter, you will know how to perform formal evidence acquisitions for analysis, which will be covered in later chapters.

In this chapter, we will cover the following topics:

- Using the `fdisk` command for partition recognition
- Creating strong hashes for evidence integrity
- Drive acquisition using DC3DD
- Drive acquisition using DD
- Drive acquisition using Guymager
- Drive and memory acquisition using FTK Imager in Wine
- RAM and paging file acquisition using Belkasoft RAM Capturer in Wine

Using the fdisk command for partition recognition

For anyone using Kali as a standalone operating system, the process of mounting a removable drive is straightforward, where you insert the removable storage drive and then run the `fdisk` command. However, for those using Kali as a virtual machine or VM, you will first need to ensure that the removable storage is recognized by VirtualBox. To do so, follow these steps:

1. Click on **Devices** in the VirtualBox window running Kali Linux, and then click on the **USB** option, as seen in the following screenshot, which will display a list of all detected USB devices. You can take a screenshot or photo of this list for comparison purposes.

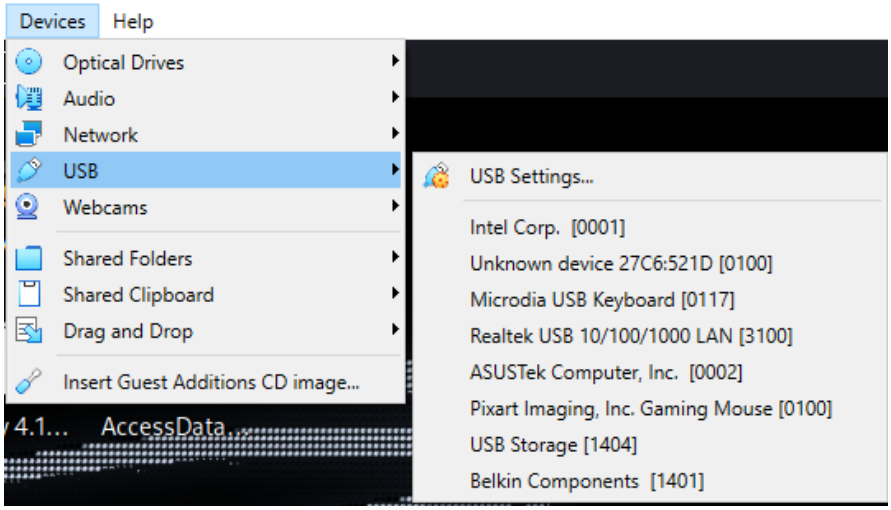


Figure 8.1 – The VirtualBox Manager devices menu

2. In the preceding step, there are eight detected devices. You can now plug in your 2-GB microSD card and click on **Devices** | **USB** again to see what your microSD card is listed as. In the following screenshot, we can see a new entry on our list of USB devices, **Generic Mass Storage Device [0100]**. Click on this entry in the list to mount the device in Kali Linux.

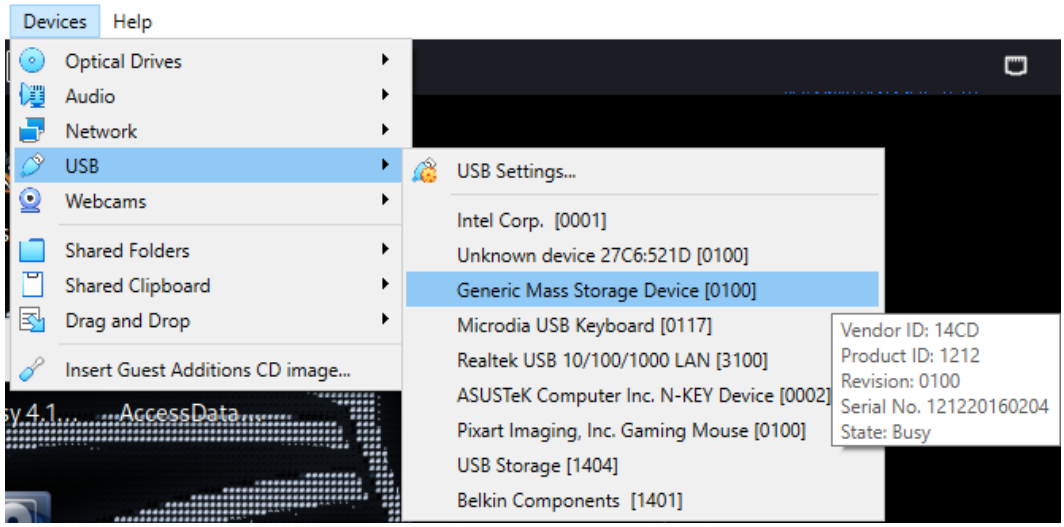


Figure 8.2 – The VirtualBox Manager USB devices menu

Once the device is recognized by VirtualBox and mounted in Kali, you should see your device showing as an icon on your Kali desktop. In the following screenshot, we can see that the 8-GB microSD card is now mounted and available for use by Kali.

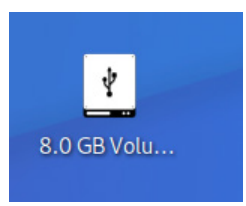


Figure 8.3 – The storage media desktop icon

3. Once our disk has been mounted, we can now use the `fdisk` command to view the drive partition details.

Now that we've successfully mounted our removable storage drive, we will learn how to identify the device and partitions using the `fdisk` command in the next section.

Device identification using the fdisk command

For this exercise, an 8-GB Sony Pro Duo card was connected via an external USB 3.0 card reader to my standalone Kali Linux desktop. Feel free to use any other type of storage media device for this exercise, as the process is exactly the same regardless of the device type used. Before we begin any acquisition processes, we will first run the `sudo fdisk -l` command to list all the attached storage devices to

differentiate between them. This is necessary, as we will not be using the storage device names given to them and will, instead, be using their disk identification information.

Open a new Terminal, type the following command, and press *Enter*:

```
sudo fdisk -l
```

In the following screenshot, the primary drive is listed as `sda` and the attached Sony Pro Duo card is listed as `sdb`.

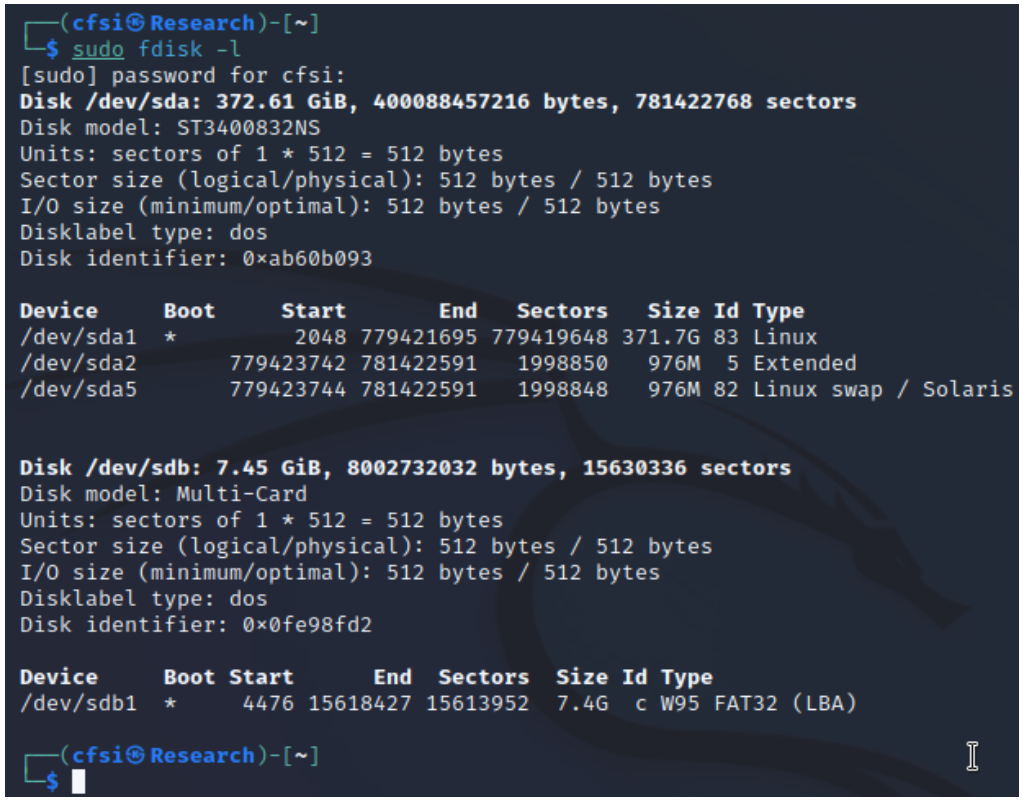


Figure 8.4 – The fdisk –l command output

As seen in the preceding screenshot, the details of the Sony ProDuo card are as follows:

- **Disk**: `sdb`
- **Size**: `7.4 GB`
- **Sector size**: `512 bytes`
- **Filesystem**: `FAT32`

As seen in the previous screenshot, Kali Linux recognizes two devices:

- `sda`: The primary hard disk with three partitions
- `sdb`: The storage drive to be forensically acquired or imaged

Users new to Kali or any Linux version or variation may find that the drive and partition recognition and naming conventions in Kali are different from those on Windows devices.

A typical device in Linux can be addressed or recognized as `/dev/sda`, whereas drives in Windows are usually recognized as `Disk 0` and `Disk 1`, and so on:

- `/dev`: This refers to the path of all devices and drives that can be read from or written to, recognized by Linux
- `/sda`: This refers to **SCSI** (short for **Small Computer Systems Interface**), SATA, and USB devices

`sd` stands for **SCSI Mass-Storage Driver**, with the following letter (for example, a, b, etc.) representing the drive number:

- `sda`: Drive `0` or the first drive recognized
- `sdb`: The second drive or storage media

While Windows recognizes partitions as primary, logical, and extended, Linux partitions are recognized as numbers that follow the drive letter:

- `sda1`: Partition 1 on the first disk (`sda`)
- `sda2`: Partition 2 on the first disk
- `sdb1`: Partition 1 on the second disk (`sdb`)
- `sdb2`: Partition 2 on the second disk

More information on Linux device naming conventions can be found at `https://www.dell.com/support/kbdoc/en-tt/000132092/ubuntu-linux-terms-for-your-hard-drive-and-devices-explained#:~:text=Under%20Linux%2C%20the%20original%20naming,address%2Dwise%20and%20so%20on`.

Now that we can identify the drives and partitions within a Linux file system, let's learn how to create hashes of the evidence for integrity purposes in the next section.

Creating strong hashes for evidence integrity

To provide proof that evidence was not tampered with, a cryptographic algorithm must be run against the evidence drive before, during, and after a forensic acquisition. These algorithms produce an output string (or hash output) of hexadecimal characters (a–f and 0–9) of various lengths, depending on their strength.

Common cryptographic algorithms are as follows:

- MD5: Message Digest 5
- SHA-1: Secure Hash Algorithm version 1
- SHA-256: SHA-2 256-bit

> **Note**
> More information on cryptographic hashes can be found at https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm.

In Kali Linux, we can use the md5sum, sha1sum, or sha256sum commands, followed by the path of the device, to create a hash output of the evidence/input file. For example, to create a SHA-256 hash, we would use the following command, where sdx represents the drive we are trying to acquire or image:

```
Sha256sum /dev/sdx.
```

Although I mentioned the md5sum command, I recommend using a stronger algorithm, such as SHA-1 or SHA-256, as MD5 is much older and can be compromised:

- MD5 strength: 128-bit hash value
- SHA-1 strength: 160-bit hash value
- SHA-256 strength: 256-bit value
- SHA-3 strength: 256-bit value (but faster than SHA-2)

To create an MD5 hash output against my Pro Duo card (sdb), I used the following command:

```
sudo md5sum /dev/sdb.
```

To create an SHA-1 hash output against my Pro Duo card (sdb), I used the following command:

```
sudo sha1sum /dev/sdb.
```

To create an SHA-256 hash output against my Pro Duo card (sdb), I used the following command:

```
sudo sha256sum /dev/sdb.
```

In the following screenshot, I have run all the preceding commands. Note that the `SHA-256` output is the longest.

Figure 8.5 – Cryptographic algorithm output hashes

Important note

When performing forensic acquisition or forensic imaging of any drive or storage media, the hash output of the original file must always match the output of the forensically acquired forensic image, which assures the integrity of the evidence and the copy of the evidence (the forensic image).

Now that we've distinguished between drives and are certain of which drive is to be imaged (`sdb`), we can begin the forensic imaging using DC3DD. Although I have used an older 8-GB Pro Duo storage drive to demonstrate the usage of DC3DD, you can use any drive (portable or otherwise) to practice using the tools in this chapter. Be sure to use the `fdisk -l` command to identify your drives and partitions before continuing.

Drive acquisition using DC3DD

The first tool we will use for acquisition is called **DC3DD** (developed by the **department of Defense Cyber Crime Center**). DC3DD is a patch of the very popular **Data Dump** (**DD**) tool, used for forensic acquisition and hashing.

These are the features of DD:

- Bitstream (raw) disk acquisition and cloning
- Copying disk partitions
- Copying folders and files
- Hard disk drive error checking
- Forensic wiping or forensic and secure deletion of all data on hard disk drives

DC3DD is updated whenever DD updates. DC3DD offers the best of DD with more features, including the following:

- On-the-fly hashing (or hashing done instantly by the tool), using more algorithm choices (`MD5`, `SHA-1`, `SHA-256`, and `SHA-512`)
- A meter to monitor progress and acquisition time
- Writing of errors to a file
- Splitting of output files
- Verification of files
- Wiping of output files (pattern wiping)

DC3DD must be installed manually in Kali Linux. First, we'll update our version of Kali Linux by using the `apt-get update` command.

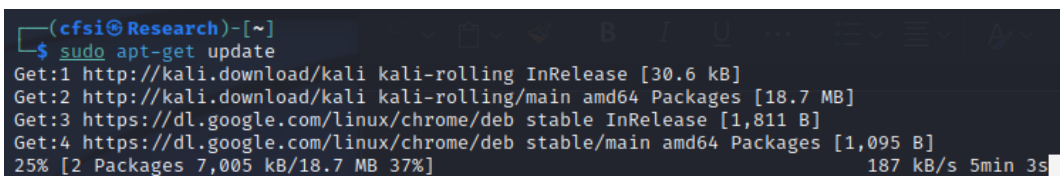


Figure 8.6 – Updating Kali Linux

Once Kali updates, you can manually install DC3DD by typing the `sudo apt-get install dc3dd` command.

```
┌──(cfsi㉿Research)-[~]
└─$ sudo apt-get install dc3dd
[sudo] password for cfsi:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following NEW packages will be installed:
  dc3dd
0 upgraded, 1 newly installed, 0 to remove and 749 not upgraded.
Need to get 121 kB of archives.
After this operation, 501 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 dc3dd amd64 7.2.646-5
+b1 [121 kB]
Fetched 121 kB in 1s (151 kB/s)
Selecting previously unselected package dc3dd.
(Reading database ... 312296 files and directories currently installed.)
Preparing to unpack .../dc3dd_7.2.646-5+b1_amd64.deb ...
Unpacking dc3dd (7.2.646-5+b1) ...
Setting up dc3dd (7.2.646-5+b1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for kali-menu (2022.3.1) ...
```

Figure 8.7 – Installing dc3dd

DC3DD is a CLI tool and can be easily run in Kali Linux by first opening a Terminal and typing `dc3dd`. To start with, I recommend using the `dc3dd --help` command, which lists the available parameters used with `dc3dd`:

```
┌──(cfsi㉿Research)-[~]
└─$ dc3dd --help
usage:

        dc3dd [OPTION 1] [OPTION 2] ... [OPTION N]

                *or*

        dc3dd [HELP OPTION]

        where each OPTION is selected from the basic or advanced
        options listed below, or HELP OPTION is selected from the
        help options listed below.
```

Figure 8.8 – The dc3dd help option

As shown in the previous screenshot that shows the use of the `dc3dd --help` command, typical usage of the DC3DD command looks like this:

```
dc3dd [option 1] [option 2] ... [option n]
```

To create a forensic image of my 8-GB drive, I've used the following options within `dc3dd`:

```
sudo dc3dd if=/dev/sdb hash=sha1 log=dc3dd8gb of=8gbproduo.dd
```

The following screenshot shows the output of the preceding command.

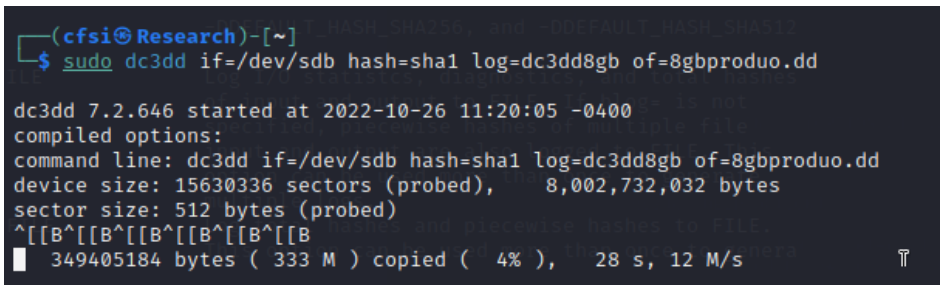


Figure 8.9 – The drive acquisition command within dc3dd

The following list explains the output of the screenshot shown in *Figure 8.9*:

- `if`: This specifies the *input file*, which is the device we will be imaging.
- `hash`: This specifies the type of hash algorithm we will be using for integrity verification. In this case, I have used the older `MD5` hash.
- `log`: This specifies the name of the log file that logs the details of the device and the acquisition, including errors.
- `of`: This specifies the output file name of the forensic image created by DC3DD. Although a `.dd` image file type was specified in this example, other formats are recognized by DC3DD, including `.img`, as shown in a later example.

The device size (in sector and bytes) should be noted and later compared to the output results for the device field. Once the acquisition process has been completed, the input and output results are displayed.

In *Figure 8.10*, the last line displays the progress and status of the acquisition process, showing the amount of data copied, the elapsed time in seconds, and the speed of the imaging process in Mbps:

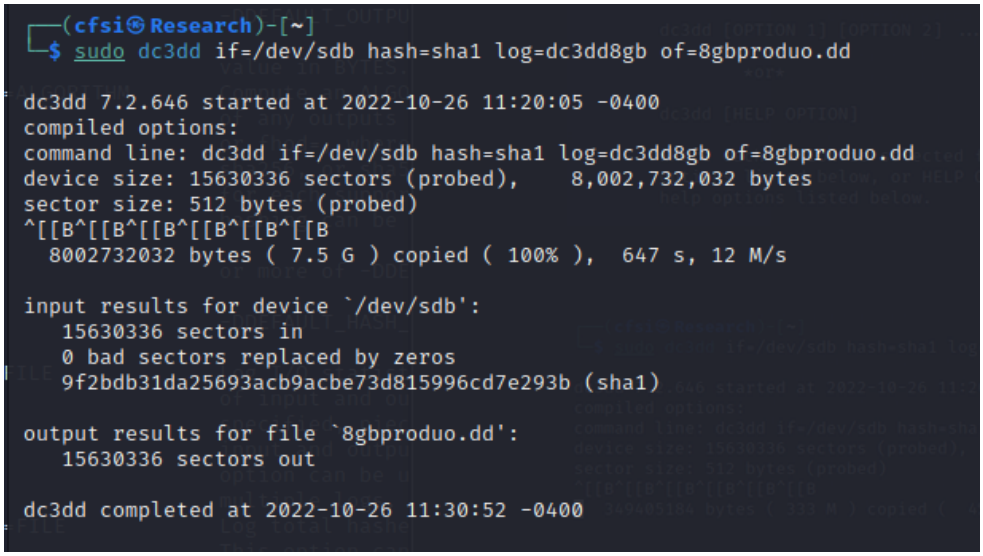


Figure 8.10 – The completed dc3dd output

The larger the drive or file to be acquired, the lengthier the time taken to do so. Might I suggest you get yourself a cup of coffee or a refreshing beverage, or even have a look at some other wonderful titles available from Packt at `https://www.packtpub.com/`.

Analyzing the results in *Figure 8.10*, we can see that the same amount of sectors (`15630336`) have been imaged, with no bad sectors being replaced by zeros. We can also see that the exact `SHA1` hash was created for the image, assuring us that an exact copy was created without modification.

In the Terminal, we can also use the `ls` command to list the directory contents to ensure that the DC3DD output file (`8gbproduo.dd`) and log (`dc3dd8gb`) have been created:

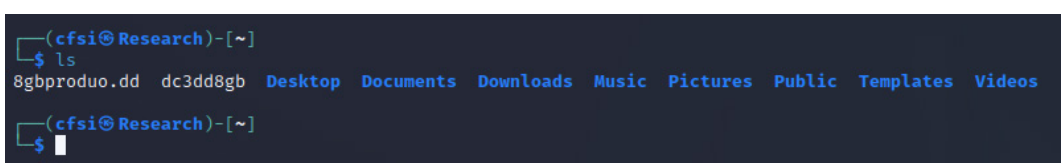


Figure 8.11 – The home directory listing using the ls command

To access our forensic image and log file, we can go to our `/home` directory by clicking on the folder icon (in the top-left corner) on the desktop and then clicking on **Open Folder**.

Within the `Home` folder, the first file, `8gbproduo.dd`, is the output image created by DC3DD using the `of=8gbproduo.dd` command. The last file, `dc3dd8gb`, is the log file, created when we used the `log=dc3dd8gb` command:

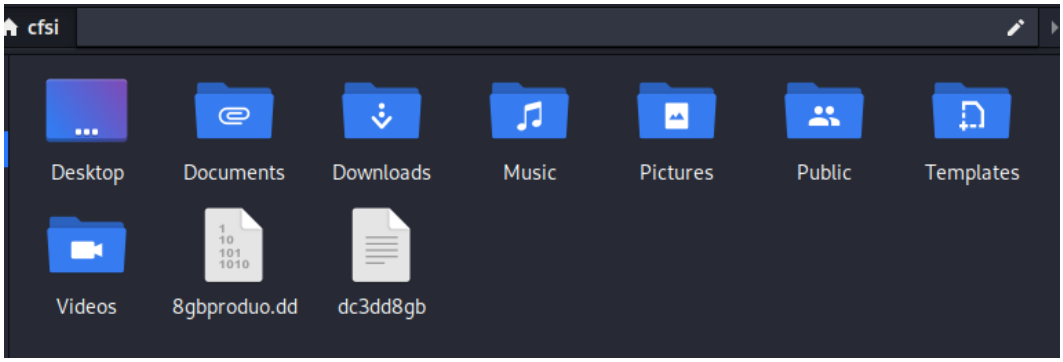


Figure 8.12 – The forensic image and log file in the Home folder

It's important to keep this log file to have a record of the acquisition process and its results, which were displayed on the screen upon completion:

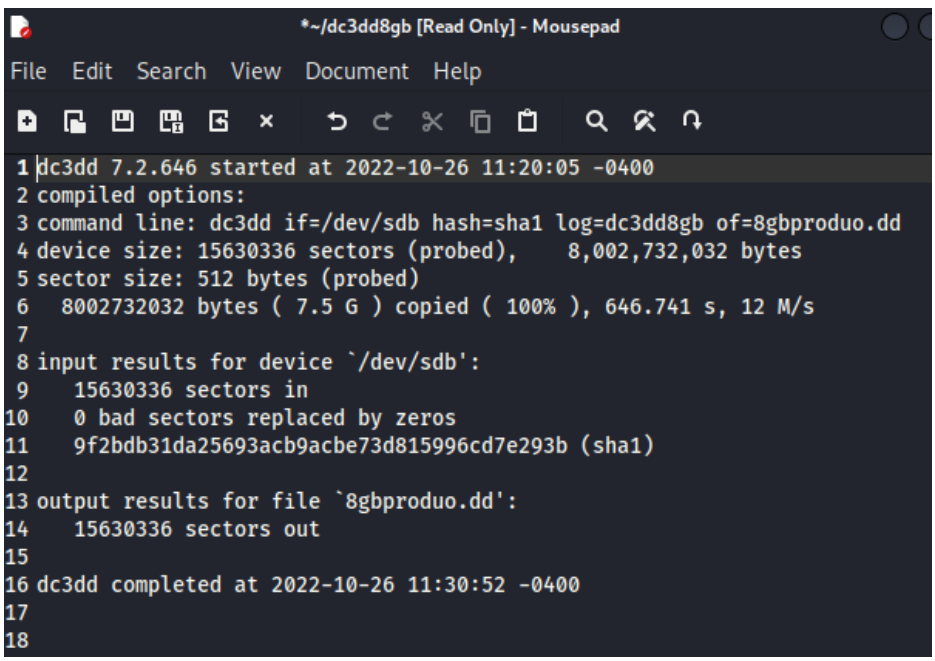


Figure 8.13 – The contents within the log file

In *Chapter 12*, *Autopsy Forensic Browser,* and *Chapter 13*, *Performing a Full DFIR Analysis with the Autopsy 4 GUI*, we will analyze acquired forensic images using Autopsy; however, the acquired evidence images can also be copied or directly cloned to another device if an investigator so wishes.

As an example, we could clone the forensic image acquired previously (`8gbproduo.dd`) onto a new drive, recognized as `sdc`. The command used to perform this task would be as follows:

```
dc3dd if=8gbproduo.dd of=/dev/sdc log=drivecopy.log
```

When copying an image to a drive, the destination drive size should be of equal size or larger than the image file.

Verifying the hash output of image files

To verify the hash output of `sdb`, the following command can be used:

```
sudo sha1sum /dev/sdb
```

The following screenshot shows the output of the preceding command:

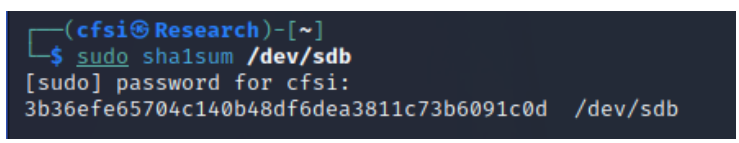


Figure 8.14 – sha1sum output

You can also use the following command:

```
cat 8gbproduo.dd | sha1sum
```

The following screenshot shows the output of the preceding command:

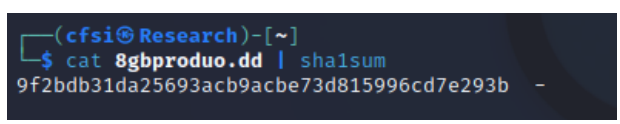


Figure 8.15 – cat sha1sum output

Erasing a drive using DC3DD

We've seen the power of DC3DD as a very impressive forensic acquisition tool, but I'd also like to go one step further and introduce you to its capabilities as a data-wiping tool.

DC3DD can wipe data and erase drives by overwriting data in three ways:

- Overwriting and filling data and drives with zeroes. The command used is as follows:

```
dc3dd wipe=/dev/sdb
```

The following screenshot shows the output of the preceding command:

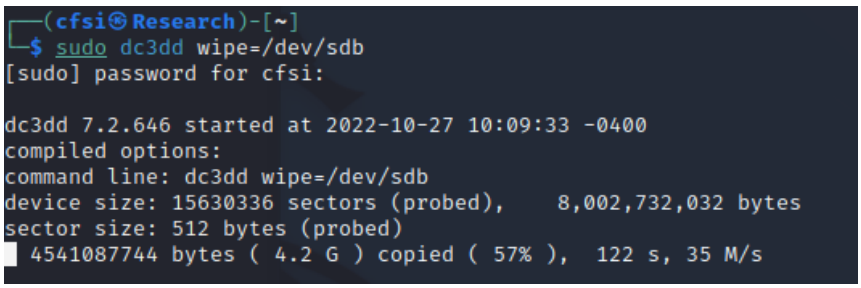


Figure 8.16 – dc3dd wipe command output

- Overwriting and filling data and drives, using a hexadecimal pattern and the `pat` option. The command used is as follows:

```
dc3dd wipe=/dev/sdb pat=10101010
```

The following screenshot shows the output of the preceding command:

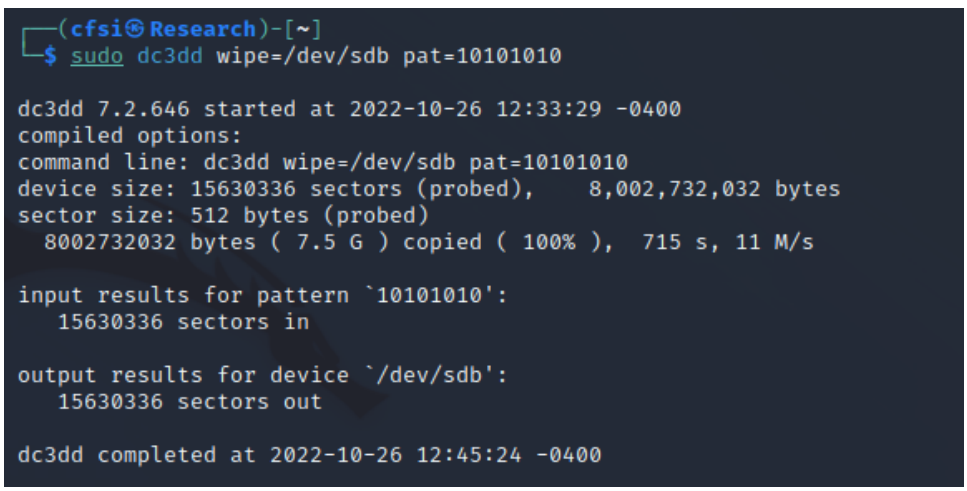


Figure 8.17 – The dc3dd wipe command with pattern output

- Overwriting and filling the data and drives, using a text pattern and the `tpat` option. The command used is as follows:

```
dc3dd wipe=/dev/sdb tpat=CFSI
```

The following screenshot shows the output of the preceding command:

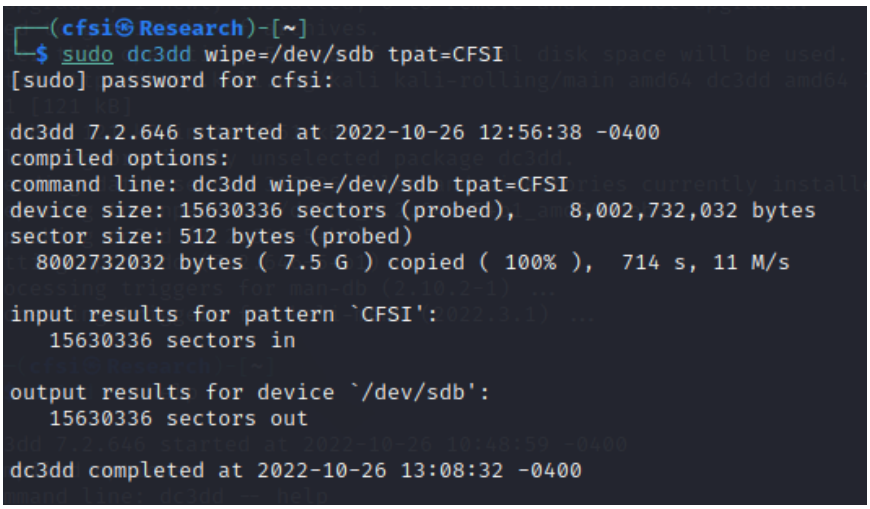


Figure 8.18 – The dc3dd wipe command with text pattern output

We've covered the usage of the `dc3dd` tool and the various methods to forensically wipe media. Let's now look at another tool called DD that can also be used for forensic data copying and wiping.

Drive acquisition using DD

Before we get started using DD, I need to again draw your attention to one of the features of DD, the ability to wipe data, partitions, and drives. Hence, you may find that DD is sometimes fondly referred to as the **data destroyer**. Be sure to always first identify your devices, partitions, input and output files, and parameters when using DD and DC3DD.

For the exercises in this chapter, I'll be using an older but functional 2-GB flash drive for the acquisition process using DC3DD.

Should you also wish to use the DD tool, the commands and usage are very much the same.

You may want to first ensure that you can access the dd tool by running `dd --help`. If the `dd` command cannot be found, update Kali by running the `apt-get update` command, and then run the `dd --help` command again.

```
root@kali:~# dd --help
Usage: dd [OPERAND]...
  or:  dd OPTION
Copy a file, converting and formatting according to the operands.

  bs=BYTES        read and write up to BYTES bytes at a time (default: 512);
                  overrides ibs and obs
  cbs=BYTES       convert BYTES bytes at a time
  conv=CONVS      convert the file as per the comma separated symbol list
  count=N         copy only N input blocks
  ibs=BYTES       read up to BYTES bytes at a time (default: 512)
  if=FILE         read from FILE instead of stdin
  iflag=FLAGS     read as per the comma separated symbol list
  obs=BYTES       write BYTES bytes at a time (default: 512)
  of=FILE         write to FILE instead of stdout
  oflag=FLAGS     write as per the comma separated symbol list
  seek=N          skip N obs-sized blocks at start of output
  skip=N          skip N ibs-sized blocks at start of input
  status=LEVEL    The LEVEL of information to print to stderr;
                  'none' suppresses everything but error messages,
                  'noxfer' suppresses the final transfer statistics,
                  'progress' shows periodic transfer statistics

N and BYTES may be followed by the following multiplicative suffixes:
c =1, w =2, b =512, kB =1000, K =1024, MB =1000*1000, M =1024*1024, xM =M,
GB =1000*1000*1000, G =1024*1024*1024, and so on for T, P, E, Z, Y.

Each CONV symbol may be:

  ascii     from EBCDIC to ASCII
  ebcdic    from ASCII to EBCDIC
  ibm       from ASCII to alternate EBCDIC
  block     pad newline-terminated records with spaces to cbs-size
  unblock   replace trailing spaces in cbs-size records with newline
  lcase     change upper case to lower case
  ucase     change lower case to upper case
  sparse    try to seek rather than write the output for NUL input blocks
  swab      swap every pair of input bytes
```

Figure 8.19 – dd help options

To perform image acquisition, I've used this command:

```
dd if=/dev/sdb of=produo8gb.img bs=65536 conv=noerror,sync
```

The following screenshot shows the output of the preceding command:

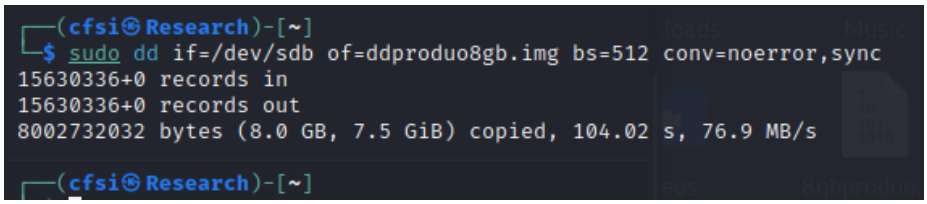


Figure 8.20 – Drive acquisition using dd

Let's do a breakdown of the individual options in the preceding command:

- `If`: The input file (`sdb` device)
- `Of`: The output file (the name of the forensic image)
- `Bs`: The block size (the default size is 512)
- `conv=noerror, sync`: Continue the imaging even if there are errors (`noerror`), and if there are errors, null-fill the blocks (`sync`).

In the preceding command, the `.img` output file extension was specified; however, you can use another format, such as `.iso`, by specifying the file extension in the output file's `(of)` option.

We can view the created files by using the `ls` command, where we can also see the two images.

The following screenshot shows the output of the preceding command:

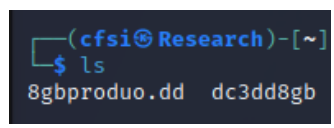


Figure 8.21 – The evidence acquisition files in the Home directory

We will now explore another very popular acquisition tool called Guymager, which offers many of the same features in a **Graphical User Interface** (**GUI**).

Drive acquisition using Guymager

Guymager is another standalone acquisition tool that can be used to create forensic images and also perform disk cloning. Developed by Guy Voncken, Guymager is completely open source, has many of the same features as DC3DD, and is also only available for Linux-based hosts. While some investigators may prefer CLI tools, Guymager is a GUI tool and for beginners, so it may be preferred.

For this acquisition, I'll also use the very same 2-GB flash drive used in the DC3DD examples, at the end of which we can compare results. It's also important to remember to continue using your write-blocker when acquiring and creating forensic images of evidence and drives, in an effort to not write data to the drives or modify the original evidence files.

As previously done in the DC3DD acquisition, we should first ensure that we are familiar with the devices attached to our machine, using the `fdisk -l or sudo fdisk -l` command.

Running Guymager

Guymager can be started by using the menu in Kali and clicking on the **Applications** menu at the top, then the **11 - Forensics** option, and then expanding the **Forensic Imaging Tools** menu, as shown in the following screenshot:

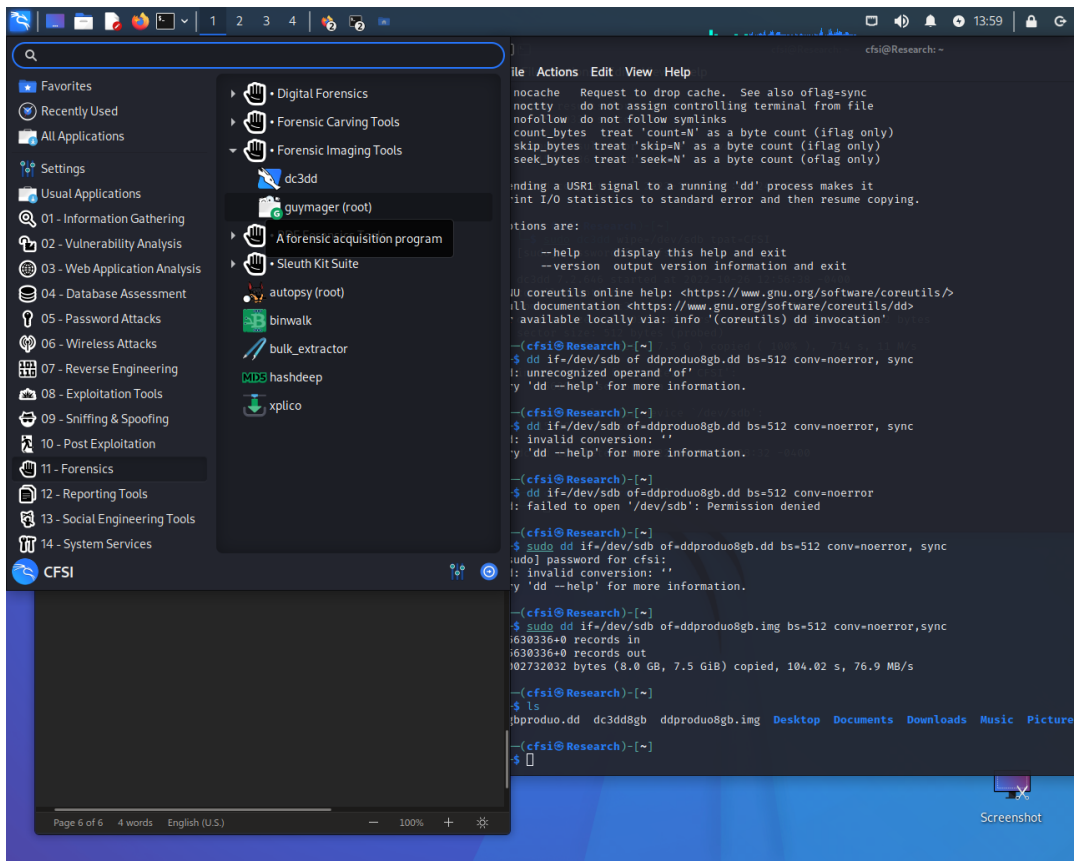


Figure 8.22 – The Kali Linux forensics menu

The Guymager application runs and then displays the existing drives recognized in Kali Linux. As shown in the following screenshot, the details of the 2-GB flash drive being used are shown, including the following:

- **Linux device**: Recognized as `/dev/sdb`
- **Model**: `USB_Flash_Memory`
- **State**: Shown as **Idle**, as the image acquisition has not yet begun
- **Size**: **2.0GB**
- **Serial number**: `001CC0C60D`... (this will be unique for each drive)
- **Hidden areas**: **unknown**

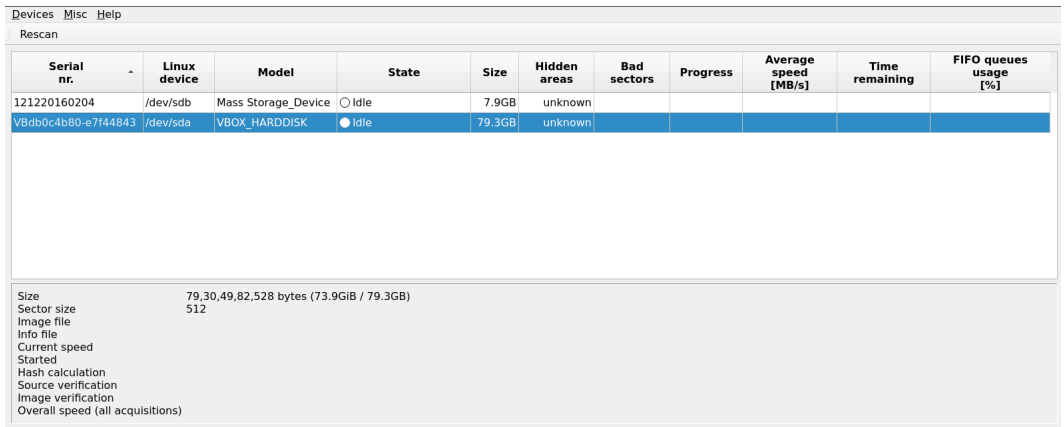


Figure 8.23 – Guymager interface

Should your device not be listed in Guymager or you need to add an additional device, click the **Rescan** button in the top-left corner of the application to detect the device.

Acquiring evidence with Guymager

To begin the acquisition process, right-click on the evidence drive (`/dev/sdb` in this example) and select **Acquire image**. Note that the **Clone device** option is also available if you wish to clone the evidence drive to another. Again, as previously mentioned, when cloning a device, the capacity of the destination device must be equal to or exceed that of the source (original) evidence drive:

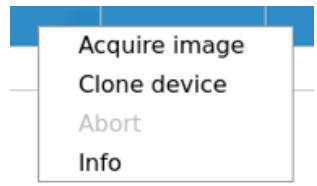


Figure 8.24 – Acquisition and clone options in Guymager

Before the actual acquisition process starts, the investigator is prompted to enter details about themselves and the evidence under the following two sections:

- **File format**:
  - **File extensions**: `.dd`, `.xxx`, and `.Exx`
  - **Split size**: Allows the investigator to choose the size of multiple image parts
  - **Case management information**: The case number, evidence number, examiner name, description, and notes

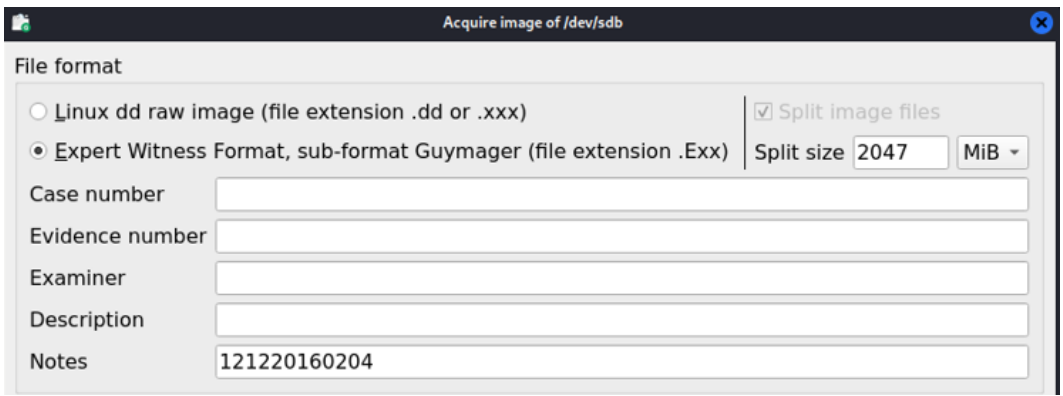


Figure 8.25 – Guymager file format options

- **Destination**:
  - **Image directory**: The location of the created image file and log (the info file)
  - **Image filename**: The name of the image file

Figure 8.26 – Guymager destination folder options

Hash calculation/verification

Multiple hashing algorithms can be selected and calculated, allowing the investigator to choose from `MD5`, `SHA-1`, and `SHA-256`.

The **Re-read source after acquisition for verification** option verifies the source.

The **Verify image after acquisition** option verifies the destination.

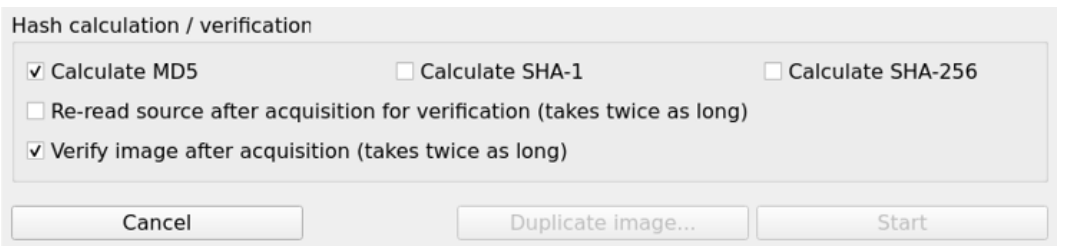


Figure 8.27 – Guymager cryptographic algorithm options

At the bottom of *Figure 8.27*, note that there are two grayed-out options.

Guymager adds a convenient **Duplicate image...** button (grayed-out in *Figure 8.27*) to create duplicate copies without having to repeat the data entry process.

For new users, you may want to specify the directory where the image file will be saved. In the destination section, click on the **Image directory** button and choose your location. For this acquisition, I've chosen the `Desktop` directory as the location for both the image and the log/info file.

The following screenshot shows the data that I've used for the Guymager acquisition, having chosen `Desktop` as the image directory and the `MD5` and `SHA-1` hashing algorithms:

Figure 8.28 – Guymager acquisition image folder options

Once the **Start** button is clicked (refer to *Figure 8.29*), note that the state changes from **Idle** to **Running**. The **Progress** field also now displays a progress bar:

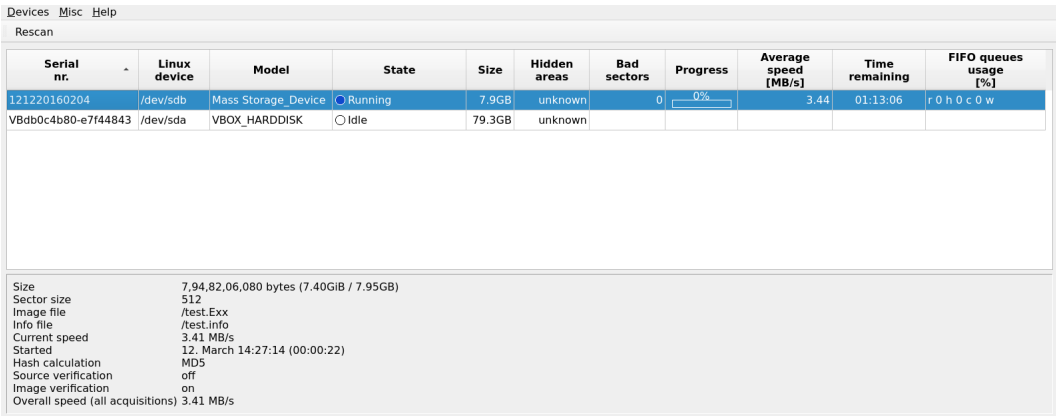


Figure 8.29 – The Guymager drive acquisition process

Taking a closer look at the details in the lower-left corner of the screen, we can see the size, image and info file paths, names and extensions, current speed, and chosen hash calculations. We also see that **Image verification** is turned **on**:

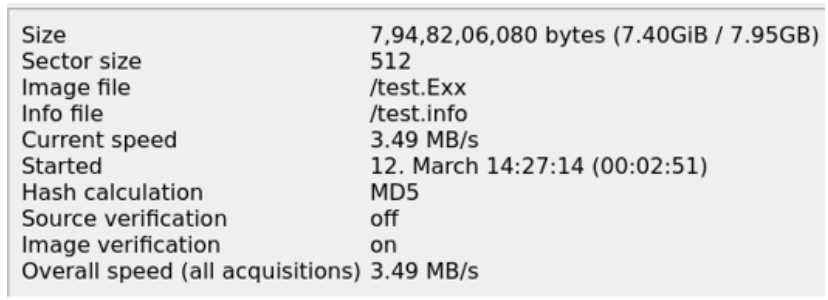


Figure 8.30 – Guymager process details

Once the acquisition process is completed, the color of the **State** field button changes from blue to green, indicating that the acquisition process is finished, and it also displays **Finished - Verified & ok** if verification options were selected in the **Hash verification / calculation** pane. The progress bar also displays **100%**:

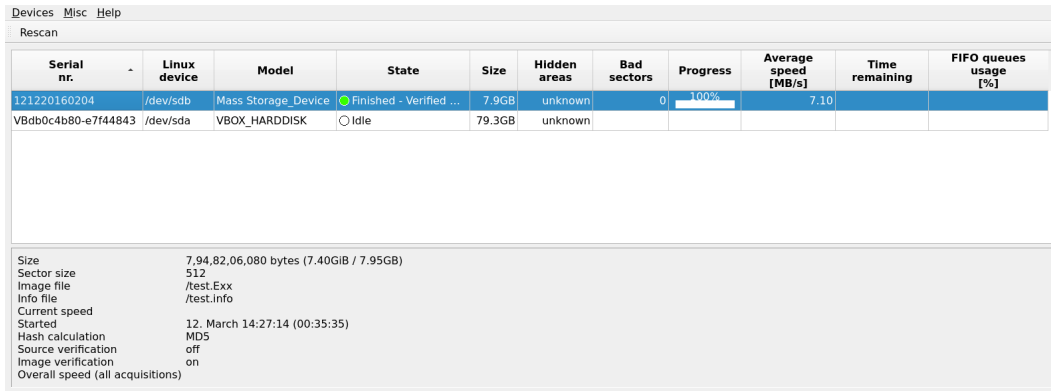


Figure 8.31 – The Guymager completed acquisition process

Our output file and info file can be found on the desktop, as this was specified in the the evidence. We'll now look into this by verifying and comparing hash algorithms next.

Exploring the .info file

Double-clicking on the info file in the image directory window allows us to inspect a variety of details about the acquisition process, from start to completion, including the hashed outputs.

This info file contains much more data than the log file produced by DC3DD, including the case management details.

Let's have a closer look at the hash details within the `.info` file:

We can see that the `MD5` and `SHA-1` hashes have been created and verified, as shown in the last line of the following screenshot:

```
MD5 hash                : 7d9171d9c5aabf743799ce4a323a9d45
MD5 hash verified source  : --
MD5 hash verified image   : 7d9171d9c5aabf743799ce4a323a9d45
SHA1 hash               : --
SHA1 hash verified source  : --
SHA1 hash verified image   : --
SHA256 hash             : --
SHA256 hash verified source: --
SHA256 hash verified image : --
Image verification OK. The image contains exactly the data that was written.
```

Figure 8.32 – Guymager log file hash outputs

This completes the evidence and drive acquisition processes using Guymager, which can be much simpler to use than the previous command-line tools, namely `dc3dd` and `dd`.

Let's have a look at another acquisition tool called FTK Imager, which can also be used within Kali Linux.

Drive and memory acquisition using FTK Imager in Wine

There are several tools for Windows systems that you may wish to take advantage of to be able to capture the memory and paging files on a Windows device. The forensic images can then be opened on your Kali machine for analysis, using **Volatility 3** for memory analysis and **Autopsy** for drive analysis. Let's first look at installing and using FTK Imager within Wine in Kali Linux.

Installing FTK Imager

FTK (**Forensic Toolkit**) Imager is a free Windows tool for the live acquisition of memory (RAM), the paging file, and drive images.

Follow these steps to install FTK Imager in Kali Linux to create forensic acquisitions:

1. First, download FTK Imager from the official website at `https://go.exterro.com/l/43312/2022-08-23/f7rytx`. Enter all relevant detail on the registration page. Once all fields are completed, click on the **Submit** button, and you will be prompted to download the application.
2. Once downloaded, click on the **Home** folder icon, then the `Downloads` folder, and then select **Open Folder**, as shown here:

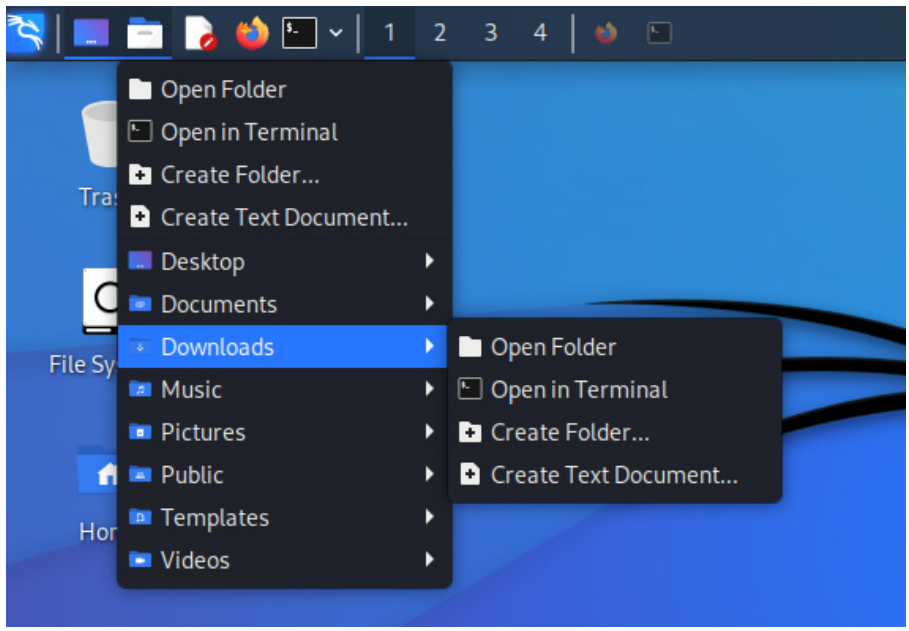


Figure 8.33 – The Kali folder menu

3. Next, right-click on the `AccessData_FTK_Imager` file you downloaded, choose **Open With**, and then select **Open with "Wine Windows Program Loader"**, as shown in the following screenshot.

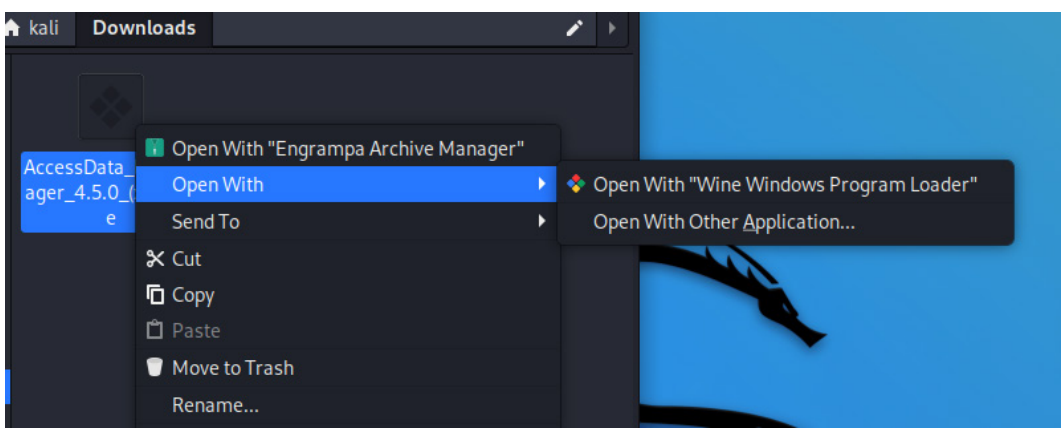


Figure 8.34 – Opening FTK Imager in Wine

4. FTK Imager will now begin installing on your Kali Linux machine. Click **Next** to proceed.

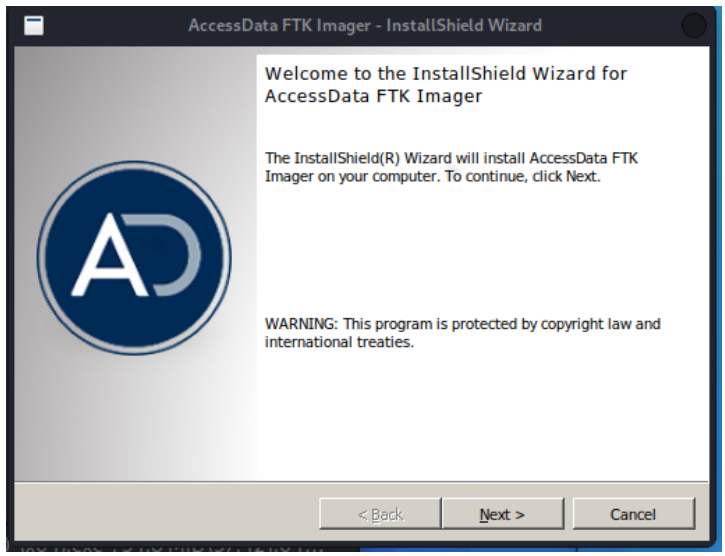


Figure 8.35 – The FTK Imager installer

5. Accept the license agreement, accept the destination folder by clicking **Next**, and then click on the **Install** button. Once the installation is complete, click on the **Finish** button to launch FTK Imager, as shown here.

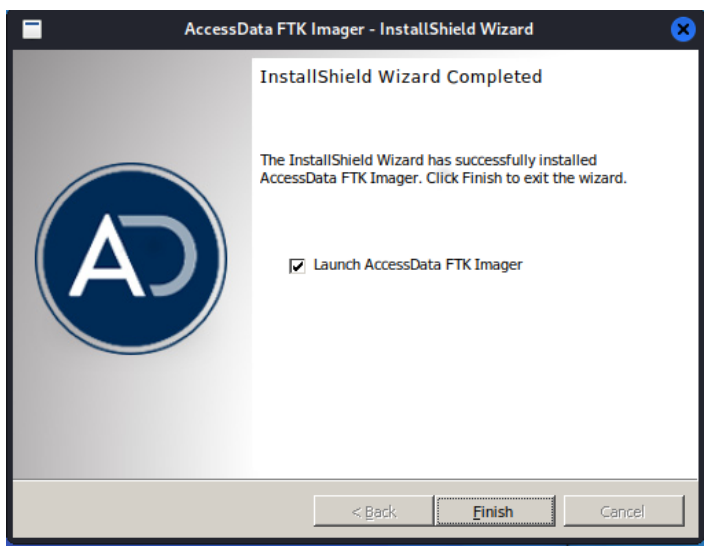


Figure 8.36 – FTK Installation complete

6. You may be prompted to download the Wine Gecko installer, which is required for some applications to run correctly. Click on **Install** to continue.

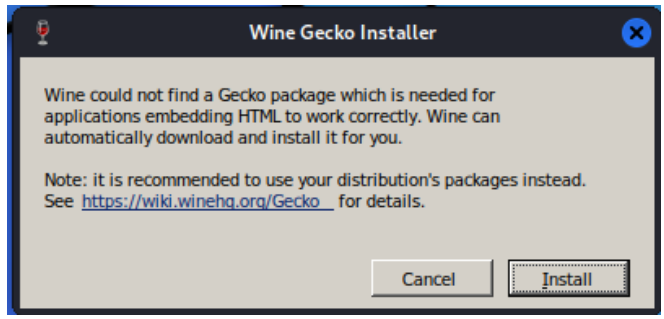


Figure 8.37 – The Wine Gecko Installer

FTK Imager should now be installed and running on your Kali Linux machine.

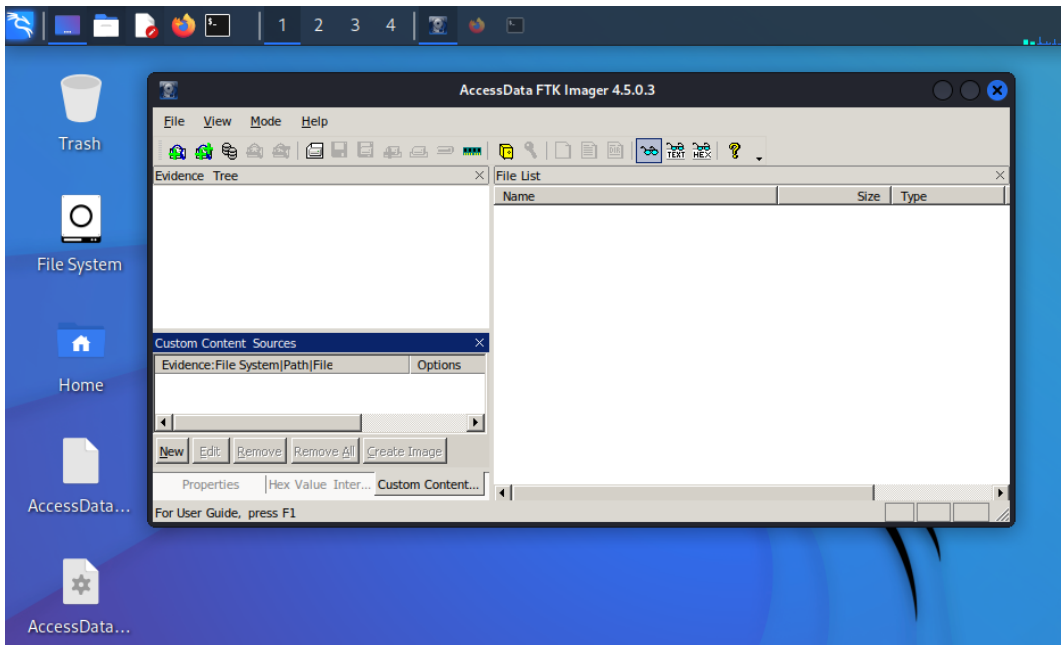


Figure 8.38 – The FTK Imager interface in Kali Linux

7. To view the options for imaging and acquisition, click on **File**.

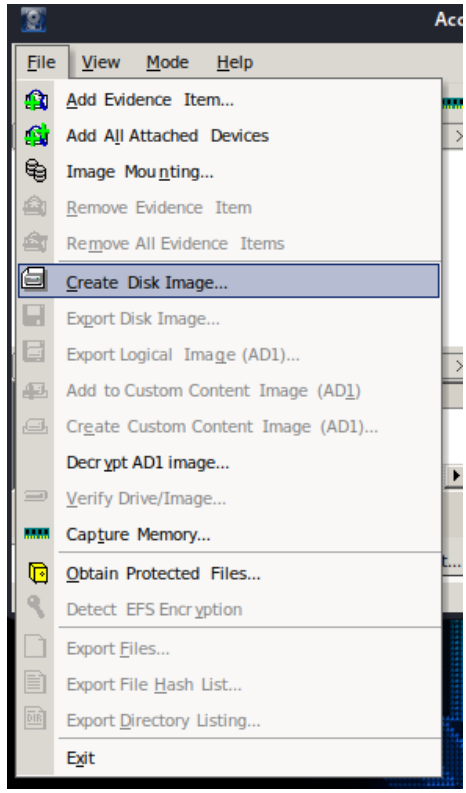


Figure 8.39 – The FTK Imager File menu

8. Within the **File** menu, we are presented with several options within FTK Imager for evidence acquisition and analysis. The **Create Disk Image…** option allows you to perform a forensic acquisition of physical and logical drives, contents of a folder, and CDs and DVDs. Click on the **Create Disk Image…** option.

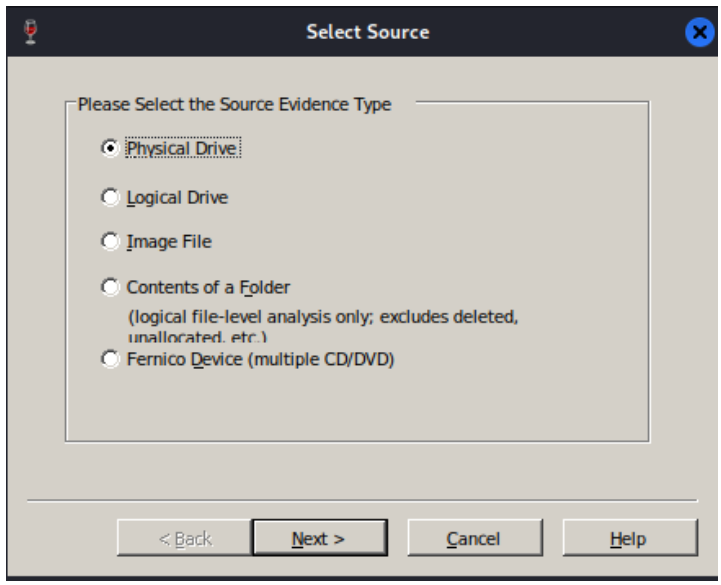


Figure 8.40 – FTK Imager source selection options

9. Click on **Next** to continue. I've selected a 32-GB Kingston physical drive to acquire. You can select any drive of your choice. Once selected, click on **Finish**.

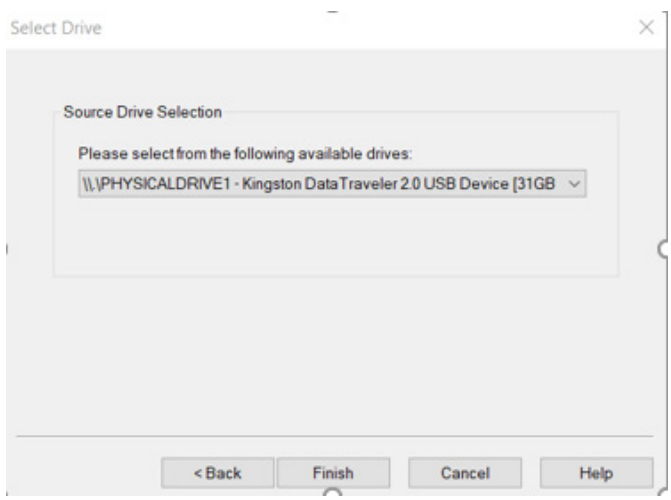


Figure 8.41 – Source Drive Selection

10. Next, we'll need to choose a destination to save the image file. Click on **Add**, choose the image type (**Raw**, **SMART**, **E01**, or **AFF**), and then click on **Next**.

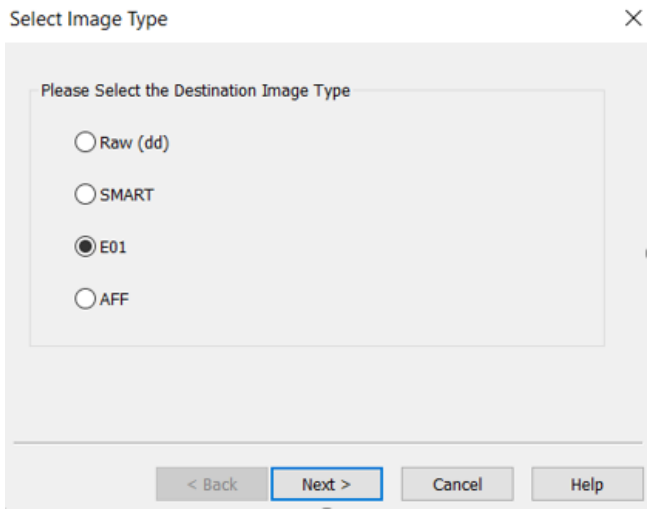


Figure 8.42 – Image acquisition type

11. Complete the form by filling in the **Evidence Item Information** fields, and then click on **Next**.

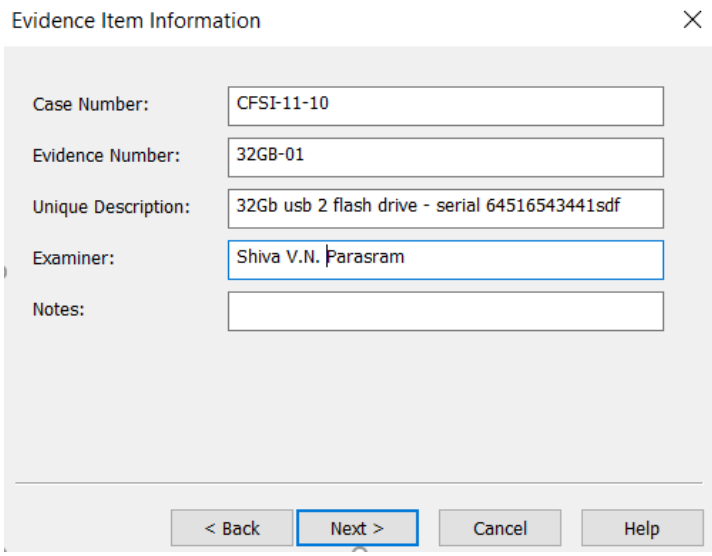


Figure 8.43 – Acquisition evidence details

12. Lastly, select the image destination folder and type a filename with an extension.

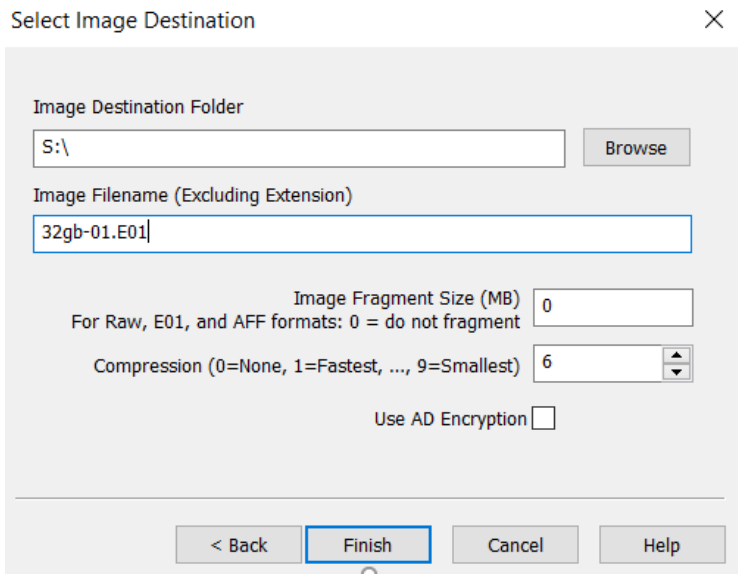


Figure 8.44 – The image destination folder

I've specified the image fragment size to be a value of `0`. This tells the software to not fragment or split the image file. Click on **Finish** and then **Start** to begin the acquisition process.

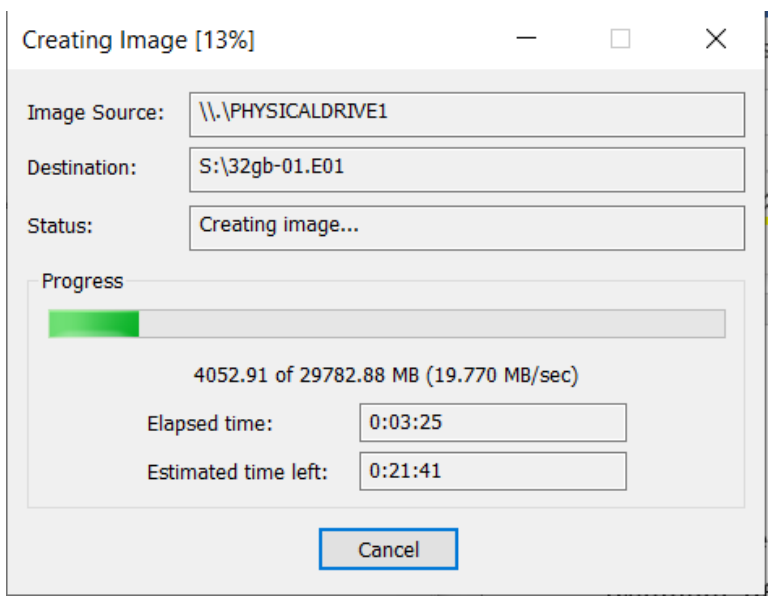


Figure 8.45 – The FTK Imager evidence acquisition process

The created disk image can now be analyzed using tools of your choice, such as Autopsy, which will be covered in *Chapter 10*, *Memory Forensics and Analysis with Volatility 3*, and *Chapter 11*, *Artifact, Malware, and Ransomware Analysis*.

RAM acquisition with FTK Imager

We can also perform live acquisition with FTK Imager, whereby we acquire the RAM and paging file:

1. Click on **File** and **Memory Capture**.

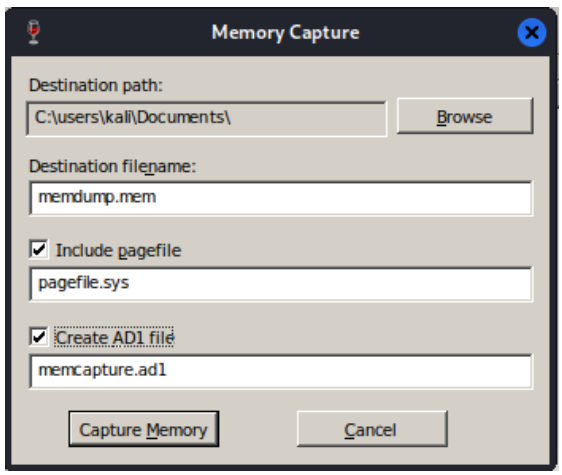


Figure 8.46 – The FTK Imager memory and paging file acquisition process

2. Next, select the destination path, and specify a filename for the memory dump (`.mem`) file. To include the pagefile, check the **Include pagefile** checkbox.
3. Click on **Capture Memory** to begin the acquisition process.

The status bar indicates when the process is completed. This is usually not a lengthy process compared to the static dive acquisition process. Using this within a VM may be problematic; however, it can be used on your standalone installation of Kali with Wine installed or any Windows machine.

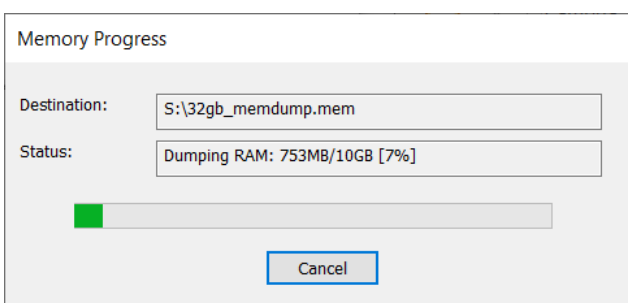


Figure 8.47 – The FTK memory acquisition process

This was a fairly simple process of acquiring RAM using FTK Imager. Let's look at another tool called RAM Capturer that can also be used for RAM acquisition.

RAM and paging file acquisition using Belkasoft RAM Capturer

Belkasoft is a company that creates forensic tools and also has a full suite of tools available for forensic acquisition and analysis, along with its free RAM capturer tool, which can be downloaded at `https://belkasoft.com/ram-capturer`. This tool is best used in Windows but is mentioned here because of its popularity and speed when performing memory and paging file analysis.

After browsing the `https://belkasoft.com/ram-capturer` page, click on the **Download Now** button, enter your email address, and click on **Proceed**. An email with the download link should be sent to you within 24 hours.

Once downloaded and extracted on your Windows machine, choose the appropriate version (`x86` or `x64`) and launch the environment.

The GUI is as simple as it gets with Belkasoft RAM Capturer. You are prompted to specify an output folder path, and from there, it captures the memory and paging file after clicking on **Capture!**.

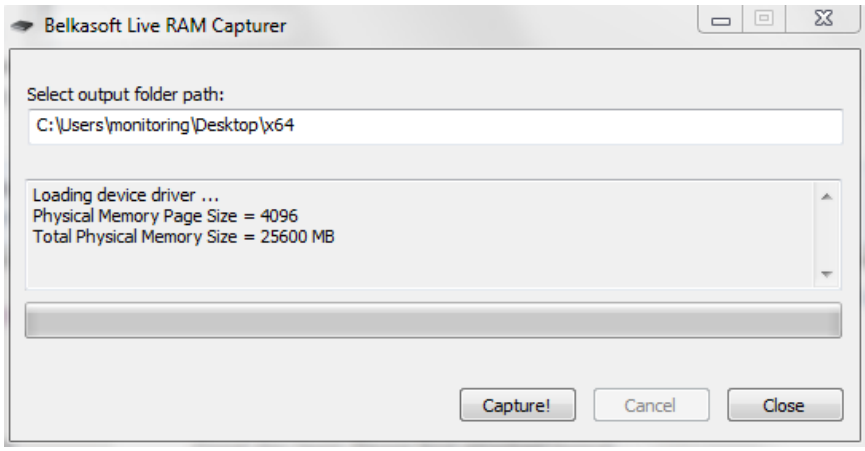


Figure 8.48 – Belkasoft Live RAM Capturer acquisition

The tool takes a few minutes to perform the acquisition, and from there, you can hash and analyze using the tools of your choice. This concludes our usage of RAM Capturer, one of the simplest tools available for RAM acquisition.

Summary

In this chapter, we covered several tools that can be used natively in Kali Linux and another tool called FTK Imager, which is native to Windows but can be installed in Kali Linux once Wine has been installed, for the acquisition of digital evidence. We first learned about the importance of being able to identify your devices so that you can accurately acquire a forensic copy or image of an evidence file using the `fdisk - l` command. For forensic analysis, bitstream copies of the evidence are needed, as these provide an exact copy of the evidence, bit by bit, which is why we used tools such as DC3DD, DD, and Guymager.

Firstly, we used DC3DD, an enhancement of the data dump tool, and through the Terminal, we performed quite a few tasks, including device imaging, hashing, verification, and drive wiping. We also performed acquisition using DD, which is very similar to DC3DD.

Our third tool, Guymager, has built-in case management abilities and also has many functional similarities to DC3DD, but it comes as a GUI tool and may be easier for beginners.

All the tools covered in this chapter deliver accurate and forensically sound results. For those that may not constantly work with Guymager, DD, and DC3DD, Guymager may be the easier tool to use, given that all acquisition options including cloning are readily available through the GUI, along with an easy-to-read log, which provides case management details. For advanced uses such as drive wiping, however, you may wish to use DC3DD. In the end, however, the choice remains yours.

We also looked at FTK Imager and Belkasoft Ram Capturer. FTK Imager runs on Windows and can acquire RAM and disk images, but it can easily be installed in Kali using Wine, whereas Belkasoft RAM Capturer (also for Windows) performs only RAM acquisition.

Not bad for our first forensics tools in Kali Linux! Next, we'll move on to some analysis and file recovery tools. Exciting stuff!

9

File Recovery and Data Carving Tools

Now that we've learned how to create forensic images of evidence, let's look at the file recovery and data carving process, using specific tools in Kali Linux.

File carving retrieves data and files from unallocated space using specific characteristics such as the file structure and file headers, instead of traditional metadata created by, or associated with, filesystems. A simple way to think of file carving is to think of an ice sculpture. It starts off with a huge block of ice, which, when given to a skilled individual, can be chipped away into a piece of art. In the same way, DFIR investigators and analysts can create a forensic image using any of the tools mentioned in the previous chapter, and then use a variety of tools to extract useful data and files from the acquired forensic image.

In this chapter, we'll cover the following topics:

- File basics
- Downloading sample files for the labs in this chapter
- Performing file recovery and data carving using Foremost
- Performing image recovery with Magicrescue
- Data carving with Scalpel
- Data extraction with `bulk_extractor`
- NTFS recovery using `scrounge_ntfs`
- JPEG recovery using Recoverjpeg

File basics

Before we get started on the practical aspects, let's briefly go through some common terminology.

When we last covered filesystems in *Chapter 6, Understanding File Systems and Storage*, we learned that various operating systems use their own filesystems to be able to store, access, and modify data. Storage media use file systems to do the very same.

Metadata, or data about data, helps an operating system identify data. Metadata includes technical information, such as the creation and modification dates, and the file type of the data. This data makes it much easier to locate and index files.

As the name implies, **unallocated space** is an area of storage media that has been marked by an operating system or file table as empty, or unallocated to any file or data. Although the location of, and information about, the file is not present and sometimes corrupted, there are still characteristics about the file that reside in its header and footer that can identify the file, or even fragments of the file.

Even if a file extension has been changed or is missing altogether, file headers contain information that can identify the file type and we can attempt to carve the file by analyzing the header and footer information. Data carving is quite a lengthy process and should be done using automated tools to save time. It also helps if the investigator has an idea of what file types they are looking for, giving them a better focus and saving time. Nevertheless, this is forensics, and we know that time and patience are key.

Some common file types, as displayed in hexadecimal format within the file headers, include the following:

- **Joint Photographic Experts Group (JPEG)**: FF D8 FF E0
- **Portable Document Format (PDF)**: 25 50 44 46

While more on the analysis of files and headers will be looked at in *Chapter 12*, *Autopsy Forensic Browser,* and *Chapter 13*, *Performing a Full DFIR Analysis with the Autopsy 4 GUI*, using Autopsy, let's first download the sample files we will be using for data carving in Kali Linux.

Downloading the sample files

Before we begin our DFIR file recovery and data carving activities, I thought it would be a good idea to first have all sample files downloaded, which will allow us to jump straight into the specifics of using each tool without the process being interrupted.

We will be using the following sites, which generously offer free datasets and acquisitions to the public. A special thank you to all that have compiled and invested their time and resources to make these datasets available to us:

- Digital Forensics Tool Testing Images: `https://dftt.sourceforge.net/`
- **Computer Forensic Reference DataSets** (**CFReDS**): `https://cfreds.nist.gov/`
- Digital Corpora: `https://digitalcorpora.org/`

- Foremost sample file: `https://cfreds-archive.nist.gov/FileCarving/Images/L0_Documents.dd.bz2`
- Scalpel: `http://prdownloads.sourceforge.net/dftt/11-carve-fat.zip?download`
- Bulk Extractor: `https://digitalcorpora.s3.amazonaws.com/corpora/drives/nps-2010-emails/nps-2010-emails.E01`

Although only the preceding sample files will be used for our lab purposes, you are certainly free to download other sample files from the aforementioned sites and try to carve those images, using the following tools where applicable.

File recovery and data carving with Foremost

Foremost is a simple and effective CLI tool that carves and recovers files by reading the headers and footers of the files. We can start Foremost by clicking on **Applications | 11 - Forensics | Foremost**. However, I prefer starting Foremost from the Terminal within the folder containing our sample acquired files, which will simplify the entire process without errors. So, let's get started:

1. If you haven't yet downloaded the sample forensic acquisition file, you can do so now by clicking on this link:

   `https://cfreds-archive.nist.gov/FileCarving/Images/L0_Documents.dd.bz2`

2. Once this file has been downloaded, I recommend creating a *new folder* in the `Downloads` folder by right-clicking in the `Downloads` folder and clicking on **Create Folder**. We will call this folder `Foremost`. We will also create folders for each tool as we move along in this chapter to avoid having a cluttered `Downloads` folder, and for the sake of having an organized DFIR workspace.

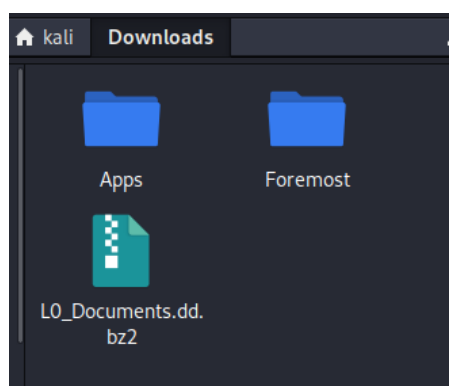


Figure 9.1 – File downloads with the Downloads folder

3. Drag the `L0_Documents.dd.bz2` file into the `Foremost` folder that you just created. By default, the `L0_Documents.dd.bz2` file will be downloaded to your `Downloads` folder. The downloaded file is a compressed `BunZip (.bz2)` format and contains a `datadump (.dd)` acquisition file that must be extracted.

   To extract the `.dd` file, right-click on the `L0_Documents.dd.bz2` file and click on the **Extract Here** option, as shown here.

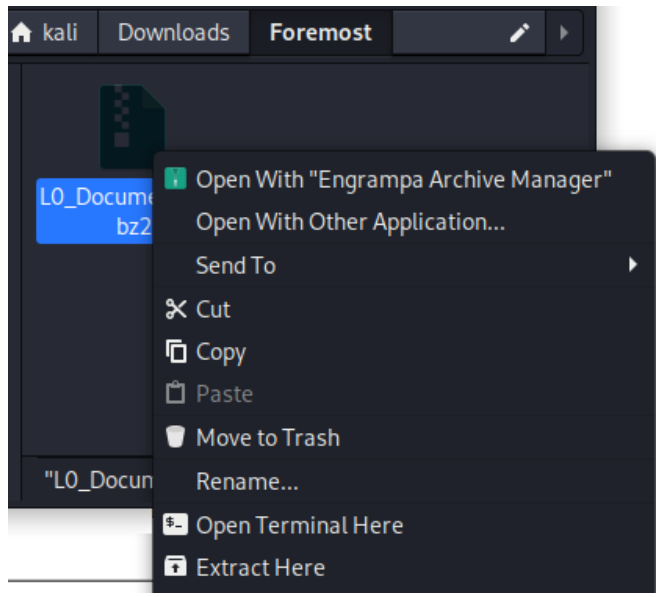


Figure 9.2 – Extracting the downloaded files

You should now see the `L0_Documents.dd` evidence file in your `Foremost` folder.

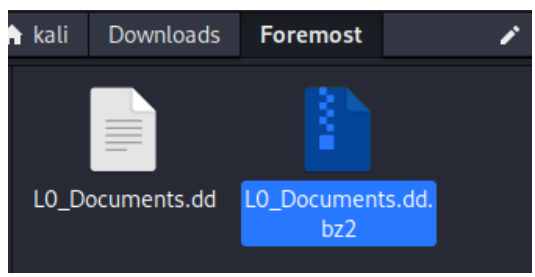


Figure 9.3 – The extracted image file

4. Here's the fun part! Let's click on the **New Terminal** icon at the top of the screen to open a new Terminal and begin carving and recovering files from our forensic acquisition sample.

   Once in the Terminal, we must change our working directory (using the `cd` command) to the `Foremost` folder within our `Downloads` folder. To do this, type the following commands in the Terminal and press *Enter*:

   ```
   cd /Downloads/Foremost
   ```

> **Important note**
>
> All commands are *case-sensitive* and should be typed exactly as I have typed them.

5. Next, type `ls` to list and show the files to ensure that we are in the correct directory, where our files are located.

   ```
   ls
   ```

   In the following screenshot, I've entered the aforementioned commands and can see the files we have downloaded and extracted.

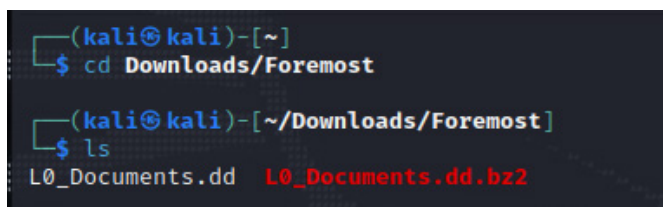


Figure 9.4 – Using the ls command

6. To have a better understanding of Foremost and the switches used, try browsing the **Foremost System Manager's Manual**. This can be done by entering the following command:

   ```
   man foremost
   ```

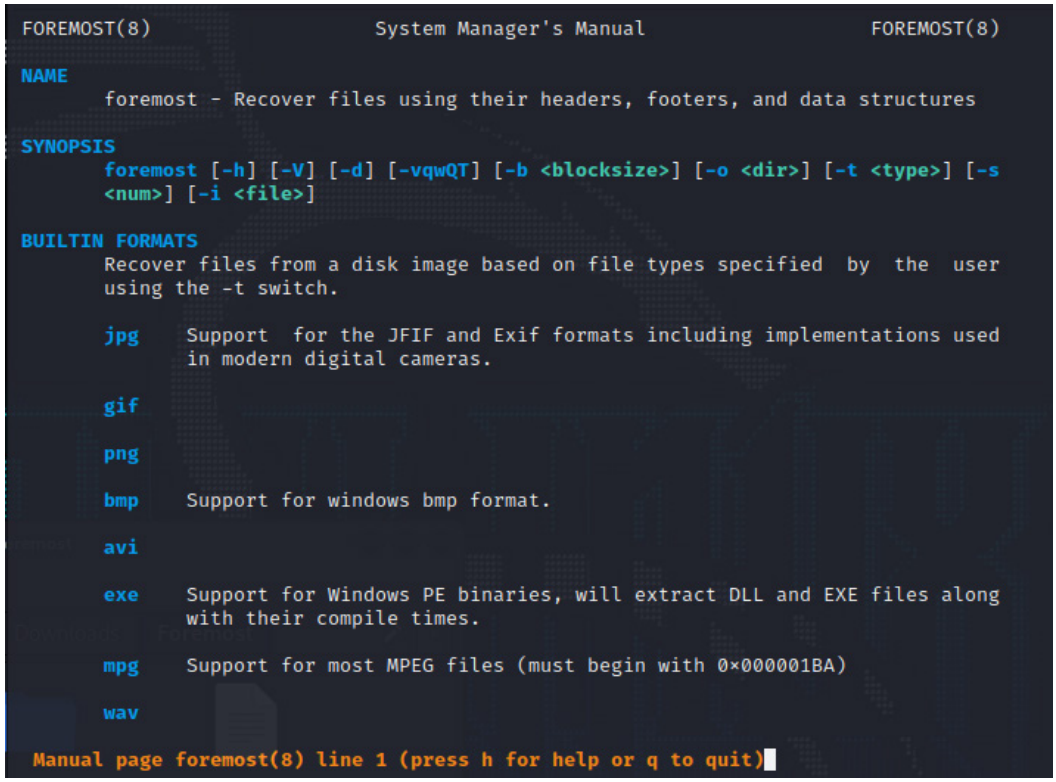


Figure 9.5 – The Foremost help manual

7. The syntax for using Foremost is as follows:

```
foremost -i (forensic image) -o (output folder) -options
```

8. Using the preceding format, let's enter the following command and options to carve files from the evidence file and recover them into a folder of our choice:

```
foremost -i L0_Documents.dd -o L0_Documents_Recovery
```

In this example, we have specified the `L0_documents.dd` file as the input file (`-i`) and specified an empty folder named `L0_Documents_Recovery` as the output file (`-o`). Additionally, other switches can be specified as needed.

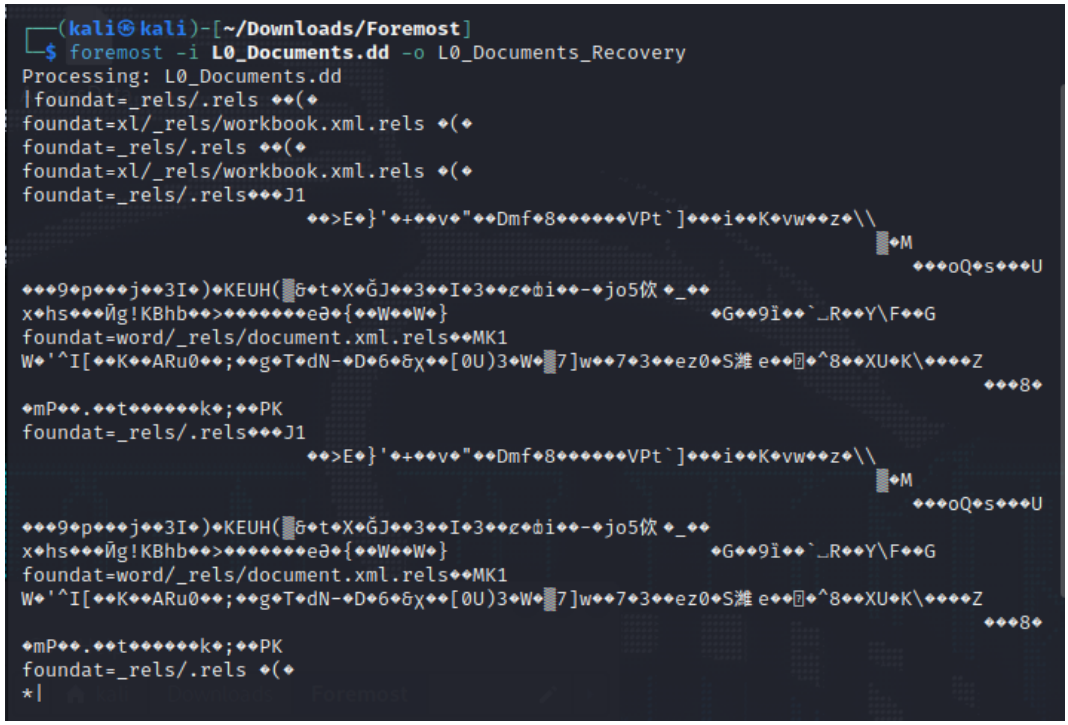


Figure 9.6 – Foremost command output

Although the characters in the preceding output may be unreadable, the carved files will be clearly categorized and summarized in the specified output folder.

9. To view our carved files, click on the **Home** folder icon at the top of the screen, click on the `Downloads` folder, and then click on **Open Folder**. Double-click on your `Foremost` folder and then open the `L0_Documents_Recovery` folder.

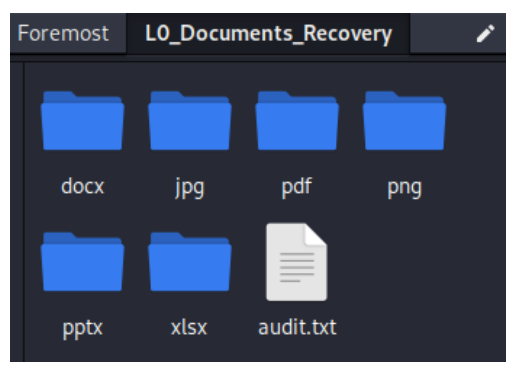


Figure 9.7 – The Foremost recovery folder

10. Foremost has automatically created subfolders for the relevant file types. To view a list of all files carved and recovered, open the `audit.txt` file.

```
File: L0_Documents.dd
Start: Fri Oct 28 10:58:48 2022
Length: 40 MB (42490880 bytes)

Num     Name (bs=512)       Size      File Offset     Comment

0:      00081300.jpg        27 KB     41625792
1:      00041137.xlsx       22 KB     21062144
2:      00051183.xlsx       13 KB     26205696
3:      00061210.docx       4 KB      31339520
4:      00071219.docx       3 KB      36464128
5:      00081227.pptx       881 KB    41588224
6:      00081364.png        15 KB     41658610        (256 x 256)
7:      00010000.pdf        3 MB      5120000  (PDF is Linearized)
8:      00026204.pdf        2 MB      13416448        (PDF is Linearized)
Finish: Fri Oct 28 10:58:48 2022

9 FILES EXTRACTED

jpg:= 1
zip:= 5
png:= 1
pdf:= 2
--------------------------------------------

Foremost finished at Fri Oct 28 10:58:48 2022
```

Figure 9.8 – The contents of the audit.txt file

11. The `audit.txt` file tells us that there were nine files carved/recovered, including images, MS Office, and PDF documents. Feel free to browse the various subfolders to view the carved files.

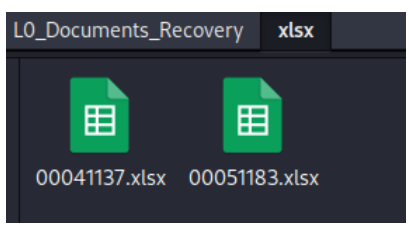


Figure 9.9 – Recovered .xlsx documents

Congratulations on your first successful DFIR recovery! Let us now move on to another recovery tool called Magic Rescue.

Image recovery with Magicrescue

Magicrescue is an older tool that is still very useful, as it quickly scans and recovers certain file types that must be manually specified. For this lab, I'll be using a 32-GB SanDisk flash drive, which once contained several audio, video, and image files that were all deleted by accident:

1. I'll first run the `fdisk` command to view the device information by typing the following:

```
sudo fdisk -l
```

The `fdisk` command output seen in the following screenshot shows that the flash drive is recognized as `sdc2`.

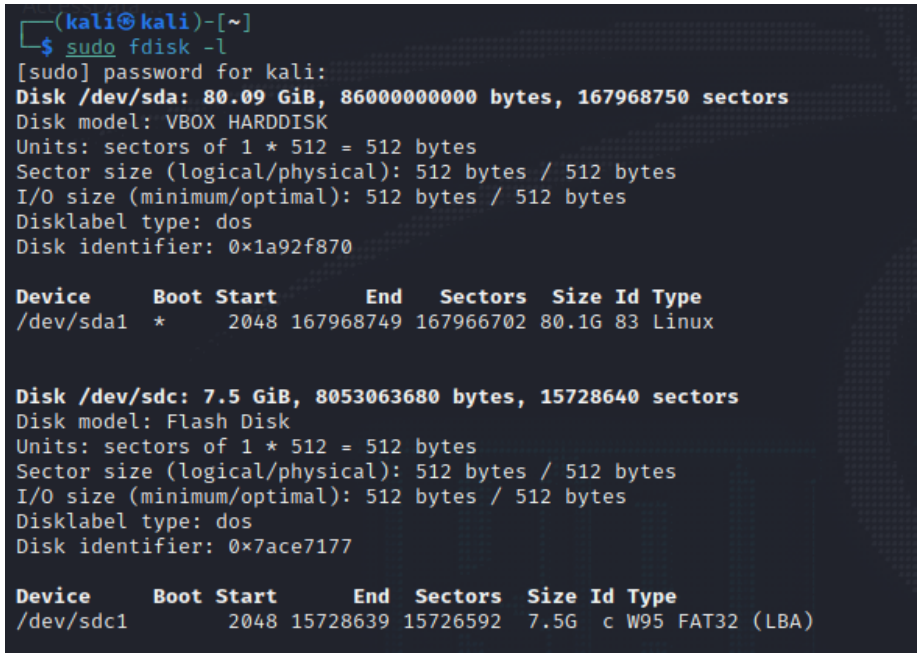


Figure 9.10 – fdisk command output showing disk information

2. Before we continue, it is useful to view the magicrescue help manual by typing the following:

```
man magicrescue
```

The following figure shows the usage options in magicrescue:

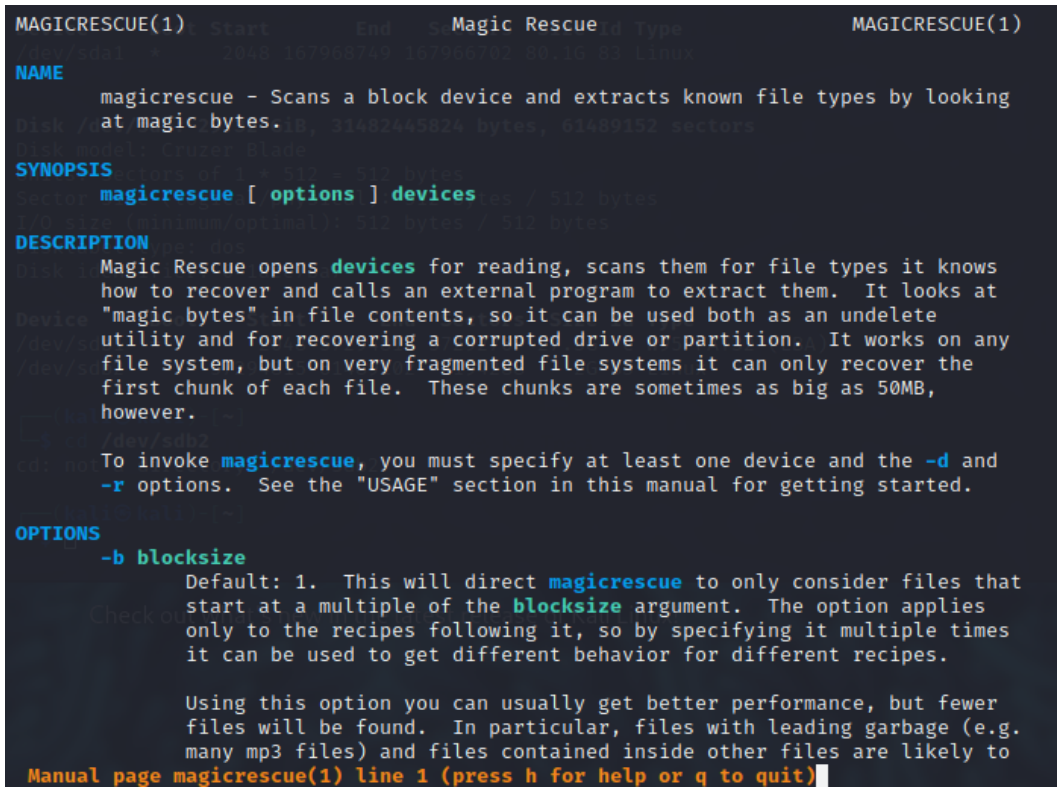


Figure 9.11 – Magicrescue usage options

3. To run magicrescue and search and recover files, we must first know which file types can be recovered. I've navigated to the `usr` | `share` | `magicrescue` | `recipes` folder, which shows the recipes or file types available for recovery by magicrescue. These *recipes*, as they are called, must be manually specified when using the `magicrescue` command.

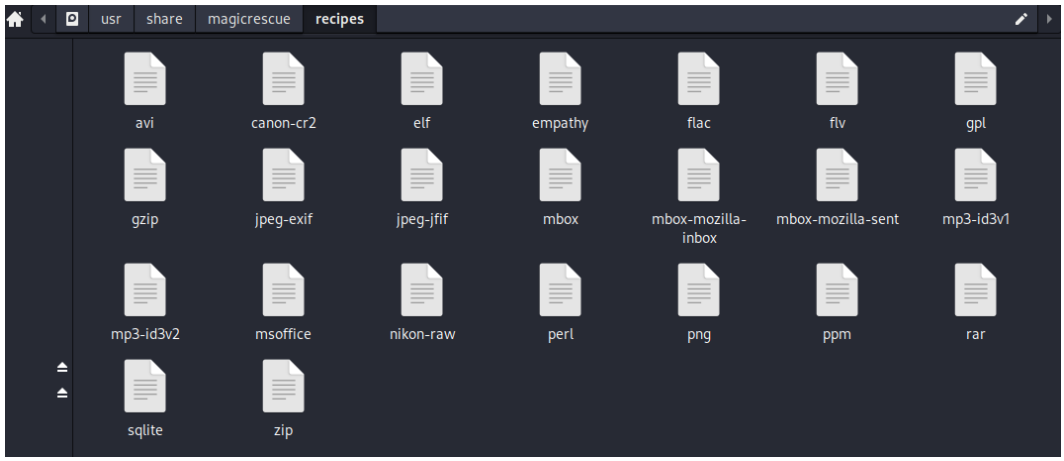


Figure 9.12 – Magicrescue recipe types

4. To use magicrescue to recover JPEG files from my flash drive, recognized as `sdb2`, I'll first create a folder on my desktop called `Rescue`, which will be the destination to which to save my recovered files.
5. In a new Terminal, change to the `Desktop` directory by typing the following:

```
cd Desktop
```

6. Then, create a new folder by typing the following:

```
mkdir Rescue
```

7. Change to the `Rescue` directory by typing the following:

```
cd Rescue
```

The following screenshot shows the output within the Terminal when changing directories.

Figure 9.13 – Changing working directories

8. To use Magic Rescue to recover files from my `sdb2` drive into the `Rescue` folder on the desktop, I'll type the following command with specific recipe options for image files:

```
sudo magicrescue -r png -r jpeg-jfif -r jpeg-exif -d
recovered -M io /dev/sdc > logs
```

Once run, Kali will output the details of the recovered files, as shown in the following screenshot.

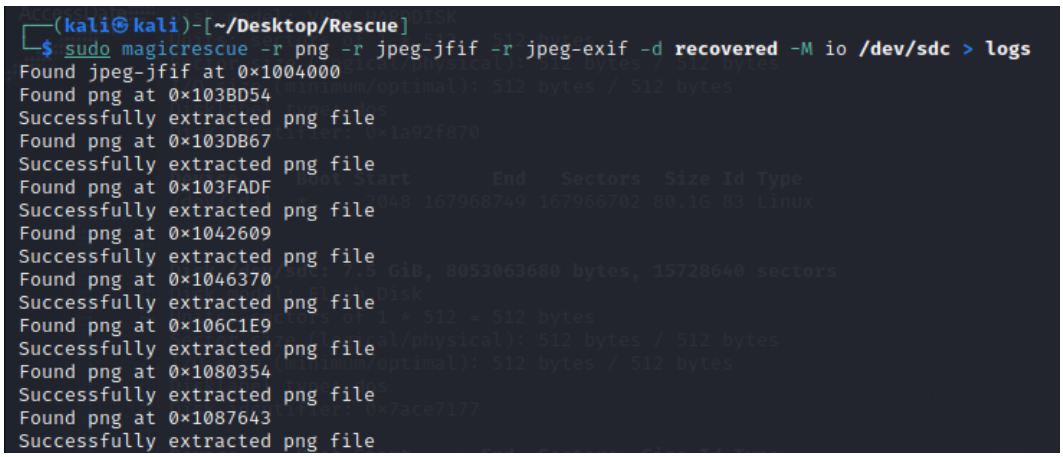


Figure 9.14 – Magicrescue file carving and recovery

9. We can now open the `Rescue` folder on the desktop to view the recovered files. As shown in the following screenshot, several deleted images were quickly recovered.

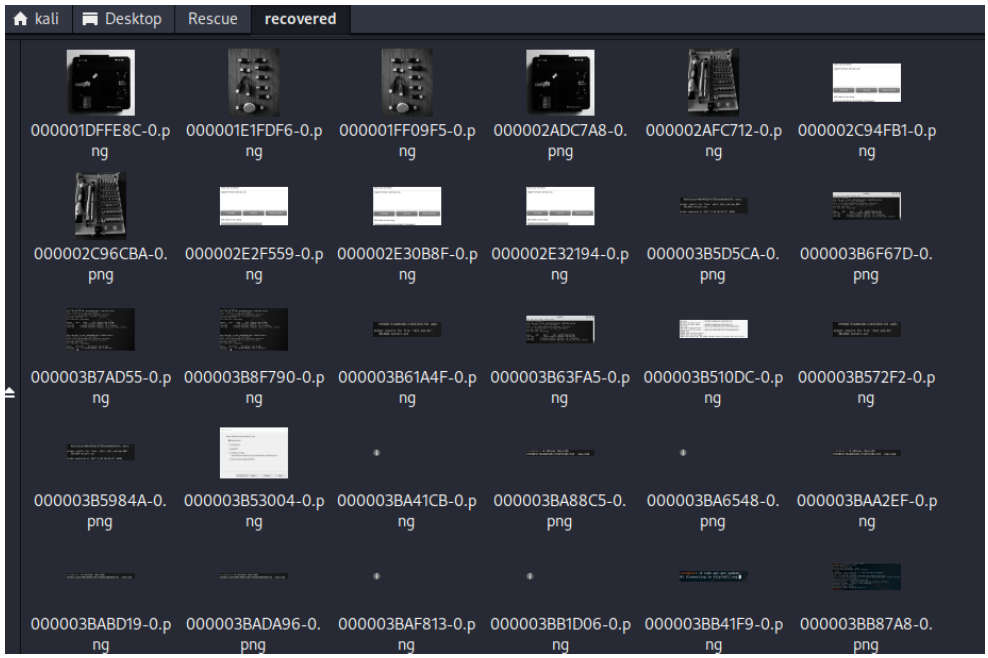


Figure 9.15 – Carved and recovered files

Let's now move on to another popular and powerful file carving tool called Scalpel.

Data carving with Scalpel

Scalpel was created as an improvement of a much earlier version of Foremost. Scalpel aims to address the high CPU and RAM usage issues of Foremost when carving data.

Unlike Foremost, file types of interest must be specified by an investigator in the Scalpel configuration file. This file is called `scalpel.conf` and is located at `etc/scapel/`:

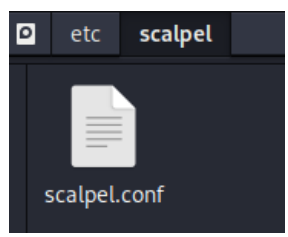


Figure 9.16 – The Scalpel configuration file

To specify the file types, the investigator must remove the comments at the start of the line containing the file type, as all supported file types are commented out with a hashtag at the beginning of the file type. The following screenshot shows the default Scalpel configuration file (`scalpel.conf`) with all file types commented out. Note that each line begins with a hashtag:

```
GIF and JPG files (very common)
#       gif     y       5000000         \x47\x49\x46\x38\x37\x61    \x00\x3b
#       gif     y       5000000         \x47\x49\x46\x38\x39\x61    \x00\x3b
#       jpg     y       5242880         \xff\xd8\xff???Exif         \xff\xd9   REVERSE
#       jpg     y       5242880         \xff\xd8\xff???JFIF         \xff\xd9   REVERSE
#
#
PNG
#       png     y       20000000        \x50\x4e\x47?       \xff\xfc\xfd\xfe
#
#
BMP     (used by MSWindows, use only if you have reason to think there are
#       BMP files worth digging for. This often kicks back a lot of false
#       positives
#
#       bmp     y       100000   BM??\x00\x00\x00
#
TIFF
#       tif     y       200000000       \x49\x49\x2a\x00
TIFF
#       tif     y       200000000       \x4D\x4D\x00\x2A
#
#-----------------------------------
# ANIMATION FILES
#-----------------------------------
#
AVI (Windows animation and DiVX/MPEG-4 movies)
#       avi     y       50000000 RIFF????AVI
#
```

Figure 9.17 – Scalpel file types within the conf file

I've removed the hashtags at the beginning of some of the lines to let Scalpel know to search for these specific file types; this also reduces the time taken to otherwise search for all supported file types. The following screenshot (*Figure 9.17*) shows that Scalpel will be searching for GIF, PNG, and TIFF files, as the comments have been removed. I have also removed hashtags from all video, MS Office, email, and PDF file types to allow Scalpel to search for additional file types. If you are having issues with saving the changes to the file, this may have to be done as a user with root access:

1. For this lab, we will use the `11-carve-fat.dd` evidence file that we previously downloaded. If you did not do so, you can do so now by clicking on the following link:

   `http://prdownloads.sourceforge.net/dftt/11-carve-fat.zip?download`

   The file downloads as a `.zip` file and must be extracted before being used.

2. In the `Downloads` folder, right-click on the `11-carve-fat.zip` file and click on the **Extract Here** option, which will then extract the `11-carve-fat.zip` file to the `Downloads` folder.
3. Let's use the Terminal to change directories to the `Downloads` folder like we did when using Foremost by typing the following:

```
cd Downloads
```

4. To view the available options and their usage in Scalpel, type the following command:

```
scalpel -h
```

As we can see in the following screenshot, we have many options available to us for carving and recovery when using `scalpel`.

```
Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.
Carves files from a disk image based on file headers and footers.

Usage: scalpel [-b] [-c <config file>] [-d] [-h|V] [-i <file>]
               [-m blocksize] [-n] [-o <outputdir>] [-O num] [-q clustersize]
               [-r] [-s num] [-t <blockmap file>] [-u] [-v]
               <imgfile> [<imgfile>] ...

-b  Carve files even if defined footers aren't discovered within
    maximum carve size for file type [foremost 0.69 compat mode].
-c  Choose configuration file.
-d  Generate header/footer database; will bypass certain optimizations
    and discover all footers, so performance suffers.  Doesn't affect
    the set of files carved.  **EXPERIMENTAL**
-h  Print this help message and exit.
-i  Read names of disk images from specified file.
-m  Generate/update carve coverage blockmap file.  The first 32bit
    unsigned int in the file identifies the block size. Thereafter
    each 32bit unsigned int entry in the blockmap file corresponds
    to one block in the image file.  Each entry counts how many
    carved files contain this block. Requires more memory and
    disk.  **EXPERIMENTAL**
-n  Don't add extensions to extracted files.
-o  Set output directory for carved files.
-O  Don't organize carved files by type. Default is to organize carved files
    into subdirectories.
-p  Perform image file preview; audit log indicates which files
    would have been carved, but no files are actually carved.
-q  Carve only when header is cluster-aligned.
-r  Find only first of overlapping headers/footers [foremost 0.69 compat mode].
-s  Skip n bytes in each disk image before carving.
```

Figure 9.18 – Scalpel usage options

5. To run Scalpel against our sample evidence file, type the following command:

```
scalpel -o scalpelOutput/ 11-carve-fat.dd
```

The following screenshot shows the output of the previously used command.

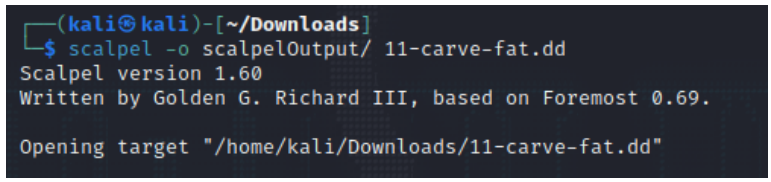


Figure 9.19 – Scalpel carving command

The following screenshot shows the continued output of the command that we just used.

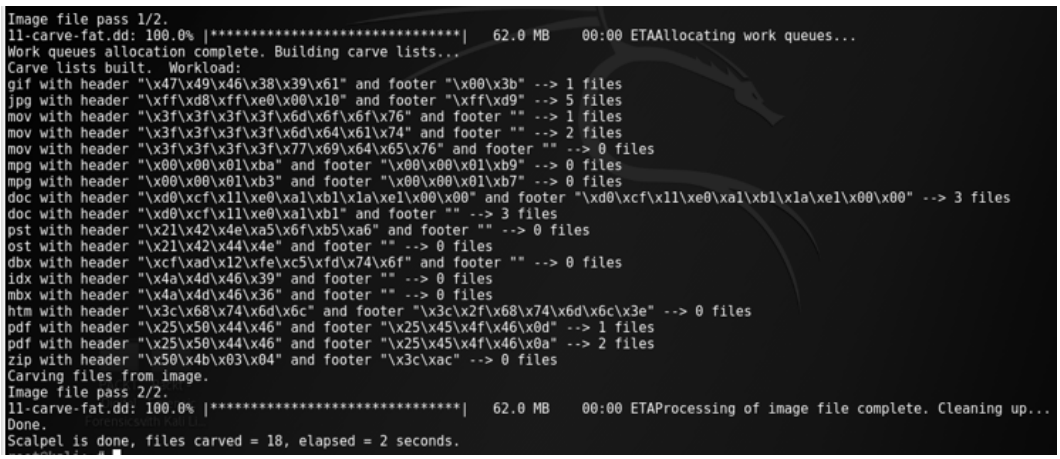


Figure 9.20 – Scalpel command output

In the previous screenshot (*Figure 9.20*), we can see that Scalpel builds a carve list, showing the file type with header and footer information, as well as the number of files carved.

Taking a closer look at the last few lines produced by the Scalpel output, we can see that the carving process was 100% complete, with 18 files being carved:

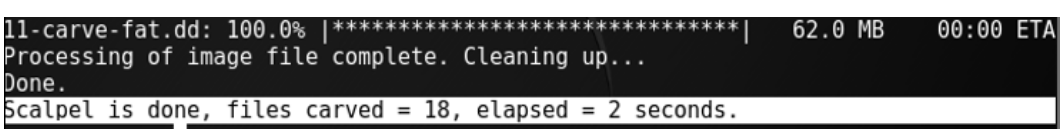


Figure 9.21 – A snippet of the Scalpel command output

6. We can now open the `scalpelOutput` folder to view our carved files. The results of the Scalpel output are similar to that of Foremost, with both output folders containing various subfolders with carved files, along with an `audit.txt` file with details of the findings:

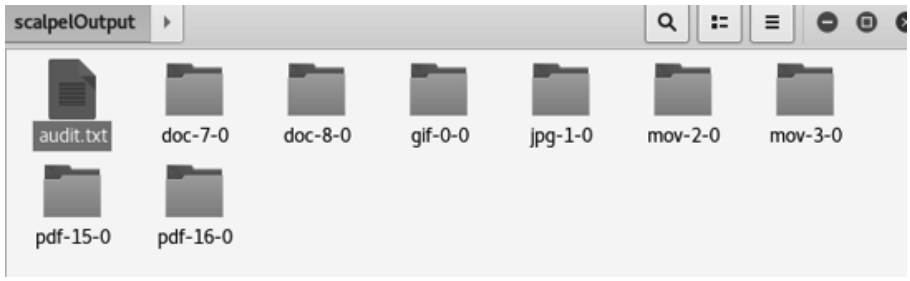


Figure 9.22 – Carved contents within the scalpelOutput folder

Foremost, magicrescue, and Scalpel, as we've seen so far, are quite impressive at file carving and data recovery; however, they are limited to specific file types only. For further extraction of data, we will now look at another tool called `bulk_extractor`.

Data extraction with bulk_extractor

The `bulk_extractor` tool extracts several additional types of information that can be very useful in investigations. Although `bulk_extractor` is quite capable of recovering and carving image, video, and document files, other data that can be carved and extracted by `bulk_extractor` includes the following:

- Credit card numbers
- Email addresses
- URLs
- Online searches
- Social media profiles and information

For this example, we will work with a freely available evidence file named `nps-2010-emails.E01`:

1. The `nps-2010-emails.E01` file can be downloaded directly from the digital corpora website, which allows the use of forensic evidence images for forensic research purposes.

   If not already downloaded, the file can be downloaded at `https://digitalcorpora.s3.amazonaws.com/corpora/drives/nps-2010-emails/nps-2010-emails.E01`.

2. Once downloaded, open a new Terminal and change to the `Downloads` folder, just as we previously did. You can view the available options and usage by typing the following:

```
bulk_extractor -h
```

The following screenshot shows the output of the previously used command.

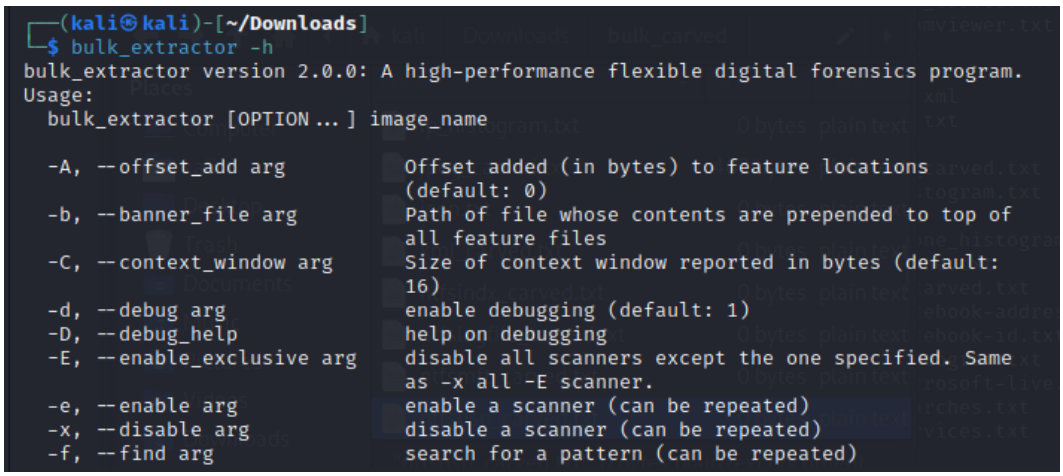


Figure 9.23 – bulk_extractor usage options

3. Like Foremost and Scalpel, the syntax for using `bulk_extractor` is quite simple and requires that an output folder (`-o`) and the forensic image be specified. Let us now use `bulk_extractor` to carve the evidence file and save all discovered files and information to a folder named `bulk_carved` in the `Downloads` folder, by typing the following command:

```
bulk_extractor -o bulk_carved nps-2010-emails.E01
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ bulk_extractor -o bulk_carved nps-2010-emails.E01
opening nps-2010-emails.E01

bulk_extractor version: 2.0.0
Input file: "nps-2010-emails.E01"
Output directory: "bulk_carved"
Disk Size: 10485760
Scanners: aes base64 elf evtx exif facebook find gzip httplogs json kml_carved msxml n
et ntfsindx ntfslogfile ntfsmft ntfsusn pdf rar sqlite utmp vcard_carved windirs winln
k winpe winprefetch zip accts email gps
Threads: 2
going multi-threaded ... ( 2 )
bulk_extractor      Fri Oct 28 14:22:21 2022

available_memory: 6097850368
bytes_queued: 0
depth0_bytes_queued: 0
depth0_sbufs_queued: 0
elapsed_time:  0:00:00
estimated_date_completion: 2022-10-28 14:22:20
estimated_time_remaining: n/a
fraction_read: 0.000000 %
max_offset: 0
```

Figure 9.24 – bulk_extractor command output

4. Once completed, `bulk_extractor` indicates that all threads have finished and provides a summary of the process and even some findings. The last line in the following screenshot lists 67 found email features:

```
Phase 2. Shutting down scanners
Computing final histograms and shutting down ...
Phase 3. Generating stats and printing final usage information
All Threads Finished!
Elapsed time: 2.006 sec.
Total MB processed: 10
Overall performance: 5.228 MBytes/sec 2.614 (MBytes/sec/thread)
sbufs created:   62785
sbufs unaccounted: 0
Time producer spent waiting for scanners to process data:         0:00:00 (0.00 second
s)
Time consumer scanners spent waiting for data from producer:      0:00:00 (0.04 second
s)
Average time each consumer spent waiting for data from producer:  0:00:00 (0.00 second
s)
*** More time spent waiting for reader. You need faster I/O for improved performance.
Total email features found: 67
```

Figure 9.25 – A snippet of the bulk_extractor output

5. To view the output and findings of `bulk_extractor`, we can change our directory to the `bulk_carved` folder in `Downloads` and use the `ls -l` command. In the following screenshot, we can see that several text files such as `domain_histogram.txt`, `domain.txt`, `email_domain_histogram.txt`, and `email.txt` contain data, as their file sizes are greater than zero:

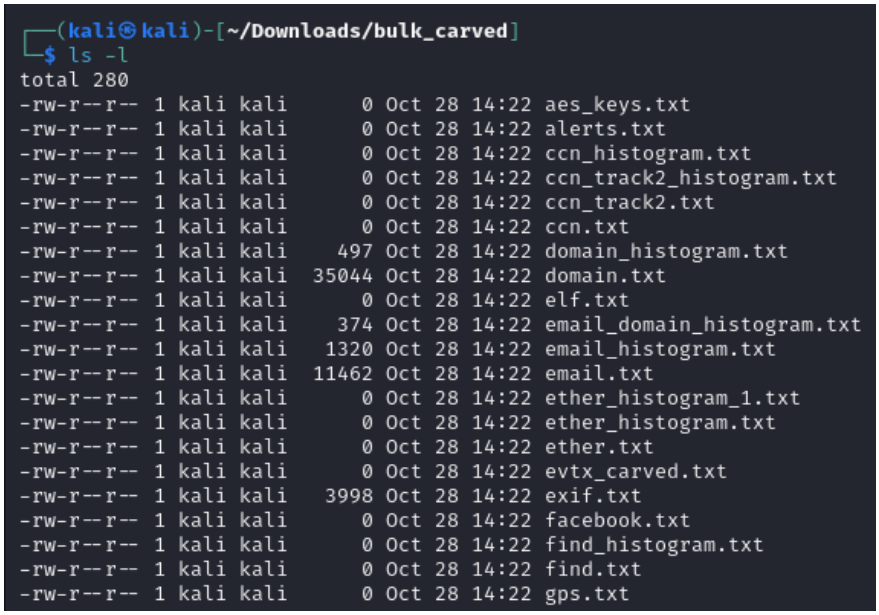


Figure 9.26 – A list of files carved by bulk_extractor

6. We can also open our `bulk_carved` folder and double-click on text files containing data. Note that not all files will contain information. Files with a file size of *zero* will contain no information.

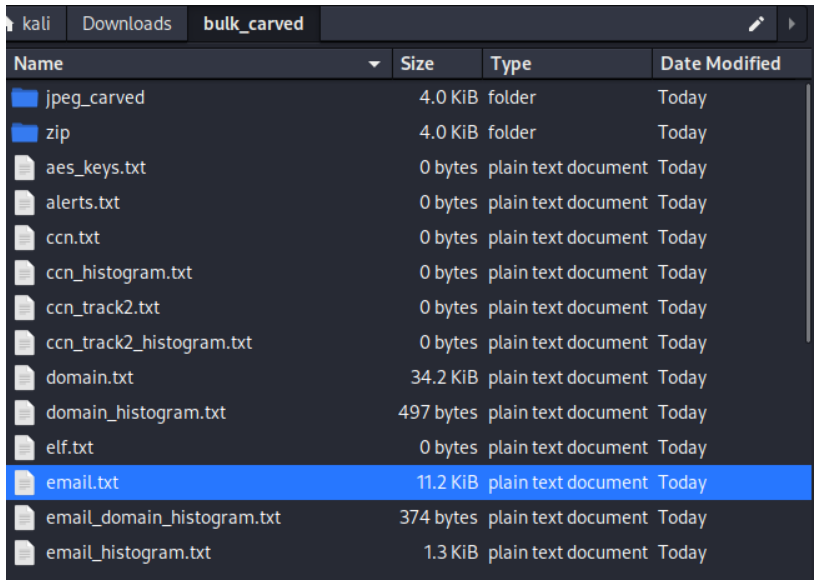


Figure 9.27 – Carved files within the bulk_extractor output folder

Let's have a look at the `email.txt` file, as shown in the following screenshot. We can see several email addresses and attachment details were carved by `bulk_extractor`.

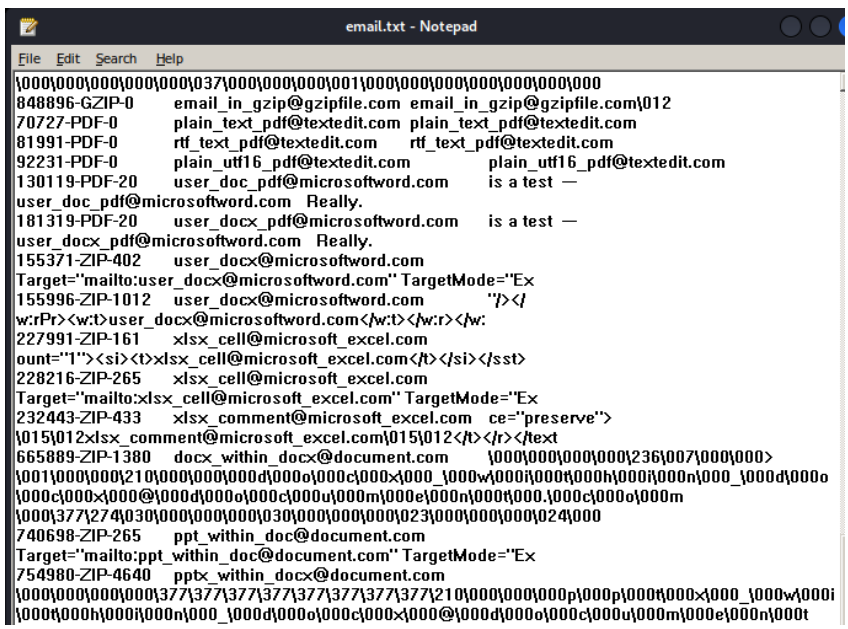


Figure 9.28 – Content within the carved email.txt file

Now that we've covered some of the most popular file carving tools, let's have a look at a file recovery tool called `scrounge-ntfs`.

NTFS recovery using scrounge-ntfs

`scrounge-ntfs` is a tiny tool that comes with the default version as well as the `Everything` metapackages of Kali Linux. A tiny but powerful tool, `scrouge-ntfs` is a command-line tool for recovering drives formatted as **NTFS** (short for **New Technology File System**), generally used by Windows devices. Most storage media, including flash drives and storage media, can also be formatted as NTFS using default formatting tools within Windows systems.

It's quite common for hard drives to have a corrupt **MFT** (short for **Master File Table**). This table stores information about all files, including size, permissions, content, and time stamps within NTFS drives. If this table becomes corrupt or is missing, the contents of the drive may be inaccessible, or the drive may not boot if it contains a bootable operating system.

Regardless of the reason for an NTFS drive becoming corrupted, `scrounge-ntfs` was built for the specific purpose of recovering files within an NTFS partition. For this lab, I'll be using an external hard drive formatted as NTFS, which has also had all its contents deleted:

1. Let's first see what our device is listed as in Kali by running the following:

   ```
   sudo fdisk -l
   ```

   The following screenshot shows the output of the previously typed command.

   ```
   Disk /dev/sdb: 465.76 GiB, 500107862016 bytes, 976773168 sectors
   Disk model: 2115
   Units: sectors of 1 * 512 = 512 bytes
   Sector size (logical/physical): 512 bytes / 4096 bytes
   I/O size (minimum/optimal): 4096 bytes / 4096 bytes
   Disklabel type: dos
   Disk identifier: 0x01ea53d4

   Device     Boot Start       End   Sectors   Size Id Type
   /dev/sdb1        16065 976768064 976752000 465.8G  f W95 Ext'd (LBA)
   /dev/sdb5        16128 976768064 976751937 465.8G  7 HPFS/NTFS/exFAT
   ```

   Figure 9.29 – fdisk output

   As seen here, my attached drive is listed as `sdb` too.

2. Before we begin the recovery process, I'd like to create a folder called `Scrounge_recovery` on the desktop where all `scrounge-ntfs`-recovered files will be saved.

   To do so, I'll switch to the `Desktop` directory by typing the following:

   ```
   cd Desktop
   ```

3. Then, I'll create the new directory by typing the following:

```
mkdir Scrounge_recovery
```

4. I'll now change my output directory to avoid having to specify the directory name when typing the `scrounge-ntfs` command, by typing the following:

```
cd Scrounge_recovery
```

The following screenshot shows the output of the previously typed command.

Figure 9.30 – Changing working directories

Note in the preceding screenshot (*Figure 9.30*) that we can see my current directory is `/Desktop/Scrounge_recovery`.

5. Let's now view the usage and options of `scrounge_ntfs` by typing the following:

```
Scrounge-ntfs -h
```

The following screenshot shows the output of the previously typed command.

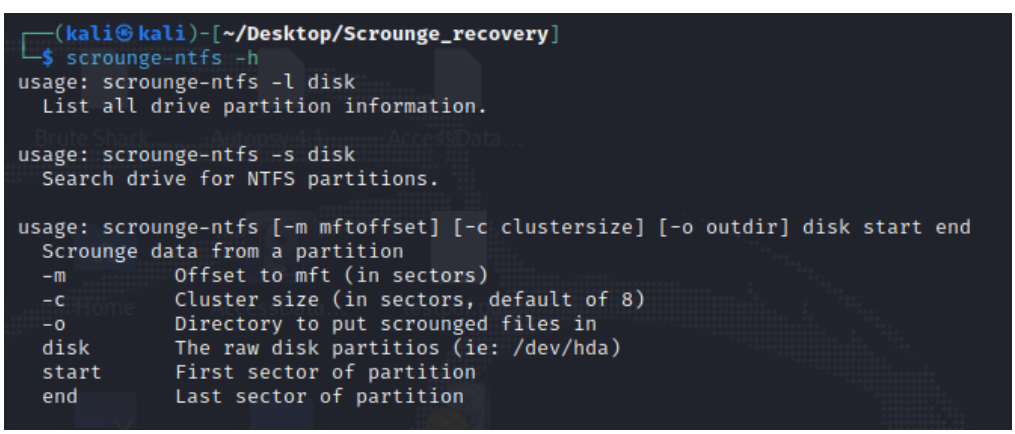


Figure 9.31 – scrounge-nfts usage options

6. Let's list all drive partitions by typing the following:

```
sudo scrounge-ntfs -l /dev/sdb
```

The following screenshot shows the output of the previously typed command.

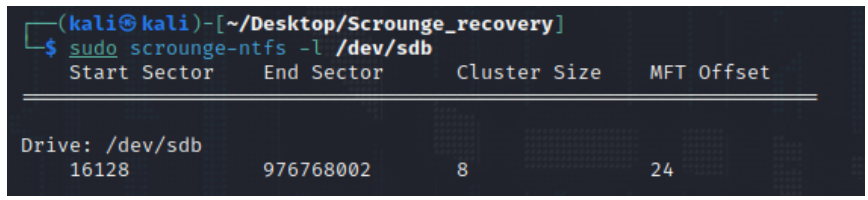


Figure 9.32 – The drive details

Now that we can see the specific sector, cluster, and offset values (which I've also listed here for clarity), we can use the format specified in the preceding `scrounge-ntfs -h` output (*Figure 9.32*):

- **Start Sector**: `16128`
- **End Sector**: `976768002`
- **Cluster Size**: `8`
- **MFT Offset**: `24`

7. The default usage for `scrounge-ntfs` is as follows:

```
sudo scrounge-ntfs [-m offset] [-c clustersize] [-o
outdir] disk start end
```

Let's break down the preceding options:

- `-o [outdir]` = the output directory to save all recovered files to
- `disk` = the drive to recover
- `start` = the start sector
- `end` = the end sector

8. For my drive, the command will be the following:

```
sudo scrounge-ntfs -m 629456 -c 8 /dev/sdb/ 2048 15726592
```

I did not specify an outdir or output directory command, as I am already within the output directory of `Scrounge_recovery`, which I previously created.

9. Once you have double-checked your values to ensure they are correct, hit the *Enter* key.

```
┌──(kali㉿kali)-[~/Desktop/Scrounge_recovery]
└─$ sudo scrounge-ntfs -m 24 -c 8 /dev/sdb 16128 976768002
[Scrounging via MFT ... ]
[Processing MFT ... ]
\$TxfLog.blf
\$TxfLogContainer00000000000000000001
\$TxfLogContainer00000000000000000002
\$RECYCLE.BIN
\$RECYCLE.BIN\S-1-5-21-1054828521-904020205-1243921375-24603
\$RECYCLE.BIN\S-1-5-21-1054828521-904020205-1243921375-24603\desktop.ini
\System Volume Information
\System Volume Information\WPSettings.dat
\System Volume Information\IndexerVolumeGuid
\$RECYCLE.BIN\S-1-5-21-2653766897-1606394338-1728658101-500\$R10G6QE\ISOs
\$RECYCLE.BIN\S-1-5-21-2653766897-1606394338-1728658101-500\$R10G6QE
\.Trash-1000\files\Microsoft SQL Server 2016 13.0.4001.0 (Service Pack 1)
```

Figure 9.33 – The scrounge-ntfs command and output

In the preceding output, you can see the files being recovered.

10. I can now list all files in the `Scrounge_recovery` directory or double-click on `Scrounge_directory` on the desktop to view my recovered files:

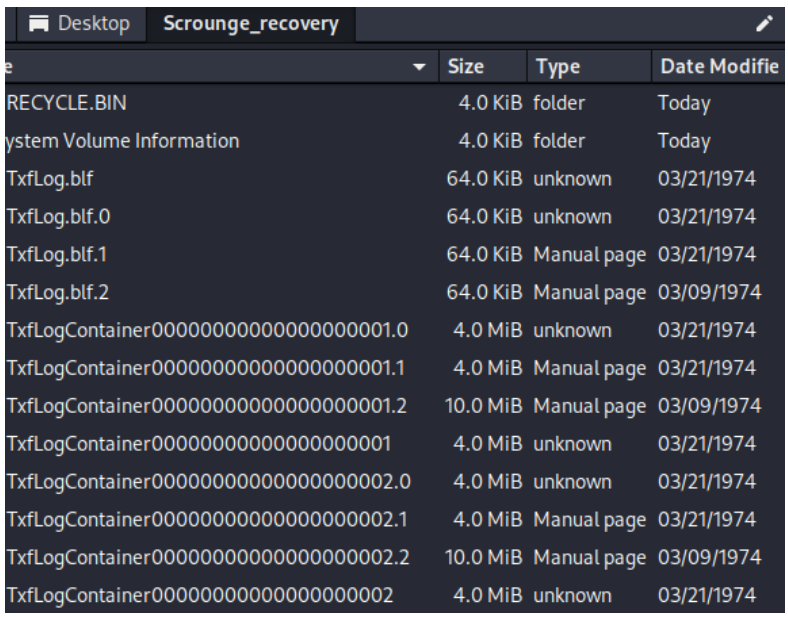


Figure 9.34 – The recovered files

As seen in this section, `scrounge-ntfs` is a very useful tool for NTFS drive and partition recovery. Let's move on to our final tool for this chapter called Recoverjpeg, which can be used to quickly recover image files.

Image recovery using Recoverjpeg

The last tool that I'll cover in this chapter is **recoverjpeg**, which, as the name implies, recovers JPEG images. This tool quickly and easily recovers deleted photos from any storage media type. For this lab, I'll be using a formatted 32-GB flash drive that once contained family photos I'd like to recover:

1. I'll first create a folder called `Recoverjpeg` on my desktop where all recovered files will be saved to. This helps keep things organized, as not specifying an output folder for all recovered files to be saved results in the recovered files being saved to the `Home` or current working directory, which can clutter your workspace.

   In the following screenshot, I've typed the `cd` command followed by `mkdir Recoverjpeg` to create this folder on my desktop:

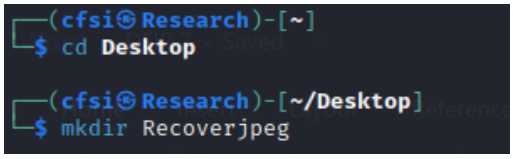


Figure 9.35 – Creating the output folder

2. Next, I'll install the `recoverjpeg` tool by typing the following command:

```
sudo apt-get install recoverjpeg
```

   The following screenshot shows the output of the previously typed command.

```
┌──(cfsi㉿Research)-[~/Desktop]
└─$ sudo apt-get install recoverjpeg
[sudo] password for cfsi:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  exif fonts-urw-base35 ghostscript graphicsmagick graphicsmagick-ima
  libgraphicsmagick-q16-3 libwmflite-0.2-7
Suggested packages:
  fonts-texgyre ghostscript-x graphicsmagick-dbg
The following NEW packages will be installed:
  exif ghostscript graphicsmagick graphicsmagick-imagemagick-compat g
  libwmflite-0.2-7 recoverjpeg
The following packages will be upgraded:
  fonts-urw-base35
1 upgraded, 8 newly installed, 0 to remove and 1090 not upgraded.
Need to get 8,913 kB of archives.
After this operation, 9,965 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Figure 9.36 – Installing recoverjpeg

3. Note that the last line in the preceding screenshot prompts us to press *Y* to continue the installation. Press *Y* and then *Enter* to proceed.
4. Next, I'll make sure that my flash drive is recognized in Kali by typing `sudo fdisk -l`:

```
┌──(cfsi㉿Research)-[~/Desktop]
└─$ sudo fdisk -l
Disk /dev/sda: 372.61 GiB, 400088457216 bytes, 781422768 sectors
Disk model: ST3400832NS
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xab60b093

Device     Boot     Start       End   Sectors   Size Id Type
/dev/sda1  *         2048 779421695 779419648 371.7G 83 Linux
/dev/sda2       779423742 781422591   1998850   976M  5 Extended
/dev/sda5       779423744 781422591   1998848   976M 82 Linux swap / Solaris


Disk /dev/sdb: 29.32 GiB, 31482445824 bytes, 61489152 sectors
Disk model: Cruzer Blade
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1ace7ba7

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sdb1  *        2048 57294814 57292767 27.3G  c W95 FAT32 (LBA)
/dev/sdb2       57294815 61489102  4194288    2G 83 Linux
```

Figure 9.37 – fdisk command output

In the preceding screenshot, we can see that my Cruzer Blade 32-GB flash drive is listed as `/dev/sdb`.

5. I'll now change directories to the `Recoverjpeg` folder that I previously created, where all recovered files will be stored, by typing `cd Recoverjpeg`:

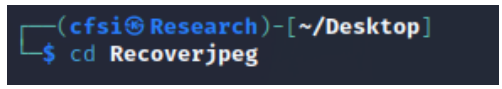


Figure 9.38 – Changing to the output directory

6. To view the usage options for Recoverjpeg, type `recoverjpeg`:

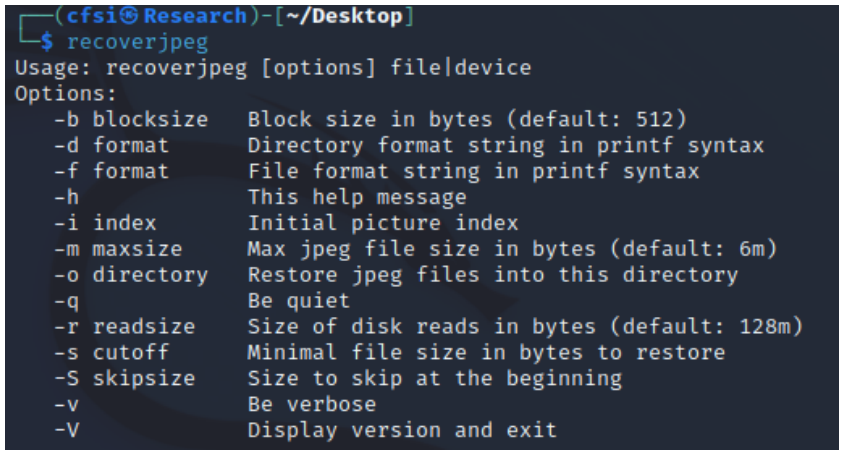


Figure 9.39 – recoverjpeg usage options

7. Now that I know which drive I'd like to recover images from, I now use `recoverjpeg` to scan and recover images by typing the following command:

```
sudo recover jpeg /dev/sdb
```

The following screenshot shows the output of the previously typed command.

Figure 9.40 – Using the recoverjpeg command

Important note

Depending on the size of the drive or image, this can take quite a few minutes. Once the recovery is complete, recoverjpeg will display the number of files recovered. In *Figure 8.44*, we can see that a total of 2,021 files were recovered.

```
└─$ sudo recoverjpeg /dev/sdb
Restored 2021 pictures
```

Figure 9.41 – recoverjpeg process completion

The recovered files were saved in the `Recoverjpeg` folder on the desktop but will not have their original names. Instead, all recovered images will be named in numeric order, starting from `image00000.jpg`, as shown in the following screenshot.

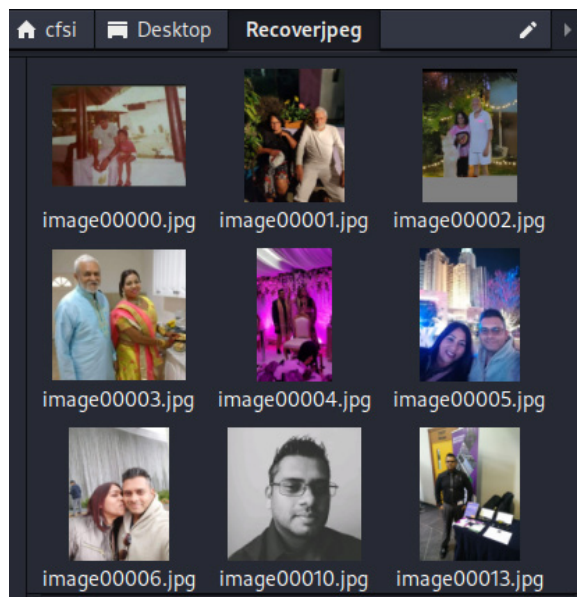


Figure 9.42 – Recovered jpeg photos within the output folder

I hope you found this tool as useful and easy to use as I do. This brings us to the end of our chapter. I encourage you to practice using these tools, as you will find that data recovery is one of the most common aspects of DFIR.

Summary

In this chapter, we learned about file recovery and data carving using popular open source tools in Kali Linux. We first performed file carving using the very impressive Foremost, which carved the entire downloaded forensic image for supported file types within the file header and footers. We then did the same using Magicrescue and Scalpel, but we had to make slight modifications by selecting the file types we wished to carve. Both Foremost and Scalpel presented us with an `audit.txt` file, summarizing the carve list and its details along with subfolders containing the actual evidence.

`bulk_extractor` is a wonderful tool that carves data and also finds useful information, such as email addresses, visited URLs, Facebook URLs, credit card numbers, and a variety of other information. `bulk_extractor` is also great for investigations requiring file recovery and carving, together with either Foremost or Scalpel, or even both.

Lastly, we looked at file recovery using `scrounge_NTFS` and `recoverjpeg` to recover data and images from drives rather than forensic images.

Now that we've covered file carving and recovery, let's move on to something more analytical. In the next chapter, we'll take a look at exploring RAM and the paging file as part of memory forensics, using the very powerful volatility. See you there!

10

Memory Forensics and Analysis with Volatility 3

In the previous chapters, we looked at the various methods for data carving and file recovery. In this chapter, we'll look at the analysis of content stored in **Random Access Memory** (**RAM**) using the very powerful Volatility 3. RAM is volatile, meaning that the data in RAM is easily lost when there is no longer an electrical charge or current going to the RAM. With the data on RAM being the most volatile, it ranks high in the order of volatility and must be forensically acquired and preserved as a matter of high priority.

Many types of data and forensic artifacts reside in RAM and the paging file. As discussed in previous chapters, login passwords, user information, running and hidden processes, malware, and even encrypted passwords are just some of the many types of interesting data that can be found when performing RAM analysis, further compounding the need for memory forensics.

In this chapter, we will look at the very powerful Volatility framework and its many uses in memory forensics, including the following topics:

- What's new in Volatility 3
- Downloading a sample memory dump file for analysis
- Installing Volatility 3 in Kali Linux
- Memory dump analysis using Volatility 3

What's new in Volatility 3

The Volatility framework is an open source, cross-platform incident response framework that comes with many useful plugins that provide the investigator with a wealth of information from a snapshot of memory, also known as a **memory dump**. The concept of Volatility has been around for a decade, and apart from analyzing running and hidden processes, it is also a very popular choice for malware analysis.

To create a memory dump, several tools, such as **Belkasoft RAM Capturer**, **FTK Imager**, **dd**, **dc3dd**, **CAINE**, **Helix**, and **LiME** (short for **Linux Memory Extractor**), can be used to acquire the memory image or memory dump (which we previously did in *Chapter 8*, *Evidence Acquisition Tools*) and then analyzed using various tools known as plugins within the Volatility framework.

The Volatility framework can be run on any **Operating System** (**OS**), that is, 32- and 64-bit, that supports Python, including the following:

- Windows XP, 7, 8, 8.1, and Windows 10
- Windows Server 2003, 2008, 2012/R2, and 2016
- Linux 2.6.11-4.2.3 (including Kali, Debian, Ubuntu, and CentOS) and macOS Leopard (10.5.x) and Snow Leopard (10.12.x)

Volatility supports several memory dump formats (both 32- and 64-bit), including the following:

- Windows crash and hibernation dumps (Windows 7 and earlier)
- VirtualBox
- VMware – `.vmem` dump
- VMware saved state and suspended dumps – `.vmss/.vmsn`
- Raw physical memory – `.dd`
- Direct physical memory dump over IEEE 1394 FireWire
- **Expert Witness Format** (**EWF**) – `.E01`
- **QEMU** (**Quick Emulator**)

Volatility even allows for conversion between these formats and boasts of being able to accomplish everything similar tools can.

In previous editions of this book, we used Volatility 2, which was based on the now-deprecated Python 2. Volatility 3 is based on Python 3 and has the following changes:

- Faster scans and processing of memory dumps
- Support for newer OSs
- More efficient plugins
- Operating-specific plugins

Before we begin using Volatility 3, let's first download all the sample memory dump files that we will be analyzing using Volatility.

Downloading sample memory dump files

For this chapter, we'll be using a memory dump called `cridex.vmem`, which we will be analyzing using a variety of Volatility 3 plugins. The file can be downloaded from `http://files.sempersecurus.org/dumps/cridex_memdump.zip`.

There are many other images that are publicly available for analysis at `https://github.com/volatilityfoundation/volatility/wiki/Memory-Samples`. To practice working with the Volatility framework and further enhance your analytical skills, you may wish to download as many images as you like and use the various plugins available in Volatility.

Let's first download and extract our sample memory dump, which we will later move to our Volatility installation folder for analysis. If you haven't already downloaded the file, please do so now.

I've downloaded the `cridex.vem` sample file to my `Downloads` folder. To extract the file, right-click on the file and click on **Extract Here** as you have done with previously downloaded files.

We will now install Volatility 3 and then copy the `cridex.vmem` memory dump file to our installation folder for analysis.

Now that we have downloaded all our sample files, let's install Volatility 3 on our Kali machines.

Installing Volatility 3 in Kali Linux

Volatility is no longer installed in Kali Linux by default and instead must be manually installed:

1. Let's first download Volatility 3 from the official site at `https://www.volatilityfoundation.org/releases-vol3`.
2. Be sure to click on the `.zip` file link within the **Volatility 3** tab, as seen in the following screenshot:

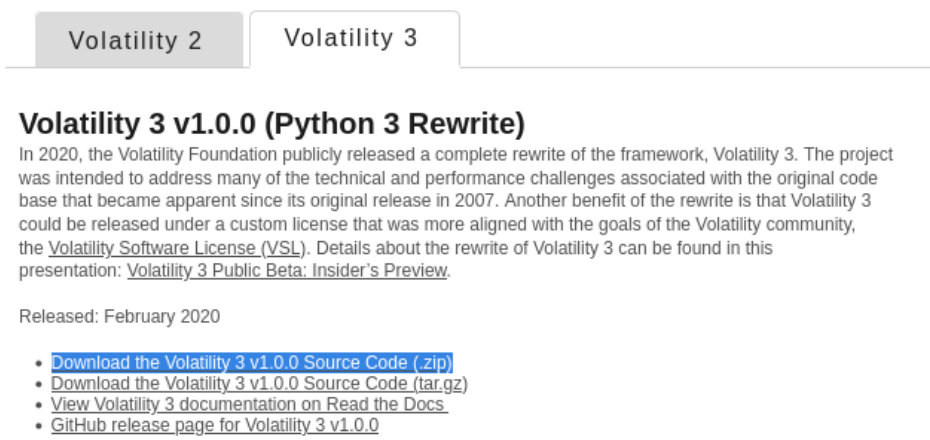


Figure 10.1 – Volatility downloads page

I've chosen to save the file in my `Downloads` folder. I've also extracted the files by right-clicking on the `.zip` file and then selecting **Extract Here**, as seen in the following screenshot.

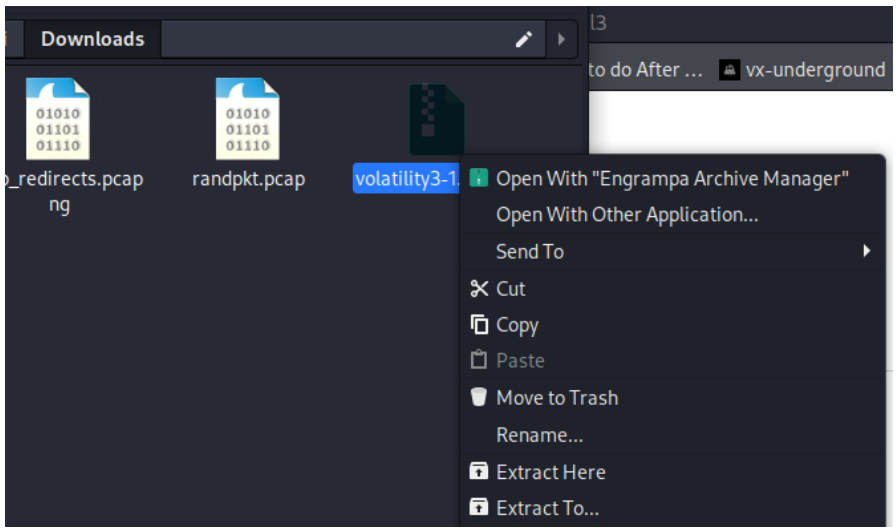


Figure 10.2 – Extracting the Volatility 3 ZIP file

> **Note**
> I've also renamed the extracted folder `volatility3` to simplify things. Be sure to take note of the folder name exactly as it is typed as it will be required later.

3. Before installing Volatility 3, I highly recommend updating your Kali installation to ensure that all files are the most current. Run the `sudo apt-get update` command to do this.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [18.7 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [43.0 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [111 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [161 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [235 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [897 kB
]
Fetched 63.2 MB in 27s (2,341 kB/s)
Reading package lists... Done
```

Figure 10.3 – Updating Kali Linux

4. Now that our system has been updated, let's install Python 3 by typing the following command:

```
sudo apt install python3
```

The following screenshot shows the output of the preceding command when installing Volatility 3.

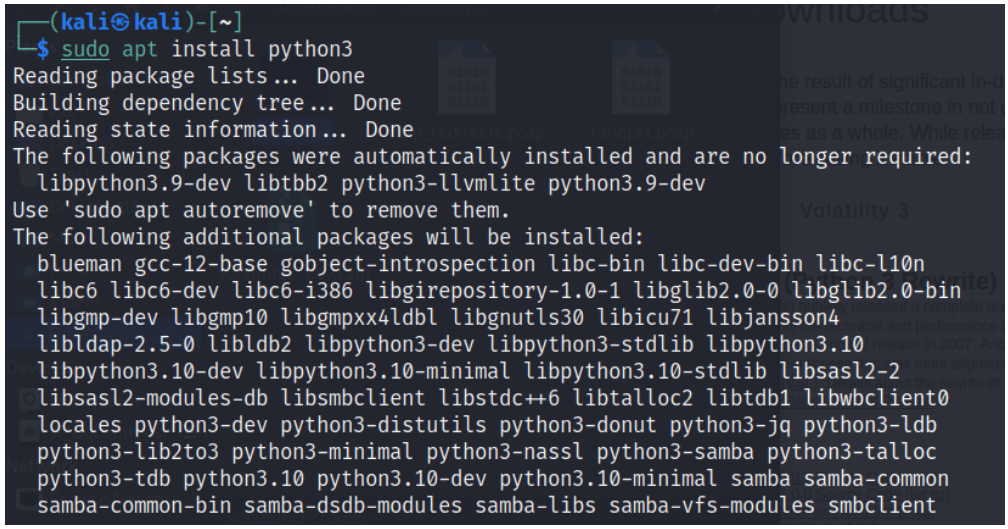


Figure 10.4 – Installing Python 3 in Kali

5. Volatility 3 also requires dependencies that must be installed for full functionality. To install all the required dependencies, type the following:

```
sudo apt install python3-pip python-setuptools build-
essential
```

The following screenshot shows the output of the preceding command when installing Python tools in Volatility 3.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install python3-pip python-setuptools build-essential
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
build-essential is already the newest version (12.9).
python-setuptools is already the newest version (44.1.1-1.2).
The following packages were automatically installed and are no longer required:
  libpython3.9-dev libtbb2 python3-llvmlite python3.9-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-wheel
The following packages will be REMOVED:
  python-pip
The following NEW packages will be installed:
  python3-pip python3-wheel
0 upgraded, 2 newly installed, 1 to remove and 1600 not upgraded.
Need to get 1,353 kB of archives.
After this operation, 2,052 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Figure 10.5 – Installing Volatility 3 dependencies

6. Let's now change directories to the folder containing all Volatility 3 files. In this instance, I have renamed my folder `volatility3` within the `Downloads` folder. To change to that directory, I'll use the following command:

```
cd Downloads/volatility3
```

The following screenshot shows the output of the preceding command.

```
┌──(kali㉿kali)-[~]
└─$ cd Downloads/volatility3
```

Figure 10.6 – Changing directories

7. We can then use the `ls` command to list and show all files within the `volatility3` folder:

```
┌──(kali㉿kali)-[~/Downloads/volatility3]
└─$ ls
development  LICENSE.txt  mypy.ini   setup.py     vol.py       volshell.spec
doc          MANIFEST.in  README.md  volatility3  volshell.py  vol.spec
```

Figure 10.7 – ls command output

In the preceding screenshot, we can see all files required to set up and run Volatility 3.

8. Within my `volatility3` folder, I can now install Volatility 3 by typing the following command:

```
sudo python3 setup.py install
```

The following screenshot shows the output of the preceding command when installing Python 3.

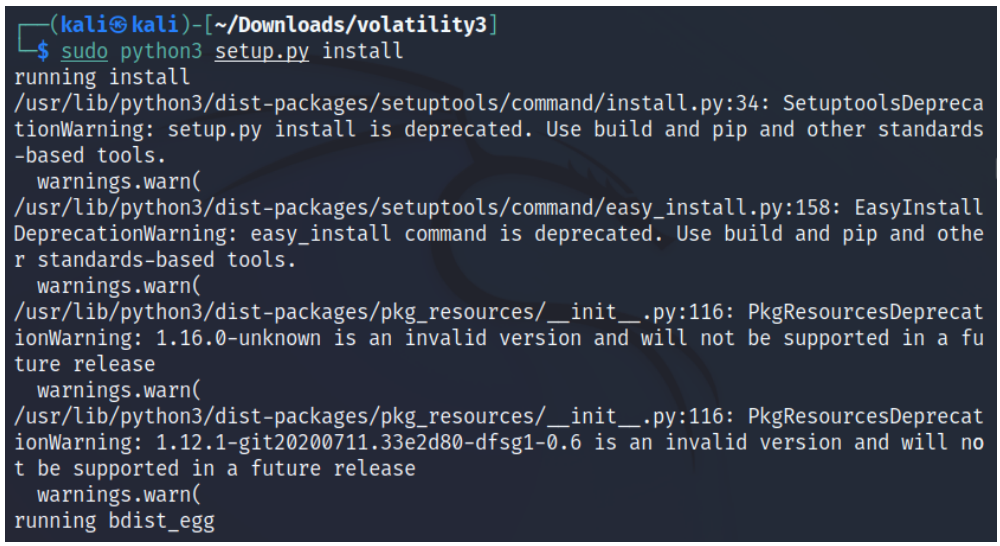


Figure 10.8 – Installing Volatility 3

9. Now that Python 3 and all the Volatility packages and prerequisites have been installed, we can verify the installation by typing the following command:

```
python3 vol.py
```

The following screenshot shows the output of the preceding command.

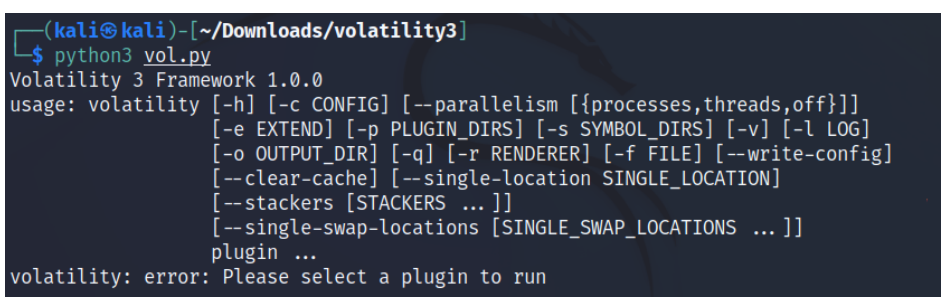


Figure 10.9 – Volatility installation verification

Volatility contains many plugins that you will need to occasionally reference. I recommend having the list of all plugins open in a separate Terminal for easy reference as this is much easier than having to scroll to the top of the Terminal to find the Volatility plugin commands.

10. Open a new Terminal within the `volatility3` directory and type the following command to view all plugins:

```
python3 vol.py -h
```

The following screenshot shows the output of the preceding command.

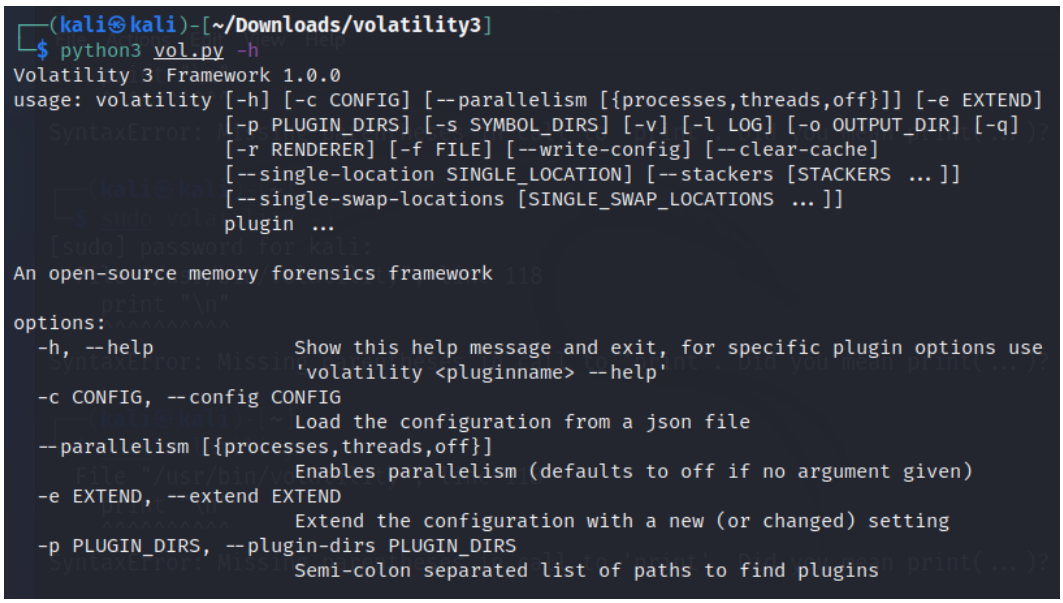


Figure 10.10 – Volatility help command

Volatility 3 now uses **OS-specific** plugins for Linux, Mac, and Windows, as seen in the snippet of the `python3 vol.py -h` output here:

```
mac.psaux.Psaux     Recovers program command line arguments.
mac.pslist.PsList   Lists the processes present in a particular mac memory image.
mac.pstree.PsTree   Plugin for listing processes in a tree based on their parent
                    process ID.
mac.socket_filters.Socket_filters
                    Enumerates kernel socket filters.
mac.timers.Timers   Check for malicious kernel timers.
mac.trustedbsd.Trustedbsd
                    Checks for malicious trustedbsd modules
mac.vfsevents.VFSevents
                    Lists processes that are filtering file system events
timeliner.Timeliner
                    Runs all relevant plugins that provide time related information
                    and orders the results by time.
windows.bigpools.BigPools
                    List big page pools.
windows.callbacks.Callbacks
                    Lists kernel callbacks and notification routines.
windows.cmdline.CmdLine
                    Lists process command line arguments.
```

Figure 10.11 – OS-specific plugins in Volatility 3

11. Before we begin the analysis of our previously downloaded `cridex.vmem` sample memory dump file, let's copy the file from its current directory in the `Downloads` folder and paste it into the `volatility3` folder. This makes access to our memory dump file easier by not having to specify a lengthy path to the file each time we need to use a plugin.
12. Keeping the previous Terminal open, let's open a new Terminal and change directories to our `volatility3` folder and issue the `ls` command to ensure that our `cridex.vmem` memory dump sample can be found within that directory, as seen in the following screenshot.

```
┌──(kali㉿kali)-[~/Downloads/volatility3]
└─$ ls
build                MANIFEST.in          volshell.py
cridex_memdump.zip   mypy.ini             volshell.spec
cridex.vmem          README.md            vol.spec
development          setup.py             'wannacry pw- infected'
dist                 volatility3          'wannacry pw- infected.7z'
doc                  volatility3.egg-info  wcry.raw
LICENSE.txt          vol.py
```

Figure 10.12 – Contents of the volatility3 directory

Here's the exciting part. Let's do some DFIR analysis using Volatility 3 and see what we can find!

Memory dump analysis using Volatility 3

For those of you who may have read previous editions of this book where we used Volatility 2, or are just familiar with using Volatility 2, you will notice that Volatility 3 is a bit different as far as the plugins are concerned. You may also notice that the speed at which the plugins work is also faster in Volatility 3.

For this lab, we'll take a very structured approach using the various plugins in Volatility 3. We'll first look at process and service identification, gather some user information, have a look at registry information, and discover any malware that may be running on the device.

Using Volatility 3 is quite simple. Once you are in the Volatility directory, the commands to use the plugins are essentially the same apart from the plugin name.

The syntax is as follows:

```
python3 -f (dump name) (OS.plugin)
```

In the preceding example, `-f` specifies the filename of the dump, which in our case is `cridex.vmem`, and `OS.plugin` will be the plugin that we would like to run against our memory dump. Let's use the `info` plugin as our first analysis task.

Image and OS verification

Although no longer required in Volatility 3, it is useful to identify the version of the OS of the device from which the memory dump was created to ensure that we use the correct plugins, as they are now specific to various OSs, which we learned earlier on in this chapter.

Let's find out what OS was running on the system by using the `info` plugin:

```
python3 vol.py -f cridex.vmem windows.info
```

The following screenshot shows the output of the preceding command.

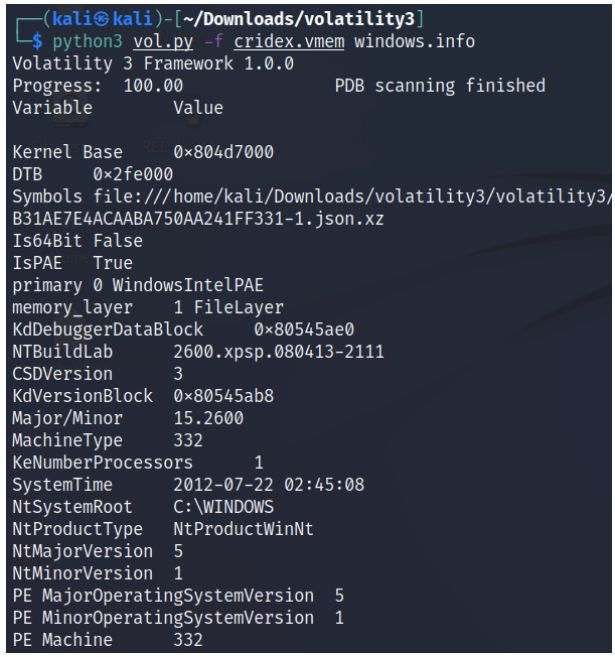


Figure 10.13 – Volatility 3 info plugin output

The output for the `info` plugin is lengthy; however, I've included a snippet of the output in the following figure, where we can see that this memory dump was taken from a Windows XP Service Pack 3 machine:

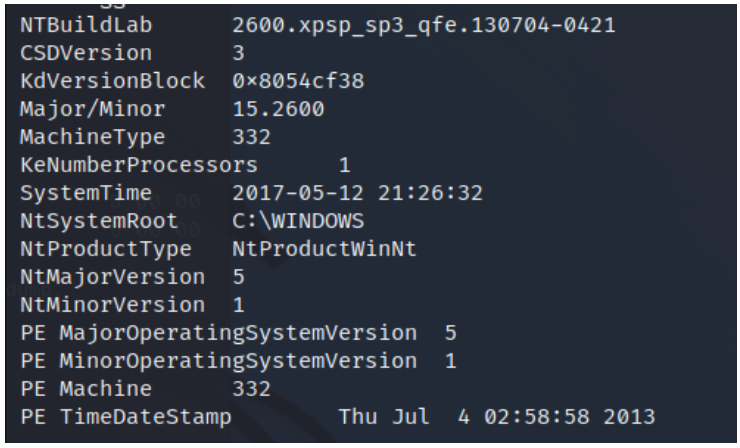


Figure 10.14 – info plugin snippet

This tells us that we must only use Windows plugins against this dump for our analysis. Let's now attempt to identify running processes and services.

Process identification and analysis

Let's officially start our DFIR memory dump analysis by attempting to identify and link connected processes, their IDs, times started, and offset locations within the memory image. We will be using several plugins but will begin with the following three plugins to get us started:

- `pslist`
- `pstree`
- `psscan`

The pslist plugin

This tool not only displays a list of all running processes but also gives useful information such as the **Process ID** (**PID**) and the **Parent PID** (**PPID**), and shows the time the processes were started.

The command to run the `pslist` plugin is as follows:

```
python3 vol.py -f cridex.vmem windows.pslist
```

In the following screenshot, we can see that the `System`, `smss`, `csrss`, `winlogon.exe`, `services.exe`, `lsass.exe`, `svchost.exe`, and `explorer.exe` services were all started first and then followed by a few others. Notice any suspicious-looking services?

```
┌──(kali㉿kali)-[~/Downloads/volatility3]
└─$ python3 vol.py -f cridex.vmem windows.pslist
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0×823c89c8      53      240     N/A     False   N/A     N/A     Disabled
368     4       smss.exe        0×822f1020      3       19      N/A     False   2012-07-22 02:42:31.000000      N/A     Disabled
584     368     csrss.exe       0×822a0598      9       326     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
608     368     winlogon.exe    0×82298700      23      519     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
652     608     services.exe    0×81e2ab28      16      243     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
664     608     lsass.exe       0×81e2a3b8      24      330     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
824     652     svchost.exe     0×82311360      20      194     0       False   2012-07-22 02:42:33.000000      N/A     Disabled
908     652     svchost.exe     0×81e29ab8      9       226     0       False   2012-07-22 02:42:33.000000      N/A     Disabled
1004    652     svchost.exe     0×823001d0      64      1118    0       False   2012-07-22 02:42:33.000000      N/A     Disabled
1056    652     svchost.exe     0×821dfda0      5       60      0       False   2012-07-22 02:42:33.000000      N/A     Disabled
1220    652     svchost.exe     0×82295650      15      197     0       False   2012-07-22 02:42:35.000000      N/A     Disabled
1484    1464    explorer.exe    0×821dea70      17      415     0       False   2012-07-22 02:42:36.000000      N/A     Disabled
1512    652     spoolsv.exe     0×81eb17b8      14      113     0       False   2012-07-22 02:42:36.000000      N/A     Disabled
1640    1484    reader_sl.exe   0×81e7bda0      5       39      0       False   2012-07-22 02:42:36.000000      N/A     Disabled
788     652     alg.exe 0×820e8da0      7       104     0       False   2012-07-22 02:43:01.000000      N/A     Disabled
1136    1004    wuauclt.exe     0×821fcda0      8       173     0       False   2012-07-22 02:43:46.000000      N/A     Disabled
1588    1004    wuauclt.exe     0×8205bda0      5       132     0       False   2012-07-22 02:44:01.000000      N/A     Disabled

┌──(kali㉿kali)-[~/Downloads/volatility3]
└─$
```

Figure 10.15 – The plslist plugin output

The PID identifies the process and the PPID identifies the parent of the process. Looking at the `pslist` output, we can see that the `winlogon.exe` process has a PID of `608` and a PPID of `368`. The PPIDs of the `services.exe` and `lsass.exe` processes (directly after the `winlogon.exe` process) are both `608`, indicating that `winlogon.exe` is in fact the PPID for both `services.exe` and `lsass.exe`.

For those new to PIDs and processes themselves, a quick Google search can assist with identification and description information. It is also useful to become familiar with many of the startup processes to be able to readily point out processes that may be unusual or suspect.

The timing and order of the processes should also be noted as these may assist in investigations. If we scroll down a bit, we can also tell that `explorer.exe` with a PID of `1484` is the PPID of `reader_sl.exe`.

Let's dig a bit deeper using the `pstree` plugin.

The pstree plugin

Another process identification command that can be used to list processes is the `pstree` plugin. This plugin shows the same list of processes as the `pslist` plugin but indentation is also used to identify child and parent processes.

Run the `pstree` plugin by typing the following:

```
python3 vol.py -f cridex.vmem windows.pstree
```

In the following screenshot, the asterisks represent the tree structure. One asterisk indicates the PID and more than one asterisk indicates it is a child process.

```
└─$ python3 vol.py -f cridex.vmem windows.pstree
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime

4       0       System  0×8205bda0      53      240     N/A     False   N/A     N/A
* 368   4       smss.exe        0×8205bda0      3       19      N/A     False   2012-07-22 02:42:31.000000      N/A
** 584  368     csrss.exe       0×8205bda0      9       326     0       False   2012-07-22 02:42:32.000000      N/A
** 608  368     winlogon.exe    0×8205bda0      23      519     0       False   2012-07-22 02:42:32.000000      N/A
*** 664 608     lsass.exe       0×8205bda0      24      330     0       False   2012-07-22 02:42:32.000000      N/A
*** 652 608     services.exe    0×8205bda0      16      243     0       False   2012-07-22 02:42:32.000000      N/A
**** 1056       652     svchost.exe     0×8205bda0      5       60      0       False   2012-07-22 02:42:33.000000      N/A
**** 1220       652     svchost.exe     0×8205bda0      15      197     0       False   2012-07-22 02:42:35.000000      N/A
**** 1512       652     spoolsv.exe     0×8205bda0      14      113     0       False   2012-07-22 02:42:36.000000      N/A
**** 908        652     svchost.exe     0×8205bda0      9       226     0       False   2012-07-22 02:42:33.000000      N/A
**** 1004       652     svchost.exe     0×8205bda0      64      1118    0       False   2012-07-22 02:42:33.000000      N/A
***** 1136      1004    wuauclt.exe     0×8205bda0      8       173     0       False   2012-07-22 02:43:46.000000      N/A
***** 1588      1004    wuauclt.exe     0×8205bda0      5       132     0       False   2012-07-22 02:44:01.000000      N/A
**** 788        652     alg.exe 0×8205bda0      7       104     0       False   2012-07-22 02:43:01.000000      N/A
**** 824        652     svchost.exe     0×8205bda0      20      194     0       False   2012-07-22 02:42:33.000000      N/A
1484    1464    explorer.exe    0×8205bda0      17      415     0       False   2012-07-22 02:42:36.000000      N/A
* 1640  1484    reader_sl.exe   0×8205bda0      5       39      0       False   2012-07-22 02:42:36.000000      N/A
```

Figure 10.16 – pstree plugin output

Let's look at the first asterisk, which represents the `smss.exe` process with a PID of `386`. We can see that under this process, there are other processes with more than one asterisk that have a PPID of `386`, indicating that they are all child processes of `smss.exe`. Similarly, if we look further down, we can see that `explorer.exe` with a PID of `1484` is the parent process of `reader_sl.exe` with a PPID of `1484`. It takes some time to adjust to this format but it helps simplify parent and child processes in DFIR analysis.

The psscan plugin

The `psscan` command displays inactive and even hidden processes that can be used by malware, such as rootkits, and are well known for doing just that to evade discovery by users and antivirus programs.

Let's run the `psscan` plugin by typing the following:

```
python3 vol.py -f cridex.vmem windows.psscan
```

The following screenshot shows the output of the preceding command when running the `psscan` plugin.

```
└─$ python3 vol.py -f cridex.vmem windows.psscan
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset  Threads Handles SessionId       Wow64   CreateTime      ExitTime

908     652     svchost.exe     0×2029ab8       9       226     0       False   2012-07-22 02:42:33.000000
664     608     lsass.exe       0×202a3b8       24      330     0       False   2012-07-22 02:42:32.000000
652     608     services.exe    0×202ab28       16      243     0       False   2012-07-22 02:42:32.000000
1640    1484    reader_sl.exe   0×207bda0       5       39      0       False   2012-07-22 02:42:36.000000
1512    652     spoolsv.exe     0×20b17b8       14      113     0       False   2012-07-22 02:42:36.000000
1588    1004    wuauclt.exe     0×225bda0       5       132     0       False   2012-07-22 02:44:01.000000
788     652     alg.exe 0×22e8da0       7       104     0       False   2012-07-22 02:43:01.000000      N/A
1484    1464    explorer.exe    0×23dea70       17      415     0       False   2012-07-22 02:42:36.000000
1056    652     svchost.exe     0×23dfda0       5       60      0       False   2012-07-22 02:42:33.000000
1136    1004    wuauclt.exe     0×23fcda0       8       173     0       False   2012-07-22 02:43:46.000000
1220    652     svchost.exe     0×2495650       15      197     0       False   2012-07-22 02:42:35.000000
608     368     winlogon.exe    0×2498700       23      519     0       False   2012-07-22 02:42:32.000000
584     368     csrss.exe       0×24a0598       9       326     0       False   2012-07-22 02:42:32.000000
368     4       smss.exe        0×24f1020       3       19      N/A     False   2012-07-22 02:42:31.000000
1004    652     svchost.exe     0×25001d0       64      1118    0       False   2012-07-22 02:42:33.000000
824     652     svchost.exe     0×2511360       20      194     0       False   2012-07-22 02:42:33.000000
4       0       System  0×25c89c8       53      240     N/A     False   N/A     N/A     Disabled
```

Figure 10.17 – psscan output

So far, there's nothing that stands out. Let's keep digging.

The modscan plugin

The `modscan` plugin displays a list of all modules present in the memory image. This helps us identify the path and directory for processes, system files, and **Dynamic Link Library** (**DLL**) files.

Run the `modscan` plugin by typing the following:

```
python3 vol.py -f cridex.vmem windows.modscan
```

As seen in the following screenshot, the `modscan` plugin lists all modules running that were not available in the previous process scans.

```
└─$ python3 vol.py -f cridex.vmem windows.modscan
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
Offset  Base    Size    Name    Path    File output

0×59ca40        0×89607b8d      0×89662c46                      Disabled
0×5a3890        0×6600000c      0×8d50a045                      Disabled
0×5a3e06        0×400   0×66000010                      Disabled
0×20296b8       0×f7c6f000      0×4000  ndisuio.sys     \SystemRoot\system32\DRIVERS\ndisuio.sys
d
0×202fe80       0×f8b46000      0×3000  ndistapi.sys    \SystemRoot\system32\DRIVERS\ndistapi.sys
d
0×20350c8       0×f89b2000      0×7000  HIDPARSE.SYS    \SystemRoot\system32\DRIVERS\HIDPARSE.SYS
d
0×2078108       0×f8982000      0×5000  flpydisk.sys    \SystemRoot\system32\DRIVERS\flpydisk.sys
d
0×2085008       0×bff50000      0×3000  framebuf.dll    \SystemRoot\System32\framebuf.dll       Di
0×20858d8       0×f877a000      0×f000  redbook.sys     \SystemRoot\system32\DRIVERS\redbook.sys
```

Figure 10.18 – modscan plugin output

Let's keep looking.

The getsids plugin

All users can also be uniquely identified by a **Security Identifier** (**SID**). The `getsids` command has four very useful items in the order in which the processes were started (refer to the `pslist` and `pstree` command screenshots in *Figures 10.15* and *10.16*).

The format for the `getsids` command output is as follows:

```
[Process] (PID) [SID] (User)
```

The first result in the list, for example, lists the following:

```
System (4) : S - 1 - 5- 18 (User)
```

Where:

- `System` is the process
- `(4)` is the PID
- `S - 1 - 5- 18` is the SID
- `User` is the local system

To run the `getsids` plugin, type the following:

```
python3 vol.py -f cridex.vmem windows.getsids
```

The following screenshot shows the output of the preceding command when running the `getsids` plugin.

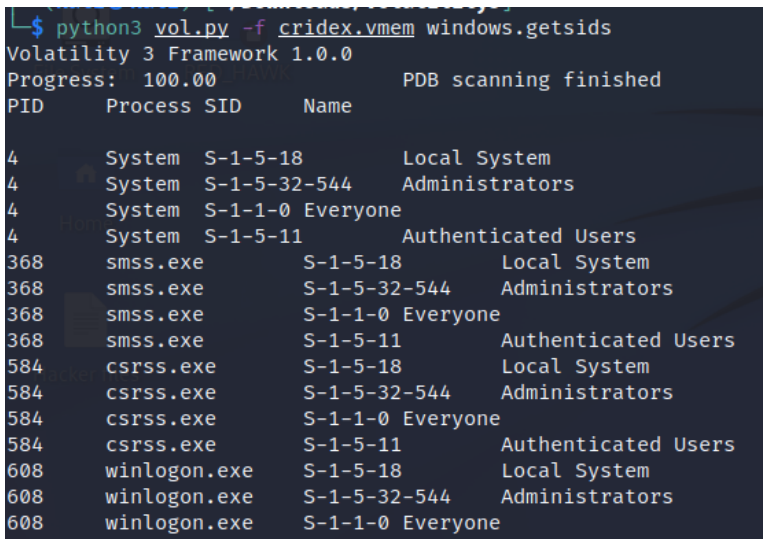


Figure 10.19 – getsids plugin output

If we scroll further down, we can see that the `reader_sl.exe` process was started by a user named Robert with the SID of `S-1-5-21`.

```
1640    reader_sl.exe   S-1-5-21-789336058-261478967-1417001333-1003    Robert
1640    reader_sl.exe   S-1-5-21-789336058-261478967-1417001333-513     Domain Users
1640    reader_sl.exe   S-1-1-0 Everyone
1640    reader_sl.exe   S-1-5-32-544    Administrators
1640    reader_sl.exe   S-1-5-32-545    Users
1640    reader_sl.exe   S-1-5-4 Interactive
1640    reader_sl.exe   S-1-5-11        Authenticated Users
1640    reader_sl.exe   S-1-5-5-0-53426 Logon Session
1640    reader_sl.exe   S-1-2-0 Local (Users with the ability to log in locally)
```

Figure 10.20 – getsids output snippet

The envars plugin

Let's continue our analysis using the `envars` plugin, which displays process environment variables and nicely maps all processes to paths and users.

Run the `envars` plugin by typing the following:

```
python3 vol.py -f cridex.vmem windows.envars
```

```
└─$ python3 vol.py -f cridex.vmem windows.envars
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     Process Block   Variable        Value

368     smss.exe        0x110048        Path    C:\WINDOWS\System32
368     smss.exe        0x110048        SystemDrive     C:
368     smss.exe        0x110048        SystemRoot      C:\WINDOWS
368     smss.exe        0x110048        ve      C:
368     smss.exe        0x110048        SystemRoot      C:\WINDOWS
368     smss.exe        0x110048        oot     C:\WINDOWS
584     csrss.exe       0x110048        ComSpec C:\WINDOWS\system32\cmd.exe
584     csrss.exe       0x110048        FP_NO_HOST_CHECK        NO
584     csrss.exe       0x110048        NUMBER_OF_PROCESSORS    1
584     csrss.exe       0x110048        OS      Windows_NT
584     csrss.exe       0x110048        Path    C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
584     csrss.exe       0x110048        PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
584     csrss.exe       0x110048        PROCESSOR_ARCHITECTURE  x86
584     csrss.exe       0x110048        PROCESSOR_IDENTIFIER    x86 Family 6 Model 23 Stepping 6, GenuineIntel
584     csrss.exe       0x110048        PROCESSOR_LEVEL 6
584     csrss.exe       0x110048        PROCESSOR_REVISION      1706
584     csrss.exe       0x110048        SystemDrive     C:
584     csrss.exe       0x110048        SystemRoot      C:\WINDOWS
584     csrss.exe       0x110048        TEMP    C:\WINDOWS\TEMP
584     csrss.exe       0x110048        TMP     C:\WINDOWS\TEMP
584     csrss.exe       0x110048        windir  C:\WINDOWS
```

Figure 10.21 – envars plugin output

Scrolling through the lengthy output down to the `reader_sl.exe` process, we can find some very useful information about the process, path and directories, computer name architecture, drive information, and temporary file location. Good stuff.

```
1640    reader_sl.exe   0x20048 ALLUSERSPROFILE C:\Documents and Settings\All Users
1640    reader_sl.exe   0x20048 APPDATA C:\Documents and Settings\Robert\Application Data
1640    reader_sl.exe   0x20048 CLIENTNAME      Console
1640    reader_sl.exe   0x20048 CommonProgramFiles      C:\Program Files\Common Files
1640    reader_sl.exe   0x20048 COMPUTERNAME    ACCOUNTING12
1640    reader_sl.exe   0x20048 ComSpec C:\WINDOWS\system32\cmd.exe
1640    reader_sl.exe   0x20048 FP_NO_HOST_CHECK        NO
1640    reader_sl.exe   0x20048 HOMEDRIVE       C:
1640    reader_sl.exe   0x20048 HOMEPATH        \Documents and Settings\Robert
1640    reader_sl.exe   0x20048 LOGONSERVER     \\ACCOUNTING12
1640    reader_sl.exe   0x20048 NUMBER_OF_PROCESSORS    1
1640    reader_sl.exe   0x20048 OS      Windows_NT
1640    reader_sl.exe   0x20048 Path    C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
1640    reader_sl.exe   0x20048 PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
1640    reader_sl.exe   0x20048 PROCESSOR_ARCHITECTURE  x86
1640    reader_sl.exe   0x20048 PROCESSOR_IDENTIFIER    x86 Family 6 Model 23 Stepping 6, GenuineIntel
1640    reader_sl.exe   0x20048 PROCESSOR_LEVEL 6
1640    reader_sl.exe   0x20048 PROCESSOR_REVISION      1706
1640    reader_sl.exe   0x20048 ProgramFiles    C:\Program Files
1640    reader_sl.exe   0x20048 SESSIONNAME     Console
1640    reader_sl.exe   0x20048 SystemDrive     C:
1640    reader_sl.exe   0x20048 SystemRoot      C:\WINDOWS
1640    reader_sl.exe   0x20048 TEMP    C:\DOCUME~1\Robert\LOCALS~1\Temp
1640    reader_sl.exe   0x20048 TMP     C:\DOCUME~1\Robert\LOCALS~1\Temp
1640    reader_sl.exe   0x20048 USERDOMAIN      ACCOUNTING12
1640    reader_sl.exe   0x20048 USERNAME        Robert
1640    reader_sl.exe   0x20048 USERPROFILE     C:\Documents and Settings\Robert
1640    reader_sl.exe   0x20048 windir  C:\WINDOWS
```

Figure 10.22 – envars output snippet

Let's do a bit of registry analysis and see what we can come up with.

The hivelist plugin

The `hivelist` plugin lists the registry hives present at the time the memory dump was taken and will also show logged-in users. The `hivelist` command shows the details of virtual and physical addresses along with the easier readable plaintext names and locations.

To run this plugin, type the following:

```
vol.py -f cridex.vmem windows.registry.hivelist
```

The following screenshot shows the output of the preceding command when running the `hivelist` plugin.

```
└─$ python3 vol.py -f cridex.vmem windows.registry.hivelist
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
Offset  FileFullPath    File output

0×e18e5b60      \Device\HarddiskVolume1\Documents and Settings\Robert\Local Settings\Application Data\Microsoft
\Windows\UsrClass.dat   Disabled
0×e1a19b60      \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT        Disabled
0×e18398d0      \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Mic
rosoft\Windows\UsrClass.dat     Disabled
0×e18614d0      \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT  Disabled
0×e183bb60      \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\M
icrosoft\Windows\UsrClass.dat   Disabled
0×e17f2b60      \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT        Disabled
0×e1570510      \Device\HarddiskVolume1\WINDOWS\system32\config\software        Disabled
0×e1571008      \Device\HarddiskVolume1\WINDOWS\system32\config\default Disabled
0×e15709b8      \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY        Disabled
0×e15719e8      \Device\HarddiskVolume1\WINDOWS\system32\config\SAM     Disabled
0×e13ba008              Disabled
0×e1035b60      \Device\HarddiskVolume1\WINDOWS\system32\config\system  Disabled
0×e102e008              Disabled
```

Figure 10.23 – hivelist plugin output

Password dumping

The location of the **Security Accounts Manager** (**SAM**) file is also listed using the `hivelist` plugin, shown in the following screenshot (*Figure 10.24*). The `SAM` file contains hashed passwords for usernames in Windows machines. The path to the `SAM` file is seen in the following screenshot: `Windows\system32\config\SAM`. This file cannot be accessed by users within Windows while the system is on. This can be further used to acquire the hashed passwords in the `SAM` file to crack passwords using a wordlist along with password-cracking tools such as John the Ripper, also available in Kali Linux:

```
0×e15719e8      \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
```

Figure 10.24 – SAM file location

Let's expand on this a bit further using the `userassist` plugin.

The userassist plugin

The `userassist` plugin displays more registry information and, as we can see in the following screenshot, displays much more details about users, file locations, files accessed, and timestamps.

To run the `userassist` plugin, type the following:

```
python3 vol.py -f cridex.vmem windows.registry.userassist
```

The following screenshot shows the output of the preceding command when running the `userassist` plugin.

```
└─$ python3 vol.py -f cridex.vmem windows.registry.userassist
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
Hive Offset     Hive Name       Path    Last Write Time Type    Name    ID      Count   Focus Count     Time Fo
cused   Last Updated    Raw Data

0×e1a19b60      \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT        NTUSER.DAT\Software\Mic
rosoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count  2012-07-22 02:2
6:25.000000     Key     N/A     N/A     N/A     N/A     N/A     N/A     N/A
* 0×e1a19b60    \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT        NTUSER.DAT\Software\Mic
rosoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count  2012-07-22 02:2
6:25.000000     Value   UEME_CTLSESSION -       -       -       -       -
8b 3d 6b 0e 03 00 00 00 .=k.....
* 0×e1a19b60    \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT        NTUSER.DAT\Software\Mic
rosoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count  2012-07-22 02:2
6:25.000000     Value   UEME_CTLCUACount:ctor   1       2       N/A     N/A     N/A
01 00 00 00 02 00 00 00 ........
00 00 00 00 00 00 00 00 ........
* 0×e1a19b60    \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT        NTUSER.DAT\Software\Mic
rosoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count  2012-07-22 02:2
6:25.000000     Value   UEME_UITOOLBAR  1       1       N/A     N/A     2011-04-13 00:56:58.000000
01 00 00 00 06 00 00 00 ........
30 35 b2 b0 75 f9 cb 01 05..u...
* 0×e1a19b60    \Device\HarddiskVolume1\Documents and Settings\Robert\NTUSER.DAT        NTUSER.DAT\Software\Mic
rosoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count  2012-07-22 02:2
6:25.000000     Value   UEME_UITOOLBAR:0×4,7031 1       1       N/A     N/A     2011-04-13 00:56:58.000000
```

Figure 10.25 – userassist plugin output

Lastly, for this chapter, let's see whether we can find any malicious code or applications.

The malfind plugin

The `malfind` plugin, as the name suggests, scans the dump for any malicious embedded code. This will be covered more in detail in *Chapter 11*, *Artifact, Malware, and Ransomware Analysis*, but let's see whether we can find any embedded code within our current DFIR investigation.

To run the `malfind` plugin, type the following:

```
python3 vol.py -f cridex.vmem windows.malfind
```

The following screenshot shows the output of the preceding command when running the `malfind` plugin.

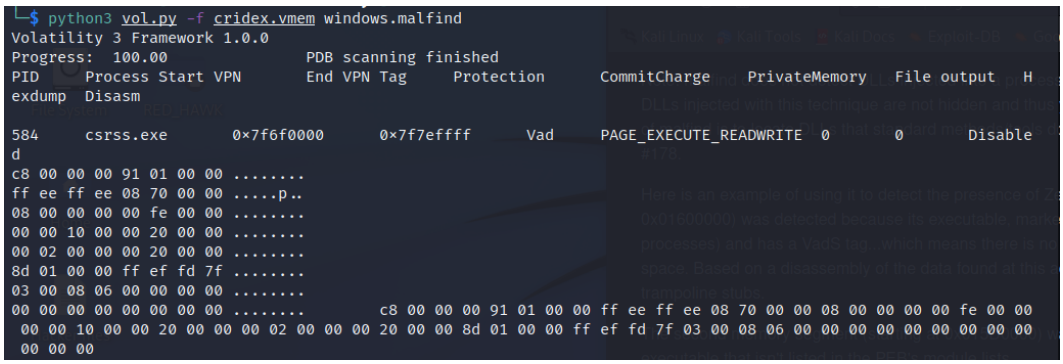


Figure 10.26 – malfind plugin output

This plugin is a bit more complex to understand but very useful once you understand the basics. The column names that we need to focus on are the following:

- `PID`
- `Process`
- `CommitCharge`
- `PrivateMemory`

Let's look closer at the `reader_sl.exe` details in the `malfind` output, as follows.

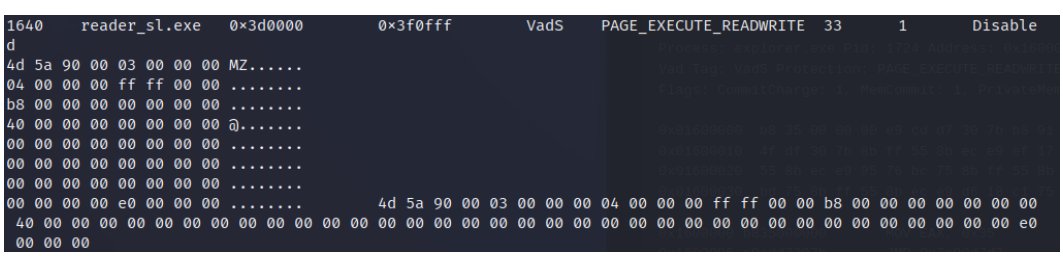


Figure 10.27 – malfind snippet

Let's analyze the output of the first line of the preceding `malfind` snippet:

- `PID: 1640`
- `Process: reader_sl.exe`
- `CommitCharge: PAGE_EXECUTE_READWRITE`
- `PrivateMemory: 33`
- `File output: 1`

In the preceding output, we see that `reader_sl.exe` is executable code and is writing hidden code in memory. We will dive deeper into malware analysis in the next chapter.

Summary

In this chapter, we looked at memory forensics and analysis using some of the many plugins available within the Volatility 3 framework. We were able to successfully perform process, registry, DLL, and even malware analysis using this versatile tool. As we've seen, Volatility can perform several important functions in DFIR analysis and should be used together with other tools we've used previously to perform in-depth and detailed forensic analyses and investigations.

Be sure to download more publicly available memory images and samples to test your skills in this area. Experiment with as many plugins as you can and, of course, be sure to document your findings and consider sharing them online.

Next up, we'll be going even deeper into Volatility as we perform ransomware analysis and use many other tools to discover and analyze various DFIR artifacts. See you in the next chapter!

11
Artifact, Malware, and Ransomware Analysis

In this chapter, we'll cover several different tools to uncover various digital artifacts, malware, and ransomware, some of which reside in RAM and the swap file, which, as we learned in the previous chapter, can be quite useful in our DFIR investigations.

To start things off, we will look into artifact analysis using tools such as p0f to identify devices and operating systems, use swap_digger for swap file analysis, and then use MimiPenguin for password dumping. Following this, we will dive into malware analysis using pdf-parser and PDFiD for PDF malware analysis, use Hybrid Analysis for malicious file analysis, and then end things off by using Volatility 3 for ransomware analysis.

The following topics will be covered in this chapter:

- Identifying devices and operating systems with p0f
- Looking at the swap_digger tool to explore Linux artifacts
- Password dumping with MimiPenguin
- PDF malware analysis
- Using Hybrid Analysis for malicious file analysis
- Ransomware analysis using Volatility 3

Identifying devices and operating systems with p0f

Let's get started with **p0f**. p0f is a small tool that can be used to passively scan and detect operating systems within a network. This scanning tool is considered passive because it does not send data to other hosts apart from **Synchronize** (**SYN**) packets. This is very useful when trying to quietly collect information about other hosts on a network in DFIR investigations.

Let's look at how to install and use p0f to detect other host operating systems on the network:

1. Depending on the version of Kali you are running (2019.3 – 2023.1), you can run the `p0f -h` command to determine whether it is preinstalled. If not, Kali will ask whether you would like to install it. Press *y* to accept and install it, as seen in the following screenshot.

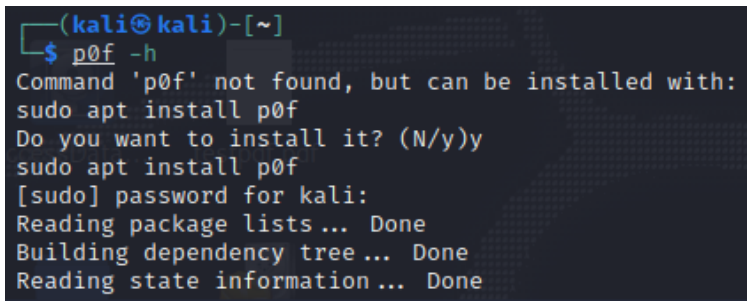


Figure 11.1 – Installing p0f in Kali

2. Run the `p0f -h` command again after installation. This displays the network interface options, operating mode, output settings, and performance-related options.

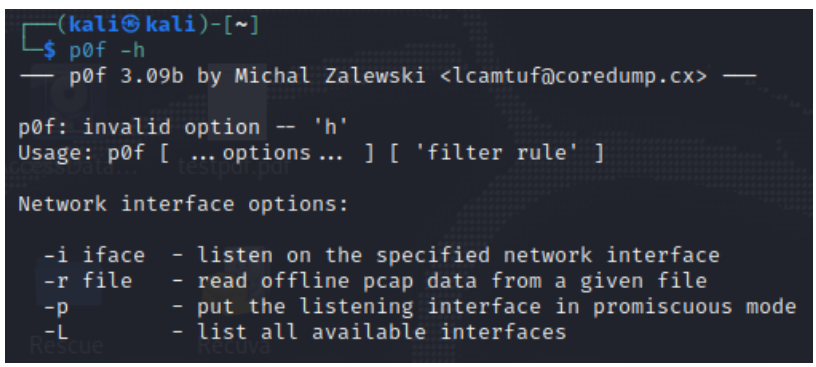


Figure 11.2 – p0f usage options

3. Once the installation is verified, you may specify an interface if you know which one you would like to use. Check your network interfaces by typing the following:

```
ifconfig
```

The following screenshot shows the output of the preceding `ifconfig` command.

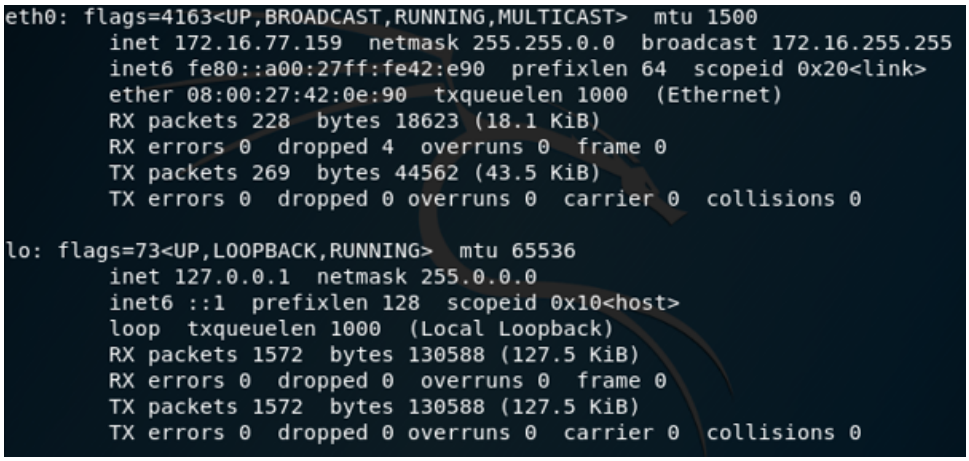


Figure 11.3 – ipconfig command output

> **Note**
> You may also use the `p0f -L` command to list all the interfaces.

4. The output in *Figure 11.3* shows that I have two interfaces with `eth0` being my Ethernet/LAN interface with an IP of `172.16.77.159` and a default `127.0.0.1` loopback address. I'll be using the `eth0` interface with `p0f` by typing the following:

```
sudo p0f -i eth0
```

The following screenshot shows the output of the preceding command.

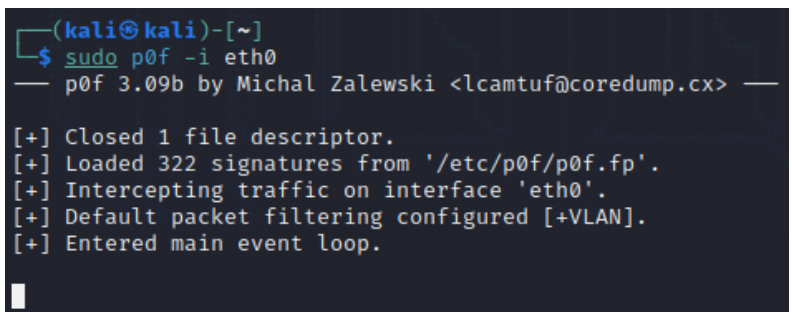


Figure 11.4 – p0f eth0 output

This may take a short while to run, but in the following screenshot, we can see that the output returns client details such as the IP address and operating system.

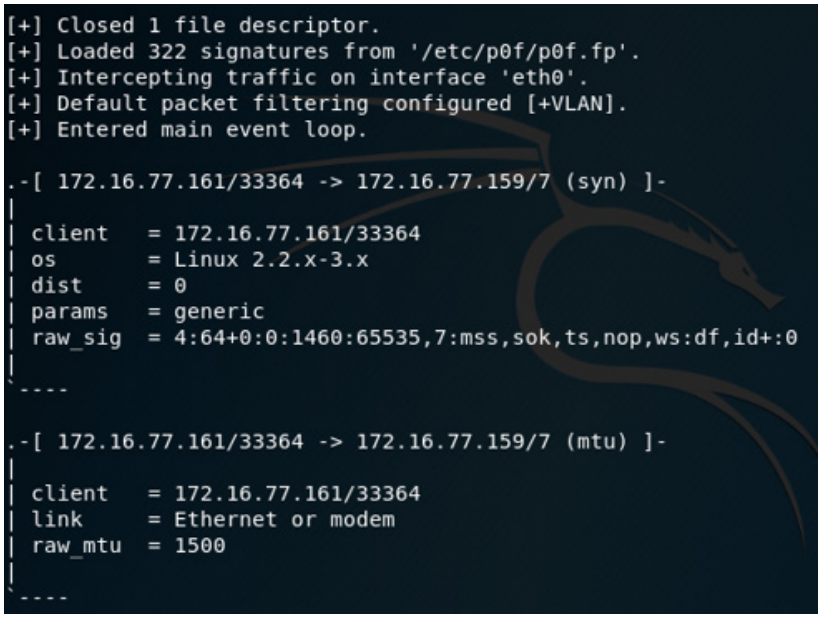


Figure 11.5 – p0f result output

Let's go a bit further by opening a browser to detect what other hosts we may be communicating with.

5. Open the web browser in Kali and you'll see the Terminal being populated with more IP information. By default, the Firefox web browser's home page carries us to the Offensive Security site and so p0f shows information about the connections and network hops to the server and information about the server.
6. Try browsing a site. I've opened `www.cfsi.co` in the browser. `p0f` updates the information in the terminal in real time and the first entry displayed shows a SYN request from `172.16.77.159` (my Kali machine) to `185.230.60.211` via port `80`. I can also see information about my Kali machine, such as the operating system (`Linux 2.2-3.x`), as fingerprinted by p0f:

```
.-[ 172.16.77.159/53382 -> 185.230.60.211/80 (syn) ]-
|
| client   = 172.16.77.159/53382
| os       = Linux 2.2.x-3.x
| dist     = 0
| params   = generic
| raw_sig  = 4:64+0:0:1460:mss*44,7:mss,sok,ts,nop,ws:df,id+:0
|
`----

.-[ 172.16.77.159/53382 -> 185.230.60.211/80 (mtu) ]-
|
| client   = 172.16.77.159/53382
| link     = Ethernet or modem
| raw_mtu  = 1500
|
```

Figure 11.6 – Updated p0f output after browsing

7. Let's get more information about the IP address `185.230.60.211`. In the terminal window, click on **File** | **New Tab**. In the new tab, type the following:

```
whois 185.230.60.211
```

In the following `whois` output, we can see that the IP points to `wix.com`, which is the host for the `www.cfsi.co` website:

```
inetnum:        185.230.60.0 - 185.230.60.255
netname:        wix_com_inc
country:        US
admin-c:        SP17239-RIPE
tech-c:         SP17239-RIPE
status:         LIR-PARTITIONED PA
mnt-by:         il-wixcom-sys-mnt
mnt-by:         il-wixcom-1-mnt
created:        2018-05-21T15:00:58Z
last-modified:  2019-10-10T07:20:09Z
source:         RIPE

person:         Stanislav Panich
address:        Namal Tel Aviv 40
phone:          +972 3 5454900
nic-hdl:        SP17239-RIPE
mnt-by:         il-wixcom-sys-mnt
created:        2018-05-09T15:30:17Z
last-modified:  2018-05-09T15:30:17Z
source:         RIPE
```

Figure 11.7 – whois output

8. Scroll through the `p0f` output to see several other pieces of information, including the uptime of the server and other IP addresses and hops along the way:

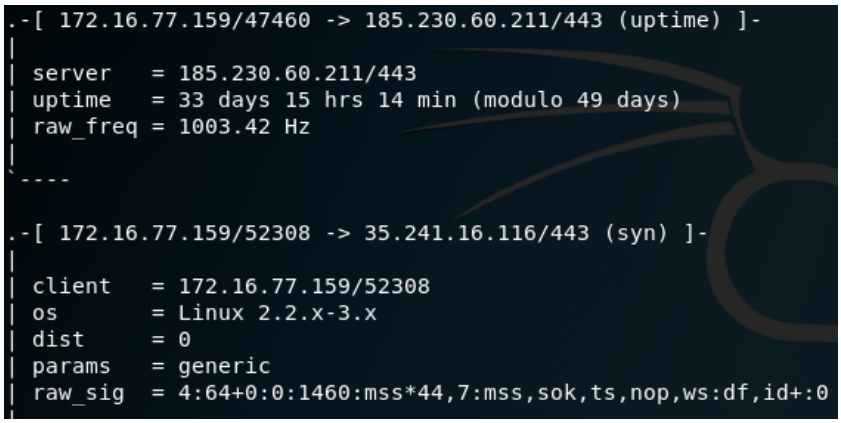


Figure 11.8 – Additional p0f output

Now that we've learned how to install and use p0f to detect other operating systems that our device is communicating with, let's move on to another tool called `swap_digger` to explore Linux artifacts.

Looking at the swap_digger tool to explore Linux artifacts

The `swap_digger` tool performs an automated analysis of the Linux swap file and can retrieve artifacts such as system passwords, usernames, and form credentials, and even Wi-Fi information such as SSIDs and perhaps even passwords if stored in the swap file.

Installing and using swap_digger

Follow these steps to install and use `swap_digger` for swap analysis:

1. Change directories to the desktop in the terminal and clone `swap_digger` to the desktop by typing the following:

```
git clone https://github.com/sevagas/swap_digger.git
```

The following screenshot shows the output of the preceding command for installing swap_digger.

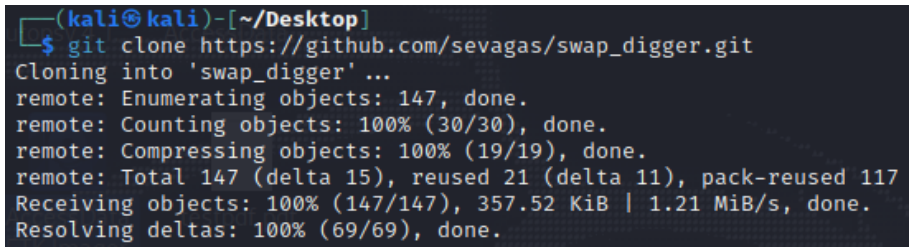


Figure 11.9 – Installing swap_digger

2. Change to the `swap_digger` directory by typing `cd swap_digger` and type the following command to ensure swap_digger will have the required access permissions:

```
chmod +x swap_digger.sh
```

3. To view all `swap_digger` usage options, type the following:

```
sudo ./swap_digger.sh -h
```

The following screenshot shows the output of the preceding command.

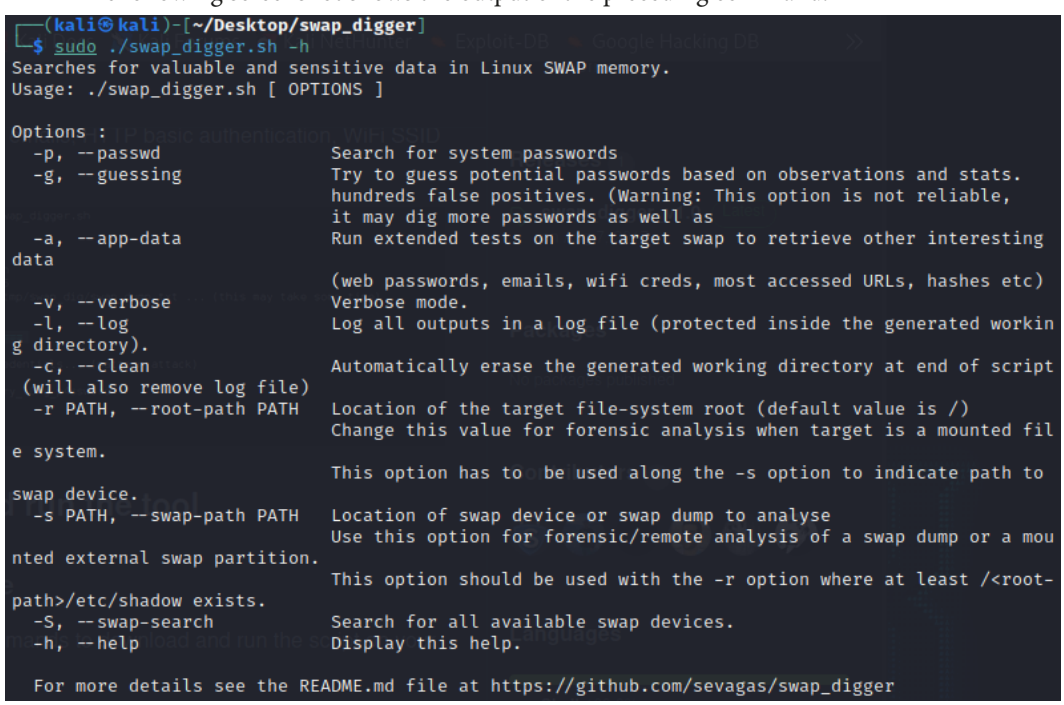


Figure 11.10 – swap_digger usage options

4. To try searching for passwords in the swap file, enter the following command:

```
sudo ./swap_digger.sh -p
```

The following screenshot shows the output of the preceding command.

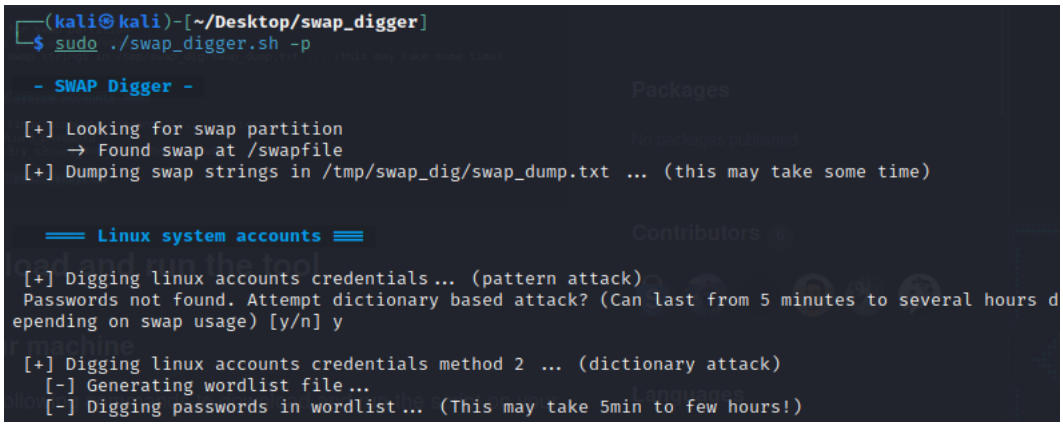


Figure 11.11 – Using swap_digger to find passwords

5. Feel free to try the other available options in `swap_digger` to discover other artifacts within the swap file of your Linux system.

Next, let's look at password dumping using the `MimiPenguin` tool.

Password dumping with MimiPenguin

The `MimiPenguin` tool is based on the very popular password-cracking tool called Mimikatz. Much like `swap_digger`, `MimiPenguin` can also retrieve artifacts running in memory by dumping memory processes that may contain unencrypted passwords in plaintext, as shown in the following steps:

1. Let's start by changing to the `Desktop` folder from our current location, and then clone `MimiPenguin` to the desktop by typing the following into a new terminal:

```
git clone https://github.com/huntergregal/mimipenguin
```

The following screenshot shows the output of the preceding command when installing MimiPenguin.

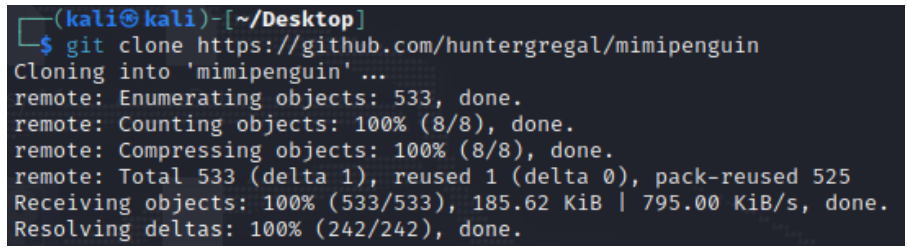


Figure 11.12 – Installing MimiPenguin

2. Change to the `mimipenguin` directory by typing `cd mimipenguin` and then show the files within by typing `ls`.

   The following screenshot shows the output of the preceding `ls` command.

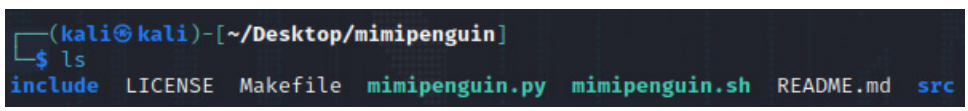


Figure 11.13 – Viewing contents of the mimipenguin folder

3. Run MimiPenguin by typing the following:

```
sudo ./mimipenguin.sh
```

   It may take a while for the password to be found. I've changed my password to something very simple for the purpose of saving time:

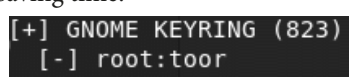


Figure 11.14 – MimiPenguin output displaying the password

Now that we've learned how to dump a password using MimiPenguin, let's go even deeper and manually analyze PDF documents for embedded malware.

PDF malware analysis

In this section, we'll have a look at PDF malware forensics and analysis. PDFs are possibly the most common form of document when sharing information as many people would rather open a PDF than an Office document, such as one in `.docx` or `.xls` format, as they are more likely to contain macros and even viruses. While PDFs are more trusted document types, it is still common to come across some that have been infected with malware or contain hidden information.

Although we won't be analyzing malicious PDFs as it may result in your system becoming infected or experiencing some adverse effects, I will still introduce you to a tool called `pdf-parser`, which can be used to inspect elements of a PDF document and pinpoint malicious code and other suspect elements.

This may be considered an advanced tool as people with programming experience typically use it to identify shellcode, streams, and filters. However, even beginners will be able to analyze the output and identify embedded executable (`.exe`) files. There are several PDF malware samples roaming the web and on Twitter. However, I urge you to not attempt to download these unless you are a professional and are doing so in an isolated environment on a sandboxed machine that if infected will not cause harm to your data or network.

Let's get started with learning how to analyze PDF documents using pdf-parser:

1. Let's first view the usage and available options of pdf-parser by typing the following:

```
pdf-parser -h
```

The following screenshot shows the output of the preceding command.

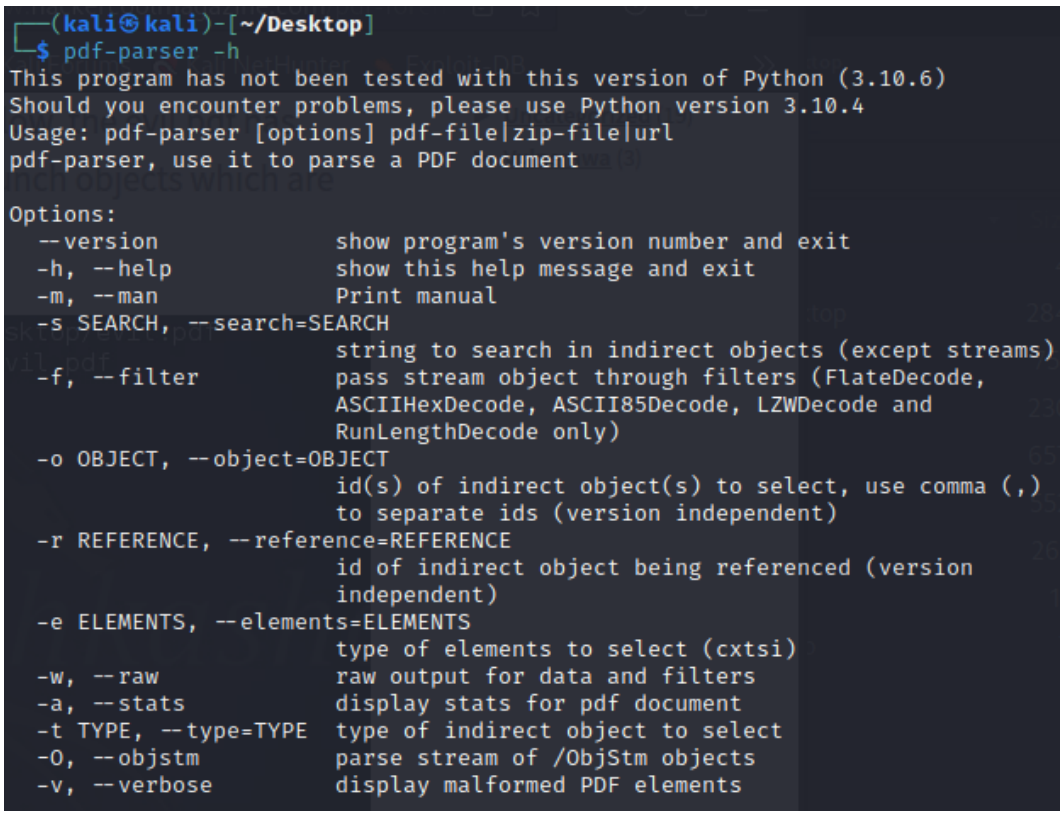


Figure 11.15 – pdf-parser usage options

2. Now, let's look at the associated statistics of a test file created by Didier Stevens (`https://blog.didierstevens.com/about/`), which I've renamed `testpdf.pdf` and saved to my desktop. We can analyze this file to determine whether it contains any embedded code or hidden malware by typing the following command:

```
pdf-parser -a testpdf.pdf
```

In the following screenshot, we can see that there is in fact an embedded file, which could also be a JavaScript file.

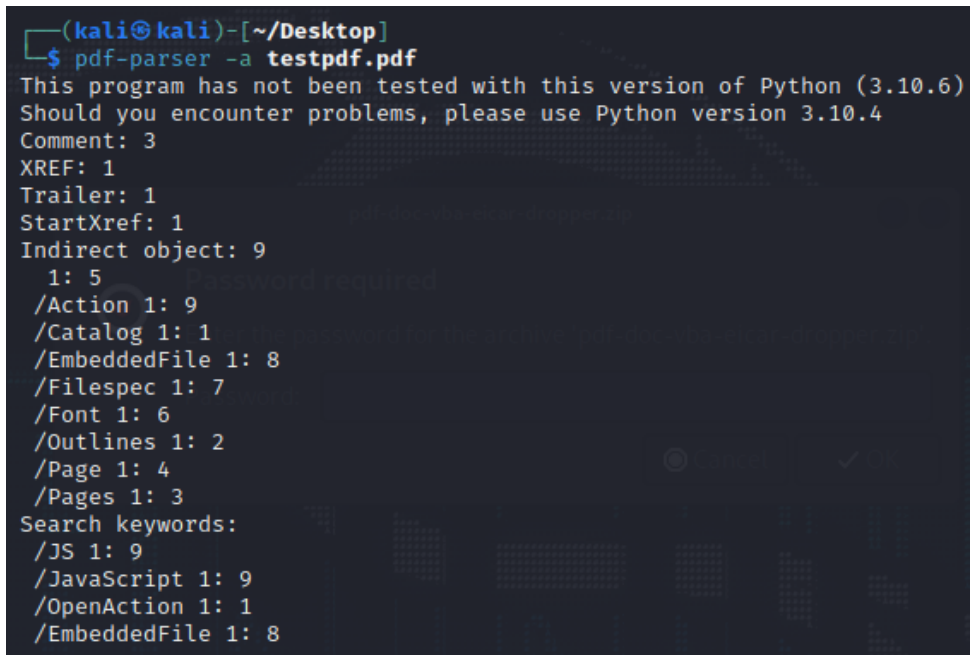


Figure 11.16 – pdf-parser results

3. Let's apply filters using the `-f` option to dig a bit deeper and see whether `pdf-parser` can identify the embedded file using the following options:

```
pdf-parser -f testpdf.pdf
```

In the following screenshot, we have confirmation that it is an embedded file. This file is a Word document called `eicar-dropper.doc`, which was embedded within the PDF file.

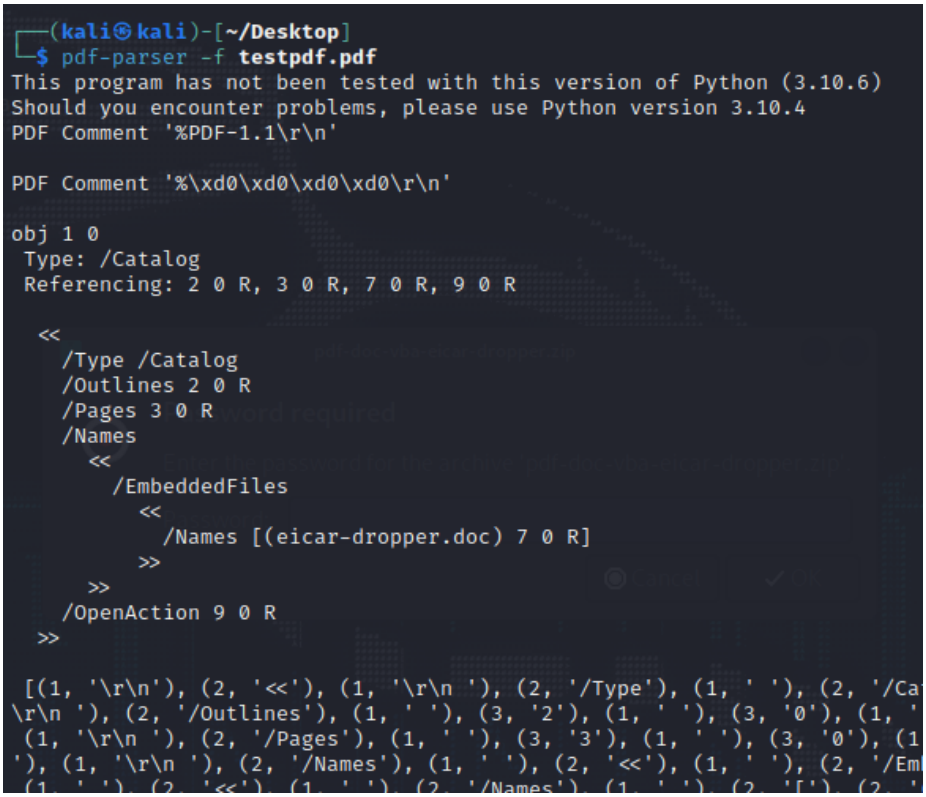


Figure 11.17 – Embedded file discovered by pdf-parser

4. We can also confirm the presence of a JavaScript file by using the PDFiD tool, running the following command:

```
pdfid testpdf.pdf
```

The following screenshot shows the output of the preceding command.

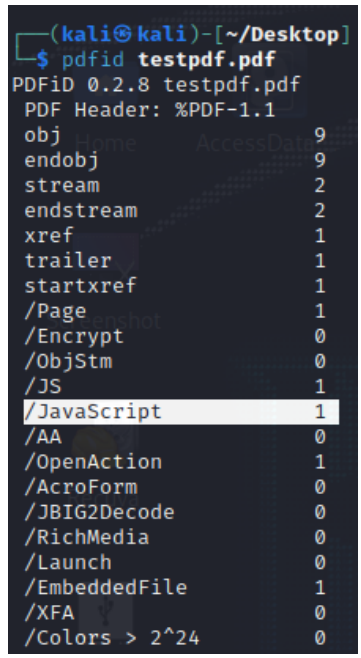


Figure 11.18 – Discovering embedded files using PDFiD

Now that we have learned how to manually inspect and analyze a PDF file that may contain malware, let's look at an online tool for automated malware analysis.

Using Hybrid Analysis for malicious file analysis

You can also use an online tool such as **Hybrid Analysis** (`https://www.hybrid-analysis.com/`) to analyze suspicious files of all types. If you suspect that a link or URL may be suspicious within an email, you can also paste the link into this site for analysis.

As an example, I'll use the very same `testpdf.pdf` file that I analyzed using `pdf-parse` and `PDFiD` in the previous section. I'll first visit the `https://www.hybrid-analysis.com` website and drag the suspect file into the upload area and click on **Analyze**, as seen in the following screenshot.

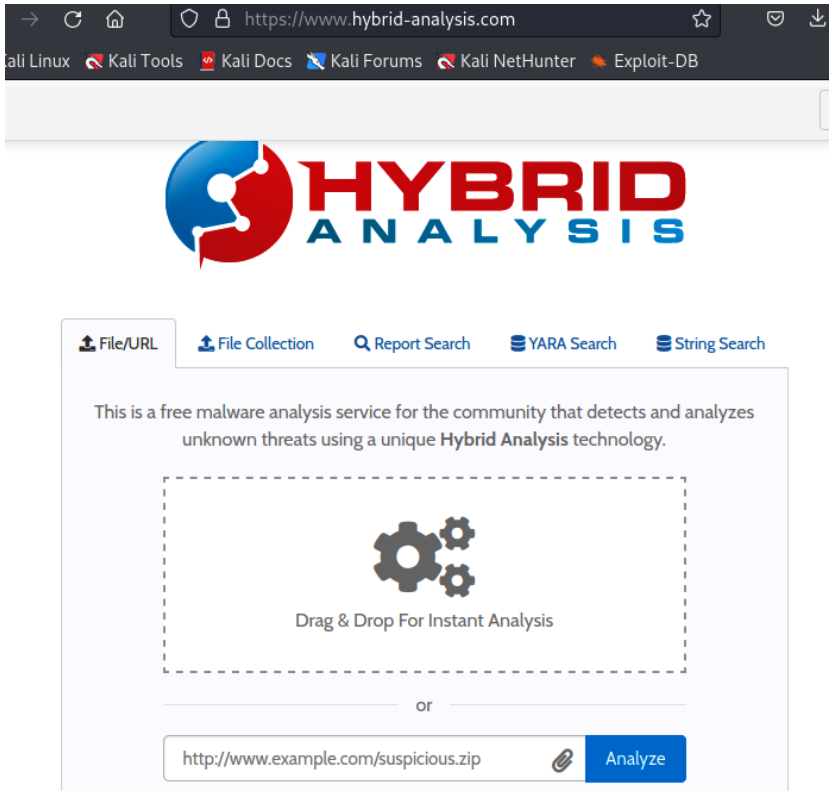


Figure 11.19 – hybrid-analysis.com website

After submitting the PDF file, the results show the file to be possibly malicious, as seen in the following screenshot:

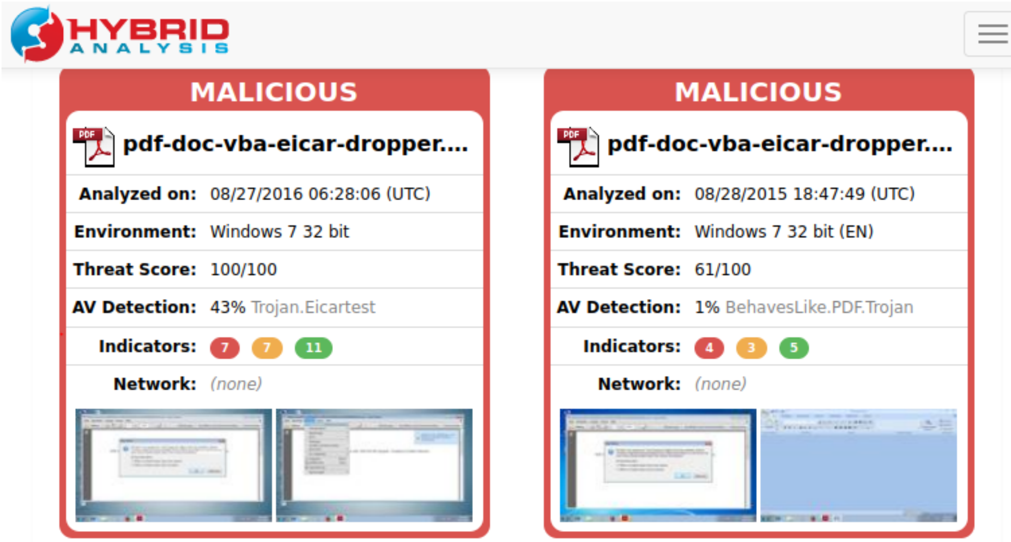


Figure 11.20 – hybrid-analysis.com file analysis and results

The details on the file's unusual characteristics and suspicious indicators are also provided, as displayed in the following screenshot:

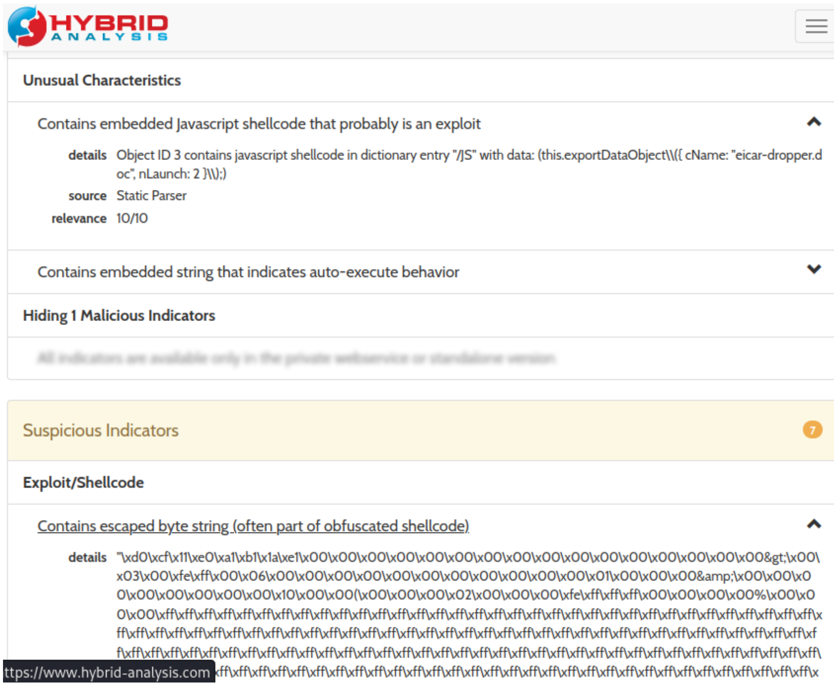


Figure 11.21 – hybrid-analysis.com results page displaying a malicious indicator

I hope you had fun learning about PDF and document malware forensics and analysis. For our last tool, we will revisit Volatility 3 to perform ransomware analysis.

Ransomware analysis using Volatility 3

For our last section, let's revisit the very powerful RAM analysis tool called Volatility 3, which we covered in *Chapter 10*, *Memory Forensics and Analysis with Volatility 3*. Feel free to take a moment to review that chapter before moving ahead.

In this lab, we'll be using a memory dump called `wcry.raw`, which contains information about a WannaCry ransomware infection on a Windows system. We will be analyzing it using a variety of Volatility 3 plugins.

Let's first download and extract our sample memory dump, which we will later move to our Volatility installation folder for analysis:

1. The WannaCry memory dump file can be downloaded from `https://mega.nz/file/7Z1ySZBT#KX5ZJKYzQgDHSa72lPFwqKL6CsZS7oQGbyyQrMTH9XY`.

   I've downloaded the WannaCry memory dump file to my `Downloads` folder, which is named `wannacry pw- infected.7z`.

2. To extract the file, right-click on the `.7z` file and click on **Extract Here** as you have done with previously downloaded files.
3. The file is password protected and you will have to type the word `infected` when prompted for the password. Once extracted, you should now have a folder within your `Downloads` folder called `wannacry pw- infected`. Double-click on the folder and you should see the `wcry.raw` memory dump file.
4. Before we begin the analysis of our downloaded `wcry.raw` sample memory dump file, let's copy the file from its current `wannacry pw- infected` directory and paste it into the `volatility3` folder, which we used in *Chapter 5*, *Installing Wine in Kali Linux*, for analysis of the `cridex.vmem` memory dump. This again makes access to our memory dump file easier by not having to specify a lengthy path to the file each time we need to use a different plugin.
5. To ensure all our Volatility 3 files and `wcry.raw` files are in the correct folder, let's open a new terminal and change directories to our `volatility3` folder, and then issue the `ls` command, as seen in the following screenshot:

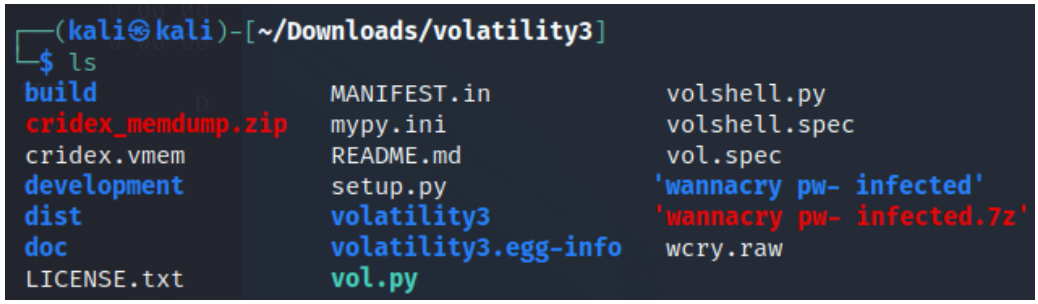


Figure 11.22 – Contents of the volatility3 directory

Now for the exciting part. Let's do some ransomware DFIR analysis using Volatility 3 and see what we can find:

1. Let's find out what operating system was running on the system by using the `info` plugin:

```
python3 vol.py -f wcry.raw windows.info
```

The following screenshot shows the output of the preceding command.

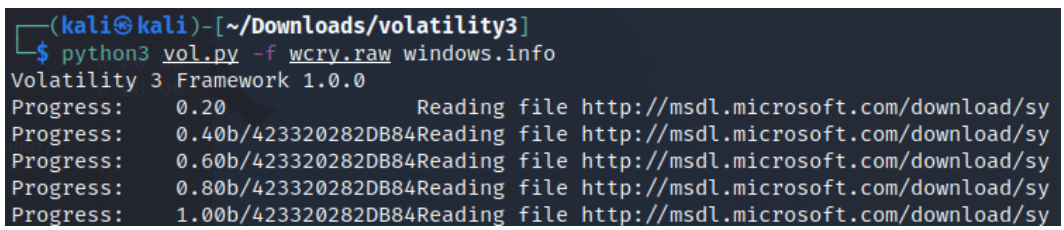


Figure 11.23 – Volatility 3 info plugin output

The output for the `info` plugin is lengthy; however, I've included a snippet of the output as follows, where we can see that this memory dump was taken from a Windows XP Service Pack 3 machine.

```
NTBuildLab      2600.xpsp_sp3_qfe.130704-0421
CSDVersion      3
KdVersionBlock  0x8054cf38
Major/Minor     15.2600
MachineType     332
KeNumberProcessors      1
SystemTime      2017-05-12 21:26:32
NtSystemRoot    C:\WINDOWS
NtProductType   NtProductWinNt
NtMajorVersion  5
NtMinorVersion  1
PE MajorOperatingSystemVersion  5
PE MinorOperatingSystemVersion  1
PE Machine      332
PE TimeDateStamp        Thu Jul  4 02:58:58 2013
```

Figure 11.24 – info plugin snippet

As done in the previous chapter, let's again do some process identification and analysis using the `pslist`, `pstree`, and `psscan` plugins individually.

The pslist plugin

Let's get the list of all running processes, using the `pslist` plugin:

```
python3 vol.py -f wcry.raw windows.pslist
```

In the following screenshot, we can see the `System`, `smss`, `csrss`, `winlogon.exe`, `services.exe`, `lsass.exe`, `svchost.exe`, and `explorer.exe` services were all started first and then followed by a few others:

```
┌──(kali㉿kali)-[~/Downloads/volatility3]
└─$ python3 vol.py -f wcry.raw windows.pslist
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0x823c8830      51      244     N/A     False   N/A     N/A     Disabled
348     4       smss.exe        0x82169020      3       19      N/A     False   2017-05-12 21:21:55.000000      N/A     Disabled
596     348     csrss.exe       0x82161da0      12      352     0       False   2017-05-12 21:22:00.000000      N/A     Disabled
620     348     winlogon.exe    0x8216e020      23      536     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
664     620     services.exe    0x821937f0      15      265     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
676     620     lsass.exe       0x82191658      23      353     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
836     664     svchost.exe     0x8221a2c0      19      211     0       False   2017-05-12 21:22:02.000000      N/A     Disabled
904     664     svchost.exe     0x821b5230      9       227     0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1024    664     svchost.exe     0x821af7e8      79      1366    0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1084    664     svchost.exe     0x8203b7a8      6       72      0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1152    664     svchost.exe     0x821bea78      10      173     0       False   2017-05-12 21:22:06.000000      N/A     Disabled
1484    664     spoolsv.exe     0x821e2da0      14      124     0       False   2017-05-12 21:22:09.000000      N/A     Disabled
1636    1608    explorer.exe    0x821d9da0      11      331     0       False   2017-05-12 21:22:10.000000      N/A     Disabled
1940    1636    tasksche.exe    0x82218da0      7       51      0       False   2017-05-12 21:22:14.000000      N/A     Disabled
1956    1636    ctfmon.exe      0x82231da0      1       86      0       False   2017-05-12 21:22:14.000000      N/A     Disabled
260     664     svchost.exe     0x81fb95d8      5       105     0       False   2017-05-12 21:22:18.000000      N/A     Disabled
740     1940    @WanaDecryptor@ 0x81fde308      2       70      0       False   2017-05-12 21:22:22.000000      N/A     Disabled
1768    1024    wuauclt.exe     0x81f747c0      7       132     0       False   2017-05-12 21:22:52.000000      N/A     Disabled
544     664     alg.exe 0x82010020      6       101     0       False   2017-05-12 21:22:55.000000      N/A     Disabled
1168    1024    wscntfy.exe     0x81fea8a0      1       37      0       False   2017-05-12 21:22:56.000000      N/A     Disabled
```

Figure 11.25 – plslist plugin output

Immediately, just using the `pslist` plugin, we can see a suspicious entry (fourth from last) called `@WanaDecryptor@` with a PID of `740` and a PPID of `1940`. To make things easy, I've included a snippet of the entry as follows:

```
┌──(kali㉿kali)-[~/Downloads/volatility3]
└─$ python3 vol.py -f wcry.raw windows.pslist
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0×823c8830      51      244     N/A     False   N/A     N/A     Disabled
348     4       smss.exe        0×82169020      3       19      N/A     False   2017-05-12 21:21:55.000000      N/A     Disabled
596     348     csrss.exe       0×82161da0      12      352     0       False   2017-05-12 21:22:00.000000      N/A     Disabled
620     348     winlogon.exe    0×8216e020      23      536     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
664     620     services.exe    0×821937f0      15      265     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
676     620     lsass.exe       0×82191658      23      353     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
836     664     svchost.exe     0×8221a2c0      19      211     0       False   2017-05-12 21:22:02.000000      N/A     Disabled
904     664     svchost.exe     0×821b5230      9       227     0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1024    664     svchost.exe     0×821af7e8      79      1366    0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1084    664     svchost.exe     0×8203b7a8      6       72      0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1152    664     svchost.exe     0×821bea78      10      173     0       False   2017-05-12 21:22:06.000000      N/A     Disabled
1484    664     spoolsv.exe     0×821e2da0      14      124     0       False   2017-05-12 21:22:09.000000      N/A     Disabled
1636    1608    explorer.exe    0×821d9da0      11      331     0       False   2017-05-12 21:22:10.000000      N/A     Disabled
1940    1636    tasksche.exe    0×82218da0      7       51      0       False   2017-05-12 21:22:14.000000      N/A     Disabled
1956    1636    ctfmon.exe      0×82231da0      1       86      0       False   2017-05-12 21:22:14.000000      N/A     Disabled
260     664     svchost.exe     0×81fb95d8      5       105     0       False   2017-05-12 21:22:18.000000      N/A     Disabled
740     1940    @WanaDecryptor@ 0×81fde308      2       70      0       False   2017-05-12 21:22:22.000000      N/A     Disabled
1768    1024    wuauclt.exe     0×81f747c0      7       132     0       False   2017-05-12 21:22:52.000000      N/A     Disabled
544     664     alg.exe 0×82010020      6       101     0       False   2017-05-12 21:22:55.000000      N/A     Disabled
1168    1024    wscntfy.exe     0×81fea8a0      1       37      0       False   2017-05-12 21:22:56.000000      N/A     Disabled
```

Figure 11.26 – Snippet of the pslist plugin showing the @WanaDecryptor@ process

Looking at the `pslist` output, we can see that the `winlogon.exe` process in *Figure 11.25* has a PID of `620` and a PPID of `348`. The PPIDs of the `services.exe` and `lsass.exe` processes (directly after the `winlogon.exe` process) are both `620`, indicating that `winlogon.exe` is in fact the PPID for both `services.exe` and `lsass.exe`.

We can also tell that `explorer.exe` with a PID of `1636` is the PPID of `tasksche.exe` and `ctfmon.exe`.

Further down, we see that `tasksche.exe` (Task Scheduler) with a PID of `1940` is the PPID of `@WanaDecryptor@`.

Let's view this a bit differently using the `pstree` plugin:

```
python3 vol.py -f wcry.raw windows.pstree
```

In the following screenshot, it is easier to see that `explorer.exe` is the parent process of `ctfmon`, `tasksche`, and `@WanaDecryptor@`:

```
└─$ python3 vol.py -f wcry.raw windows.pstree
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime

4       0       System  0x81fea8a0      51      244     N/A     False   N/A     N/A
* 348   4       smss.exe        0x81fea8a0      3       19      N/A     False   2017-05-12 21:21:55.000000      N/A
** 620  348     winlogon.exe    0x81fea8a0      23      536     0       False   2017-05-12 21:22:01.000000      N/A
*** 664 620     services.exe    0x81fea8a0      15      265     0       False   2017-05-12 21:22:01.000000      N/A
**** 1024       664     svchost.exe     0x81fea8a0      79      1366    0       False   2017-05-12 21:22:03.000000
***** 1768      1024    wuauclt.exe     0x81fea8a0      7       132     0       False   2017-05-12 21:22:52.000000
***** 1168      1024    wscntfy.exe     0x81fea8a0      1       37      0       False   2017-05-12 21:22:56.000000
**** 1152       664     svchost.exe     0x81fea8a0      10      173     0       False   2017-05-12 21:22:06.000000
**** 544        664     alg.exe 0x81fea8a0      6       101     0       False   2017-05-12 21:22:55.000000      N/A
**** 836        664     svchost.exe     0x81fea8a0      19      211     0       False   2017-05-12 21:22:02.000000
**** 260        664     svchost.exe     0x81fea8a0      5       105     0       False   2017-05-12 21:22:18.000000
**** 904        664     svchost.exe     0x81fea8a0      9       227     0       False   2017-05-12 21:22:03.000000
**** 1484       664     spoolsv.exe     0x81fea8a0      14      124     0       False   2017-05-12 21:22:09.000000
**** 1084       664     svchost.exe     0x81fea8a0      6       72      0       False   2017-05-12 21:22:03.000000
*** 676 620     lsass.exe       0x81fea8a0      23      353     0       False   2017-05-12 21:22:01.000000      N/A
** 596  348     csrss.exe       0x81fea8a0      12      352     0       False   2017-05-12 21:22:00.000000      N/A
1636    1608    explorer.exe    0x81fea8a0      11      331     0       False   2017-05-12 21:22:10.000000      N/A
* 1956  1636    ctfmon.exe      0x81fea8a0      1       86      0       False   2017-05-12 21:22:14.000000      N/A
* 1940  1636    tasksche.exe    0x81fea8a0      7       51      0       False   2017-05-12 21:22:14.000000      N/A
** 740  1940    @WanaDecryptor@ 0x81fea8a0      2       70      0       False   2017-05-12 21:22:22.000000      N/A
```

Figure 11.27 – pstree plugin output

Let's now use the `psscan` plugin to display processes that can be used by malware, such as rootkits, and are well known for doing just that to evade discovery by users and antivirus programs:

```
volatility --profile=WinXPSP2x86 -f 0zapftis.vmem psscan
```

The following screenshot shows the output of the preceding command.

```
└─$ python3 vol.py -f wcry.raw windows.psscan
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset  Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

860     1940    taskdl.exe      0x1f4daf0       0       -       0       False   2017-05-12 21:26:23.000000      2017-05-12 21:26:23.000000
536     1940    taskse.exe      0x1f53d18       0       -       0       False   2017-05-12 21:26:22.000000      2017-05-12 21:26:23.000000
424     1940    @WanaDecryptor@ 0x1f69b50       0       -       0       False   2017-05-12 21:25:52.000000      2017-05-12 21:25:53.000000
1768    1024    wuauclt.exe     0x1f747c0       7       132     0       False   2017-05-12 21:22:52.000000      N/A     Disabled
576     1940    @WanaDecryptor@ 0x1f8ba58       0       -       0       False   2017-05-12 21:26:22.000000      2017-05-12 21:26:23.000000
260     664     svchost.exe     0x1fb95d8       5       105     0       False   2017-05-12 21:22:18.000000      N/A     Disabled
740     1940    @WanaDecryptor@ 0x1fde308       2       70      0       False   2017-05-12 21:22:22.000000      N/A     Disabled
1168    1024    wscntfy.exe     0x1fea8a0       1       37      0       False   2017-05-12 21:22:56.000000      N/A     Disabled
544     664     alg.exe 0x2010020       6       101     0       False   2017-05-12 21:22:55.000000      N/A     Disabled
1084    664     svchost.exe     0x203b7a8       6       72      0       False   2017-05-12 21:22:03.000000      N/A     Disabled
596     348     csrss.exe       0x2161da0       12      352     0       False   2017-05-12 21:22:00.000000      N/A     Disabled
348     4       smss.exe        0x2169020       3       19      N/A     False   2017-05-12 21:21:55.000000      N/A     Disabled
620     348     winlogon.exe    0x216e020       23      536     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
676     620     lsass.exe       0x2191658       23      353     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
664     620     services.exe    0x21937f0       15      265     0       False   2017-05-12 21:22:01.000000      N/A     Disabled
1024    664     svchost.exe     0x21af7e8       79      1366    0       False   2017-05-12 21:22:03.000000      N/A     Disabled
904     664     svchost.exe     0x21b5230       9       227     0       False   2017-05-12 21:22:03.000000      N/A     Disabled
1152    664     svchost.exe     0x21bea78       10      173     0       False   2017-05-12 21:22:06.000000      N/A     Disabled
1636    1608    explorer.exe    0x21d9da0       11      331     0       False   2017-05-12 21:22:10.000000      N/A     Disabled
1484    664     spoolsv.exe     0x21e2da0       14      124     0       False   2017-05-12 21:22:09.000000      N/A     Disabled
1940    1636    tasksche.exe    0x2218da0       7       51      0       False   2017-05-12 21:22:14.000000      N/A     Disabled
836     664     svchost.exe     0x221a2c0       19      211     0       False   2017-05-12 21:22:02.000000      N/A     Disabled
1956    1636    ctfmon.exe      0x2231da0       1       86      0       False   2017-05-12 21:22:14.000000      N/A     Disabled
4       0       System  0x23c8830       51      244     N/A     False   N/A     N/A     Disabled
```

Figure 11.28 – psscan plugin output

The output of both the `pslist` and `psscan` commands should be compared to note similarities.

Let's run the `cmdline` plugin to map and view the paths to processes and executables:

```
python3 vol.py -f wcry.raw windows.cmdline
```

In the following screenshot, we can now be certain that there is an `@WanaDecryptor@` executable on the system that was executed at some point by a user:

```
└─$ python3 vol.py -f wcry.raw windows.cmdline
Volatility 3 Framework 1.0.0
Progress:  100.00               PDB scanning finished
PID     Process Args

4       System  Required memory at 0×10 is not valid (process exited?)
348     smss.exe        \SystemRoot\System32\smss.exe
596     csrss.exe       C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSe
rverDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Of
620     winlogon.exe    winlogon.exe
664     services.exe    C:\WINDOWS\system32\services.exe
676     lsass.exe       C:\WINDOWS\system32\lsass.exe
836     svchost.exe     C:\WINDOWS\system32\svchost -k DcomLaunch
904     svchost.exe     C:\WINDOWS\system32\svchost -k rpcss
1024    svchost.exe     C:\WINDOWS\System32\svchost.exe -k netsvcs
1084    svchost.exe     C:\WINDOWS\system32\svchost.exe -k NetworkService
1152    svchost.exe     C:\WINDOWS\system32\svchost.exe -k LocalService
1484    spoolsv.exe     C:\WINDOWS\system32\spoolsv.exe
1636    explorer.exe    C:\WINDOWS\Explorer.EXE
1940    tasksche.exe    "C:\Intel\ivecuqmanpnirkt615\tasksche.exe"
1956    ctfmon.exe      "C:\WINDOWS\system32\ctfmon.exe"
260     svchost.exe     C:\WINDOWS\system32\svchost.exe -k LocalService
740     @WanaDecryptor@ @WanaDecryptor@.exe
1768    wuauclt.exe     "C:\WINDOWS\system32\wuauclt.exe" /RunStoreAsComServer Local\[4
544     alg.exe C:\WINDOWS\System32\alg.exe
1168    wscntfy.exe     C:\WINDOWS\system32\wscntfy.exe
```

Figure 11.29 – cmdline plugin output

Let's attempt to find more information on `@WanaDecryptor@.exe` to map the infection to a user using the `envars` plugin:

```
python3 vol.py -f wcry.raw windows.envars
```

The following screenshot shows the output of the preceding command.

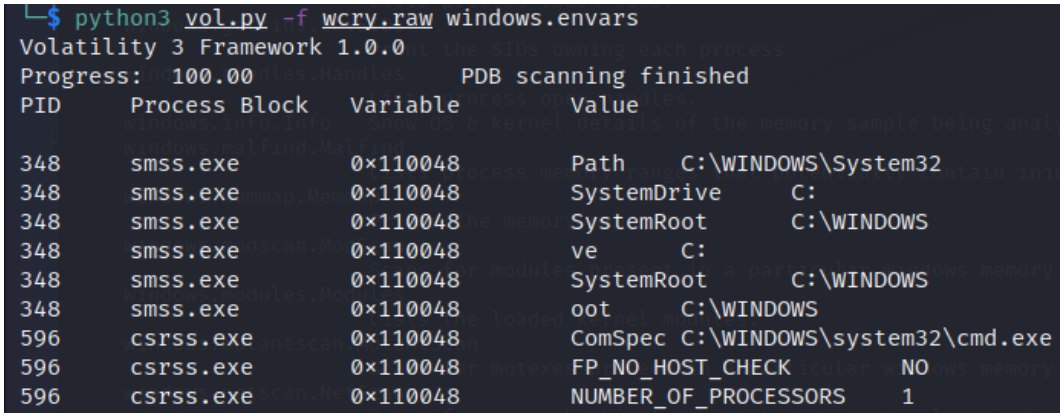


Figure 11.30 – envars plugin output

The `envars` plugin is lengthy, so I've scrolled down and taken a snippet of the `@WanaDecryptor@` processes in the following screenshot:

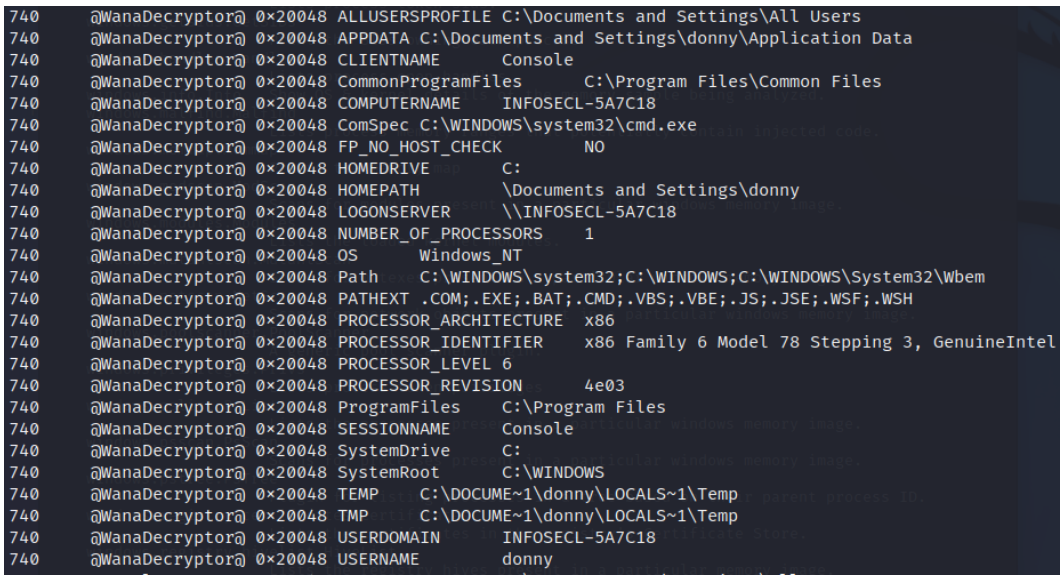


Figure 11.31 – Additional envars plugin output

In the `envars` output in *Figure 11.31*, we've found some very useful information. We can now tell that the user Donny's files have been infected with `WannaCry`/ `@WanaDecryptor@`, and we know all paths of infection.

Let's now use the `getsids` plugins to view the privileges of the processes:

```
python3 vol.py -f wcry.raw windows.getsids
```

The following screenshot shows the output of the `getsids` plugin command:

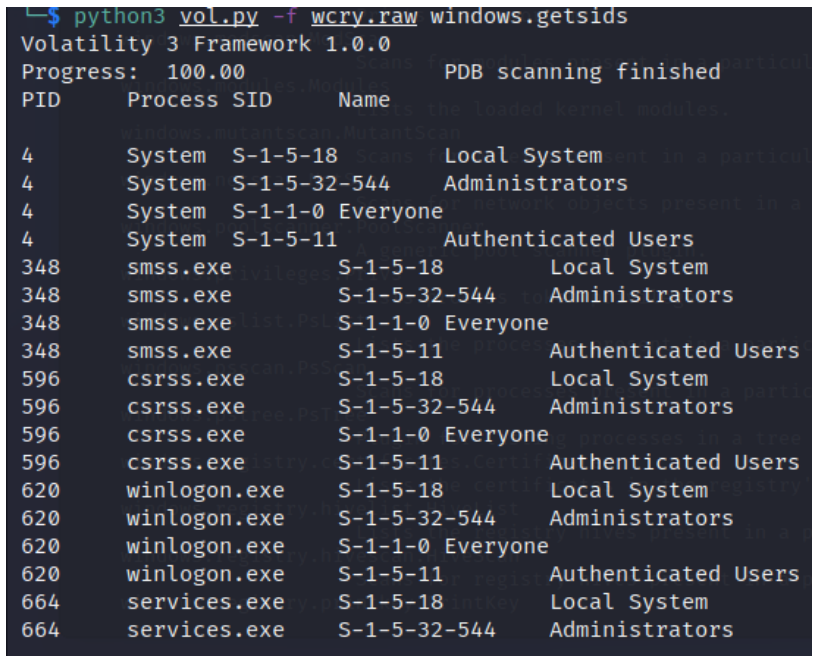


Figure 11.32 – getsids plugin output

If we scroll down a bit, we can see that the `@WanaDecryptor@` process with a PID of `740` has local and administrator user privileges.

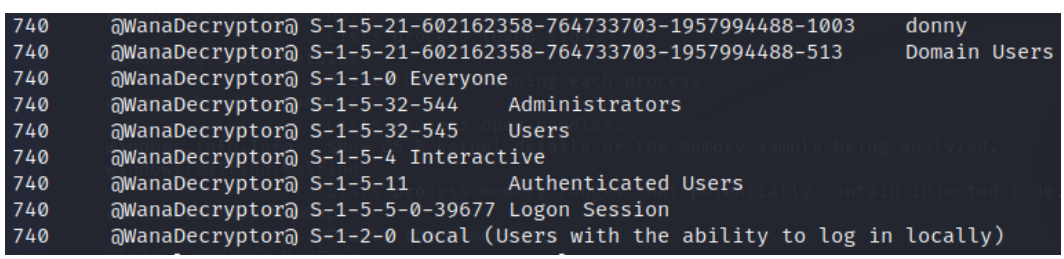


Figure 11.33 – getsids plugin output continued

Let's verify this by running the `privileges` plugin to see what access `@WanaDecryptor@` has:

```
python3 vol.py -f wcry.raw windows.privileges
```

As seen previously, the `@WanaDecryptor@` process can perform several tasks and may also have read/write access.

```
740   @WanaDecryptor@ 23    SeChangeNotifyPrivilege Present,Enabled,Default Receive notifications of changes to files
740   @WanaDecryptor@ 8     SeSecurityPrivilege     Present Manage auditing and security log
740   @WanaDecryptor@ 17    SeBackupPrivilege       Present Backup files and directories
740   @WanaDecryptor@ 18    SeRestorePrivilege      Present Restore files and directories
740   @WanaDecryptor@ 12    SeSystemtimePrivilege   Present Change the system time
740   @WanaDecryptor@ 19    SeShutdownPrivilege     Present Shut down the system
740   @WanaDecryptor@ 24    SeRemoteShutdownPrivilege       Present Force shutdown from a remote system
740   @WanaDecryptor@ 9     SeTakeOwnershipPrivilege        Present Take ownership of files/objects
740   @WanaDecryptor@ 20    SeDebugPrivilege        Present Debug programs
740   @WanaDecryptor@ 22    SeSystemEnvironmentPrivilege    Present Edit firmware environment values
740   @WanaDecryptor@ 11    SeSystemProfilePrivilege        Present Profile system performance
740   @WanaDecryptor@ 13    SeProfileSingleProcessPrivilege Present Profile a single process
740   @WanaDecryptor@ 14    SeIncreaseBasePriorityPrivilege Present Increase scheduling priority
740   @WanaDecryptor@ 10    SeLoadDriverPrivilege   Present,Enabled Load and unload device drivers
740   @WanaDecryptor@ 15    SeCreatePagefilePrivilege       Present Create a pagefile
740   @WanaDecryptor@ 5     SeIncreaseQuotaPrivilege        Present Increase quotas
740   @WanaDecryptor@ 25    SeUndockPrivilege       Present,Enabled Remove computer from docking station
740   @WanaDecryptor@ 28    SeManageVolumePrivilege Present Manage the files on a volume
740   @WanaDecryptor@ 29    SeImpersonatePrivilege  Present,Enabled,Default Impersonate a client after authentication
740   @WanaDecryptor@ 30    SeCreateGlobalPrivilege Present,Enabled,Default Create global objects
```

Figure 11.34 – Additional privileges plugin output

We can confirm this and also find specific instances of the `@WanaDecryptor@` malware using the `malfind` plugin, which will also pinpoint other processes, such as `winlogon`, that may be compromised:

```
python3 vol.py -f wcry.raw windows.malfind
```

The following screenshot shows the output of the preceding command:

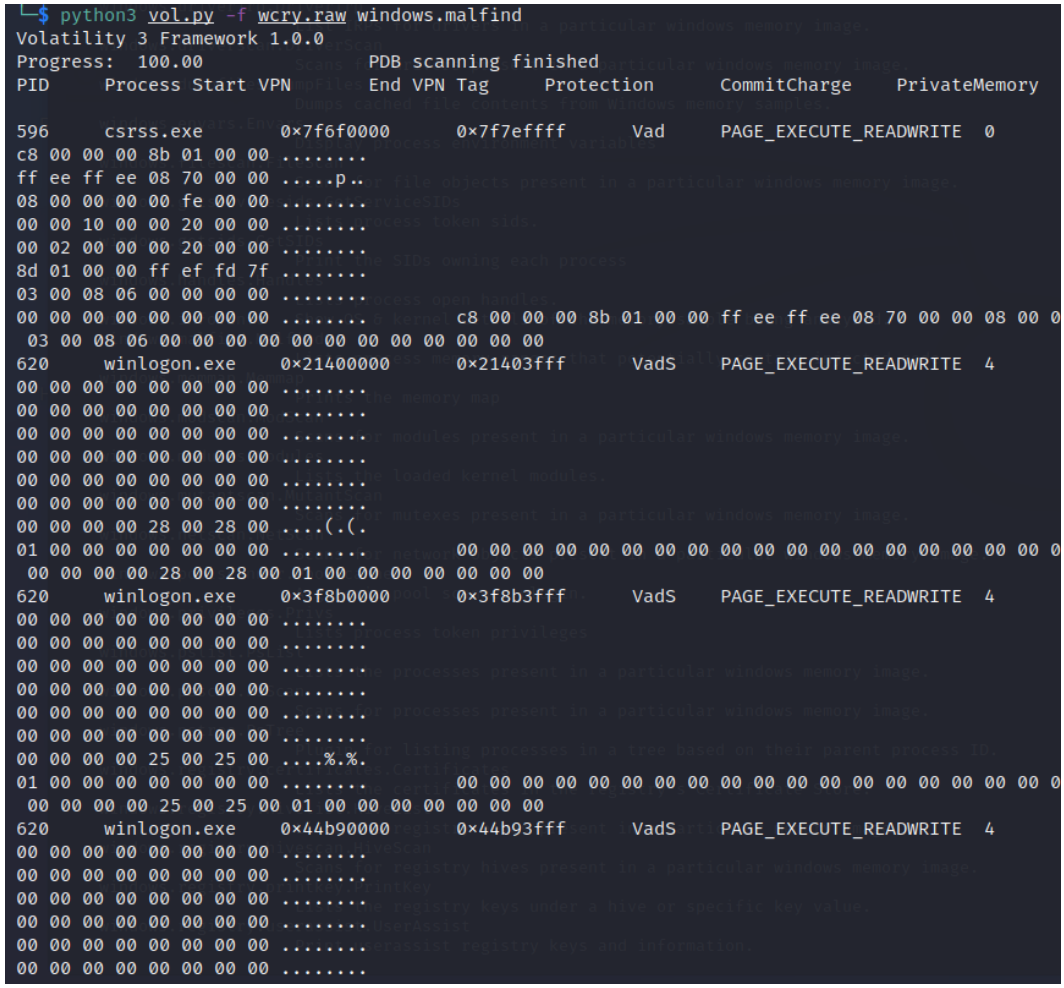


Figure 11.35 – malfind plugin output

I hope you enjoyed analyzing and finding the WannaCry ransomware using Volatility. It involves a lot of work, but with a little practice, you can easily master this very important DFIR tool.

Summary

This was certainly an intense chapter! We learned how to detect running processes and connections using `p0f` and also did some investigation using `swap_digger`, which revealed useful artifacts such as passwords within the paging file of a live Linux system. We then also used `MimiPenguin` to try dumping the current password of the system.

We then moved on to the very exciting topic of malware analysis where we discovered embedded malicious files within a PDF using `pdf-parser` and `PDFiD`, and I also introduced you to an online tool at `hybrid-analysis.com`, which I frequently use to analyze suspicious files and URLs.

Finally, we carried out an exciting lab, performing ransomware analysis using the incredibly useful Volatility 3 tool, where we found processes belonging to the WannaCry ransomware and, upon further analysis, were able to pinpoint the infected user, paths, documents, and other processes.

Next up, we will delve a bit more into automated file analysis using the Autopsy browser within Kali Linux. See you in the next chapter!

Part 4: Automated Digital Forensics and Incident Response Suites

In this part, we look at one of the most popular and powerful open source tools used by DFIR investigators and analysts, called Autopsy. Autopsy is available for both Linux and Windows, and although we previously covered only the Linux version in past editions of this book, I am absolutely delighted to show you how to get the more powerful Windows version up and running in Kali Linux using Wine in this version.

This part has the following chapters:

- *Chapter 12*, *Autopsy Forensic Browser*
- *Chapter 13*, *Performing a Full DFIR Analysis with the Autopsy 4 GUI*

12
Autopsy Forensic Browser

Autopsy and **The Sleuth Kit** go hand in hand. Both were created by Brian Carrier. The Sleuth Kit is a powerful suite of **Command-Line Interface** (**CLI**) forensic tools, whereas Autopsy is the **Graphical User Interface** (**GUI**; pronounced *gooey*) that sits on top of The Sleuth Kit and is accessed through a web browser. The Sleuth Kit supports disk image file types, including Raw **Data Dump (DD)**, **EnCase (.01)**, and **Advanced Forensic Format** (**AFF**).

The Sleuth Kit uses CLI tools to perform the following tasks:

- Find and list allocated and unallocated (deleted) files, and even files hidden by rootkits
- Reveal NTFS **Alternate Data Streams** (**ADS**) where files can be concealed within other files
- List files by type
- Display metadata information
- Create a timeline

Autopsy can be run from a live **Compact Disk** (**CD**)/**Universal Serial Bus** (**USB**) in forensic mode as part of a live analysis in live mode, or it can be used on a dedicated Kali Linux machine to investigate analysis in dead mode.

The topics we will cover in this chapter include the following:

- Introduction to The Sleuth Kit and the Autopsy forensic browser
- Downloading sample files for use and creating a case in the Autopsy browser
- Evidence analysis using the Autopsy forensic browser

Introduction to Autopsy – The Sleuth Kit

In this chapter, we will focus on the Autopsy browser, which is based on data recovery and disk analysis tools found within The Sleuth Kit. Tools in The Sleuth Kit are all command line-based and the Autopsy browser allows us to access the tools and their functionality via a GUI for easy disk and file carving, recovery, analysis, and reporting.

Autopsy offers GUI access to a variety of investigative command-line tools from The Sleuth Kit, including file analysis, image and file hashing, deleted file recovery, and case management, among other capabilities. Autopsy can be a bit tricky to install but, fortunately for us, it comes built into Kali Linux, and is also very easy to set up and use.

Although the Autopsy browser is based on The Sleuth Kit, the features of Autopsy differ when using the Windows version as compared to the Linux version. We will cover the Windows version of Autopsy using Wine in *Chapter 13, Performing a Full DFIR Analysis with the Autopsy 4 GUI*. Some of the official features offered by The Sleuth Kit and Autopsy 2.4 in Kali Linux include the following:

- **Image analysis**: Analyzing directories and files, including sorting files, recovering deleted files, and previewing files
- **File activity timelines**: Creating timelines based on timestamps of when files were written, accessed, and created
- **Image integrity**: Creating MD5 hashes of the image file used, as well as individual files
- **Hash databases**: Matching digital hashes or fingerprints of unknown files (such as suspected malicious `.exe` files) against those in the NIST **National Software Reference Library** (**NSRL**)
- **Events sequencer**: Displaying events sorted by date and time
- **File analysis**: Analyzing the entire image file to display directory and file information and contents
- **Keyword search**: Allows searching using keyword lists and predefined expression lists
- **Metadata analysis**: Allows viewing metadata details and the structure of files that are essential for data recovery

Now that we are familiar with some of the uses of Autopsy, let's first download our sample files for analysis before we begin using the tool.

Downloading sample files for use and creating a case in the Autopsy browser

In this section, we will find the links to download the required sample files that we will be analyzing using the Autopsy browser.

> **Download the image file**
>
> The image file used for analysis is publicly available for download at `http://downloads.digitalcorpora.org/corpora/scenarios/2009-m57-patents/usb/`.
>
> Be sure to note the location of the downloaded sample file, as this will be required later on.

The file we will be working with is `terry-work-usb-2009-12-11.E01`, as shown in the following screenshot:

corpora/scenarios/2009-m57-patents/usb/ files:

Name	Size	Last Modified	SHA2-256
charlie-work-usb-2009-12-11.E01	9,265,553	2020-11-22 09:40:13+00:00	7a998c556b22f5dc9426a33dcaceadd21a14c9cfb22957edce461d58a14fb40a
jo-favorites-usb-2009-12-11.E01	227,073,046	2020-11-22 09:40:15+00:00	1798e0436f99f2490dbf92f09fdf7956b0f2a6cf6cafde6a426094c9de6aec54
jo-work-usb-2009-12-11.E01	118,233,120	2020-11-22 09:40:39+00:00	d3751b55c1b88e1a5fb1940a9a41cc87e750b61e4aa59f9c76bd2f17e17c3cc2
terry-work-usb-2009-12-11.E01	33,499,203	2020-11-22 09:40:41+00:00	1600fe2bdfb2bec0b006aa9d1c0ce6d3ad0b6666141a333bf830af7800fb9230

Figure 12.1 – Sample evidence files at digitalcorpora.org

> **Please note**
>
> When investigating hard drives and devices, be sure to always follow proper acquisition procedures and use a write-blocker to avoid tampering with original evidence.

Now that we have our sample image file downloaded (or perhaps even a forensically acquired image of our own), let's proceed with the analysis using the Autopsy browser by first getting acquainted with the different ways to start Autopsy.

Starting Autopsy

Autopsy can be started in two ways:

- For the first method, we use the **Applications** menu by clicking on **Applications** | **11 - Forensics** | **autopsy (root)**:

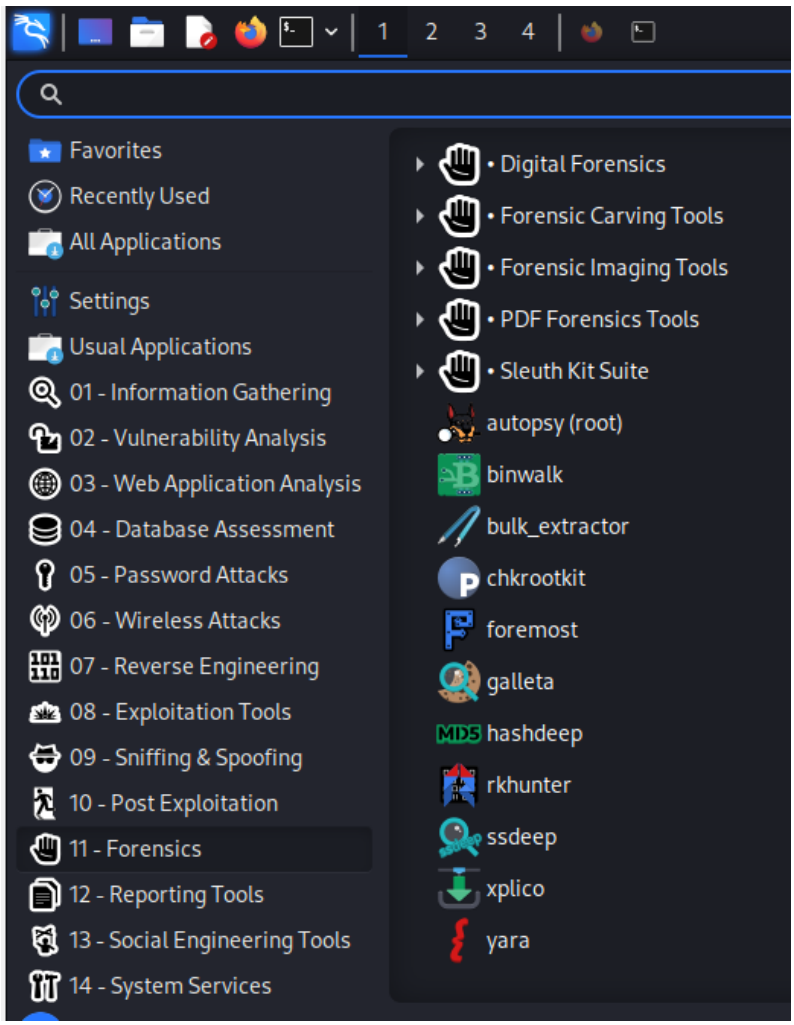


Figure 12.2 – autopsy (root) in the Kali Linux Forensics menu

- Alternatively, we can click on the search bar at the top left of the Kali Linux screen and type the word `autopsy` into it, and then click on the **autopsy (root)** icon:

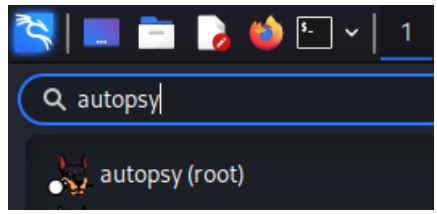


Figure 12.3 – Searching for the autopsy (root) application

Once the Autopsy icon is clicked, a new terminal is opened showing the program information along with connection details for opening the Autopsy forensic browser.

> **Note**
> The Autopsy forensic browser has not been updated for the past few years; however; we will be using the updated GUI version in the next chapter.

In the following screenshot, we can see that the version number is listed as `2.24` with the path to the `Evidence Locker` folder as `/var/lib/autopsy`:

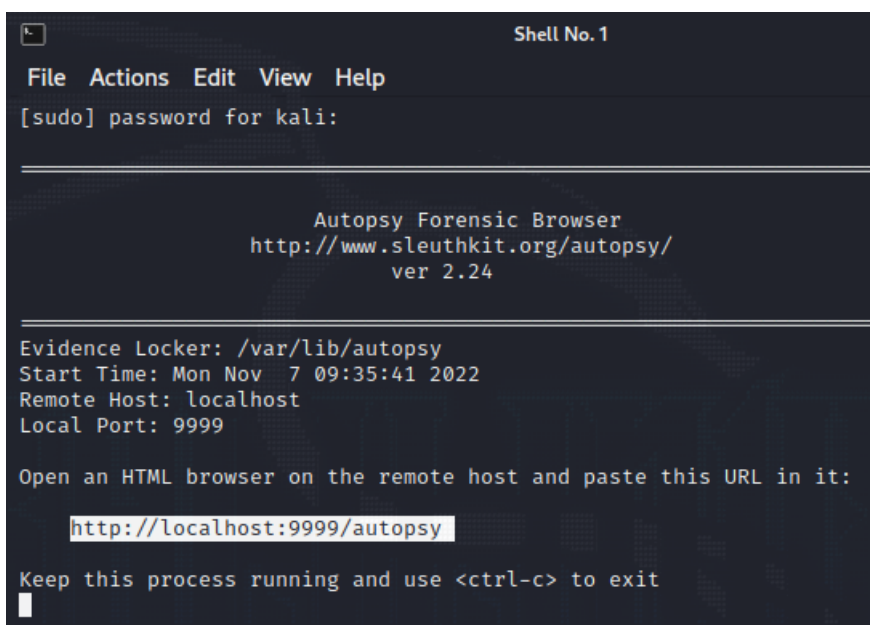


Figure 12.4 – Process to start the Autopsy forensic browser

- Alternatively, instead of clicking on the link shown in *Figure 12.4*, you can copy and paste the following link into the browser: `http://localhost:9999/autopsy`.

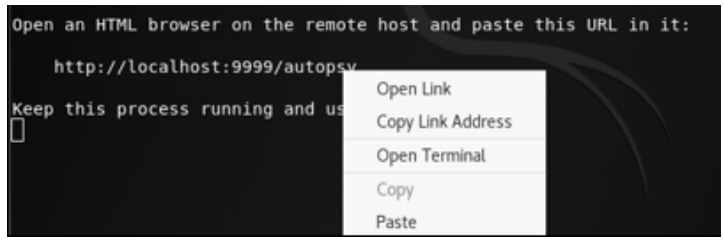


Figure 12.5 – Opening Autopsy from the terminal

Once the link is opened using either method, you will be greeted with the following web interface screen to the Autopsy forensic browser.

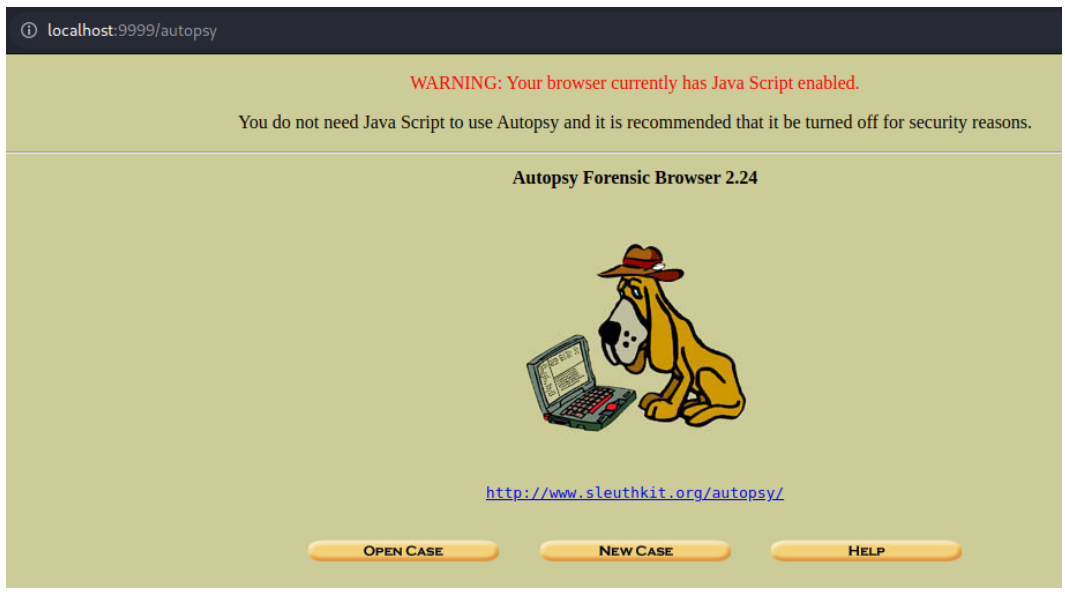


Figure 12.6 – The Autopsy forensic browser web interface

Creating a new case in the Autopsy forensic browser

To create a new case, follow the given steps:

1. When the Autopsy forensic browser opens, investigators are presented with three options: **OPEN CASE**, **NEW CASE**, and **HELP**.

   Click on **NEW CASE**:

Figure 12.7 – NEW CASE option

2. Enter details for **Case Name**, **Description**, and **Investigator Names**. For **Case Name**, I've entered `Terry_USB`, as it closely matches the image name (`terry-work-usb`) that we will be using for this investigation. Once all the information is entered, click **NEW CASE**:

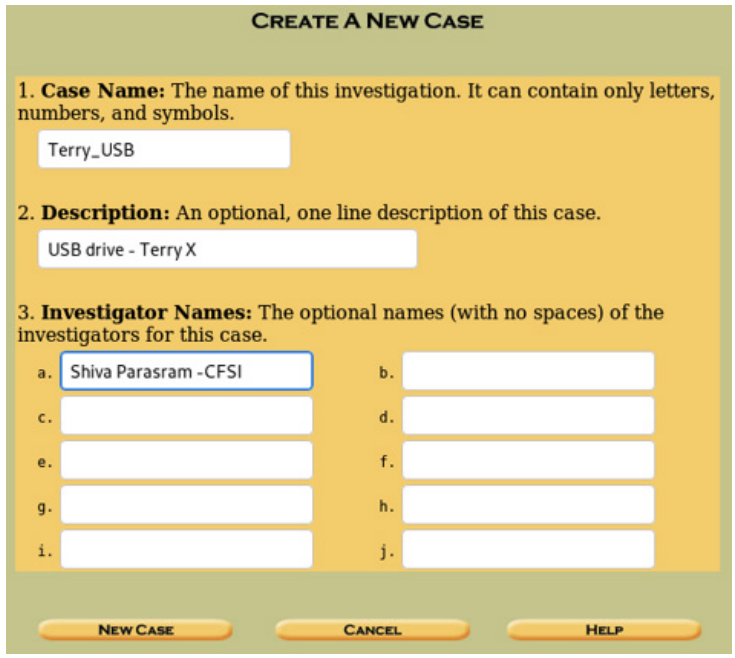


Figure 12.8 – New case details

3. The locations of the case directory and configuration file are displayed and shown as in *Figure 12.9*. It's important to take note of the case directory location, as seen in the screenshot: **Case directory (/var/lib/autopsy/Terry_USB/) created**. Click **ADD HOST** to continue:

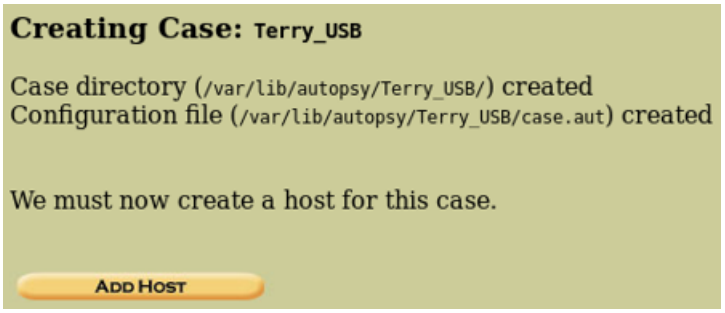


Figure 12.9 – Case location and directory details

> **Note**
> Several investigator name fields are available, as there may be instances where several investigators may be working together.

4. Enter the details for the hostname (name of the computer being investigated) and the description of the host.
5. Here are some optional settings:

   - **Time zone**: Defaults to local settings if not specified
   - **Timeskew Adjustment**: Adds a value in seconds to compensate for time differences
   - **Path of Alert Hash Database**: Specifies the path of a created database of known bad hashes
   - **Path of Ignore Hash Database**: Specifies the path of a created database of known good hashes similar to the NIST NSRL:

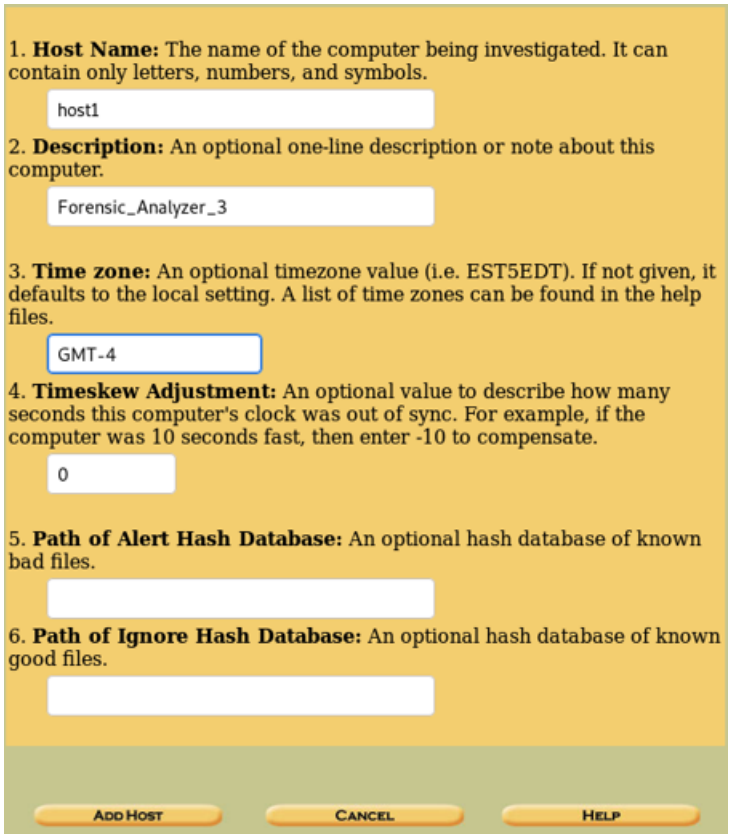


Figure 12.10 – Case details information

6. Click on the **ADD HOST** button to continue.
7. Once the host is added and directories are created, we add the forensic image we want to analyze by clicking the **ADD IMAGE** button:

Figure 12.11 – Host details

8. Click on the **ADD IMAGE FILE** button to add the image file:

Figure 12.12 – Adding the forensic image file

9. To import the image for analysis, the full path must be specified. On my machine, I've saved the image file to the default `Downloads` folder. As such, the location of the file would be `/Downloads/terry-work-usb-2009-12-11.E01`, as seen in *Figure 12.13*.

Figure 12.13 – Location and Import Method details

Note

For **Import Method**, we choose **Symlink**. This way, the image file can be imported from its current location (`Downloads`) to the `Evidence Locker` directory without the risks associated with moving or copying the image file.

10. Upon clicking **Next**, the image file details are displayed. Click on the **ADD** button to continue and then click on **OK**.

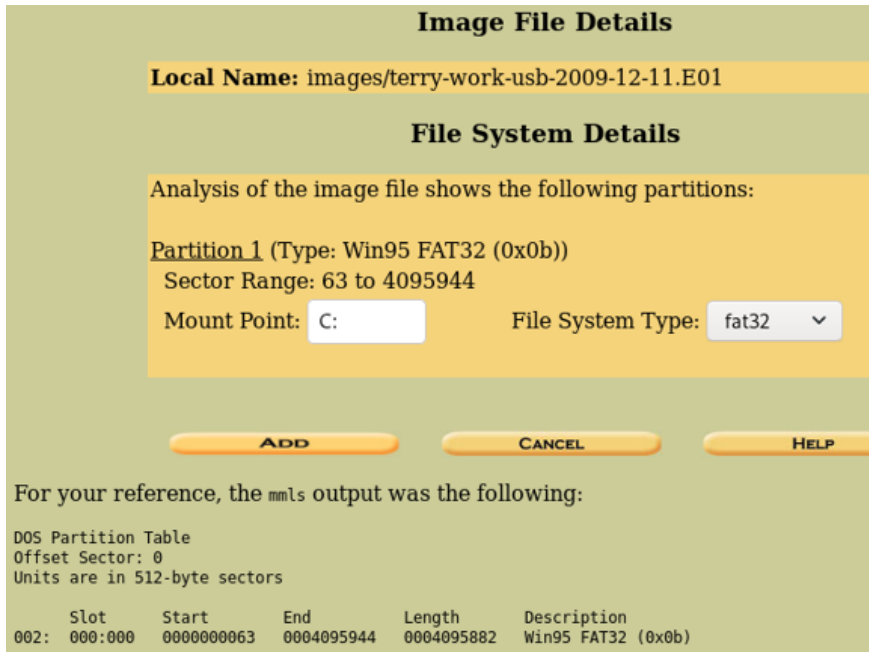


Figure 12.14 – Image File Details

11. At this point, we're just about ready to analyze the image file. Be sure to select the **C:/** option and then click on **Analyze**.

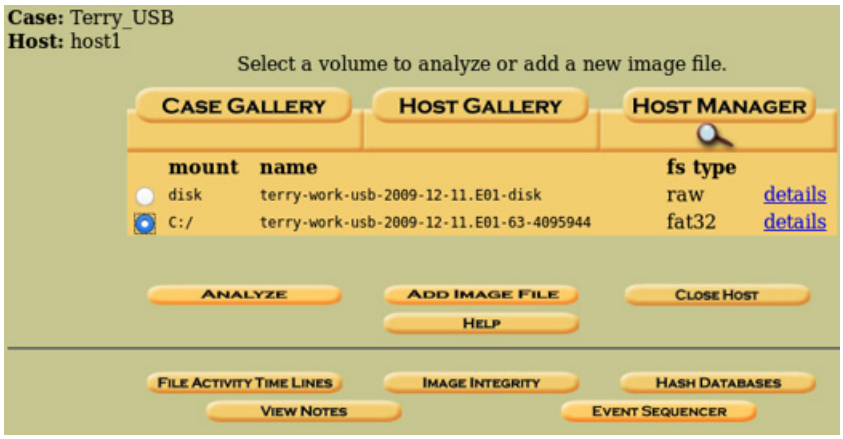


Figure 12.15 – Volume selection

Now that we have added our downloaded sample evidence file and configured all case and directory details, we can move on to the analysis of the sample evidence file.

Evidence analysis using the Autopsy forensic browser

Now that we've created our case, added host information with appropriate directories, and added our sample evidence file, we come to the analysis stage, which involves the following steps for file and drive analysis, file carving, and recovery:

1. After clicking on the **ANALYZE** button (see *Figure 12.15*), we're presented with several options in the form of tabs with which to begin our investigation:

Figure 12.16 – Analysis tab options

2. Let's look at the details of the image by clicking on the **IMAGE DETAILS** tab. In the following screenshot, we can see the volume serial number and the operating system (**OEM Name**) listed as **BSD 4.4**:

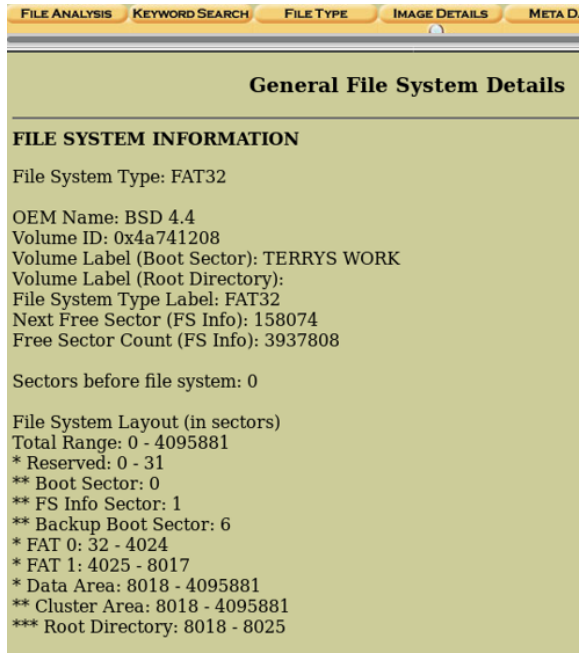


Figure 12.17 – IMAGE DETAILS tab

3. Next, we click on the **FILE ANALYSIS** tab. This tab opens into **file browsing mode**, which allows the examination of directories and files within the image. Directories within the image are listed by default in the main view area:

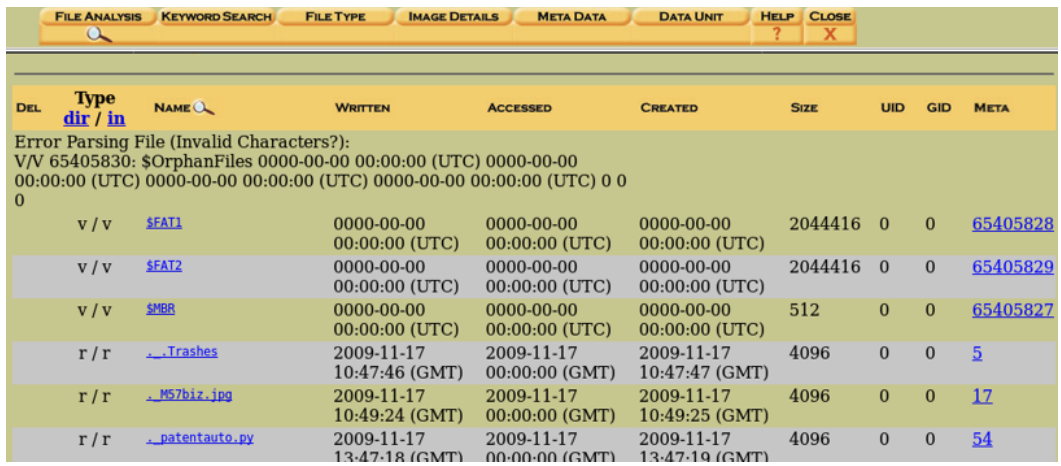
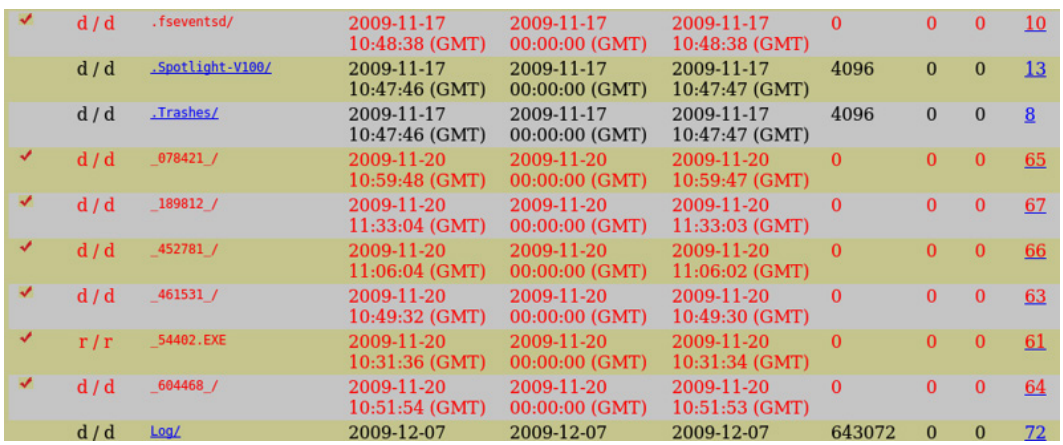


Figure 12.18 – FILE ANALYSIS tab

As seen in the preceding screenshot, for each directory and file, there are fields showing when the item was written, accessed, changed, and created, along with its size and metadata:

- **WRITTEN**: The date and time the file was last written to
- **ACCESSED**: The date and time the file was last accessed (only the date is accurate)
- **CHANGED**: The date and time the descriptive data of the file was changed
- **CREATED**: The data and time the file was created
- **META**: Metadata describing the file and information about the file

> **Note**
>
> If we scroll down a bit, we can see carved items listed in red, which are deleted files, as seen in *Figure 12.19*.



✓	d / d	.fseventsd/	2009-11-17 10:48:38 (GMT)	2009-11-17 00:00:00 (GMT)	2009-11-17 10:48:38 (GMT)	0	0	0	10
	d / d	.Spotlight-V100/	2009-11-17 10:47:46 (GMT)	2009-11-17 00:00:00 (GMT)	2009-11-17 10:47:47 (GMT)	4096	0	0	13
	d / d	.Trashes/	2009-11-17 10:47:46 (GMT)	2009-11-17 00:00:00 (GMT)	2009-11-17 10:47:47 (GMT)	4096	0	0	8
✓	d / d	_078421_/	2009-11-20 10:59:48 (GMT)	2009-11-20 00:00:00 (GMT)	2009-11-20 10:59:47 (GMT)	0	0	0	65
✓	d / d	_189812_/	2009-11-20 11:33:04 (GMT)	2009-11-20 00:00:00 (GMT)	2009-11-20 11:33:03 (GMT)	0	0	0	67
✓	d / d	_452781_/	2009-11-20 11:06:04 (GMT)	2009-11-20 00:00:00 (GMT)	2009-11-20 11:06:02 (GMT)	0	0	0	66
✓	d / d	_461531_/	2009-11-20 10:49:32 (GMT)	2009-11-20 00:00:00 (GMT)	2009-11-20 10:49:30 (GMT)	0	0	0	63
✓	r / r	_54402.EXE	2009-11-20 10:31:36 (GMT)	2009-11-20 00:00:00 (GMT)	2009-11-20 10:31:34 (GMT)	0	0	0	61
✓	d / d	_604468_/	2009-11-20 10:51:54 (GMT)	2009-11-20 00:00:00 (GMT)	2009-11-20 10:51:53 (GMT)	0	0	0	64
	d / d	Log/	2009-12-07	2009-12-07	2009-12-07	643072	0	0	72

Figure 12.19 – Carved files

4. If we scroll even further, we can see that there is an installer (`vnc-4_1_3-x86_win32.exe`) file for VNC, which was possibly downloaded to use on another machine as this operating system is not Windows.

   There is also a keylogger installation file (`xpadvancedkeylogger.exe`) listed in red, meaning it was deleted.

r / r	urlstime_machine.txt	2009-11-16 10:22:50 (GMT)	2009-11-24 00:00:00 (GMT)	2009-11-16 10:22:51 (GMT)	1538990	0	0	20
r / r	vnc-4_1_3-x86_win32.exe	2008-10-15 17:14:08 (GMT)	2009-12-07 00:00:00 (GMT)	2008-10-15 17:14:08 (GMT)	741744	0	0	75
r / r	webauto.py	2009-11-16 14:23:38 (GMT)	2009-11-24 00:00:00 (GMT)	2009-11-14 17:39:19 (GMT)	2237	0	0	6
r / r	xpadvancedkeylogger.exe	2009-12-03 09:40:44 (GMT)	2009-12-07 00:00:00 (GMT)	2009-12-03 09:41:16 (GMT)	1580660	0	0	70

Figure 12.20 – Deleted keylogger software discovered

5. The left pane of the Autopsy browser window also contains four main features that can assist in our DFIR investigation and analysis:

- **Directory Seek**: Allows the searching of directories
- **File Name Search**: Allows the searching of files by Perl expressions or filenames
- **ALL DELETED FILES**: Searches the image for deleted files
- **EXPAND DIRECTORIES**: Expands all directories for easier viewing of contents

Figure 12.21 – Additional analysis menus

6. Click on the **EXPAND DIRECTORIES** button to easily view and access all contents within the left pane and main window. The + icon next to a directory indicates that it can be further expanded to view subdirectories (++) and their contents:

Figure 12.22 – Expanded directories within the left pane

7. To view deleted files, we click on the **ALL DELETED FILES** button in the left pane. Deleted files are marked in red and also adhere to the same format of **WRITTEN**, **ACCESSED**, **CHANGED**, and **CREATED** times. In the following screenshot, we can see that the image contains several deleted files:

Figure 12.23 – Viewing all deleted files

This brings us to the end of our analysis using the Autopsy forensic browser. However, we will continue with another version known as the **Autopsy GUI v4** in the next chapter.

Summary

In this chapter, we looked at automated DFIR investigation and analysis using the Autopsy forensic browser and The Sleuth Kit. Compared to individual tools, Autopsy has case management features and supports various types of file analysis, searching, and sorting of allocated, unallocated, and hidden files. Autopsy can also perform hashing on the file and directory levels to maintain evidence integrity.

In the next chapter, we will be using the updated and much more powerful stand-alone version of the Autopsy GUI, version 4, to analyze the same file used in this chapter for comparison.

13

Performing a Full DFIR Analysis with the Autopsy 4 GUI

As we previously learned in *Chapter 12*, *Autopsy Forensic Browser*, when we used the Autopsy forensic browser, which comes with Kali Linux, Autopsy is quite a powerful tool when it comes to automated evidence and file analysis. However, the Autopsy forensic browser has some limitations, especially as it is older and not as frequently updated as the **Graphical User Interface** (**GUI**) version. The Autopsy forensic browser has been at version 2.2 for many years, whereas the Autopsy GUI is currently, at the time of writing, up to version 4.19.

In this chapter, we will focus on the Autopsy v4 GUI (also called the Autopsy 4 GUI) and analyze the very same file used in the previous chapter to compare the usage, features, and differences in analysis findings (if any). The following topics will be covered in this chapter:

- Autopsy 4 GUI features
- Installing Autopsy 4 in Kali Linux using Wine
- Downloading sample files for automated analysis
- Creating new cases and getting acquainted with the Autopsy 4 interface
- Analyzing directories and recovering deleted files and artifacts with Autopsy 4

Autopsy 4 GUI features

The Autopsy 4 GUI comes with many easy-to-use features for relatively simple and fast analysis. Autopsy 4 is one of the very few comprehensive forensic suites, is totally free to use, and is used by professionals and law enforcement.

Some of the features include the following:

- Simple and uncomplicated user interface
- Case management with all files for each case stored in their respective directories
- Available third-party add-ons and modules
- Indexed keyword searching
- Extraction of web artifacts from the browser, including cookies and browser history from Chrome and Firefox
- Automated file recovery using PhotoRec
- **Exchangeable Image File** (**EXIF**) data extraction using photo and video analysis
- Advanced timeline analysis and event viewing
- **Indicators of Compromise** (**IoC**) analysis
- Parallel task-processing for fast analysis
- Freely available for Windows, Linux, and Mac

Full details of Autopsy can be found on their home page at `https://www.sleuthkit.org/autopsy/#:~:text=Autopsy%C2%AE%20is%20a%20digital,from%20your%20camera's%20memory%20card`.

Now that we're familiar with some of the uses of the Autopsy 4 GUI, let's install it in Kali using Wine.

Installing Autopsy 4 in Kali Linux using Wine

In this example, we will be using Autopsy for Windows, which will be installed using Wine. Autopsy 4 can be downloaded for Linux; however, this may be best left to advanced users. If you haven't installed Wine on Kali Linux, you can revisit *Chapter 5*, *Installing Wine in Kali Linux*, and then return to the current chapter.

Now, let's get started with installing Autopsy 4 in Kali Linux using Wine:

1. All versions of Autopsy 4 can be found here: `https://www.autopsy.com/download/`

   To download Autopsy for Windows, you can click on the direct link here: `https://github.com/sleuthkit/autopsy/releases/download/autopsy-4.19.3/autopsy-4.19.3-64bit.msi`

   The previous version is the most current and stable 64-bit version for Windows and at the time of writing is at version 4.19.3. This tutorial should also apply to later versions as the usage has thus far remained the same over the years.

2. Once Autopsy 4.19.3 has been downloaded, you can either double-click on the downloaded file or right-click on the `autopsy-4.19.3-64bit.msi` file and select **Open with Wine Windows Program Loader** to launch the program. Doing so will launch the **Autopsy Setup** window, as seen in the following figure:

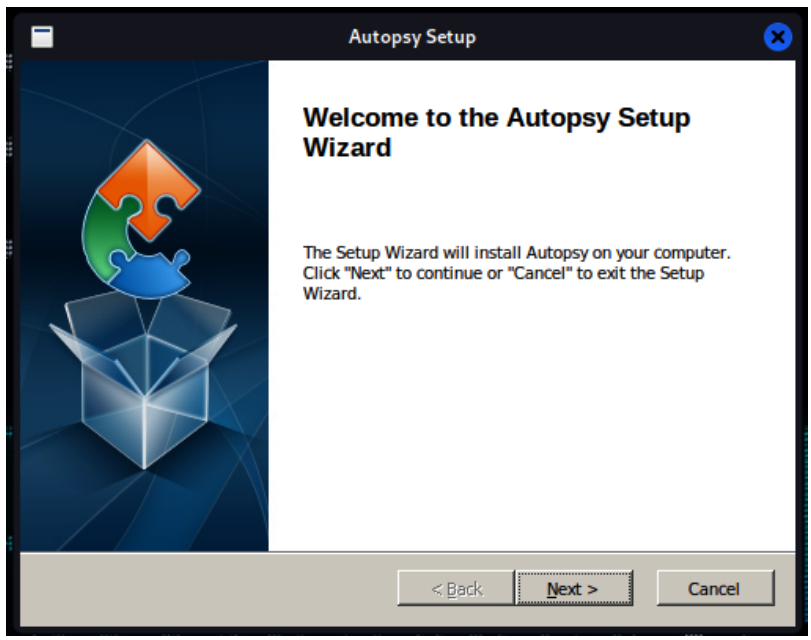


Figure 13.1 – Autopsy wizard welcome screen

3. Click on **Next** to proceed. The installation folder is listed as would usually be displayed within a Windows installation. Click on **Next** to proceed.

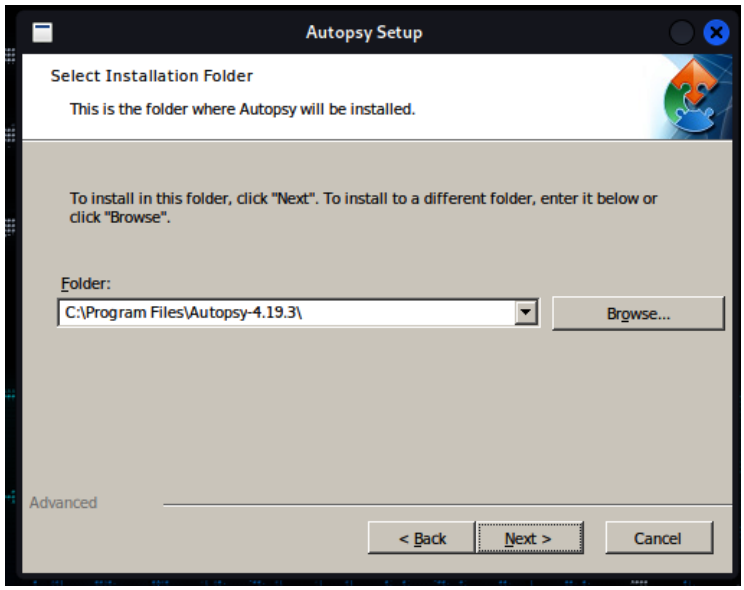


Figure 13.2 – Autopsy installation folder

4. Click on **Install** to begin the installation process.

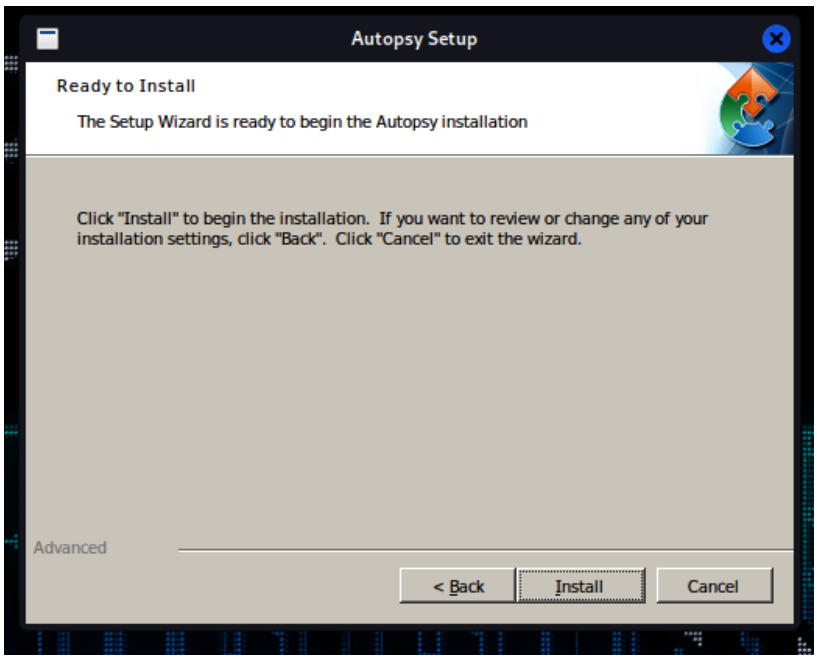


Figure 13.3 – Autopsy installation process

The installation may take a few minutes depending on how much RAM and CPU resources are assigned to your Kali environment.

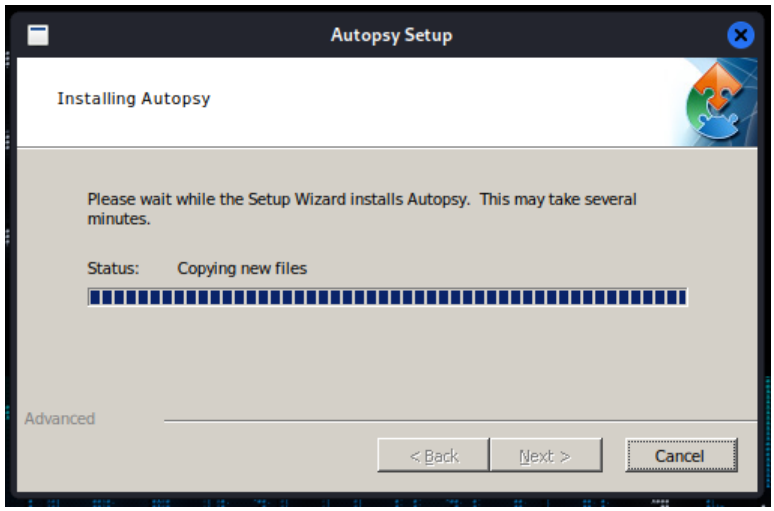


Figure 13.4 – Copy files during the installation process

5. Once all files are installed, the wizard will tell us that the installation is complete. Click on **Finish** to start Autopsy.

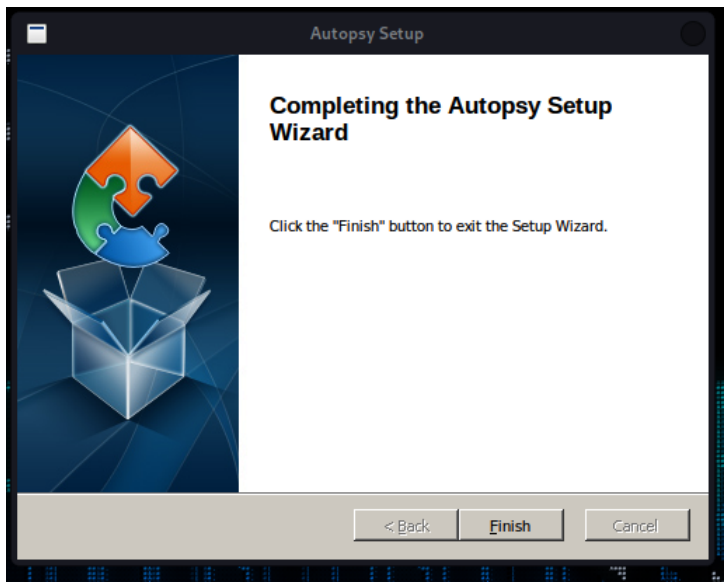


Figure 13.5 – Completed Autopsy installation

After installation, you will be presented with the Autopsy 4 splash screen. It may take a minute to load all the required modules.

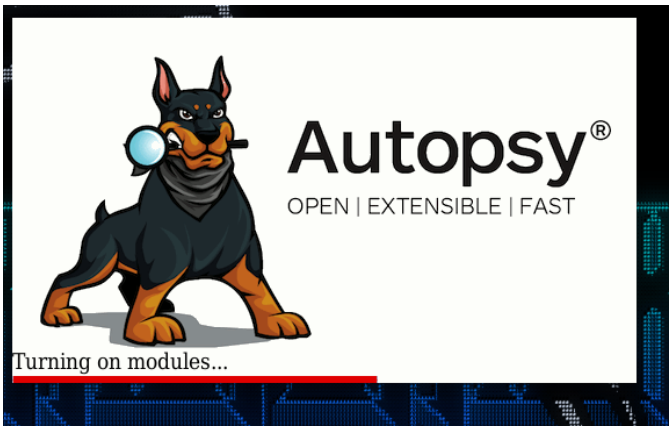


Figure 13.6 – Autopsy 4 splash screen

6. After all the modules are loaded, information about case repositories is presented, as seen in the following screenshot. Click on **OK** to continue.

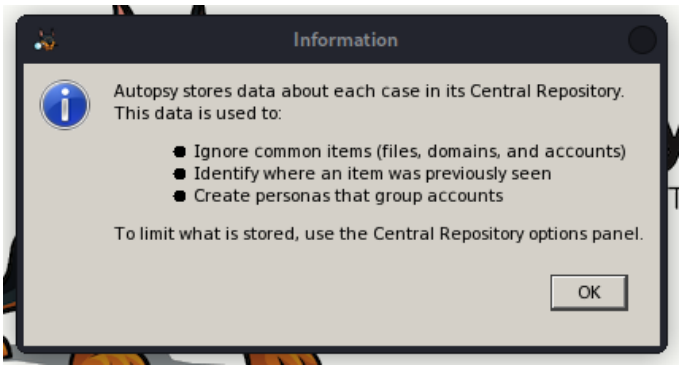


Figure 13.7 – Case repository information

We have now completed the installation and initial configuration of the Windows version of the Autopsy 4 GUI within Kali Linux using Wine. Before we proceed any further, let's ensure we have all the necessary sample files downloaded.

Downloading sample files for automated analysis

The sample evidence file we will be using will be the same `terry-work-usb-2009-12-11.E01` file downloaded in the previous chapter, which was analyzed using the Autopsy forensic browser. It can again be downloaded directly from here: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/2009-m57-patents/usb/terry-work-usb-2009-12-11.E01`.

I'd also like you to take this opportunity to download the other files for this example from the `digitalcorpora.com` website, which you can analyze on your own to become fully acquainted with Autopsy 4 as this is one of the main open source and free tools used by DFIR investigators and analysts:

- The `Charlie-work-usb` file: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/2009-m57-patents/usb/charlie-work-usb-2009-12-11.E01`
- The `Jo-favorites-usb` file: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/2009-m57-patents/usb/jo-favorites-usb-2009-12-11.E01`
- The `Jo-work-usb` file: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/2009-m57-patents/usb/jo-work-usb-2009-12-11.E01`

Again, feel free to come back to these files to improve upon your DFIR analytical skills either after reading this chapter or once you've reached the end of the book.

Let's now get back to our Autopsy 4 installation and familiarize ourselves with the interface.

Creating new cases and getting acquainted with the Autopsy 4 interface

Now that we have installed the Autopsy 4 GUI in Wine on Kali and downloaded the required files from the previous section, we can get started in the Autopsy 4 GUI by creating a new case using the sample evidence files, which we will then analyze in the next section.

Let's begin by creating a new case within the Autopsy 4 GUI:

1. Whether continuing from where we previously left off before downloading the sample evidence files or opening Autopsy 4 from the desktop or application menu, you will be greeted with the options to either begin a new case or open a case. For our purposes, let's create a new case by clicking on **New Case**:

Figure 13.8 – Creating a new case in Autopsy

2. Next, we can add a case name. I've also selected the **Documents** folder in Kali as my **Base Directory** folder to house all files and logs. Click on **Next** to continue.

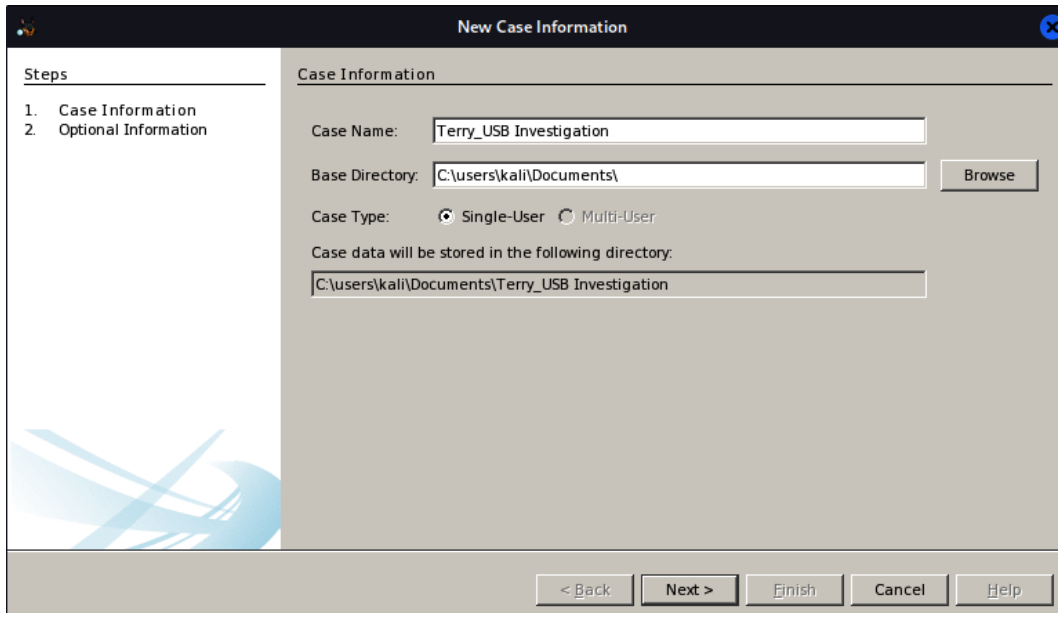


Figure 13.9 – New case details

3. We will then have to give the case a number and fill in the examiner details. When completed, click on the **Finish** button to continue.

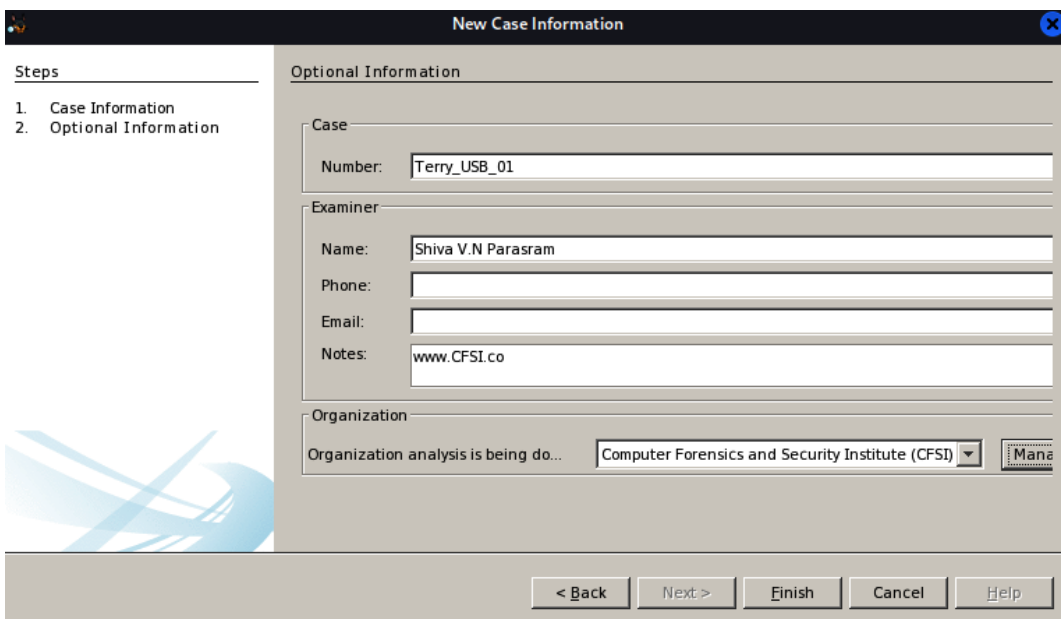


Figure 13.10 – Examiner details

Creating your new case may take several minutes as Autopsy is prepared for usage.

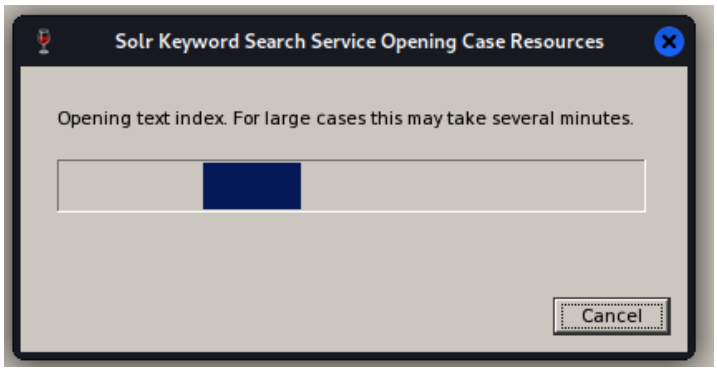


Figure 13.11 – Opening new case status display

4. Then, select the host. We can leave the default option as **Generate new host name based on data source name** for now and click on **Next** to continue.

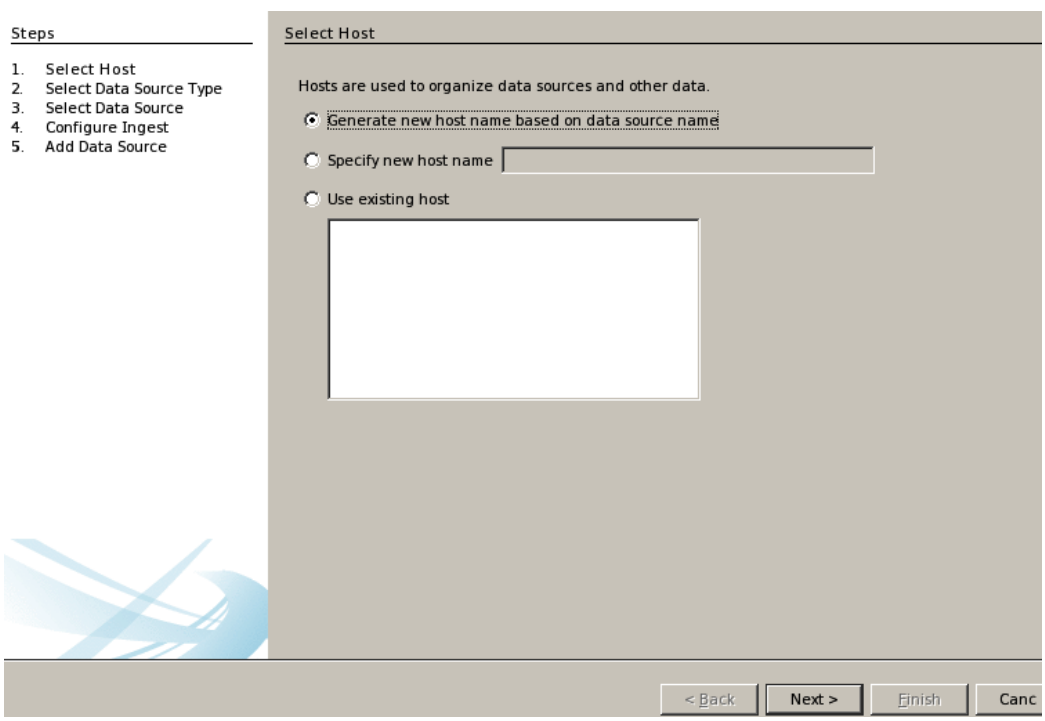


Figure 13.12 – Case host name

5. Our next step is an important one that we must pay attention to. Here, we select the data source type. Our sample file (`terry-work-usb-2009-12-11.E01`) is a forensically acquired evidence image file and so we will be choosing the first option, **Disk Image or VM File**. Once selected, click on **Next** to continue.

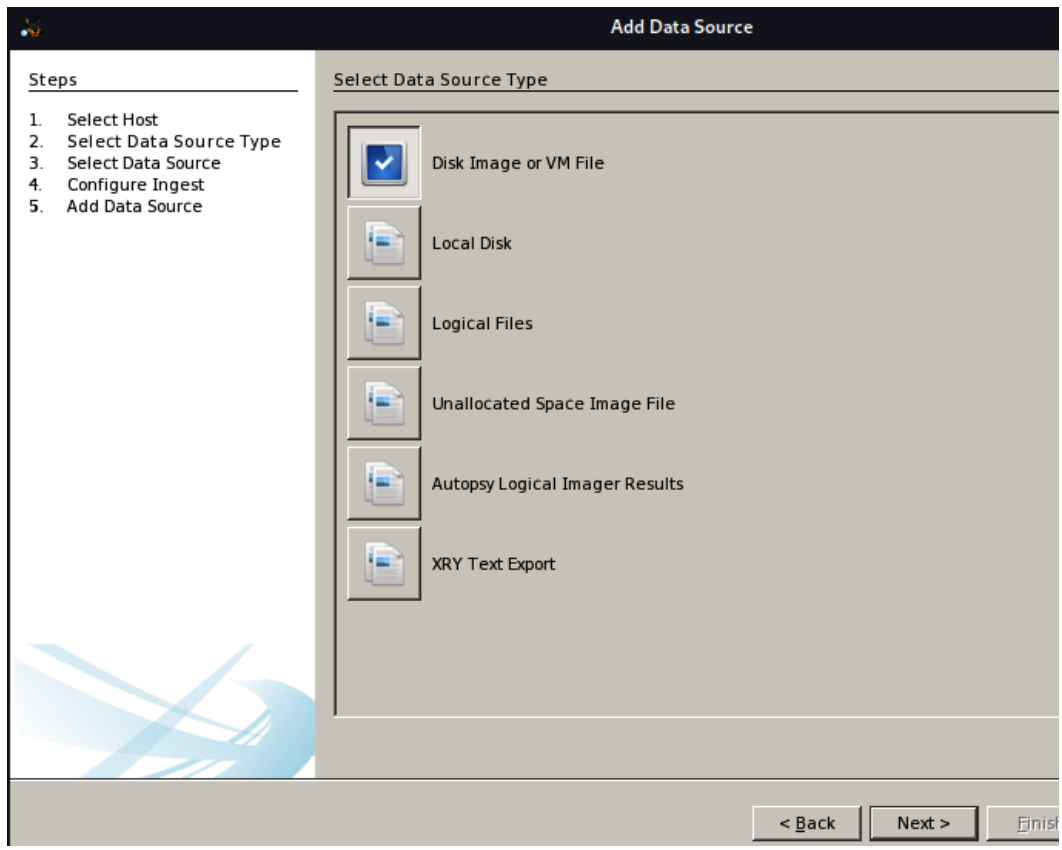


Figure 13.13 – Selecting a data source

6. We will now select the file to be analyzed. I have downloaded the sample evidence file to my `Downloads` folder in Kali Linux. You can click on the **Browse** button and navigate to the **C:\users\kali\Downloads** folder, and then select the evidence file (`terry-work-usb-2009-12-11.E01`) or type in the path of your file. Select your time zone and click **Next** to continue.

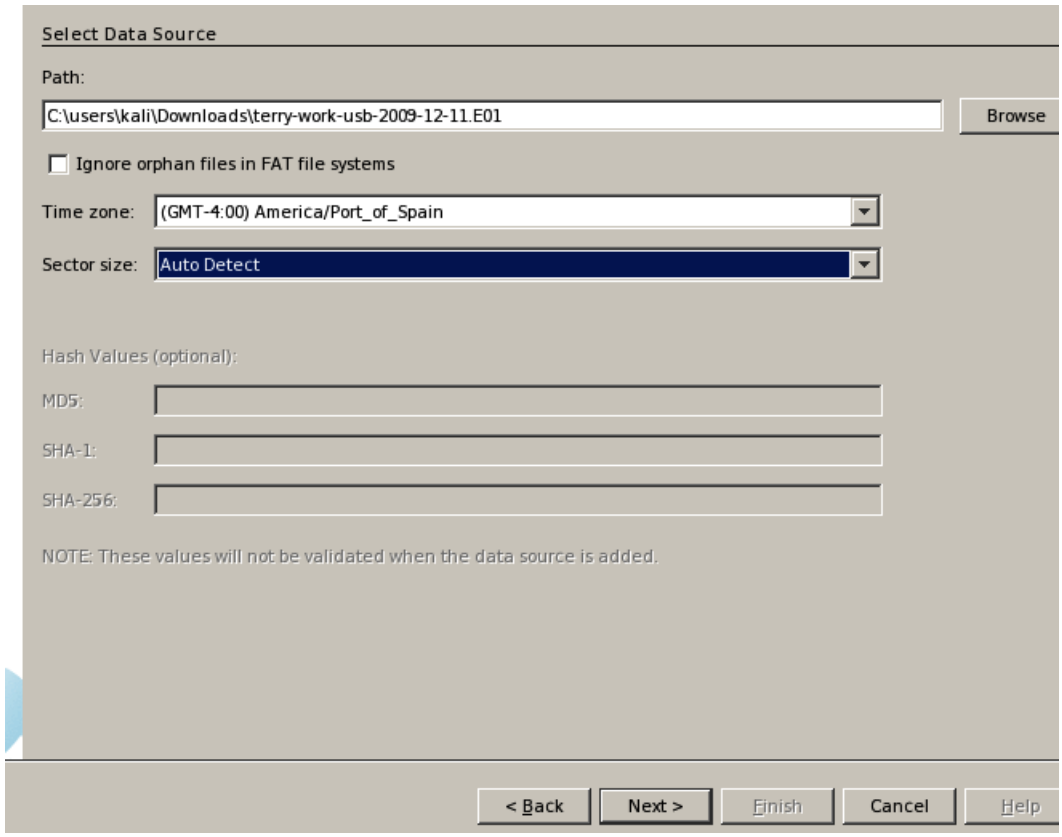


Figure 13.14 – Data source path specification

7. Next, we can configure the ingest modules that will determine the items on which Autopsy will run the selected modules to perform a fully automated DFIR analysis. I've left the default setting with all files and directories selected. Click on **Next** to continue.

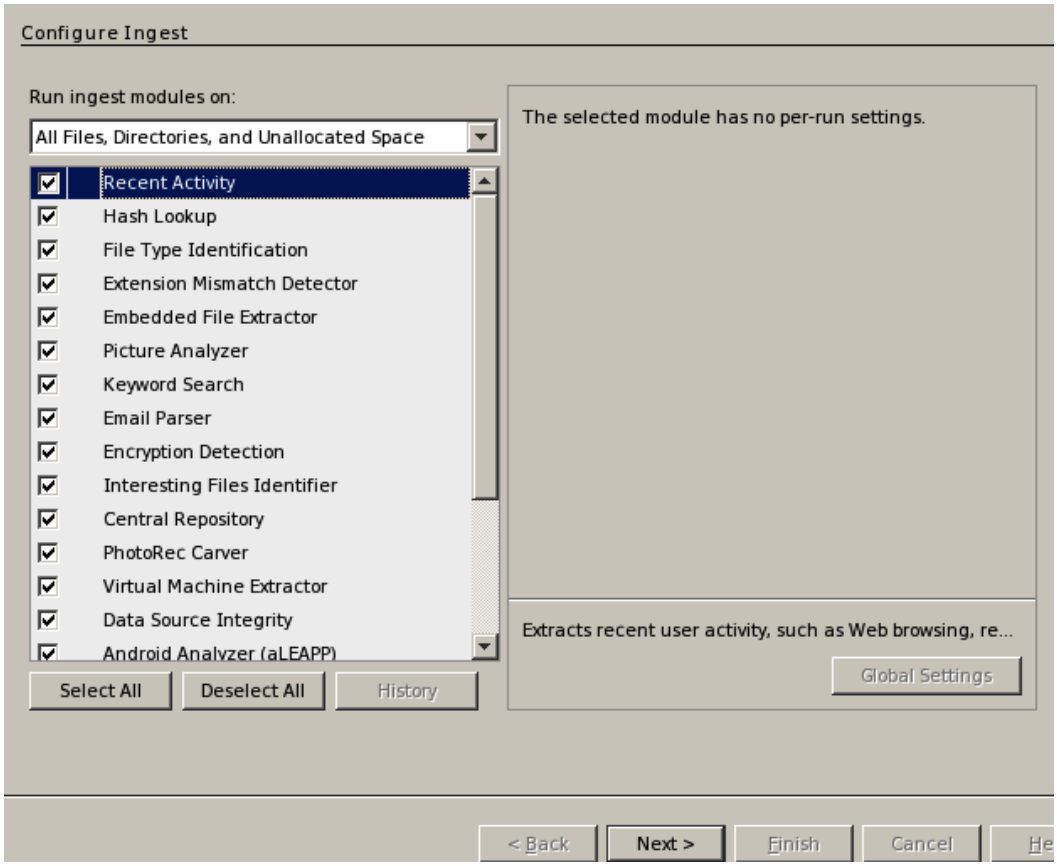


Figure 13.15 – Configuring ingest modules

Important note

On larger evidence files, disks, and images, the analysis process may be longer due to the large number of items being processed. Running all modules may not be possible. However, Autopsy will automatically cancel any modules that cannot be run if you do not manually specify them.

If Autopsy encounters any errors with modules, they will automatically be removed, as seen in the following screenshot. Click on **OK** to accept this and continue:

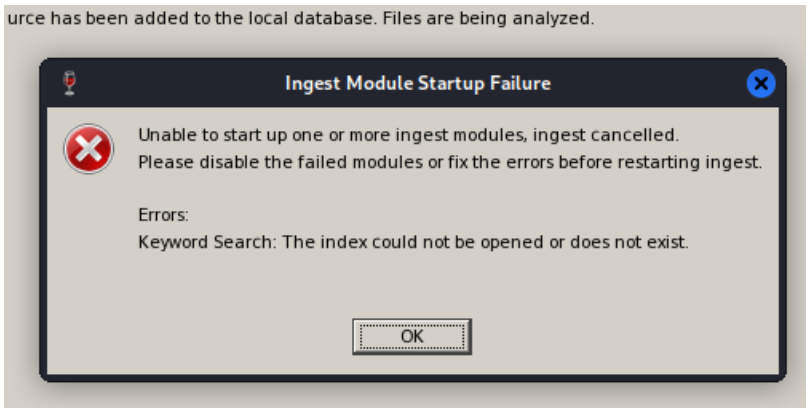


Figure 13.16 – Ingest module failure error

8. Once any error messages have been dismissed, click on the **Finish** button to begin analyzing the files:

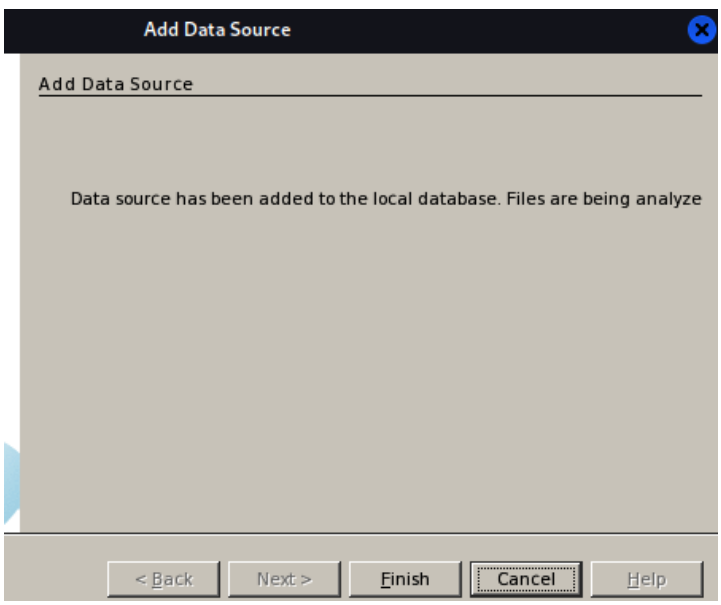


Figure 13.17 – Finalizing data source details for analysis

We can now view all analyzed files in the Autopsy interface, as seen in the following screenshot.

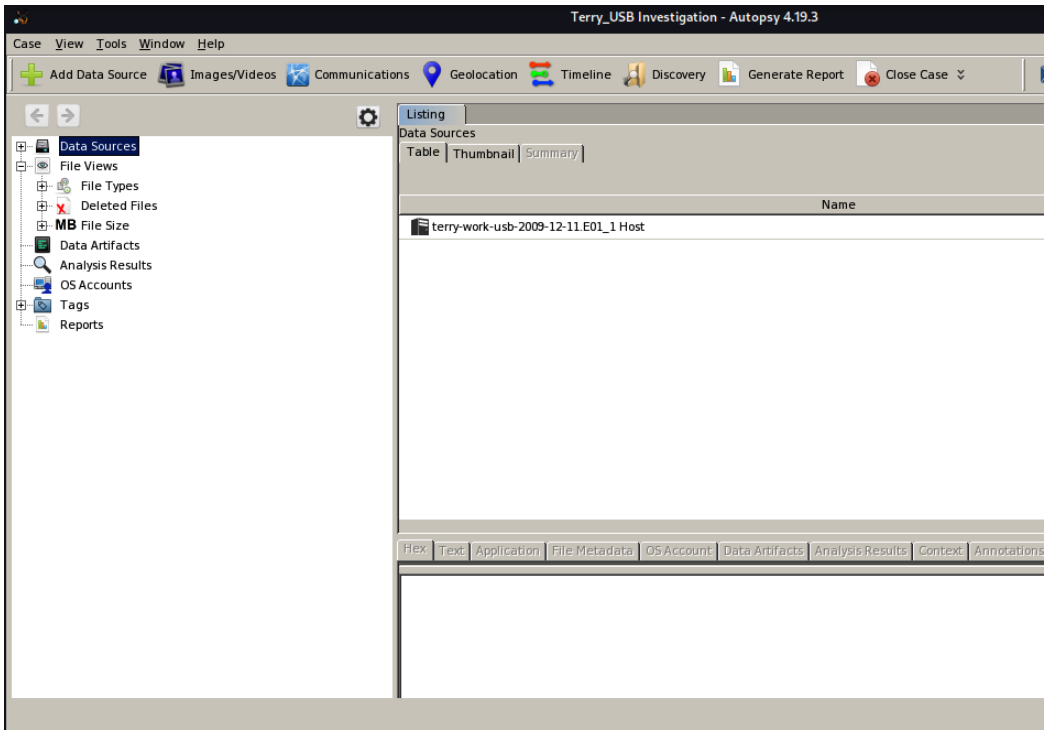


Figure 13.18 – Autopsy interface

Let's start exploring our findings in the Autopsy interface to view the results of the analysis.

Analyzing directories and recovering deleted files and artifacts with Autopsy 4

As previously mentioned, Autopsy has a very simple and uncomplicated interface. The Autopsy window is separated into three panes, as described here:

- The left pane shows the data source that was examined and analyzed and all directories and files discovered and recovered by Autopsy
- The main pane displays all the discovered files within the data source folders
- The lower pane displays file details such as **Hex**, **Text**, **File Metadata**, and **Data Artifacts**

In the following screenshot, I've expanded the **Data Sources**, **File Views**, and **Deleted Files** folders by clicking on the plus signs (+) next to each of the mentioned items, which all display sub-directories, as seen in the following screenshot:

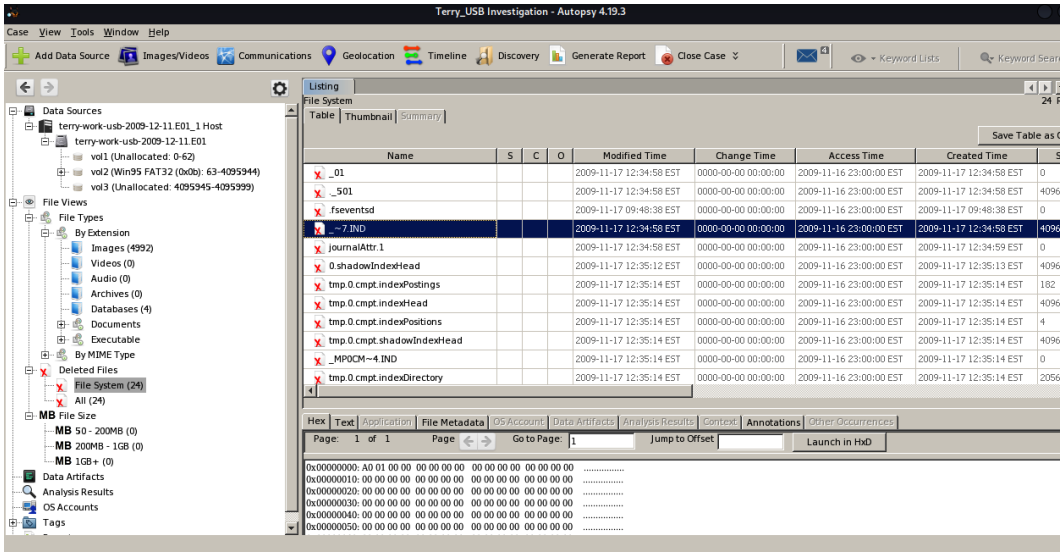


Figure 13.19 – Analyzed data source findings in Autopsy

By expanding the **Data Sources** item on the left pane, we can see that the evidence file has two volumes. Volume 1 is unallocated space and volume 2 is a Windows 95 FAT32 volume, which when expanded shows orphaned files, unallocated files, and even deleted folders, as seen in the following screenshot.

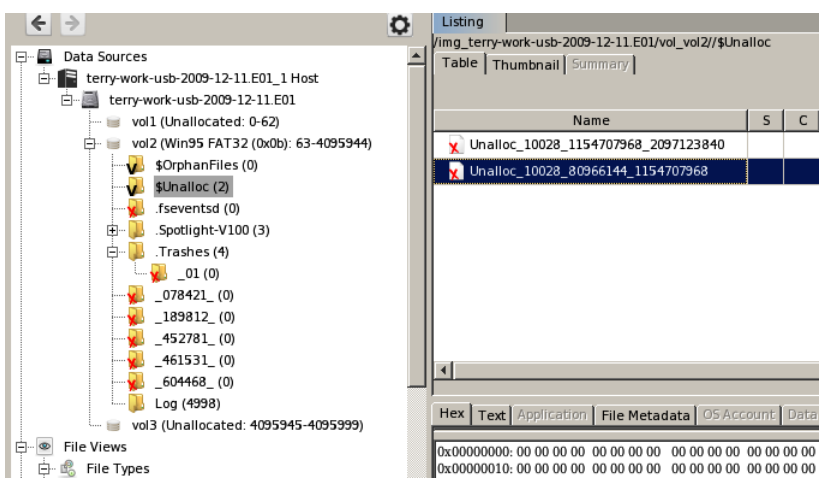


Figure 13.20 – Data Sources volumes and discovered artifacts

If we look a bit closer at **File Types**, we can see that Autopsy has categorized the file types into **Images**, **Videos**, **Audio**, **Archives**, and even **Documents** sub-folders, which may contain discovered artifacts.

In the following screenshot, we can see that Autopsy has discovered several artifact types, including 4,992 images, 4 databases, and 24 deleted files. I've clicked on the expanded **Images** folder containing 4,992 files. In the right pane, we can see the list of all images along with the times the image files were modified, accessed, and created.

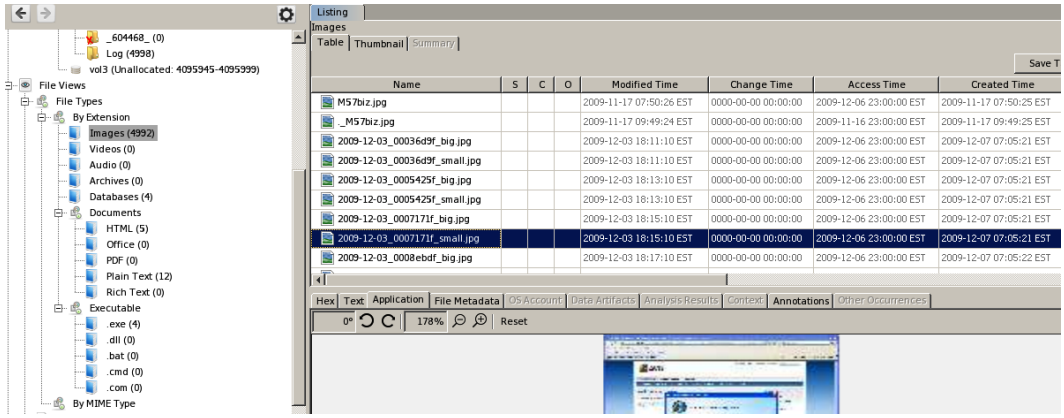


Figure 13.21 – Viewing discovered images and associated timestamps

Patience is key!

It may seem like a tedious task to go through all these files to discover any useful information or artifacts, but the key to successful DFIR investigation and analysis is patience!

Next up, I selected the **Deleted Files** folder, and then selected the **File System** folder. The contents of the **File System** folder are displayed in the main window with a red **X** next to each deleted file, as seen in *Figure 12.22*. I also scrolled down and clicked on the deleted `xpadvancedkeylogger.exe` file, which is reported as last modified on 2009-12-03 at 08:40 EST.

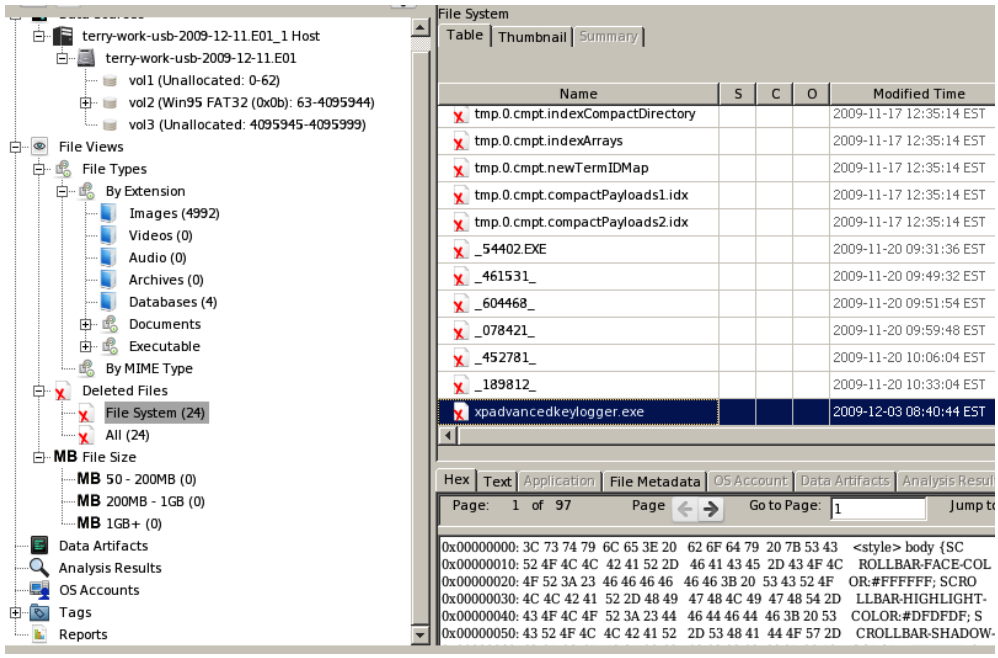


Figure 13.22 – Discovered and recovered artifacts

In the preceding screenshot, we can see how useful Autopsy can be when having to sift through deleted files as they are all grouped together and timestamped.

The final feature I'd like to show you in this very short tutorial is the **Timeline** feature. Clicking the **Timeline** button at the top of the application generates a timeline of events by month, as seen in the following screenshot.

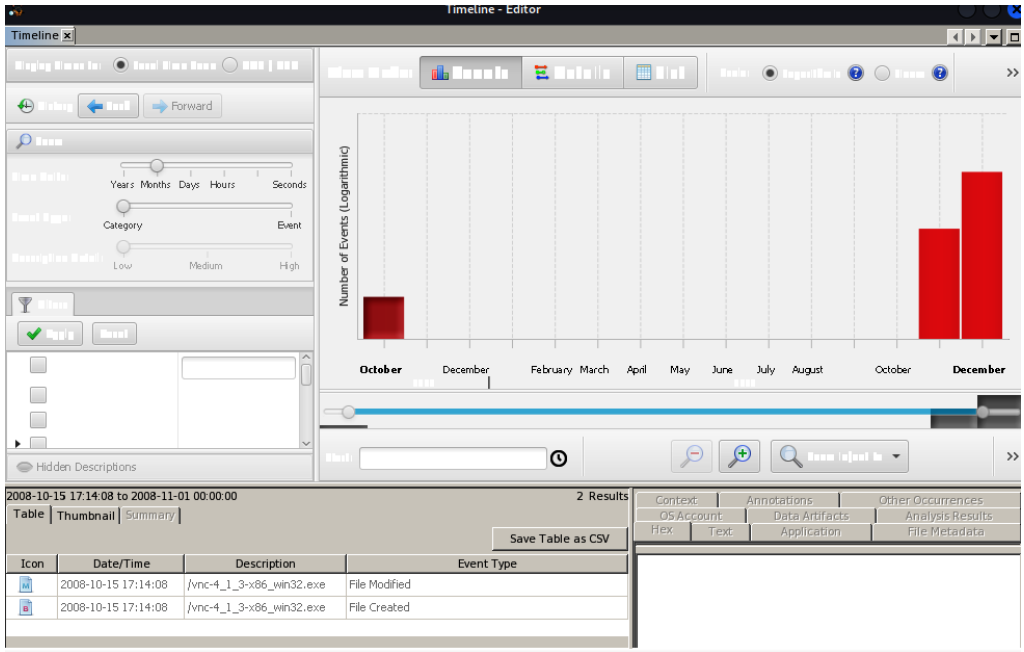


Figure 13.23 – Autopsy Timeline feature

In the preceding timeline, you can also see events by month. Let's click on the red **October** marker. In the pane below, we can see that there are two executable files that were created and modified on 2008-10-15 at 17:14. These files are `vnc` files that can be used to gain or give remote access and possibly compromise devices.

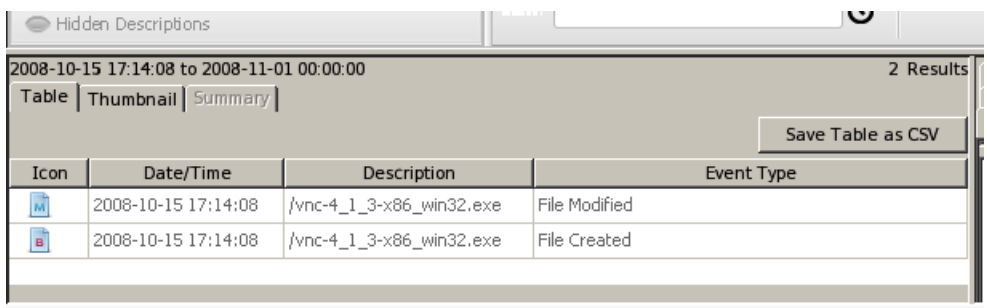


Figure 13.24 – VNC files within the Timeline feature

This brings us to the end of our automated DFIR analysis using Autopsy 4. I encourage you to return to this tool using the sample files mentioned at the beginning of the chapter and analyze the other downloaded sample files mentioned in the *Downloading sample files for automated analysis* section of this chapter.

Summary

In this chapter, we downloaded and installed the more powerful Windows version of Autopsy known as the Autopsy 4 GUI and were able to run it in Kali Linux using Wine. We also learned how simple it is to automatically analyze evidence files to find artifacts such as deleted files using this version of Autopsy. I hope you enjoyed this chapter as it showcases the much more powerful version of Autopsy when compared to the Autopsy browser, which we looked at in *Chapter 12*, *Autopsy Forensic Browser*.

Next, we will learn about some popular scanning and reconnaissance tools that, although not specifically created for DFIR purposes, are very useful in network forensics analysis and investigations. See you in *Chapter 14*, *Network Discovery Tools*!

Part 5: Network Forensic Analysis Tools

In the final part, we will use a plethora of installed and online tools to discover and examine network and online activities, including network host discovery and fingerprinting, IoT discovery and investigation, and packet capture analysis to examine network traffic.

This part has the following chapters:

- *Chapter 14*, *Network Discovery Tools*
- *Chapter 15*, *Packet Capture Analysis with Xplico*
- *Chapter 16*, *Network Forensic Analysis Tools*

14
Network Discovery Tools

In this chapter, I'd like to go over some common tools that aid in the network discovery and network information gathering and reconnaissance processes and, overall, aid in network forensic analysis. Specific network forensic analysis tools such as Xplico and NetworkMiner will be covered in *Chapter 15*, *Packet Capture Analysis with Xplico*, and *Chapter 16*, *Network Forensic Analysis Tools*, respectively.

In this chapter, we will first perform network host discovery by detailing specifics of the hosts themselves, such as **operating systems** (**OSes**) and open ports, and then use an online tool to discover external and publicly identifiable hosts.

In this chapter, we will learn about the following:

- Using netdiscover in Kali Linux to identify devices on a network
- Use **Network Mapper** (**Nmap**) in Kali Linux to identify devices on a network
- Use Nmap to fingerprint host details (ports, OSes, etc.)
- Use Shodan.io to find external IoT devices, including firewalls, CCTV systems, and servers

By the end of this chapter, you will be able to use these tools to assist in DFIR investigations that require you to collect information about locally and remotely connected devices, such as IP addresses, device types, OS details, open ports, and more.

Using netdiscover in Kali Linux to identify devices on a network

We'll start off our network host discovery by first using a very simple tool known as **netdiscover**. The netdiscover tool comes preinstalled in Kali Linux; however, you can install it on almost any version of Linux using the following command:

```
sudo apt install netdiscover
```

netdiscover, as the name suggests, is used to discover online hosts on a wired or wireless network by broadcasting **Address Resolution Protocol** (**ARP**) requests. What makes netdiscover easy to use is that it allows users to automatically scan an entire range or subnet of IP addresses on a network to detect online hosts. Another very useful feature of netdiscover is that it displays the **Media Access Control** (**MAC**) address along with the vendor of the device or **Network Interface Card** (**NIC**), which can make host discovery much simpler when trying to differentiate between end user, server, and networking devices.

Before we start using netdiscover, we should take note of the network interfaces on our scanning device. This also tells us what our IP, subnet mask, and network range are.

To do so, type the following:

```
ifconfig
```

In the following output in *Figure 14.1*, we can see that there are two interfaces, namely the Ethernet (`eth0`) and loopback (`lo`) interfaces:

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.170.105  netmask 255.255.0.0  broadcast 10.10.255.255
        inet6 fe80::71ce:b8aa:c039:d0bd  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:22:46:4f  txqueuelen 1000  (Ethernet)
        RX packets 3159246  bytes 1121334766 (1.0 GiB)
        RX errors 0  dropped 16  overruns 0  frame 0
        TX packets 325325  bytes 21899177 (20.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 108  bytes 6010 (5.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 108  bytes 6010 (5.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 14.1 – ifconfig command output

In the `eth0` interface details, we can see that my IP address is listed as `10.10.170.105`, with a subnet mask of `255.255.0.0`. We will use this interface with the `netdiscover` command.

Now that we know which interface and IP range we will use, we can proceed to use netdiscover to identify network hosts:

1. We will first start netdiscover and use the help option to view all available options and switches in netdiscover. To do so, type the following:

```
sudo netdiscover -h
```

   To use netdiscover to automatically scan for devices on our network, we will use the `-r` option to specify the subnet or range of IP addresses.

```
┌──(kali㉿kali)-[~]
└─$ sudo netdiscover -h
[sudo] password for kali:
Netdiscover 0.9 [Active/passive ARP reconnaissance tool]
Written by: Jaime Penalba <jpenalbae@gmail.com>

Usage: netdiscover [-i device] [-r range | -l file | -p] [-m file] [-F filter] [-s time]
 [-c count] [-n node] [-dfPLNS]
  -i device: your network device
  -r range: scan a given range instead of auto scan. 192.168.6.0/24,/16,/8
  -l file: scan the list of ranges contained into the given file
  -p passive mode: do not send anything, only sniff
  -m file: scan a list of known MACs and host names
  -F filter: customize pcap filter expression (default: "arp")
  -s time: time to sleep between each ARP request (milliseconds)
  -c count: number of times to send each ARP request (for nets with packet loss)
  -n node: last source IP octet used for scanning (from 2 to 253)
  -d ignore home config files for autoscan and fast mode
  -f enable fastmode scan, saves a lot of time, recommended for auto
  -P print results in a format suitable for parsing by another program and stop after ac
```

Figure 14.2 – netdiscover options

2. I'll now scan my entire `10.10.0.0` network range using the following command:

```
netdiscover -i eth0 -r 10.10.0.0/16
```

> **Note**
> Please ensure that you are using your own network range, as yours may be different from mine.

3. Now, press *Enter*. After pressing *Enter*, netdiscover will run. Allow it a minute to complete and compile a list of IPs, MAC addresses, and vendor/hostnames, as shown in the following screenshot:

> **Note**
> For privacy purposes, I have blocked parts of the MAC addresses.

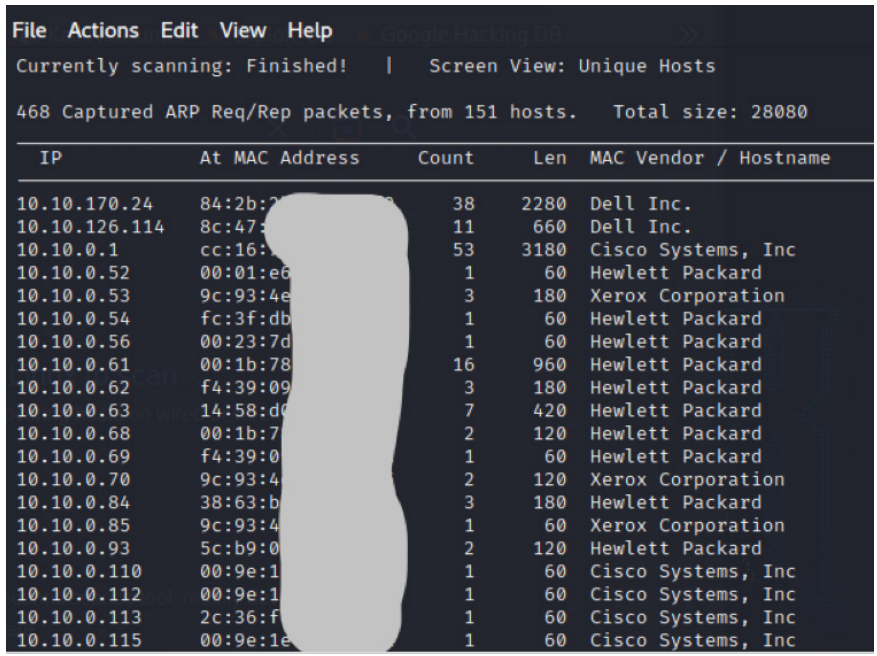


Figure 14.3 – netdiscover output

In the preceding output, we can see that netdiscover has captured `468` ARP packets and detected `151` live hosts, listed with their IPs, MAC addresses, and vendor/hostnames.

Let's now look at using Nmap to also perform host discovery.

Using Nmap to find additional hosts and devices on a network

Created by Gordon Lyon aka *Fyodor*, **Nmap** is one of the most common tools for network scanning and enumeration during vulnerability assessments and penetration tests. It is also quite useful for DFIR tasks and investigations when investigating network communications that require disclosure of host details on a network.

Nmap is also a bit of a celebrity and was used in popular movies such as *The Matrix Reloaded*, *The Bourne Ultimatum*, *Die Hard 4*, and *The Girl with the Dragon Tattoo*, and also the very popular TV series *Mr. Robot*.

Nmap is a **Command-Line Interface** (**CLI**) tool and comes pre-installed in Kali Linux. There is also a **Graphical User Interface** (**GUI**) for Windows and Mac devices, but I personally think you can access more options and features using the CLI.

Nmap has many switches and requires prior knowledge of ports, **Transmission Control Protocol/Internet Protocol** (**TCP/IP**) protocols, and TCP flags, all of which you should read more about before proceeding if you are unfamiliar with those topics. Please visit this link for more information on TCP: `https://en.wikipedia.org/wiki/Transmission_Control_Protocol`.

Let's follow these steps to use Nmap to find additional hosts and devices on a network:

1. To start nmap and view the available switches and options, we can use the following command:

```
sudo nmap -h
```

   As stated in the preceding output, the general format and usage of nmap is `nmap [Scan Type(s)] [Options] {target specification}`.

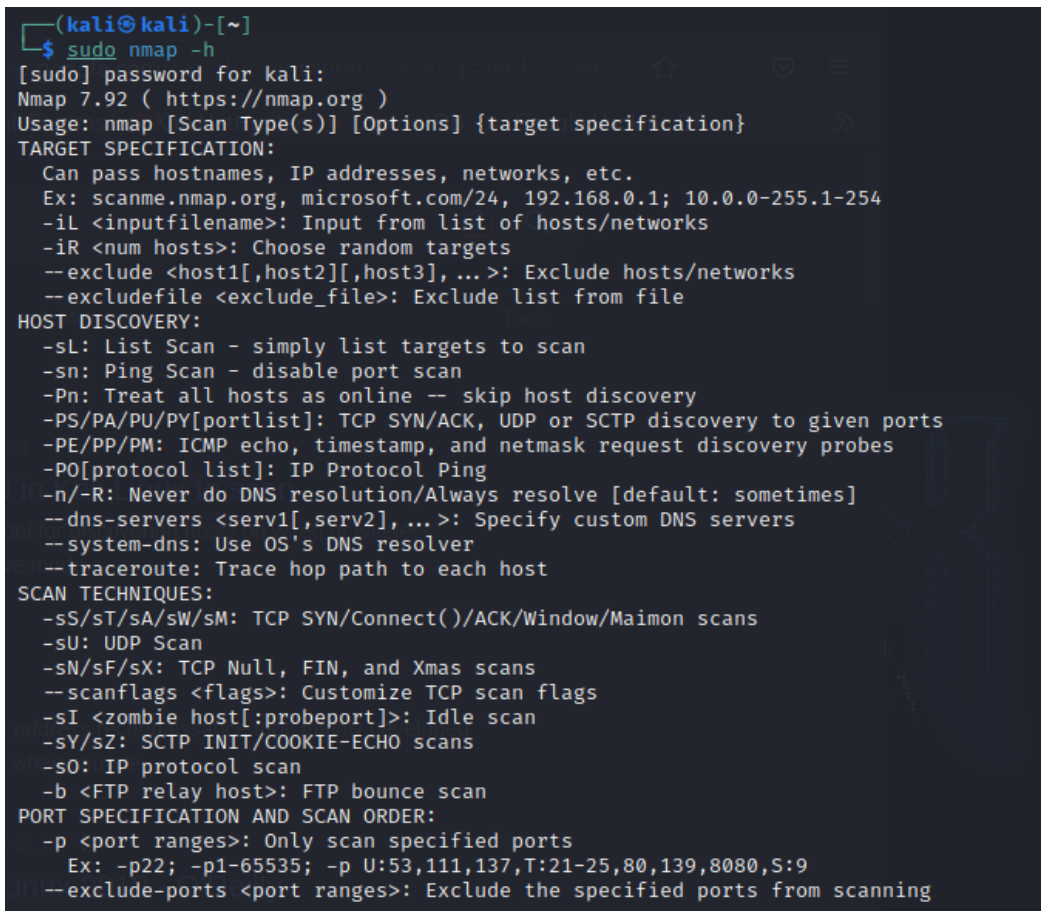


Figure 14.4 – A snippet of the nmap help option output

2. To start us off, I'll try a basic ping scan using the `-sn` option to list hosts that can be further scanned, in the same way that netdiscover found online hosts on my network in the previous section.

   To perform a basic ping scan, we can type `sudo nmap -sn` followed by the network range, which in my case is `10.10.0.0/24`. Yours may be different, so please make sure you use your own network range.

3. We can also use the `-v` option to list output verbatim as it happens in nmap:

```
sudo nmap -sn -v 10.10.0.0/24
```

   The following screenshot shows the output of the specific preceding Nmap command.

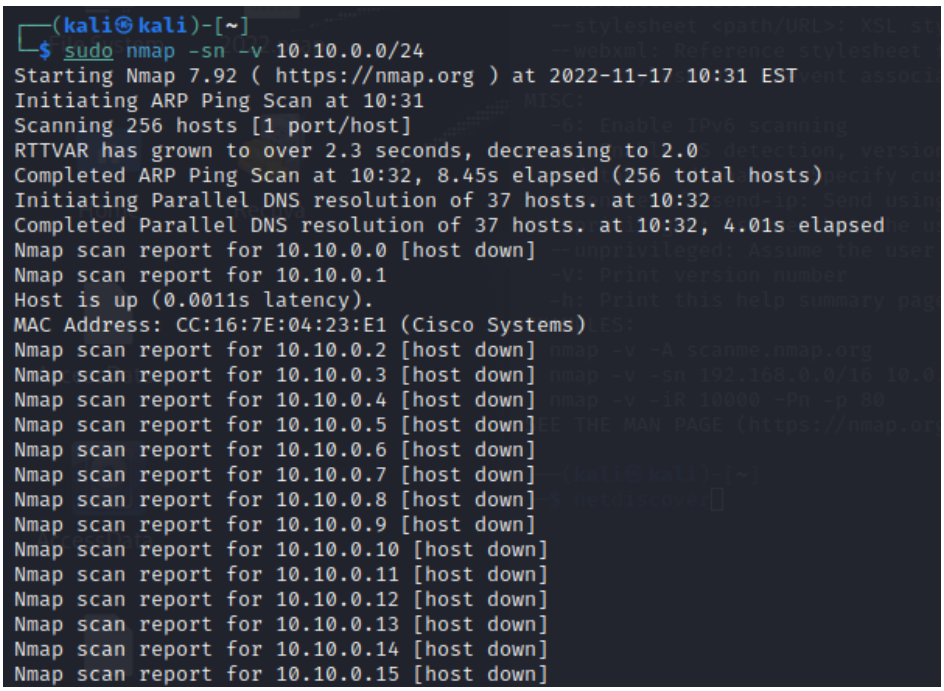


Figure 14.5 – nmap ping scan output

If we scroll down through the nmap output, we can see listed hosts that were discovered online via our ping scan. Though not as well organized as netdiscover, nmap still displays the same results, as shown here.

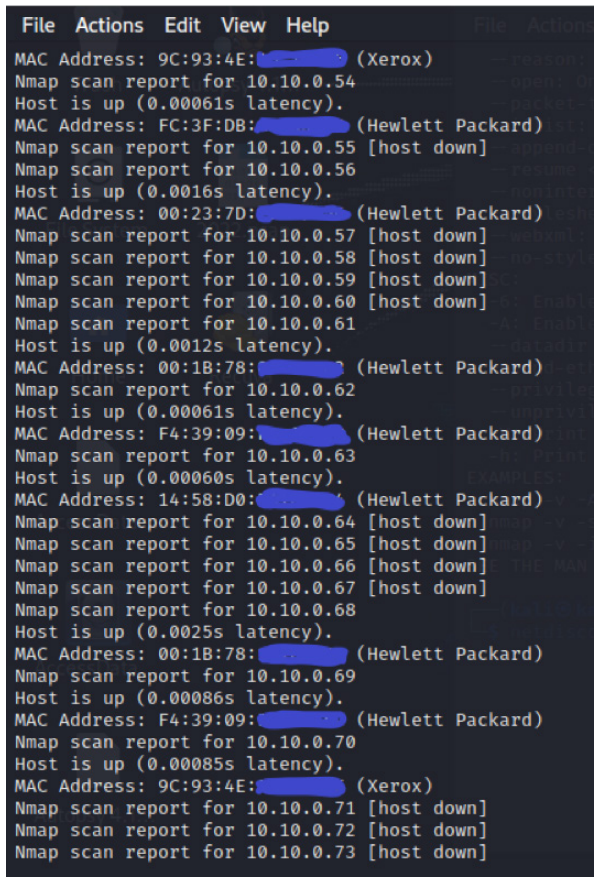


Figure 14.6 – nmap output showing discovered live hosts

In this section, we learned the basics of Nmap, and we are now capable of scanning internal networks using this CLI tool.

Let's go a bit further with our Nmap scanning now and learn how to identify ports and services of discovered devices.

Using Nmap to fingerprint host details

Now that we're familiar with how to view nmap switches and run a basic scan, let's perform host enumeration by discovering running services, ports, and even the OSes of discovered hosts, which may be useful artifacts within our DFIR investigation.

As shown in the previous section, nmap is fairly simple to use once you are aware of the various switches that can be used. Feel free to again run the `sudo nmap -h` command to view the available switches and options.

For our purposes, an `-A` option in nmap can perform the following tasks when enumerating a host or an entire network:

- OS detection
- Service version detection
- Script scanning
- Traceroute

The `-A` option can take a while to run on an entire network, so I'll run it against a single host, which was shown as alive when I previously ran the ping scan on my network.

To run this service scan as it is called, I'll run the following command:

```
sudo nmap -A -v 10.10.10.10
```

The following screenshot shows the output of the `-A` option within Nmap.

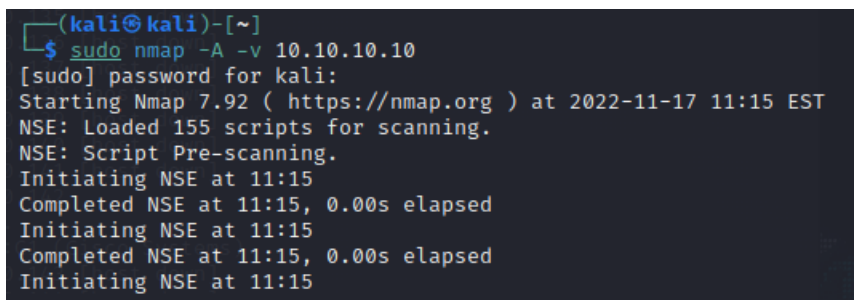


Figure 14.7 – nmap service scan

Scrolling through the service scan output, nmap has also displayed a list of open ports and has identified the running service and service versions, as shown here.

```
PORT     STATE SERVICE            VERSION
135/tcp  open  msrpc              Microsoft Windows RPC
139/tcp  open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds       Windows 10 Education 10586 microsoft-ds
3389/tcp open  ssl/ms-wbt-server?
```

Figure 14.8 – nmap output displaying open ports and running services

Scrolling a bit further down through the very same output, nmap displays details of the host, such as the host (target) name, domain, and computer name, as shown here. For privacy purposes, I've blurred some of the output.

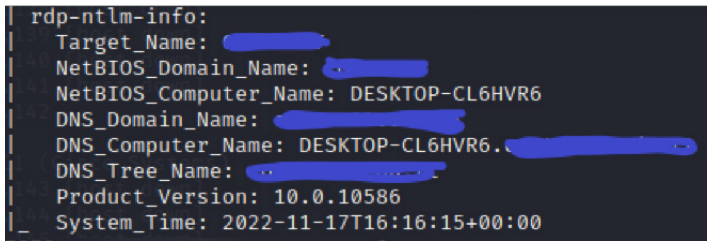


Figure 14.9 – nmap output displaying host information

Scrolling almost to the very bottom of the nmap output, I can also see the host OS details, as shown here:

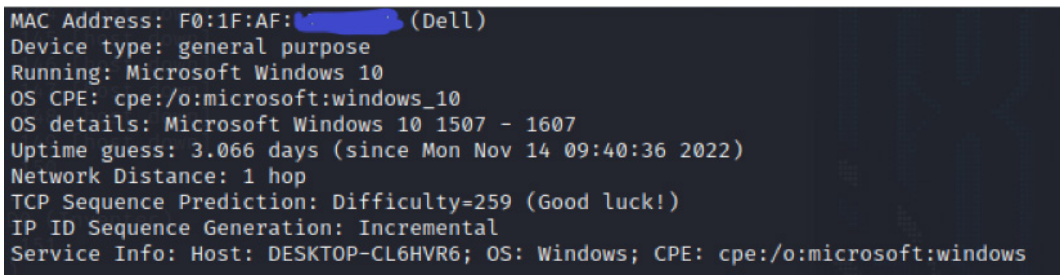


Figure 14.10 – nmap output showing the host OS details

In the preceding output, we can see that the host is a Dell device running Windows 10 (`1507 - 1607`).

We now know how to scan internal networks and identify live systems and details about those systems, such as their OS, MAC address, uptime, and even open ports, all using just Nmap.

Let's now move outside the local network and learn how to locate and identify external and IoT devices using Shodan.

Using Shodan.io to find IoT devices including firewalls, CCTV, and servers

Quickly think of a website that allows you to search for anything. Without being a psychic, I'm guessing you thought of Google. Maybe I'll stick to my day job. In the same way that Google crawls and scours the main parts of the internet and indexes websites and resources, Shodan takes a similar approach to scanning the internet and IP addresses and indexes useful information about devices that can be accessed across the web.

IoT (short for **Internet of Things**) devices are those that can connect to the internet but may not be traditional computers, such as laptops and desktops. IoT devices include smart devices, such as wearable fitness devices and trackers, smart microwaves and ovens, Amazon Echo, doorbell cameras, and even smart cars. IoT devices also extend to industrial systems and are referred to as **Industrial IoT** (**IioT**) devices. Think of a large soft drink manufacturing company as an example. There need to be systems and devices in place to count bottles, caps, and labels, check temperatures, and facilitate communications between sensors that need to be networked with each other via **Programmable Logic Controllers** (**PLCs**), **Supervisory Control and Data Acquisition** (**SCADA**), and **Industrial Control Systems** (**ICS**). This applies to every industrial sector, even nuclear power plants! It's also not uncommon to combine this **Operational Technology** (**OT**) with IioT, known as **OT-IIoT**, as OT devices can be remotely controlled via IIoT systems.

The issue here lies in securing IoT and IIoT devices as well as access, particularly remote access, to devices. As a penetration tester, I use **Shodan.io** frequently to find systems belonging to companies of different types, particularly larger organizations. This skill is also useful when conducting DFIR investigations, as information about external and IoT devices can easily be found using Shodan.io.

The team at Shodan claims that Shodan can find all devices that are connected to the internet and reveal important metadata about the devices. This can be beneficial because it allows companies to use Shodan to find out whether their devices can be publicly accessed, but it also shows attackers the same information. This is not to say that Shodan is a bad or malicious tool, as there are hundreds of other tools that can be used together to divulge the same information. Shodan just makes it much easier.

Shodan is a paid subscription; however, you can use it for free, which gives very limited results, or register a free account and gain access to two pages of results when performing IoT and IIoT searches. If you don't want to purchase one of their subscription packages, you can wait around for their annual sale (usually around Black Friday and Thanksgiving), when a limited lifetime subscription for personal use is available for $5. For this exercise, I'll be using the paid $5 subscription, but I encourage you to register with the site so that you at least have access to two pages of results at a time.

Using Shodan filters for IoT searches

Shodan searches aren't as clear-cut as a Google search, but it's not difficult to do a search for Shodan filters and try to search for specific devices. You can view the Shodan Filter Reference at `https://beta.shodan.io/search/filters`.

Here are some basic search filters you can use:

- `product`: Search the name of the hardware manufacturer or software developer
- `hostname`: Find devices based on the searched hostname
- `os`: Search for devices by OS
- `port`: Find open ports on devices

- `city`: Find devices within a specific city
- `country`: Find devices within a specific country
- `geo`: Find devices by GPS coordinates

Let's try a few searches of our own. I live on a small island in the Caribbean called Trinidad. If want to view a list of devices within Trinidad, I can use the `country` filter followed by the country code in the Shodan search bar:

```
country:"TT"
```

If unsure of your own country code, a simple web search should help you figure it out. A site that I have found to be helpful is `https://www.countrycode.org/`.

In the following screenshot, we can see the results of the country filter where `TT` represents the country code for Trinidad and Tobago. To the left of the screen, we can see a total of **124, 921** results for IoT devices found, under which there is a list of **TOP CITIES**, with the number of devices listed for each city, followed by the top open ports, top organizations, and so on. To the middle and right of the screen, we can see actual IPs and information about those devices.

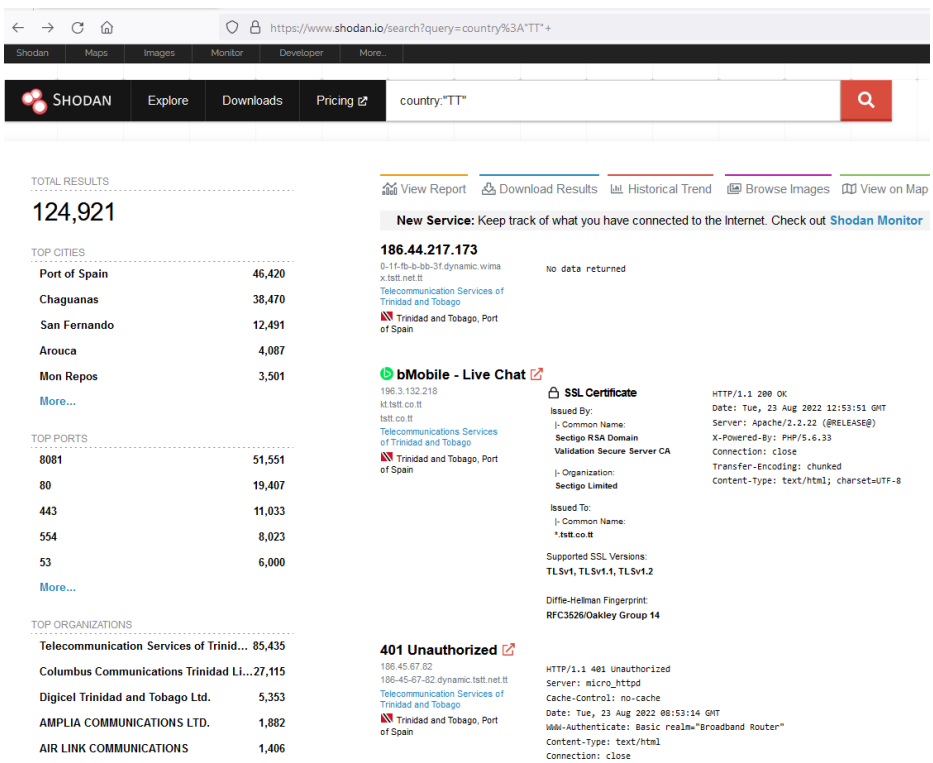


Figure 14.11 – Shodan country filter results

Due to privacy and legal concerns, I won't be clicking on the results, as this will carry us to another page, showing the company and host details and even disclosing open ports and the OSes of the devices. Be sure to do a web search and review the laws of your country and state before proceeding any further. If unsure, you may wish to first reach out to law enforcement or someone knowledgeable about your local laws and legislation.

Let's try another search for `Fortinet` firewalls within Trinidad, using the `country` filter and a `product` filter by typing the following:

```
country:"TT" product:"Fortinet"
```

In the following search results, we can see that Shodan returned 54 results for Fortinet firewalls in Trinidad. Again, for legal and privacy purposes, I won't be clicking on the IP address, as this will show even more details about the devices other than what is listed on the main page.

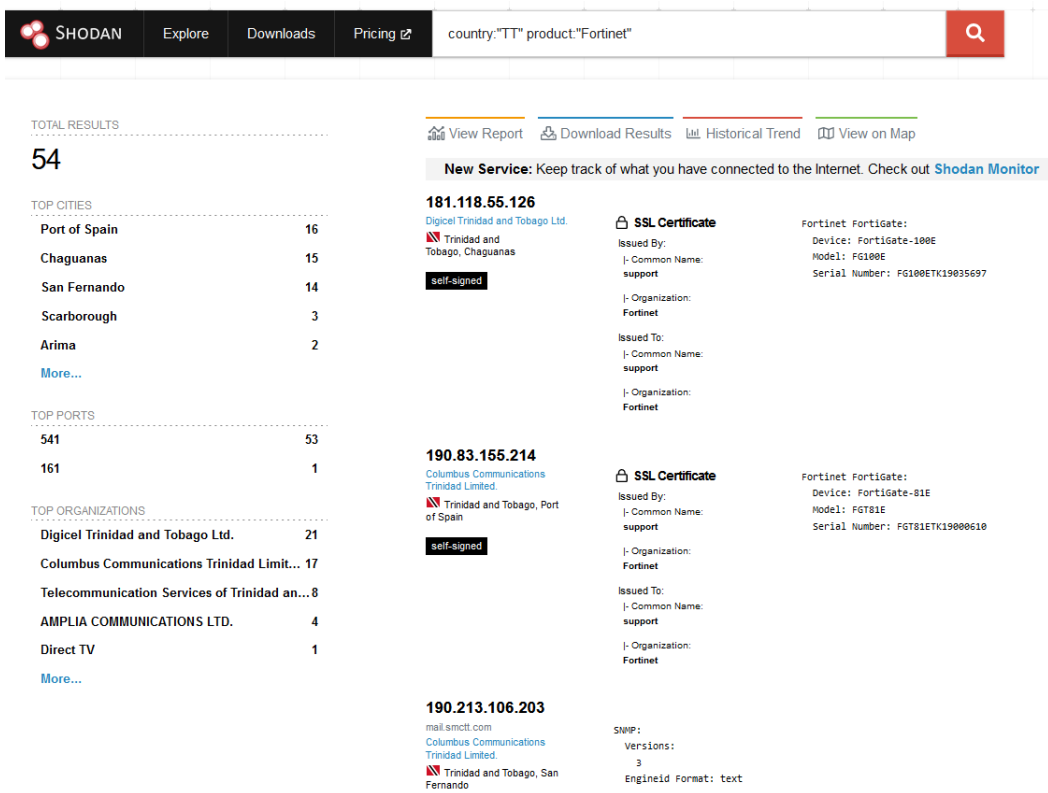


Figure 14.12 – The product filter for Fortinet firewalls in Trinidad

We can also look for specific servers using Shodan. Let's say that I want to search for Apache web servers that use version `2.2.3`. This can be achieved using the `server` filter followed by the server version by typing the following:

```
country:"TT" apache 2.2.3
```

Here, we see that Shodan has displayed 11 results, mainly found in Port of Spain and Chaguanas in Trinidad.

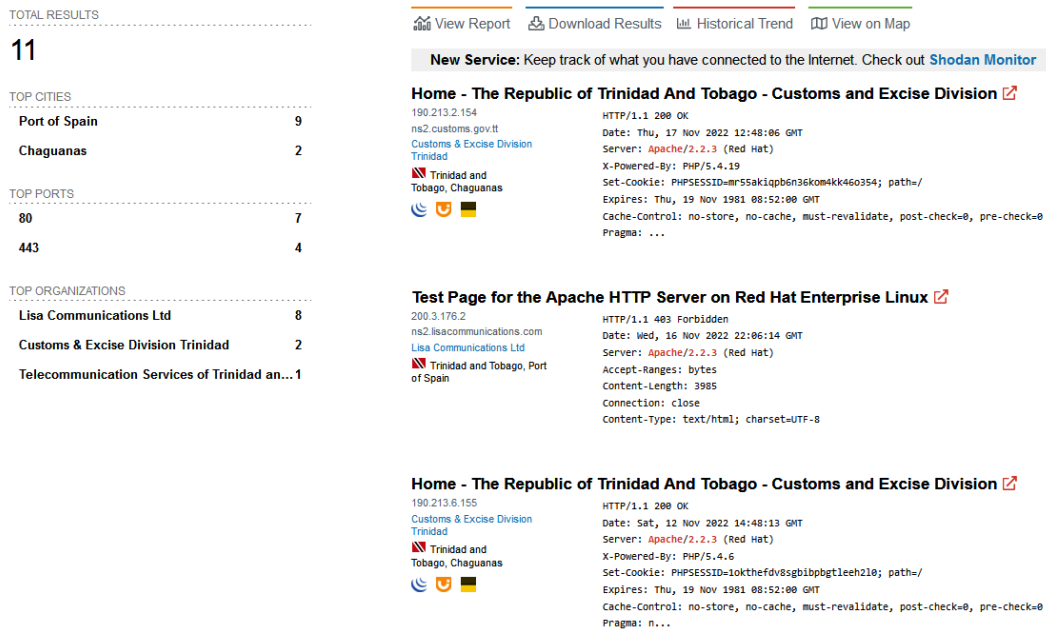


Figure 14.13 – Finding web servers with Shodan

For our last search, I'd like to search for CCTV IP cameras using Shodan. For this, we can again use the `product` filter. I'll specify the `Hikvision` camera system, as this is a very common CCTV system, by typing the following:

```
country:"TT" product:"Hikvision IP Camera"
```

As seen in this final instance, Shodan has returned 7,689 instances of Hikvision cameras. This can also be very useful if your DFIR investigation requires CCTV camera footage by alerting you to where cameras may be located nearby, and it may give a company name that you may be able to contact and ask permission to view their footage.

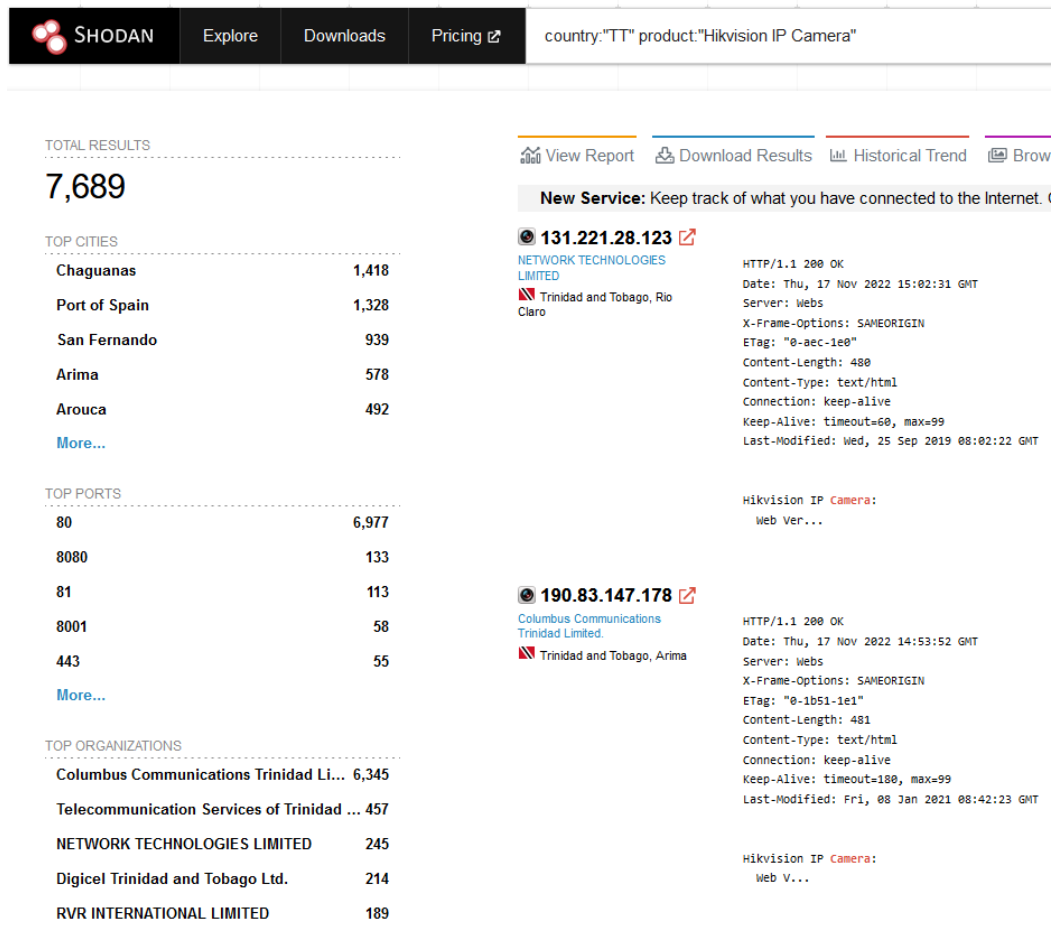


Figure 14.14 – CCTV camera discovery using Shodan

Feel free to try out more Shodan filters, but please check your local legislation where applicable to ensure that these searches are legal.

Summary

In this chapter, we looked at the networking and enumeration tools netdiscover and nmap and learned how to scan networks for open hosts and view details such as IPs, MAC addresses, and hostnames and went a bit further with Nmap to further discover more host details, such as open ports, running services and their versions, computer names, and domains. We then moved on to finding IoT devices using Shodan.io and used various search filters to find firewalls, servers, and CCTV cameras.

These tools can be very useful to anyone gathering information on local and remote devices that may be part of a network DFIR investigation.

Next up, we'll look at a **Network Forensics Analysis Tool** (**NFAT**) called **Xplico**. See you in the next chapter.

15
Packet Capture Analysis with Xplico

In this chapter, we'll look into Xplico, which is an automated **Network Forensic Analysis Tool** (**NFAT**). Xplico can be found in Kali Linux; however, I've found that within the last few releases (2019–2022), there are issues when trying to run Xplico, possibly due to upgrades within Kali itself.

Although I will explain how to start Xplico in Kali Linux for those who may have the good fortune of running it without issues, we will be using Xplico within a virtual machine running DEFT Linux 8.1, for those of us who may have difficulties running Xplico within Kali itself.

We will be covering the following main topics in this chapter:

- Installing Xplico in Kali Linux
- Installing DEFT Linux 8.1 in VirtualBox
- Downloading sample analysis files
- Starting Xplico in DEFT Linux
- Using Xplico to automatically analyze web, email, and voice traffic

Installing Xplico in Kali Linux

Follow these steps to install Xplico in Kali Linux:

1. If Xplico does not come with the version of Kali Linux you are currently using, you can first update Kali Linux and install the Kali forensics repository by typing the following command:

```
sudo apt update && sudo install kali-linux-forensics
```

2. Next, update your Kali by typing the following command:

```
sudo apt update
```

3. Once Kali has updated, install Xplico in Kali by typing the following command:

```
sudo apt-get install xplico
```

4. Press *Y* when prompted to click on **Yes** and download and install the necessary files.

   Then, we need to start the Apache and Xplico services.

5. Start the `apache2` service by typing the following command:

```
sudo apache2 start
```

6. Start the Xplico service by typing the following command:

```
sudo xplico start
```

7. To run Xplico with full administrative privileges as the root user, type the following command:

```
sudo /etc/init.d/xplico start
```

8. Type `Xplico` in the search bar and click on the first Xplico option:

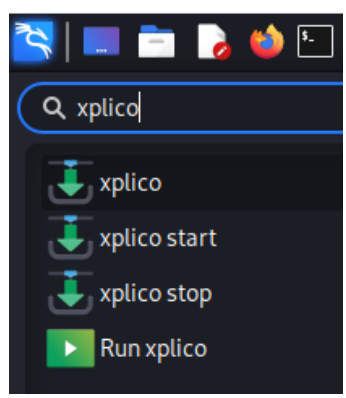


Figure 15.1 – Xplico options within the search menu

9. The Xplico interface will open in a browser. You can also do this manually by entering `http://localhost:9876/` in the browser of your choice.

If the preceding steps do not work, you can use Xplico in DEFT Linux, which we will cover in the next section.

Installing DEFT Linux 8.1 in VirtualBox

Installing DEFT Linux is very simple and only takes a few minutes. We'll have Xplico up and running in no time. For those of you who were able to start Xplico in Kali Linux, you can skip this process and go straight to the *Downloading sample analysis files* section, where we will download and begin analyzing our sample files using Xplico.

To those of you opting to install DEFT so you are able to use Xplico, let's get to it without further delay:

1. DEFT is very user-friendly and based on a lightweight version of Ubuntu called Lubuntu. You can download the DEFT Linux ISO image file at `https://archive.org/details/deft-8.2`.
2. Once downloaded, open VirtualBox and click on the **New** option to create a new virtual machine, as shown here.

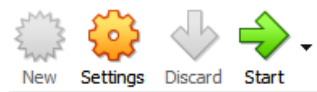


Figure 15.2 – The New option in VirtualBox

3. Enter the details in the relevant fields, as shown in the following screenshot:

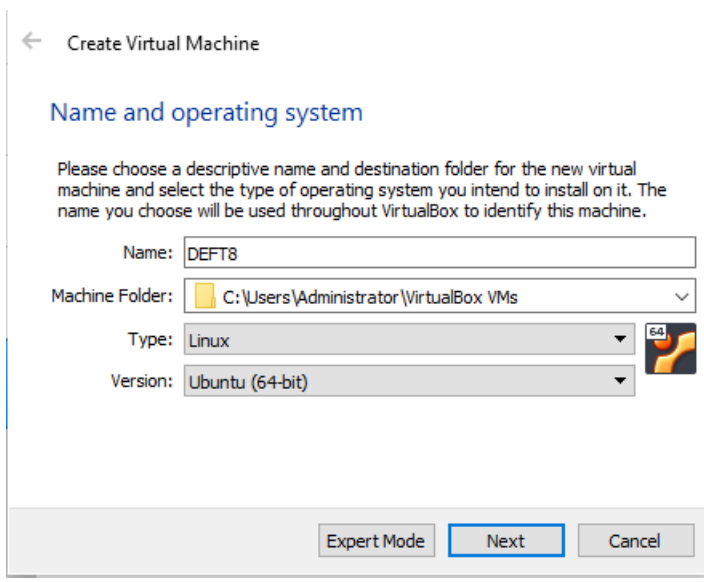


Figure 15.3 – New virtual machine details

I've allocated 6 GB of memory for the DEFT VM.

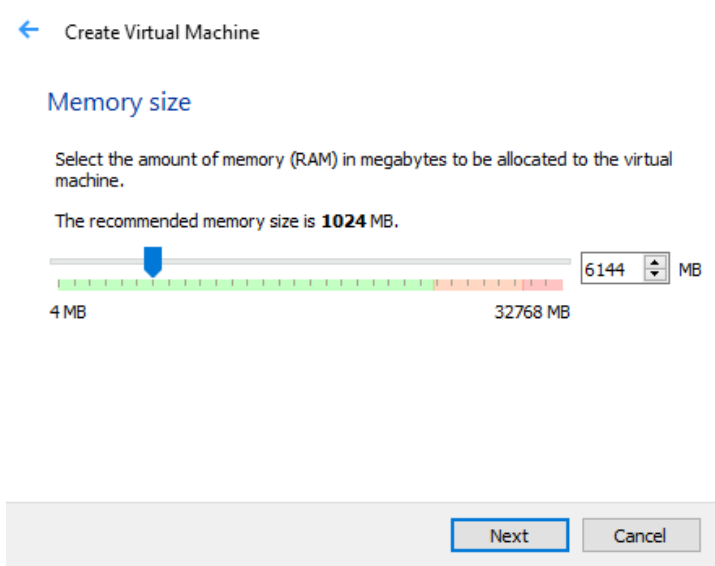


Figure 15.4 – VirtualBox memory allocation

4. For the **Hard Disk** selection, choose the **Create virtual hard disk now** option and click on **Create**.
5. For the **Hard disk file** type selection, choose the **VDI** (**VirtualBox Disk Image**) option and click on **Next**.
6. For the **Storage on physical hard disk** selection, choose the **Dynamically allocated** option, click on **Next**, and then click on **Create**.
7. Next, click on the **DEFT Linux** entry in the VirtualBox manager window and click on the green **Start** button:

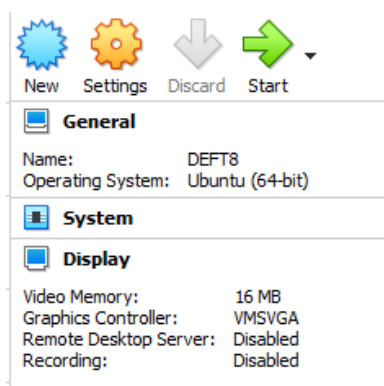


Figure 15.5 – DEFT details in the VirtualBox manager window

8. Next, we select the DEFT ISO image by clicking on the yellow folder icon in the bottom-right corner of the **Select start-up disk** window.

> **Note**
> Don't click on **Start** just yet.

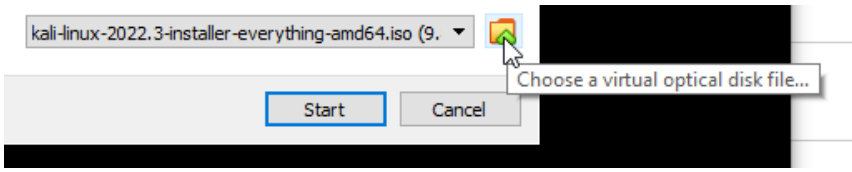


Figure 15.6 – Virtual optical disk file selection

9. Then, click on the blue **Add** button.

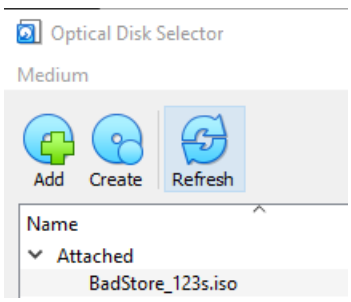


Figure 15.7 – Optical disk selection

10. Now, browse to the folder where you downloaded the `deft-8.2.iso` file, click on the file, and then click on **Open**.

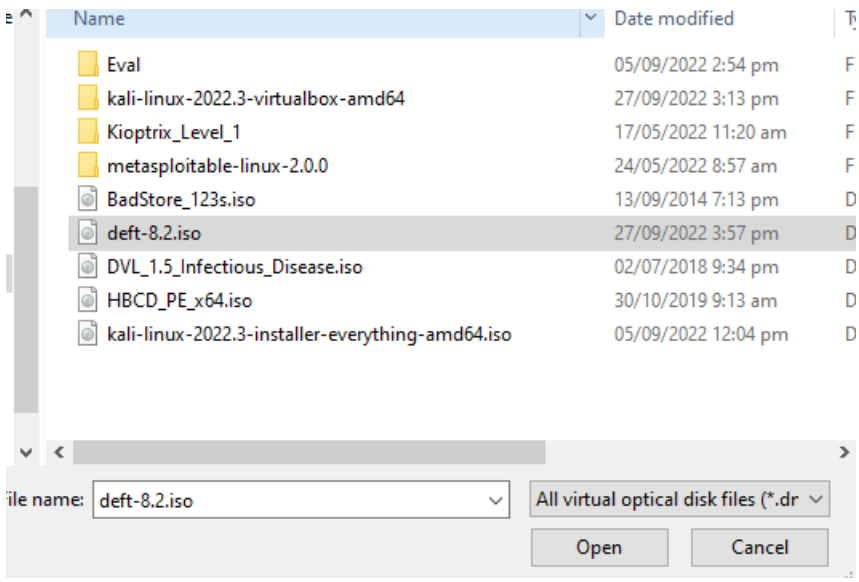


Figure 15.8 – Selecting the DEFT.iso file

You should now see `deft-8.2.iso` listed in the disk selector window.

11. Click on `deft-8.2.iso` and then click on the **Choose** button:

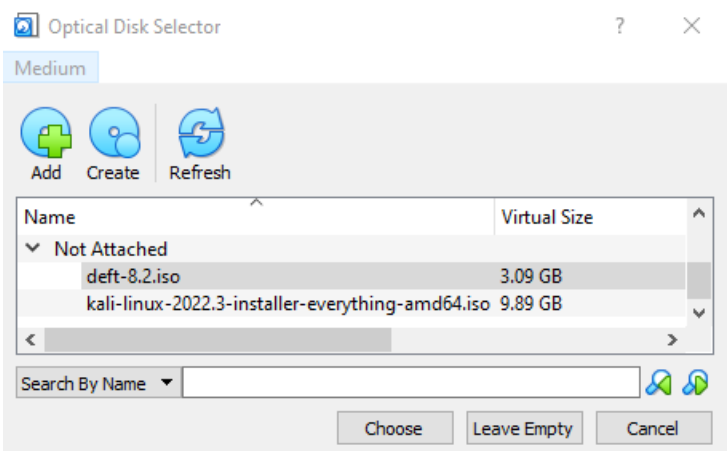


Figure 15.9 – Selecting the optical disk

12. Once chosen, you should now see the `deft-8.2.iso` file as the entry in the **Select start-up disk** window. You can now click on the **Start** button to start DEFT Linux 8.2.

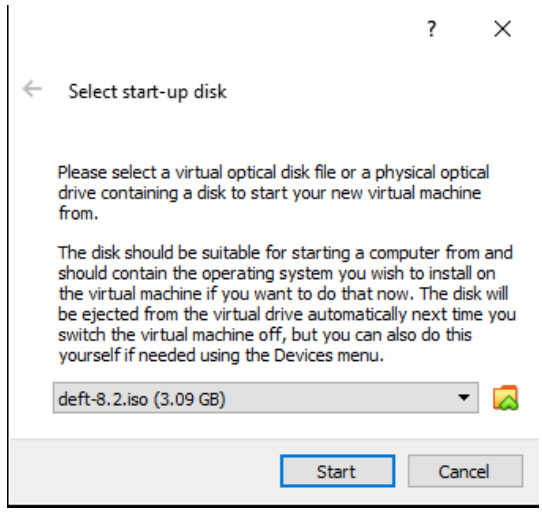


Figure 15.10 – The Select start-up disk selection window

13. Once all the preceding steps have been completed, you will now be presented with the DEFT Linux start-up screen. Select the **DEFT Linux 8 Live** option and press *Enter*.

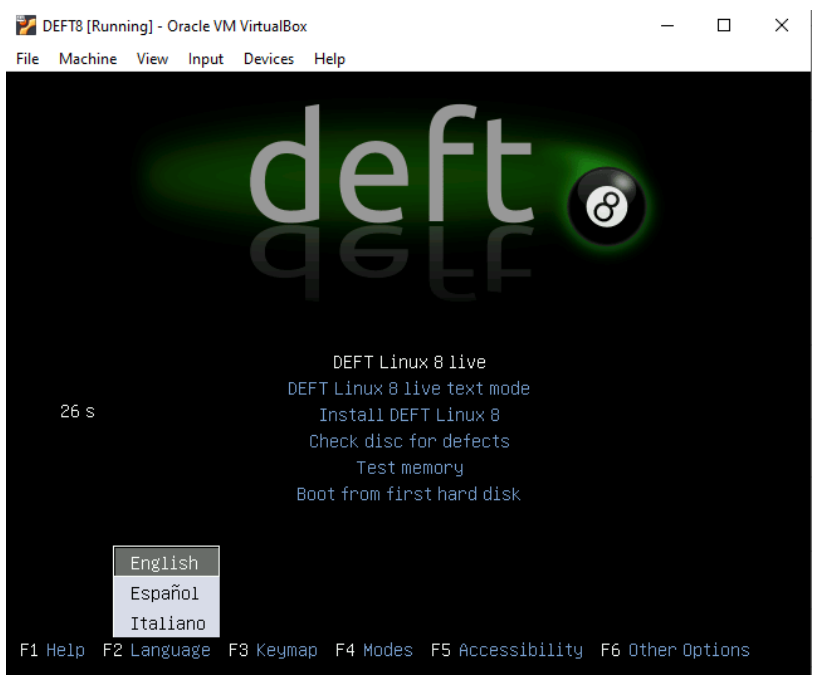


Figure 15.11 – The DEFT Linux splash screen

After all the processes have loaded, you will now have access to the DEFT Linux desktop, as shown here.

Figure 15.12 – The DEFT Linux desktop interface

Hopefully, you've arrived here without any issues. Before proceeding with our analysis, let's now download our sample files for analysis.

Downloading sample analysis files

Let's now download three sample files for automated analysis in Xplico. We will be downloading three sample packet capture (`.pcap`) files for analysis. All three files and many others are available at `https://wiki.wireshark.org/SampleCaptures`; however, you can click on the following direct links to download the three required packet capture files now:

- `http_with_jpegs.cap.gz: https://wiki.wireshark.org/uploads/__moin_import__/attachments/SampleCaptures/http_with_jpegs.cap.gz`

- `smtp.pcap`: `https://wiki.wireshark.org/uploads/__moin_import__/attachments/SampleCaptures/smtp.pcap`
- `nb6-telephone.pcap`: `https://wiki.wireshark.org/uploads/__moin_import__/attachments/SampleCaptures/nb6-telephone.pcap`

Now that we have all our sample files downloaded and ready to use, let's finally start Xplico.

Starting Xplico in DEFT Linux

We already learned how to start Xplico in Kali Linux at the beginning of the chapter, so let's now do the same for DEFT Linux:

1. In your DEFT Linux VM, click on the DEFT menu button at the bottom-left corner of the DEFT desktop, which looks like the letter **d** next to a black eightball.

Figure 15.13 – The DEFT Linux menu button

The DEFT menu will open.

2. Click on the **Service** option and then click on the **Apache start** button. A Terminal will open and the service will be started.
3. Next, go back to the **Service** option and click on the **Xplico Start** button. A Terminal will open and start the Xplico service.
4. Finally, we can start Xplico by clicking on the **DEFT** menu button/icon, navigating upward to the **DEFT** menu, going across to the **Network Forensics** menu, and then selecting **Xplico**, as shown here.

Figure 15.14 – Starting Xplico in DEFT

5. Xplico will now open in the default browser, and it can also be accessed by typing `localhost:9876` in the URL bar.

> **Important note**
> The username and password are both `xplico` (all lowercase).

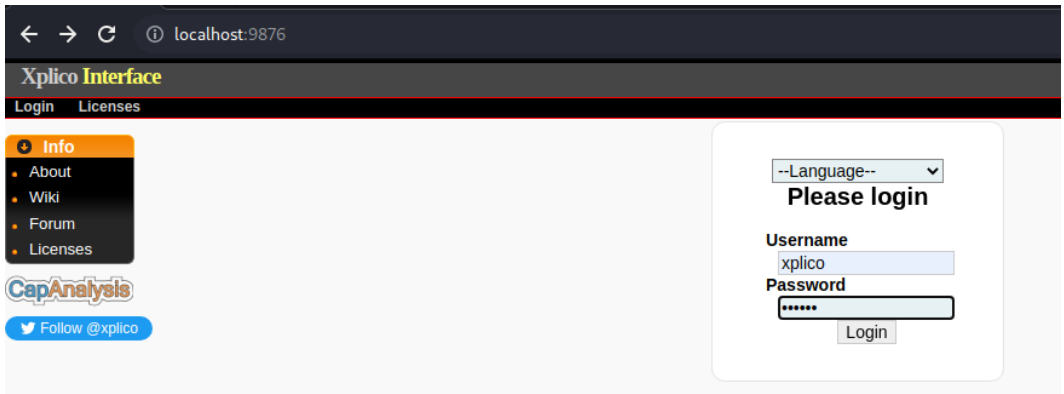


Figure 15.15 – The Xplico interface login

Now that we're logged into Xplico, let's start creating a case and session to analyze.

Using Xplico to automatically analyze web, email, and voice traffic

Once we have Xplico up and running in either Kali Linux or DEFT Linux, we can begin creating and analyzing our `.pcap` files. Xplico has a very intuitive user interface and also allows for case management of individual cases and sessions. We must first create a case and session before the `.pcap` file is uploaded and analyzed automatically for us:

1. To create a new case, click on the **New Case** option to the left of the Xplico window.

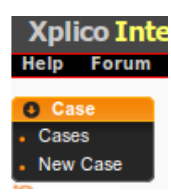


Figure 15.16 – The New Case option in Xplico

2. Next, give your case a name. I've named my case `CFSI-HTTP-2023`, as in this case, we will be analyzing the HTTP `.pcap` file we previously downloaded:

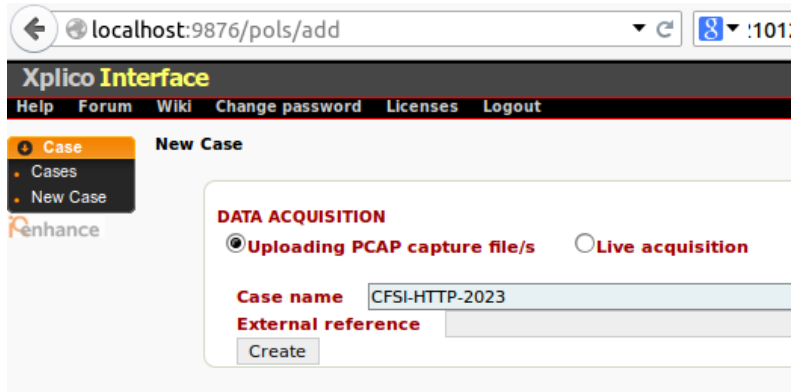


Figure 15.17 – Case name creation

3. Click on **Create** to continue and then click on the created case name to proceed. As shown here, the case name is listed as **CFSIHTTP2023**.

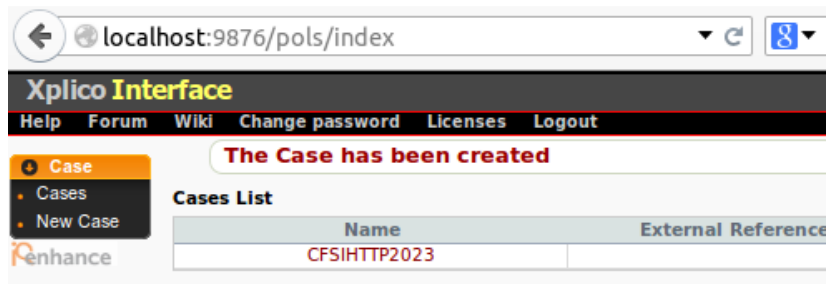


Figure 15.18 – Case lists

4. Now that a new case has been created, we must create a new session. Click on the **New Session** button in the left-hand menu in Xplico, as shown here.

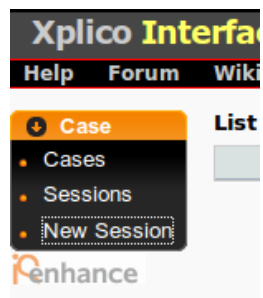


Figure 15.19 – New Session in Xplico

5. Give your session a different name than the case name and then click on **Create**:

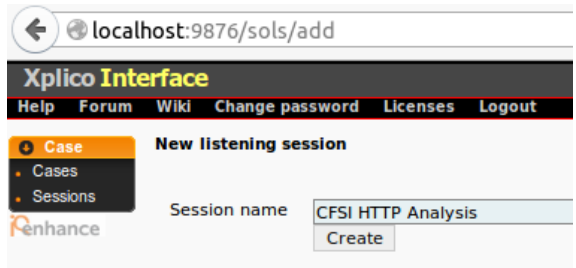


Figure 15.20 – Case session creation

6. Once the case session has been created, click on the session name, listed in red text. As shown here, my session is listed as **CFSIHTTPAnalysis**:

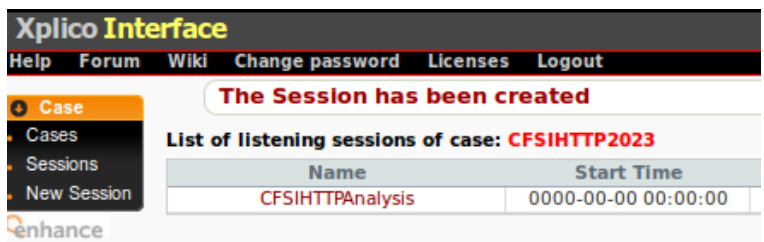


Figure 15.21 – Case session details

This will bring us to our main Xplico window, where we can upload our files for analysis.

> **Note**
> For each downloaded `.pcap` sample file, you will need to create a new case. Feel free to review this section as needed when creating new cases and sessions.

Once our case has been created, we can move on to the actual packet analysis using Xplico.

Automated web traffic analysis

In this section, we will use Xplico to perform automated web traffic and HTTP analysis:

1. Let's continue our **CFSIHTTPAnalysis** session. For this session, we will be analyzing the `http_witp_jpegs.pcap` file, which is contained within the `http_witp_jpegs.gz` file that we downloaded. To extract the `.pcap` file, right-click on the downloaded `http_witp_jpegs.gz` file and click on **Extract Here**.

Once extracted, you will now see the `http_witp_jpegs.pcap` file listed with the other downloaded files, as shown here.

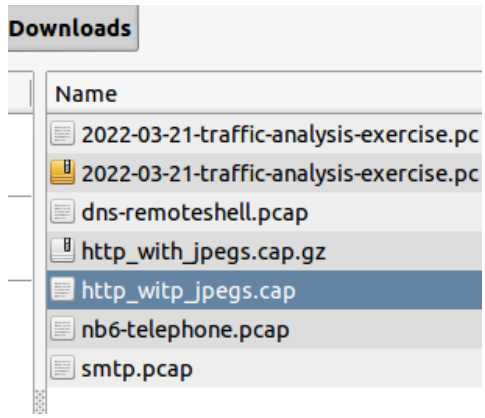


Figure 15.22 – All downloaded and extracted sample files

2. Let's now return to the Xplico web interface within our browser and click on the **Browse…** button in the **Pcap set** section, as shown here.

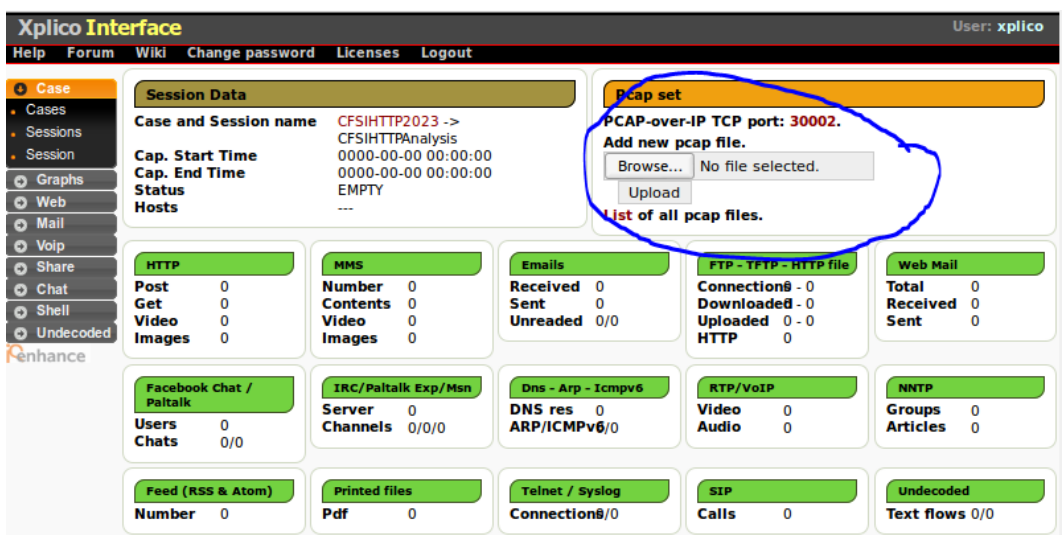


Figure 15.23 – The Pcap set section in Xplico

3. Go to `http_witp_jpegs.pcap` and click on **Open**.

4. Then, click on the **Upload** button in the **Pcap set** section to upload the file for analysis. The file may take a couple of seconds to upload, which will be displayed as a message in red text, as shown here.

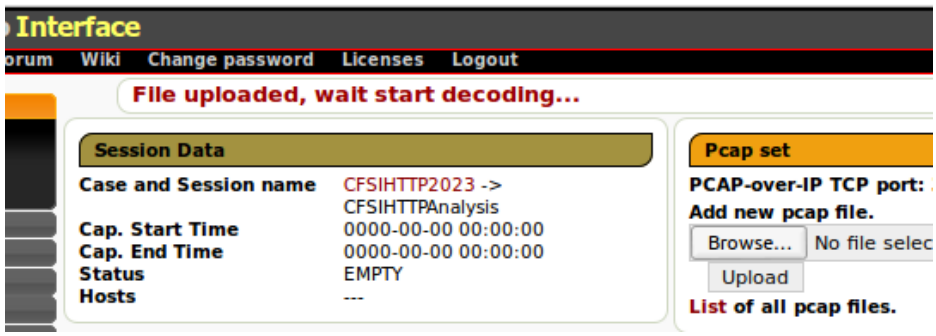


Figure 15.24 – Pcap file decoding

Once the file has been decoded and analyzed, you will see the words **DECODING COMPLETED** in the **Session Data** area of Xplico, as shown here.

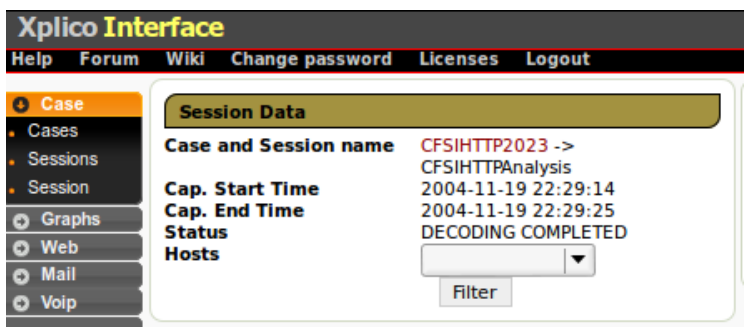


Figure 15.25 – Decoding completed

> **Important note**
>
> This process to upload and decode a file must also be done for each `.pcap` file. Feel free to return to this section as needed.

5. To analyze our uploaded files, we will utilize the menu on the left side, which allows us to inspect web, mail, **VoIP** (**Voice over Internet Protocol**), and other artifacts.

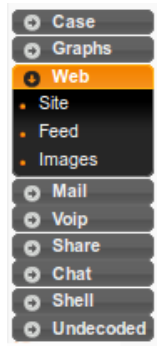


Figure 15.26 – The Xplico Analysis menu bar

6. Let's start our analysis by clicking on the **Web** menu and clicking on the **Site** option. Xplico has displayed a list of websites visited. We can also click on each website to visit them.

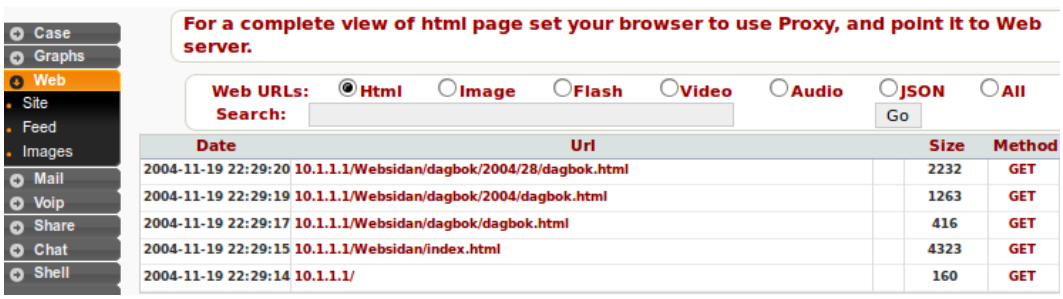


Figure 15.27 – Websites decoded by Xplico

7. Let's also click on the **Images** option on the side menu. We can see that Xplico has decoded and processed four images for us, as shown here.

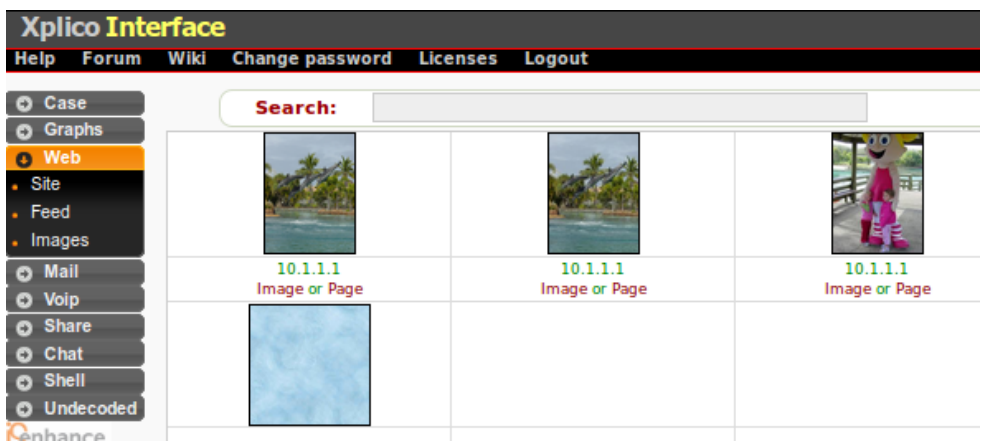


Figure 15.28 – Decoded images

Now that we know how to perform automated HTTP analysis and find various web artifacts, including images that were viewed and downloaded over the web, let's move on to some automated SMPT and email analysis using Xplico.

Automated SMTP traffic analysis

Let's create a new case to perform **Simple Mail Transfer Protocol** (**SMTP**) analysis. SMTP is used to send emails, and analysis of SMTP traffic can reveal the sender, recipient, and other details of an email, including attachments. So, let's get started with the new case creation and SMTP traffic analysis:

1. For this exercise, I've repeated the preceding steps to create a new case and session. Details of the case are as follows:
   - Case name: `CFSI-SMTP-2023`
   - External reference: `SMTP analysis`
   - Session name: `CFSI SMTP Analysis 2023`
2. Once our case and session have been created, click on the session to continue.

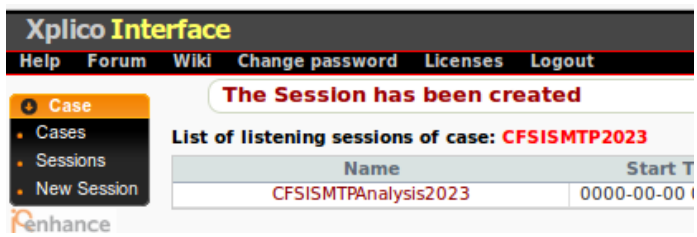


Figure 15.29 – The SMTP session

3. We can now browse the `smtp.pcap` file and upload it for processing, decoding, and analysis. You will again be notified when decoding has been completed, as shown here.

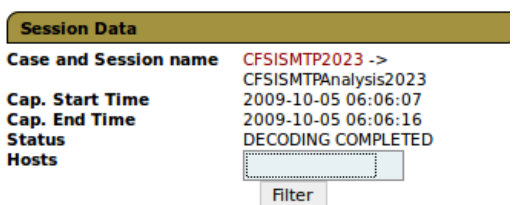


Figure 15.30 – Decoding complete

4. Let's click on the **Email** option under the **Mail** menu in the sidebar to reveal what has been decoded by Xplico.

Figure 15.31 – The decoded email

In the preceding screen, we can see that Xplico decoded an email from `gurupartap@patriots.in` to `raj_deo2002in@yahoo.com`.

5. If we click on the word **SMTP**, as shown in *Figure 15.31*, we can view the contents of the email, as shown in *Figure 15.32*.

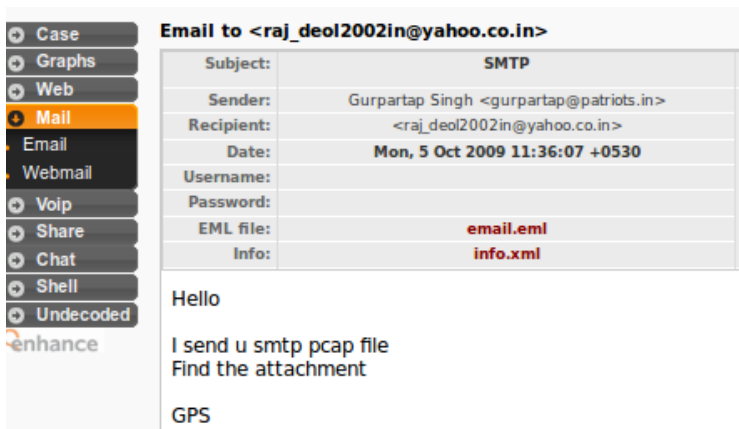


Figure 15.32 – Contents of the decoded email

Let's move on to another analysis exercise now.

Automated VoIP traffic analysis

Xplico can also decode VoIP traffic if captured and saved in a `.pcap` file. Follow these steps for this analysis:

1. Let's again create a new case and session first.

   Here are the case and session details:

   - Case name: `CFSI-Voice-2023`
   - External reference: `CFSI Voice Analysis`
   - Session name: `CFSIVoice2023`

2. Once our case and session have been created, we can then navigate to and upload the `nb6-telephone.pcap` file that we downloaded earlier.

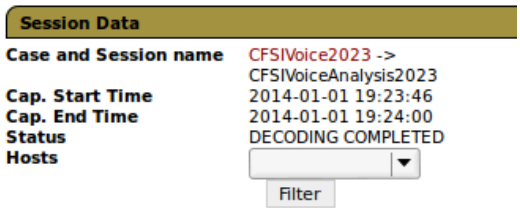


Figure 15.33 – The decoded VoIP file

3. Let's see what was revealed by Xplico by clicking on the **Sip** option in the **Voip** menu bar.

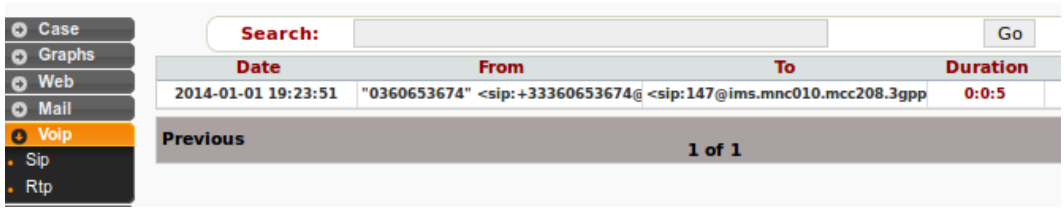


Figure 15.34 – The decoded VoIP traffic

4. As shown previously, Xplico has automatically decoded a captured VoIP conversation. You can also click on the call duration to play back the conversation, as shown in the following screenshot.

Figure 15.35 – The decoded VoIP call

You may need to install plugins to listen to the audio file. You will be prompted to install the relevant plugins automatically, as shown in *Figure 15.35*.

Summary

In this chapter, we learned how to use Xplico in both Kali Linux and DEFT Linux, which we installed separately in VirtualBox. DEFT can be a great substitute for those who may have encountered issues when installing Xplico in Kali, and it also offers many other tools that you can explore if you so wish. We learned that a new case and session must be created for each packet capture (`.pcap`) file analysis, and that Xplico does automatic decoding and analysis of `.pcap` files to reveal artifacts that are useful to our DFIR investigations. Lastly, we learned how to use Xplico to find useful artifacts such as visited websites, viewed and downloaded images, emails, and VoIP conversations. I hope you enjoyed using this automated tool. We will next look at other NFAT tools in our last chapter. See you in the next chapter.

16
Network Forensic Analysis Tools

Here we are. Our final chapter. I believe in the concept of finishing strong, so let's keep pace by continuing our DFIR journey with some **Network Forensic Analysis Tools** (**NFAT**), which I think you'll find quite useful.

We've done quite a bit of acquisition and analysis thus far including hard drive, storage, RAM and swap file analysis, malware analysis, and even a bit of network packet analysis for the purpose of acquiring, documenting, and analyzing evidence in the hope of finding or recovering artifacts. But let's go a step further into analyzing packets, protocols, and network communication, as they may also be useful artifacts that can aid us in our DFIR investigations.

On recognizing that some incidents and crimes occur online over the internet or even the **Local Area Network** (**LAN**), capturing and analyzing network traffic should be an essential part of our investigative process in discovering artifacts that may help us better understand the incident, point to sources of origin, and even in some cases, assist in extending the scope of the investigation if it is suspected that the incident may not be an isolated one.

In this chapter, we'll cover the following topics:

- Creating **packet capture** (**PCAP**) files using Wireshark
- Packet analysis using NetworkMiner
- Packet analysis using PcapXray
- Online PCAP analysis using `packettotal.com`
- Online PCAP analysis using `apackets.com`
- Reporting and presentation

Capturing packets using Wireshark

Wireshark is a very popular and well-known tool used for network and packet analysis and troubleshooting. It comes pre-installed in Kali and is relatively straightforward to use once you have an idea of filters, protocols, and color codes.

If you're new to the Wireshark protocol analyzer and packet analysis, you can find some great tutorials online, including the official documentation at `https://www.wireshark.org/docs/wsug_html_chunked/ChapterCapture.html`.

First, let's see what our network interfaces are and then begin using Wireshark:

1. We'll need to specify our interface when capturing packets after starting Wireshark. To get information on your interfaces in Kali Linux, open a Terminal and type `ifconfig`:

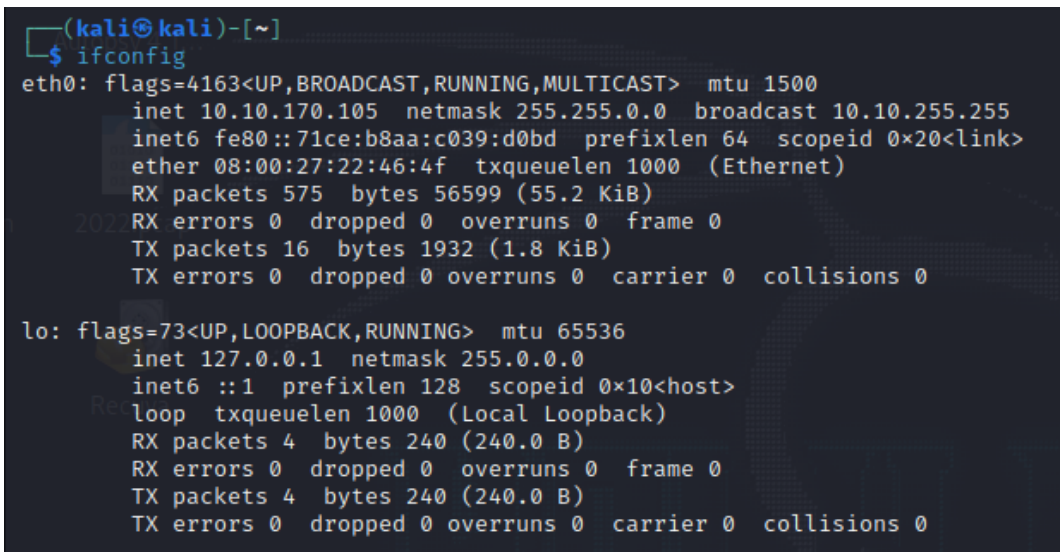


Figure 16.1 – ifconfig output

In the preceding screen capture, the `ifconfig` command displays the output for two interfaces. The interface I'll be using is my Ethernet interface, listed as `eth0`, and there is also the loopback interface listed as `lo`.

> **Please note**
>
> If you're using a wireless NIC to capture interfaces, it will be listed as `wlan0`.

2. Now that we know which of our interfaces we'll be using to capture packets and sniff the network with, we can start Wireshark by typing `sudo wireshark` in the Terminal.
3. You can also run Wireshark by clicking on **Applications** | **09-Sniffing & Spoofing** | **Wireshark**:

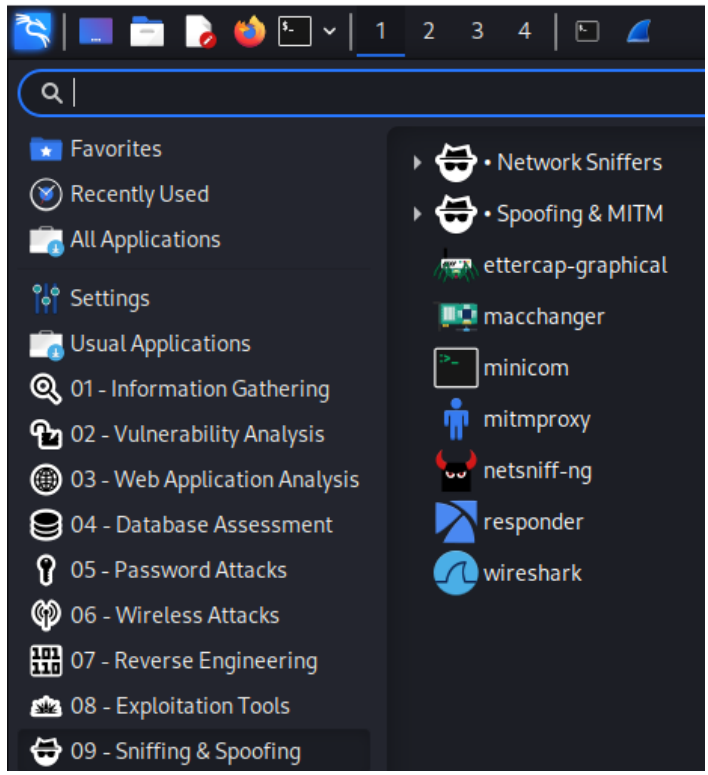


Figure 16.2 – Finding Wireshark within the Sniffing & Spoofing menu

4. After starting Wireshark, as mentioned previously, we'll need to select an interface to begin capturing packets on. In this instance, my `eth0` interface is highlighted, but be sure to select the interface you will be working with to capture packets.

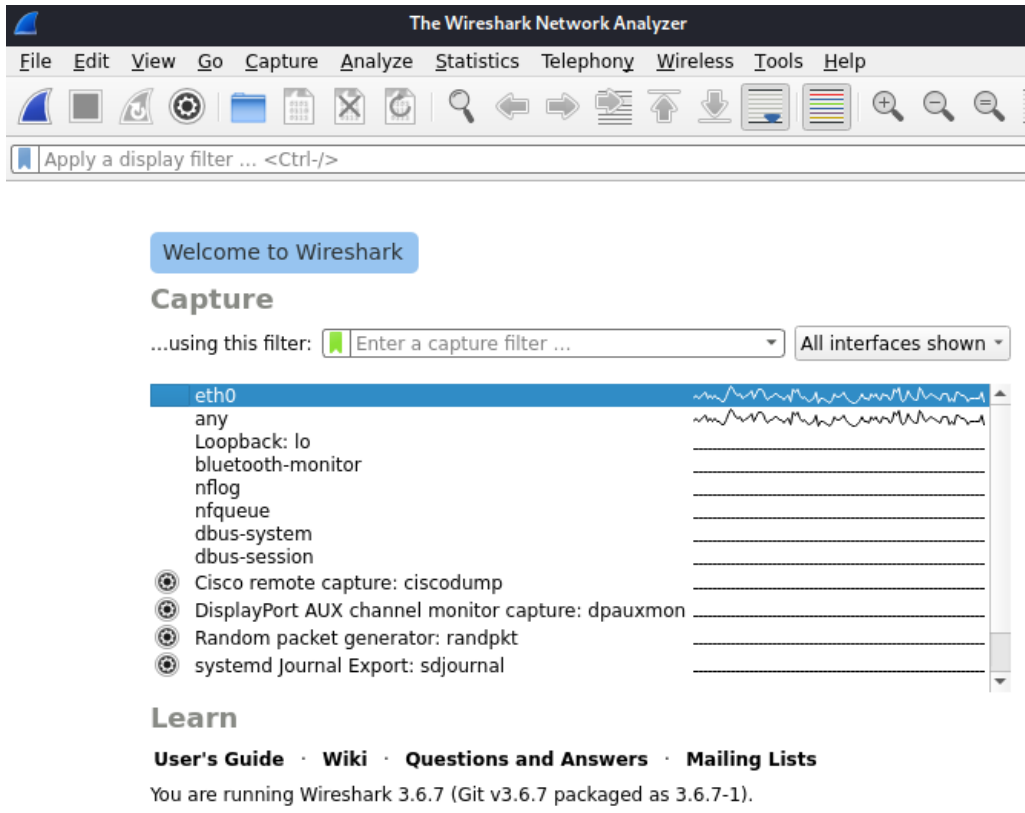


Figure 16.3 – Wireshark interfaces

5. Once the interface is selected, we can begin the packet capture process by either clicking on the *blue shark fin* icon or by clicking on **Capture | Start**:

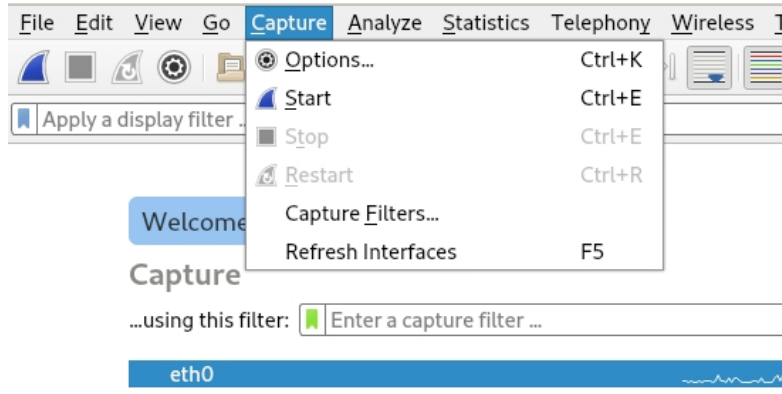


Figure 16.4 – Starting the packet capture process in Wireshark

The packet capture process automatically begins after clicking on the **Start** button to start capturing.

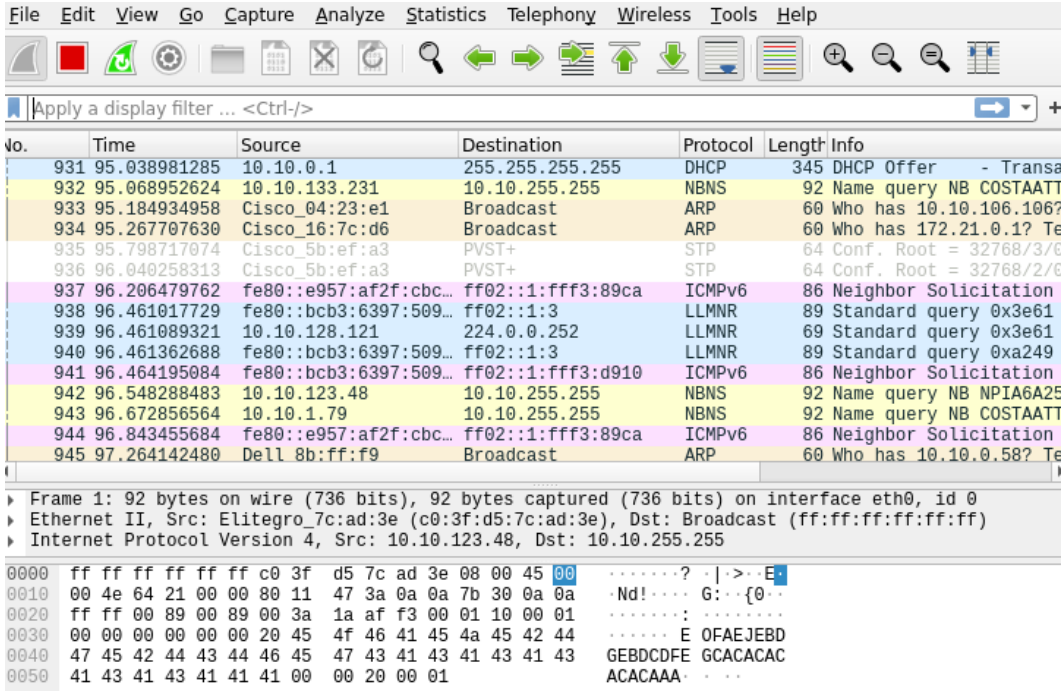


Figure 16.5 – Wireshark displaying traffic on eth0

In the preceding screenshot, we can see that Wireshark organizes the display into three sections, with the main section at the top containing rows populated with **Source**, **Destination**, **Protocol**, and other information, all color coded.

6. To stop the capture, click on the *stop* button at the top (the red square icon):

Figure 16.6 – Wireshark stop capture button

7. Be sure to save the packet capture file by clicking on **File | Save As**.

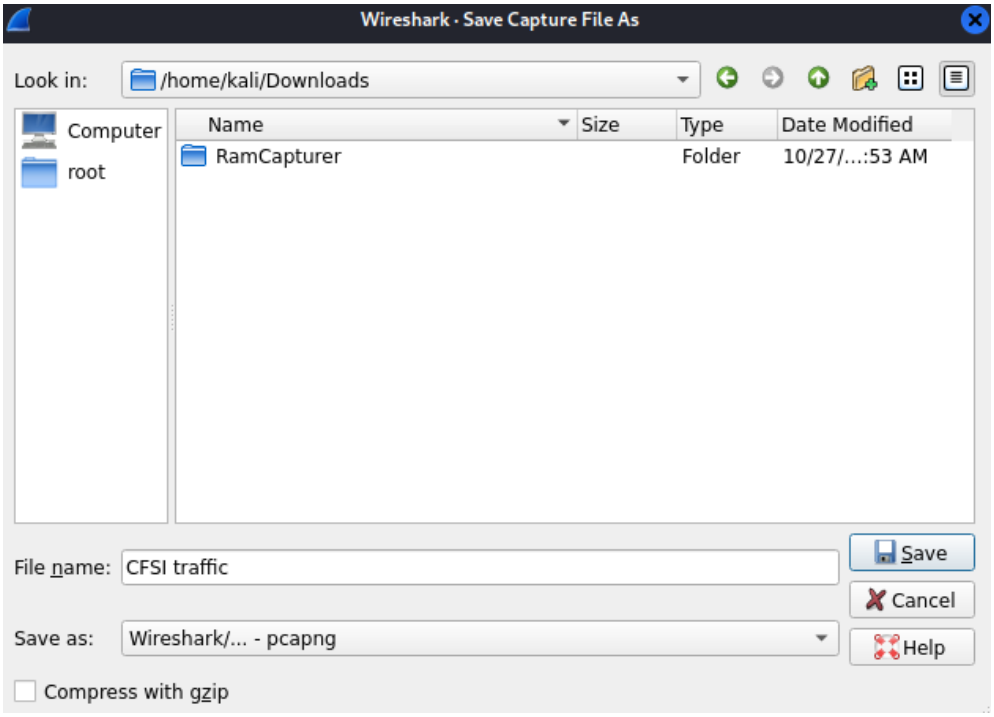


Figure 16.7 – Saving traffic within a pcapng file

8. We can also use specific Wireshark filters to sort the data, thereby presenting a logical and simpler view of the packets.

> **Wireshark filters**
> You can view a list of all Wireshark filters at `https://www.networkdatapedia.com/single-post/2019/01/29/Top-10-Wireshark-Filters`.

To set a filter for packets with a particular source or destination IP address, we use the `ip.addr==a.b.c.d` format, as shown in the following example:

```
ip.addr==172.16.79.21
```

The following figure shows the output of the preceding `ip.addr` filter:

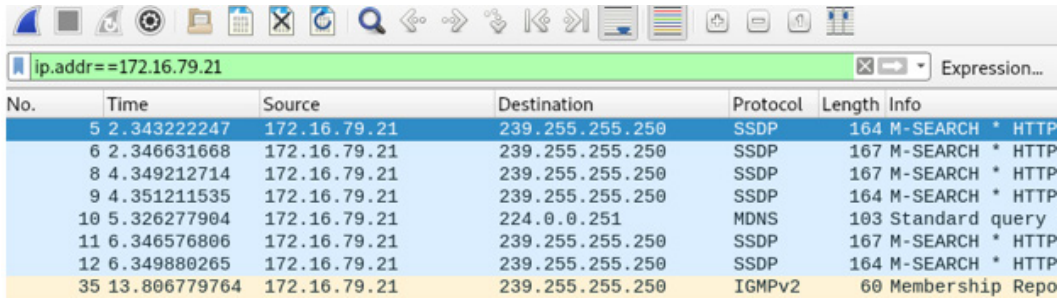


Figure 16.8 – The ip.addr Wireshark filter output

9. To set a filter for a specific TCP port, we can use the `tcp.port==` filter, as shown in this example:

```
tcp.port==443
```

The following figure shows the output of the preceding `tcp.port` filter:

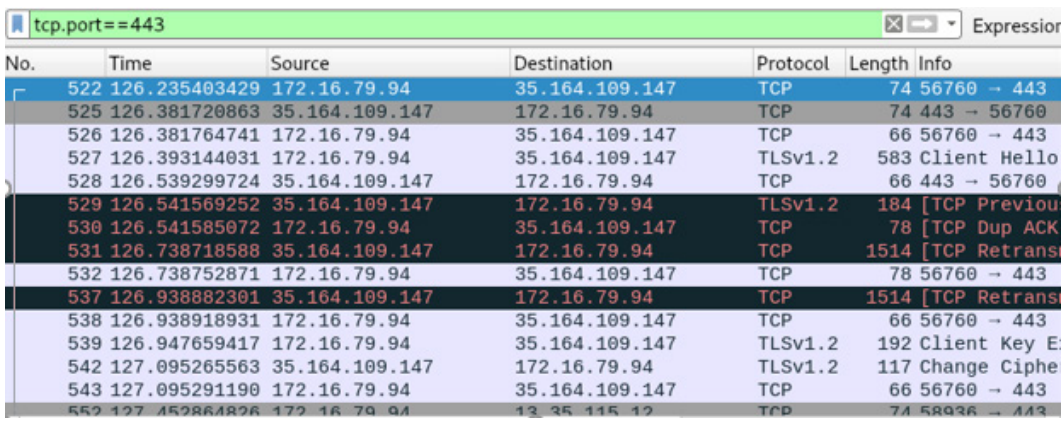


Figure 16.9 – The tcp.port filter

10. To set a filter that searches for a specific word, string, or even user, use the `frame contains` filter. For example, I visited a website (`www.malware-traffic-analysis.net`) that hosts sample PCAP files of malware infections and downloaded a sample PCAP file. To search for the word *malware*, I'll use the `frame contains malware` filter.

The following figure shows the output of the `frame contains malware` filter:

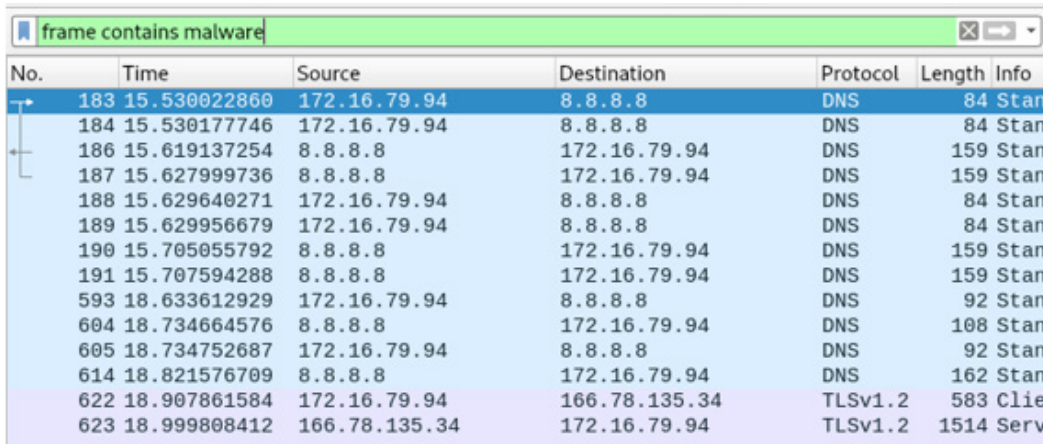


Figure 16.10 – The frame contains malware filter output

11. To set a filter for viewing a conversation between two IPs, we use the `ip.addr==a.b.c.d && ip.addr==w.x.y.z` format, as shown in the following:

```
ip.addr==172.16.79.94 && ip.addr==172.16.0.1
```

The following figure shows the output of the preceding filter:

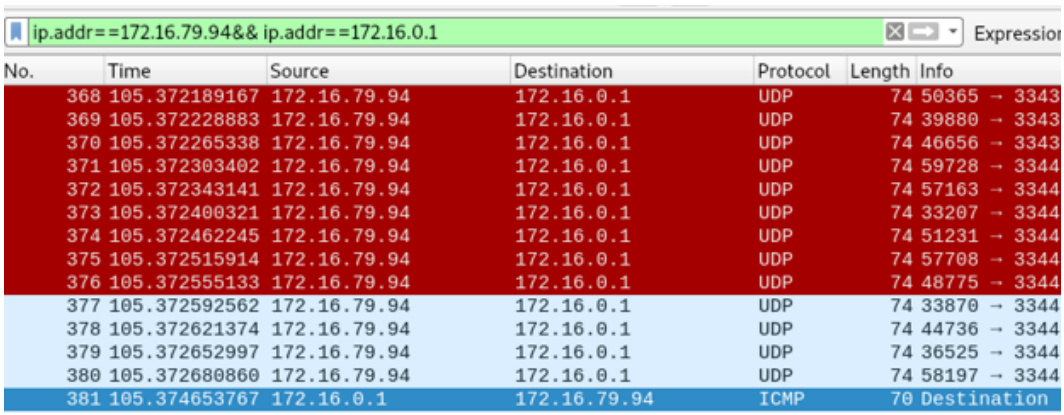


Figure 16.11 – Packet exchange between devices

Note

You can find more information on Wireshark filters here: `https://wiki.wireshark.org/DisplayFilters`.

Using Wireshark can definitely take some time to learn, but I assure you that it is very useful not just for our DFIR purposes as an NFAT but also as a network troubleshooting tool. Let's move on to a similar tool that presents the findings of packet analysis in a format that is much easier to read.

Packet analysis using NetworkMiner

Analyzing captured data from Wireshark can be a bit of a challenge to people who may be new to the protocol analyzer, as it requires knowledge of protocols, filters, and the ability to follow data streams (all of which becomes easier with practice).

NetworkMiner is an easy-to-use packet capture viewer that some users may find easier to use for `.packet` capture (PCAP) analysis, as it extracts and sorts the found data into categories of hosts (with operating system fingerprinting), files, images, messages, sessions, and more by parsing the PCAP file.

NetworkMiner comes in a free as well as a paid professional version, and can be installed on Windows and Linux.

We will now download NetworkMiner, which will be installed using Wine, and then analyze a sample PCAP file:

1. You can visit the official website for NetworkMiner at this link: `https://www.netresec.com/?page=NetworkMiner`.

   You can also download the installation file at `https://www.netresec.com/?download=NetworkMiner`.

2. Once the file has been downloaded, right-click on the file and select the **Extract Here** option. This extracts the `NetworkMiner_2-7-3` folder. Open the folder to view the extracted contents.

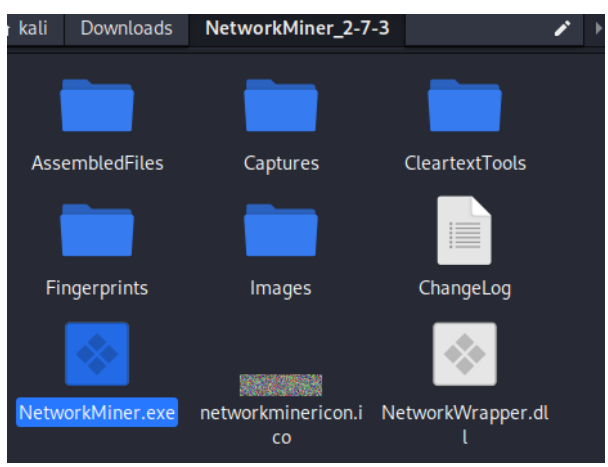


Figure 16.12 – Extracted NetworkMiner files

3. Running NetworkMiner is very simple, as we have already installed Wine. Simply double-click on `NetworkMiner.exe` to open the program.

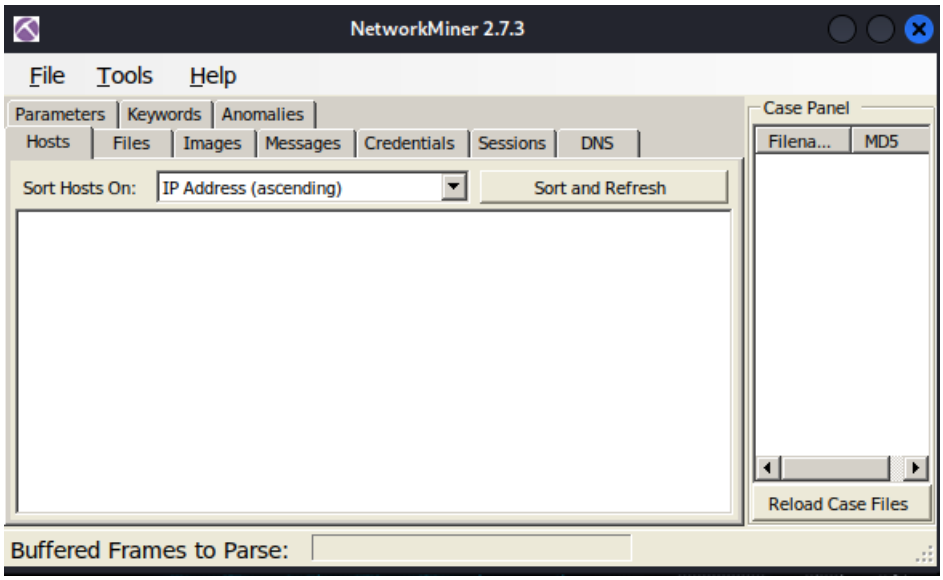


Figure 16.13 – NetworkMiner interface within Wine in Kali Linux

4. To view further documentation and videos, and get access to sample PCAP files, visit this link: `https://www.netresec.com/?page=Resources`.
5. For this analysis, we'll be using the PCAP from `https://wiki.wireshark.org/uploads/__moin_import__/attachments/SampleCaptures/http_with_jpegs.cap.gz`, which you can download and save to your Kali machine. This file was previously downloaded back in *Chapter 15*, *Packet Capture Analysis with Xplico*, for use with Xplico, so you may already have it in your `Downloads` folder.

   I've decided to use this file so that we can view the differences in findings between Xplico and NetworkMiner for comparison.
6. I've already downloaded and extracted the file to my `Downloads` folder. In the NetworkMiner program, click on **File** and **Open** and browse to the `http_witp_jpegs.cap` file in the `Downloads` folder (or wherever you may have downloaded the file to). Click on the PCAP file and then click **Open**.

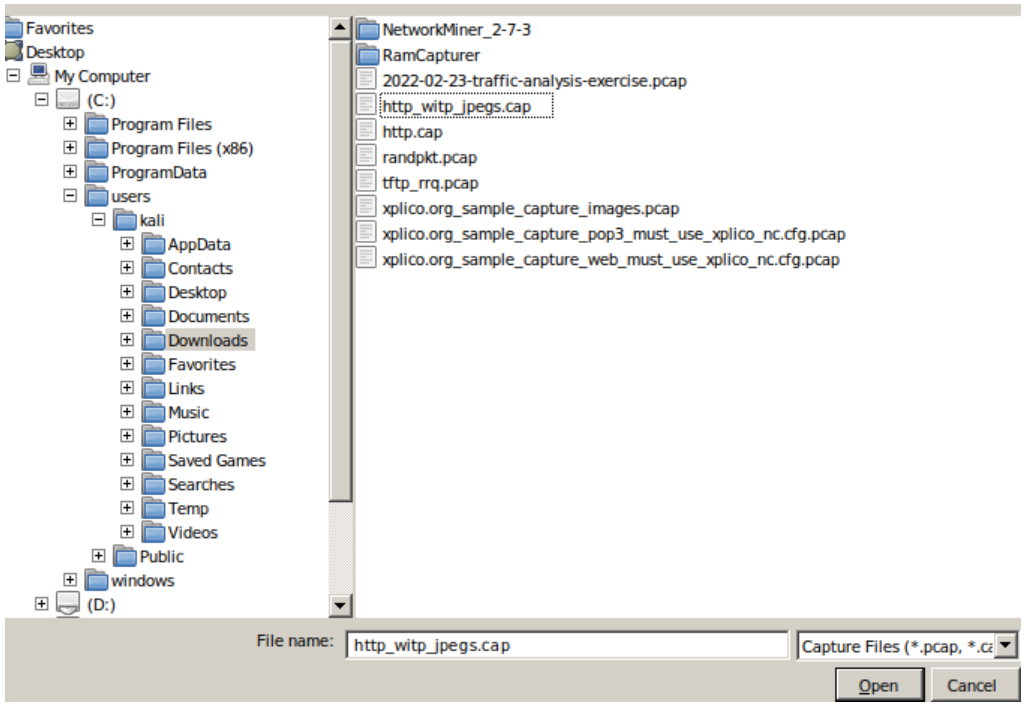


Figure 16.14 – Browsing to the sample file in Kali

7. After clicking on **Open**, NetworkMiner loads and analyzes the file and automatically categorizes the findings for us, as seen here:

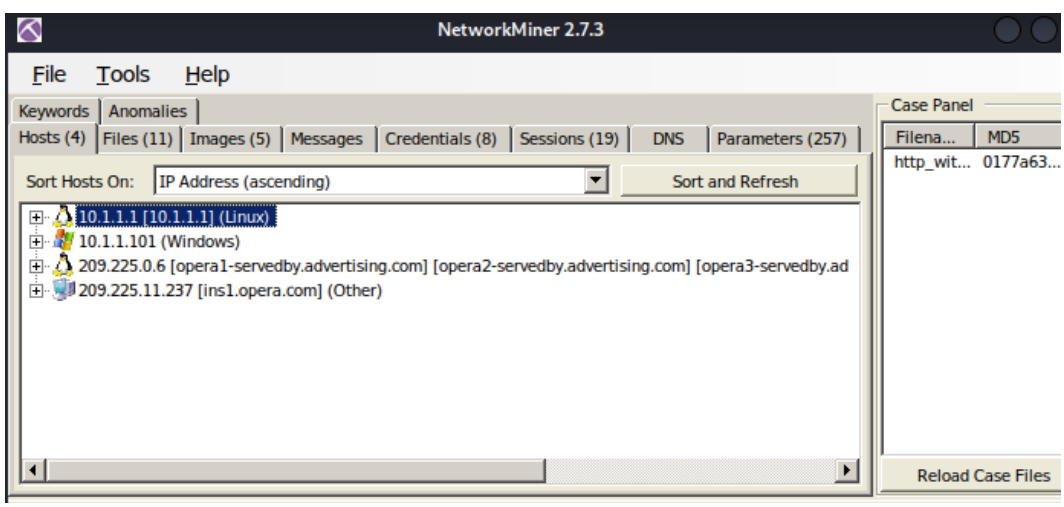


Figure 16.15 – Analyzed and categorized findings of the sample PCAP file

8. At first glance, we are presented with information about devices in the **Hosts** tab. The `.pcap` files contain several hits per category:

   - **Hosts – 4**
   - **Files – 11**
   - **Images – 5**
   - **Messages – 0**
   - **Credentials – 8**
   - **Sessions – 19**
   - **DNS – 0**
   - **Parameters – 257**

9. In the **Hosts** tab, details for each host can be seen by clicking on the expand (+) button next to the IP address. Let's expand the second IP address (`10.1.1.1`) to view the details. We can also expand the other items, such as **OS**, **Outgoing sessions**, and **Host Details**, to see all the information that NetworkMiner has automatically parsed for us.

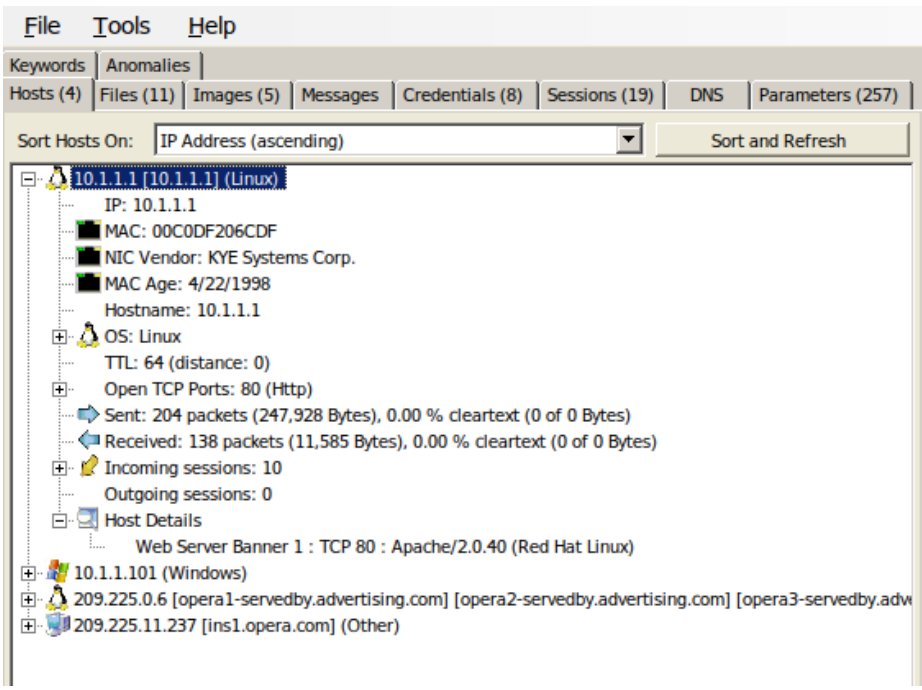


Figure 16.16 – Hosts tab findings

In the preceding screenshot, we can see details such as MAC address, NIC manufacturer, OS (Linux), open ports, outgoing sessions to servers, and even the host details, which tells us that the system is an Apache web server on Red Hat Linux.

10. Let's click on the **Files** tab next. We can see several entries and file types (`html`, `xapc`, and `jpeg`).

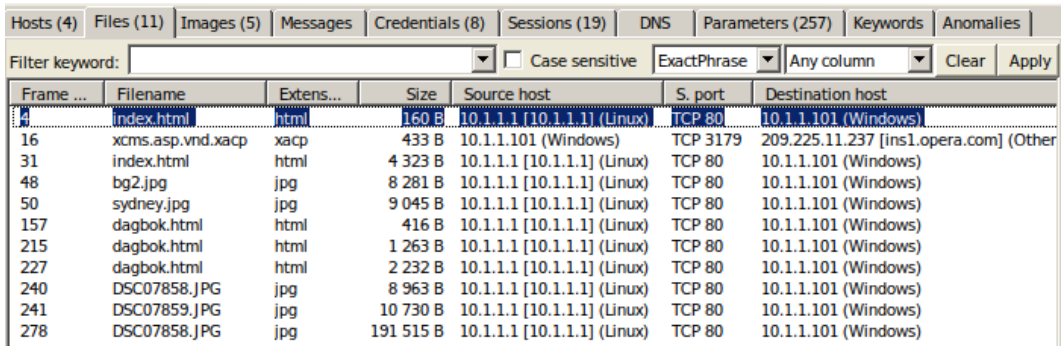


Figure 16.17 – Files tab list

11. We can also open these links and view files. Right-click on the `index.html` file and select **Open**. This opens the `.html` file in the browser. You can also do this for other files, such as the `.jpeg` files.

12. Click on the **Images** tab. In this scenario, NetworkMiner has found **5** images with previews. You may also right-click on the image and choose **Open Image** to view the image.

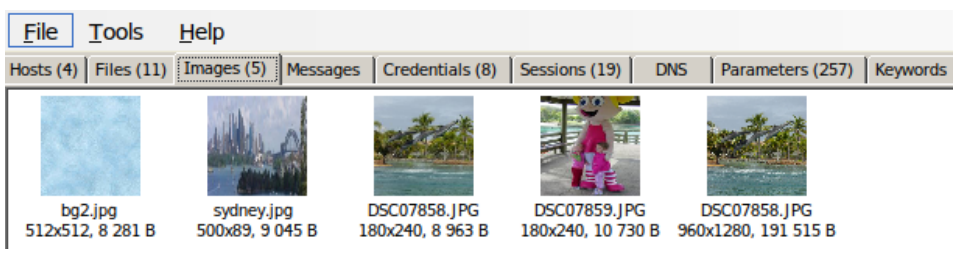


Figure 16.18 – Images tab

13. In the **Messages** tab, there are three messages that show the source and destination hosts, as well as the sender and recipient of the message/email if we scroll to the right.

14. In the **Credentials** tab, we find some very interesting artifacts. This tab shows client and server IPs, OS type, protocol, and any associated usernames and passwords that may have been used in that session, which are most likely unencrypted plain-text passwords.

15. The **Sessions** tab shows the sessions between devices at the time of the packet capture.

16. The **Keywords** tab/section allows the investigator to enter individual keywords (string or hex format) or add keywords from a text file to search for, within the list of results. If using keywords, you may have to start over by specifying a keyword list or file and then re-opening the `.pcap` file in NetworkMiner.

As we can see, NetworkMiner is a powerful packet capture analyzer that makes the analysis much easier for investigators and networking personnel by automatically parsing and categorizing the information found in the `.pcap` file.

I'm sure that many of you found NetworkMiner to be much simpler to use than Wireshark when performing packet analysis, as the findings are displayed in a simpler and easier-to-read format. Keep in mind that NetworkMiner has limitations for its free version, which is why we need to know about multiple NFAT tools. Let's move on to another tool, called PcapXray.

Packet capture analysis with PcapXray

Much like NetworkMiner, **PcapXray** is another powerful and comprehensive packet capture analysis tool. Some of the main features of this tool are the identification of malicious traffic, covert communication, web, and even Tor traffic.

We will now install and configure PcapXray within Kali Linux and then begin analyzing a packet capture file:

1. Let's install PcapXray by cloning it from GitHub by typing the following command in the Terminal:

```
sudo git clone https://github.com/Srinivas11789/PcapXray.
git
```

As usual, I've changed my directory to the desktop. When cloning PcapXray to your desktop, this will take some time, as the file is 115 MB in size.

```
┌──(kali㉿kali)-[~]
└─$ sudo git clone https://github.com/Srinivas11789/PcapXray.git
[sudo] password for kali:
Cloning into 'PcapXray' ...
remote: Enumerating objects: 1704, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 1704 (delta 3), reused 9 (delta 2), pack-reused 1689
Receiving objects: 100% (1704/1704), 115.75 MiB | 1.14 MiB/s, done.
Resolving deltas: 100% (975/975), done.
```

Figure 16.19 – Cloning PcapXray into Kali

2. Next, we need to install `graphviz` by typing the following:

```
sudo apt install graphviz
```

The following figure shows the output of the preceding command:

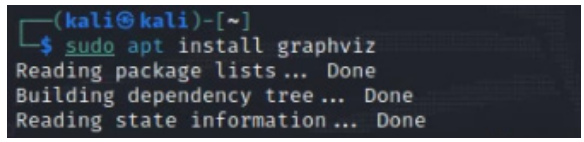


Figure 16.20 – Installing graphviz

3. Install `python3-pil.imagetk` by running the following command in the Terminal:

```
sudo apt install python3-pil python3-pil.imagetk
```

The following figure shows the output of the preceding command:

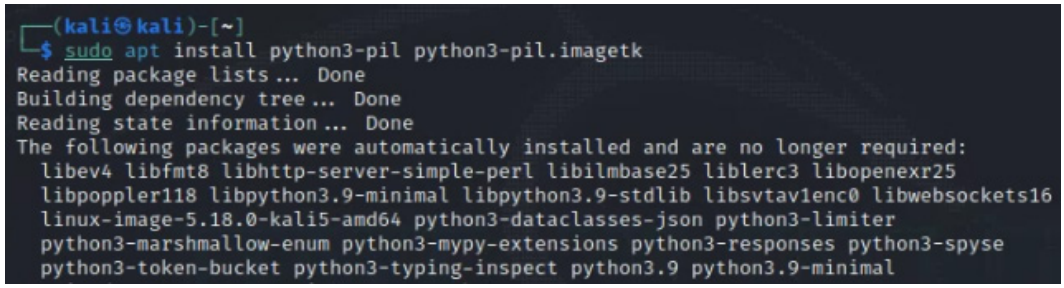


Figure 16.21 – Installing python3-pil.imagetk

4. We should now be able to see a PcapXray folder with the files required to start the application. Use the `ls` command to list and show the folders, and then change to the `PcapXray` folder using the `cd PcapXray` command.

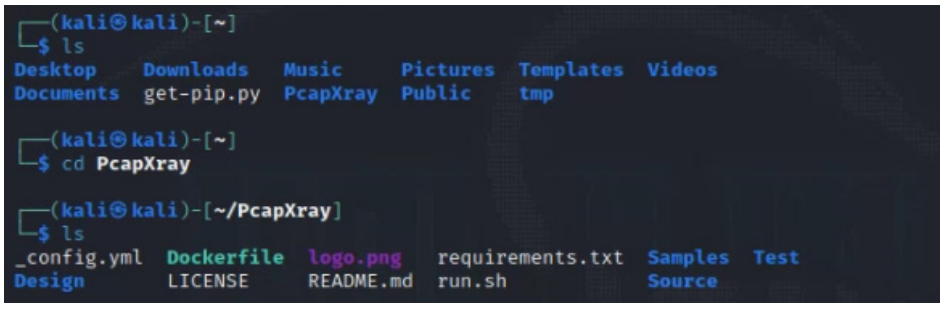


Figure 16.22 – Switching to the PcapXray directory

5. The last step before we run PcapXray will be to install all requirements for the application by typing the following command:

```
pip3 install -r requirements.txt
```

This may take some time, as it will download `pyshark`, `stem`, `pyvis`, and other required components.

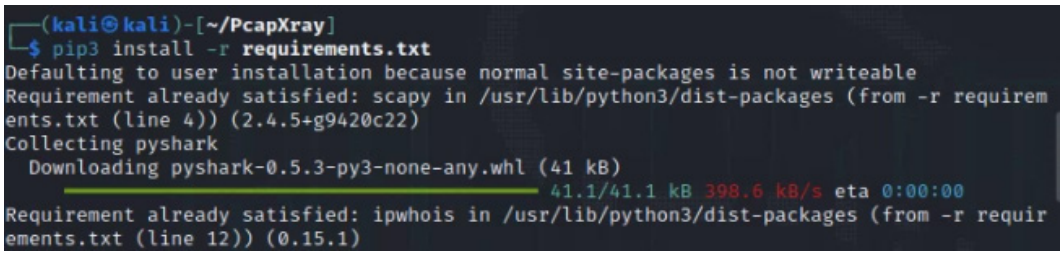


Figure 16.23 – Installing additional PcapXray requirements

6. While still within the `PcapXray` directory, we can now start the PcapXray GUI by typing the following command:

```
python3 Source/main.py
```

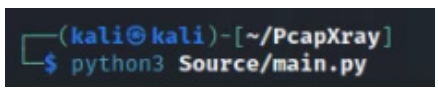


Figure 16.24 – Starting PcapXray within the Terminal

7. Let's first download a file to analyze from `https://www.malware-traffic-analysis.net/2019/07/19/index.html` named `2019-07-19-traffic-analysis-exercise.pcap`. Click on the filename to download it and then extract it so that we can begin the analysis. When prompted for a password, type the word `infected`.
8. We can now get back to the PcapXray GUI and browse to the extracted `.pcap` file and specify an output directory, as seen here:

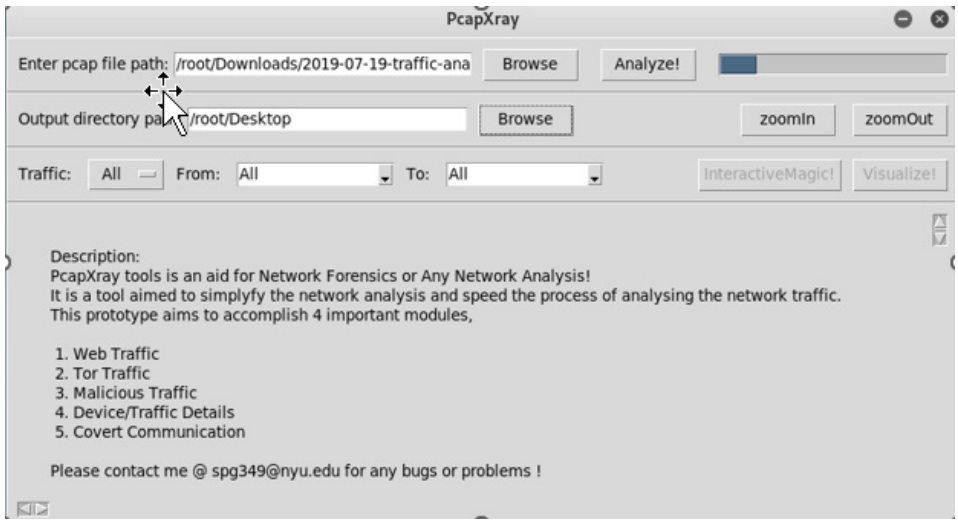


Figure 16.25 – PcapXray file upload interface

9. Click on the **Analyze!** Button, which will then take a while to perform the analysis. You should then be able to click on the **Visualize!** button when it becomes available.

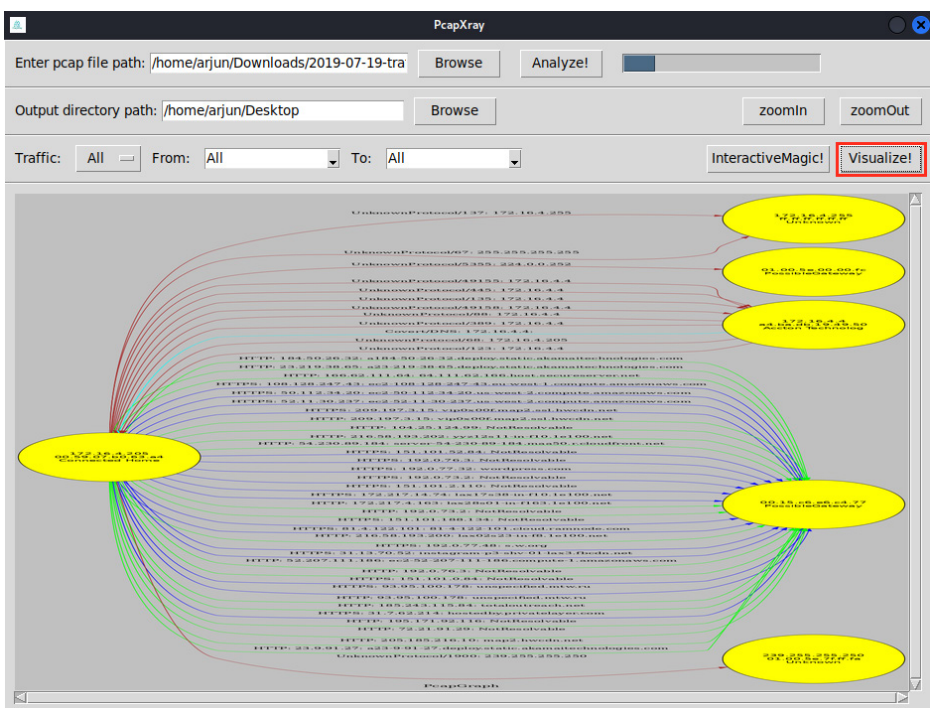


Figure 16.26 – Automated PcapXray analysis results

10. Although a bit hard to see in the preceding screenshot, it does an analysis of the traffic between the source and destinations. Click on the **InteractiveMagic!** button to see a different view of the devices and their relation to each other.

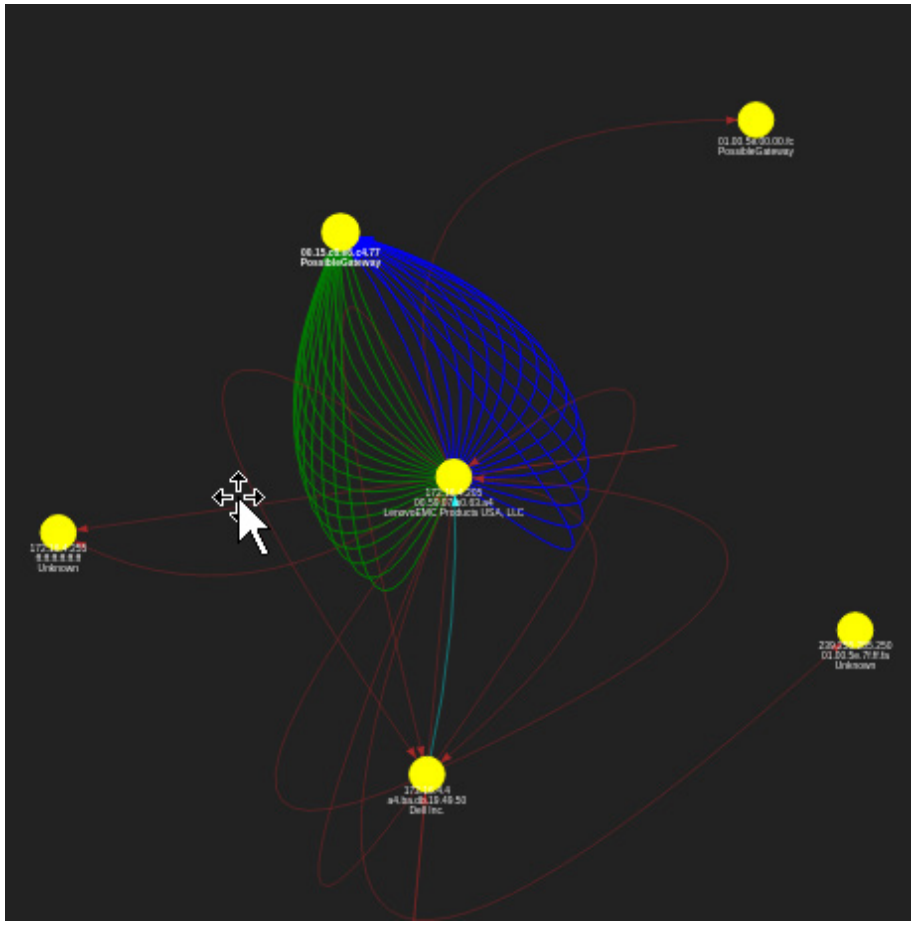


Figure 16.27 – PcapXray InteractiveMagic view

These views can help pinpoint which devices were in communication with each other, whether legitimately or covertly, and help with incident analysis.

11. We can also narrow down our view by clicking on the **All** button in the menu above the traffic visualization and choosing which type of traffic we'd like to view.

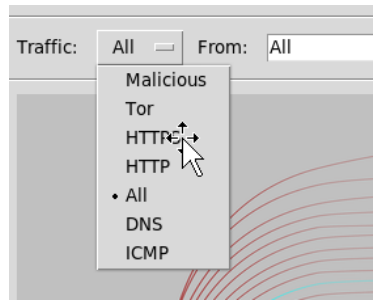


Figure 16.28 – PcapXray All Traffic view

12. Click on **Malicious** and then click on the **Visualize!** button again.

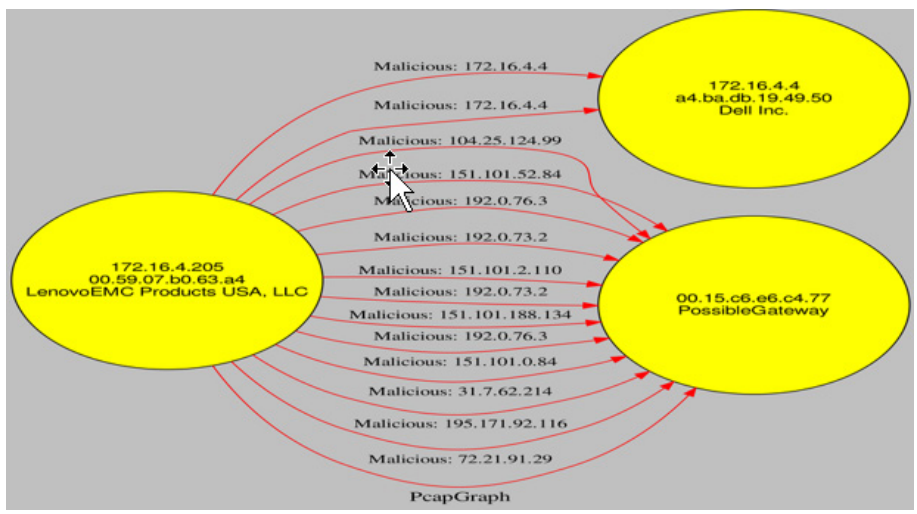


Figure 16.29 – Malicious traffic communications

Here, we can see traffic labeled as **Malicious** and the IP and gateway addresses associated with the communicating devices. If we change the traffic view to HTTPS and then click on the **Visualize!** button again, we can also see the HTTPS web traffic, and thereby, begin putting together or recreating the scenario between the malware, devices, and web traffic at the time of capture.

Let's now move on to PCAP analysis using online tools. Let's start with `packettotal.com`.

Online PCAP analysis using packettotal.com

Let's now have a look at a unique online resource for automated PCAP analysis that is freely available to us. This will be done using `www.packettotal.com`.

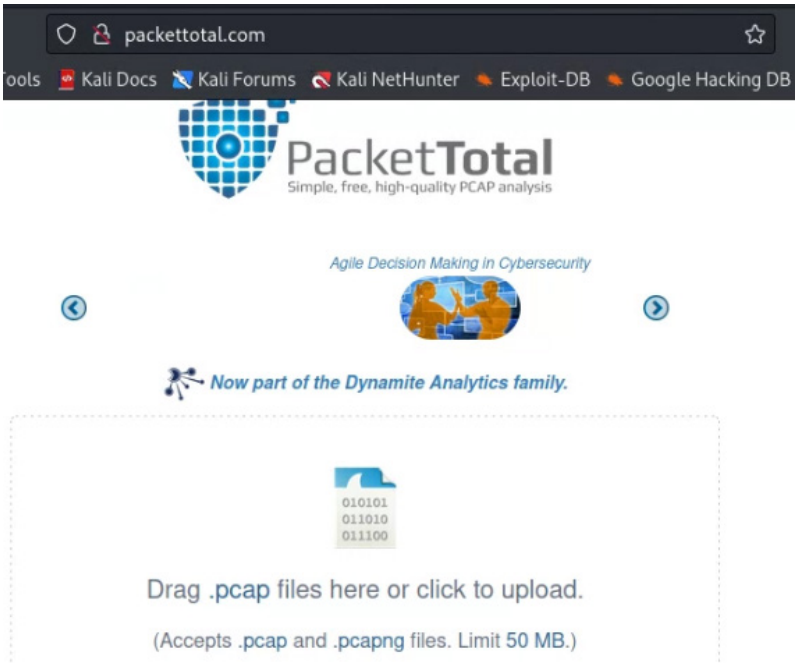


Figure 16.30 – PacketTotal file upload interface

PacketTotal is completely free, with a simple user interface that allows the user to either drag their PCAP file into the upload area or click on the **Upload** button to upload and automatically analyze the `.pcap` file. The only restriction is a limit of 50 MB on `.pcap` file uploads.

We will now download a sample packet capture file and then upload that file to PacketTotal for automated analysis:

1. Download the sample file for analysis at `https://mega.nz/file/6FciHZhY#bF1M-9kwmWLgUfQ_uq2_9k1DICIUkeY4lrT58X8XVXs`. The file is named `Testing for SQL injection.pcap`.

2. Click on the **Upload** button, browse to the downloaded file, and then click on **Open**.

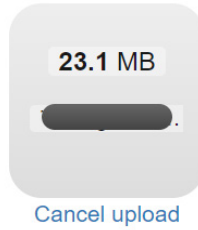


Figure 16.31 – PacketTotal PCAP file upload status

3. You'll have to click on **I'm not a robot** to continue before clicking on the **Analyze** button.

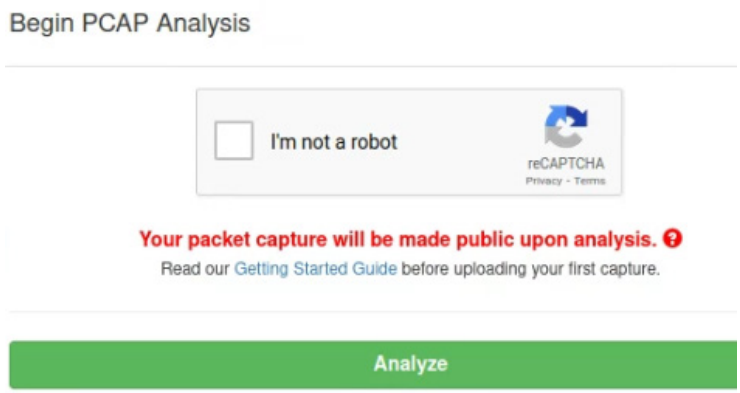


Figure 16.32 – PacketTotal reCAPTCHA verification

4. Once the analysis is complete, PacketTotal gives a very detailed view of the traffic captured. Notice the categories at the top (**Malicious Activity**, **Suspicious Activity**, **Connections**, and various protocols). The **Suspicious Activity** page is first displayed.

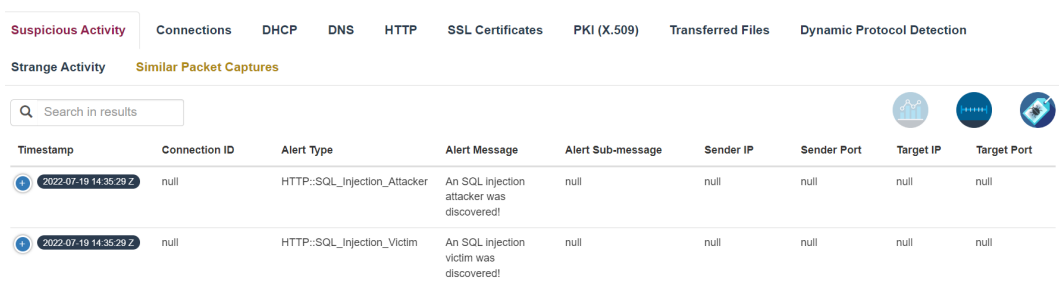
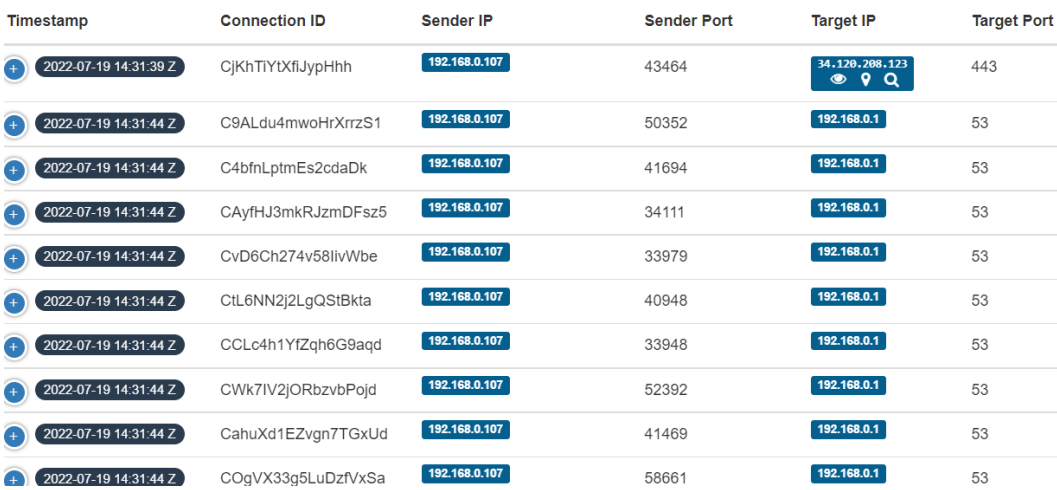


Figure 16.33 – PacketTotal Suspicious Activity tab

We can see that `HTTP::SQL_Injection_Attacker` and `HTTP::SQL_Injection_Victim` alerts were immediately detected!

5. Click on **Connections** next.

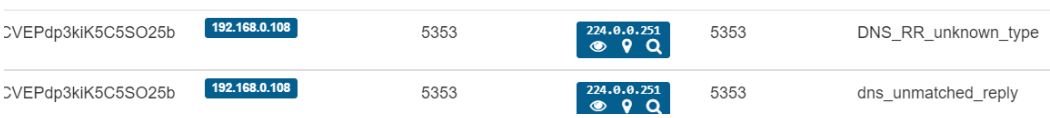
   Here, we see the **Connection ID** along with **Sender** and **Target** IPs to get more information on the `192.168.0.107` IP address.



Timestamp	Connection ID	Sender IP	Sender Port	Target IP	Target Port
2022-07-19 14:31:39 Z	CjKhTiYtXfiJypHhh	192.168.0.107	43464	34.120.208.123	443
2022-07-19 14:31:44 Z	C9ALdu4mwoHrXrrzS1	192.168.0.107	50352	192.168.0.1	53
2022-07-19 14:31:44 Z	C4bfnLptmEs2cdaDk	192.168.0.107	41694	192.168.0.1	53
2022-07-19 14:31:44 Z	CAyfHJ3mkRJzmDFsz5	192.168.0.107	34111	192.168.0.1	53
2022-07-19 14:31:44 Z	CvD6Ch274v58IivWbe	192.168.0.107	33979	192.168.0.1	53
2022-07-19 14:31:44 Z	CtL6NN2j2LgQStBkta	192.168.0.107	40948	192.168.0.1	53
2022-07-19 14:31:44 Z	CCLc4h1YfZqh6G9aqd	192.168.0.107	33948	192.168.0.1	53
2022-07-19 14:31:44 Z	CWk7IV2jORbzvbPojd	192.168.0.107	52392	192.168.0.1	53
2022-07-19 14:31:44 Z	CahuXd1EZvgn7TGxUd	192.168.0.107	41469	192.168.0.1	53
2022-07-19 14:31:44 Z	COgVX33g5LuDzfVxSa	192.168.0.107	58661	192.168.0.1	53

Figure 16.34 – PacketTotal Connections tab

6. Lastly, let us click on **Strange Activity**.



CVEPdp3kiK5C5SO25b	192.168.0.108	5353	224.0.0.251	5353	DNS_RR_unknown_type
CVEPdp3kiK5C5SO25b	192.168.0.108	5353	224.0.0.251	5353	dns_unmatched_reply

Figure 15.35 – PacketTotal Strange Activity tab

7. Feel free to continue your analysis on PacketTotal and the other tools using the freely available PCAP files at `https://www.malware-traffic-analysis.net/index.html` and `https://www.malware-traffic-analysis.net/training-exercises.html`.

Let's move on to our final tool of the chapter and the book.

Online PCAP analysis using apackets.com

For our final exercise, let's look at another online packet analysis tool, available at `https://apackets.com/`. So, let's get started with the exercise:

1. Let's download a sample file from `https://www.malware-traffic-analysis.net/2022/06/07/index.html`. The name of this file is `Emotet-epoch5-infection-with-Cobalt-Strike-.pcap`.

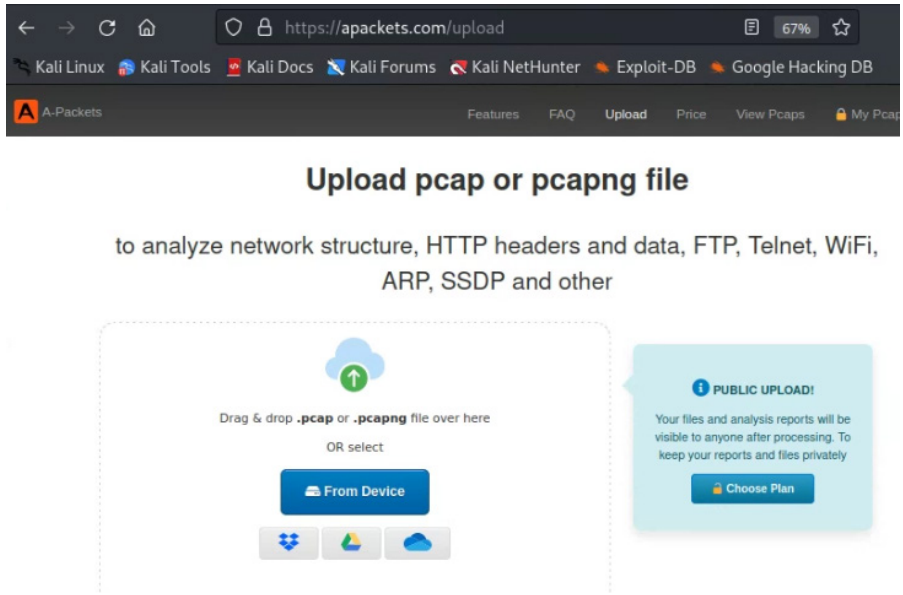


Figure 16.36 – A-Packets file upload interface

2. Click on the **From Device** button and browse to the downloaded sample file to be uploaded for analysis.

Figure 16.37 – A-Packets file upload status

3. Once A-Packets has automatically processed and analyzed our `.pcap` file, a very detailed and categorized breakdown of the findings will be presented, as seen in the following screenshot:

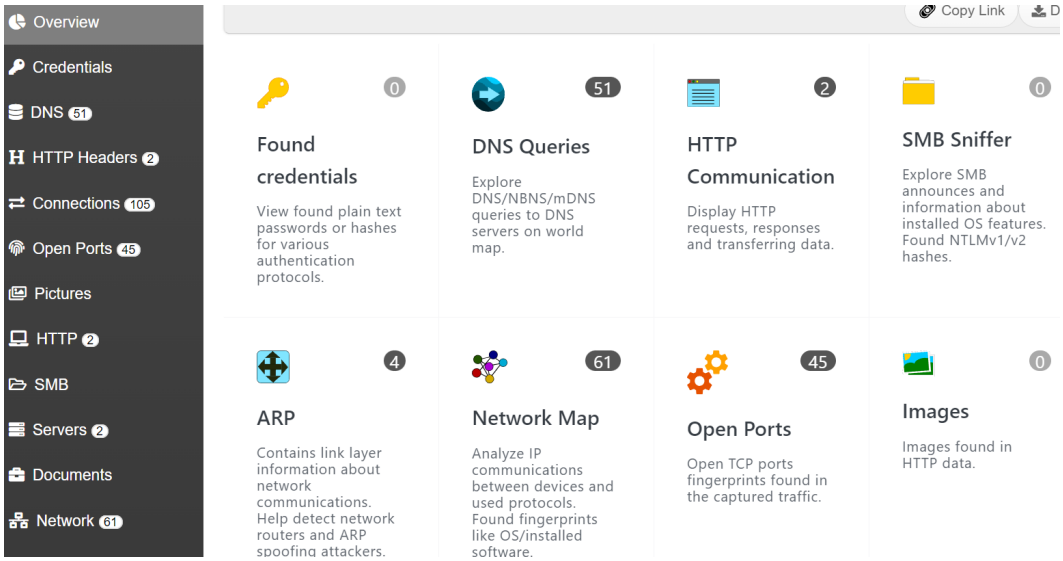


Figure 16.38 – A-Packets analyzed findings

If we scroll down a bit, we should see results for even more findings, including DNS and DHCP protocols and even Ethernet devices used in communications:

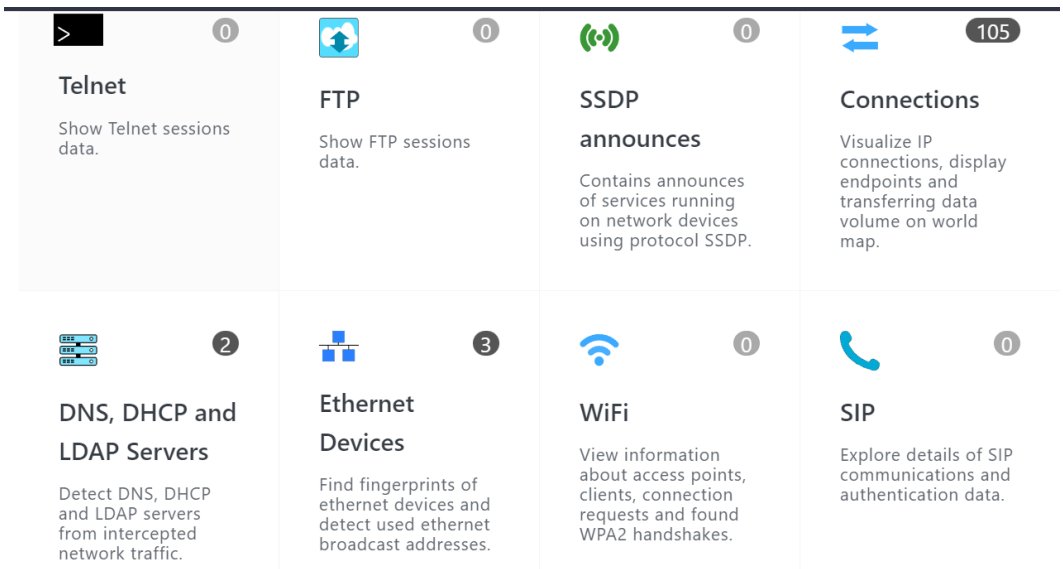


Figure 16.39 – More A-Packets analyzed findings

If you look at the menu to the left of the page, you will notice several tabs, some of which may have numbers next to each tab. These indicate the number of findings per item.

4. As an example, let's click on the **Connections** tab, which displays all connections to the local machine on which the packets were captured, as seen here:

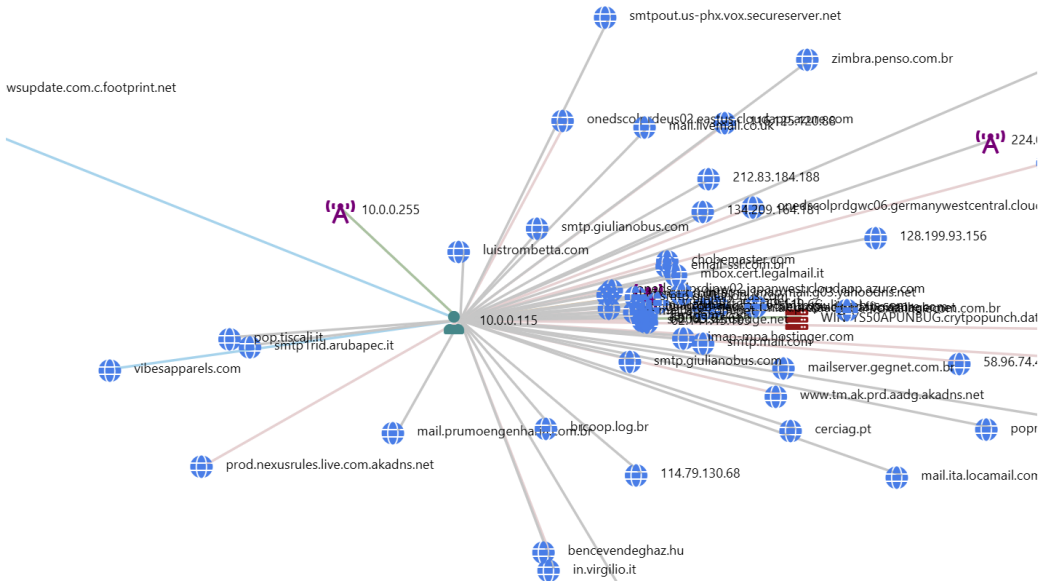


Figure 16.40 – A-Packets Connections tab findings

If you scroll down a bit, you will be presented with a geographical view of communicating devices:

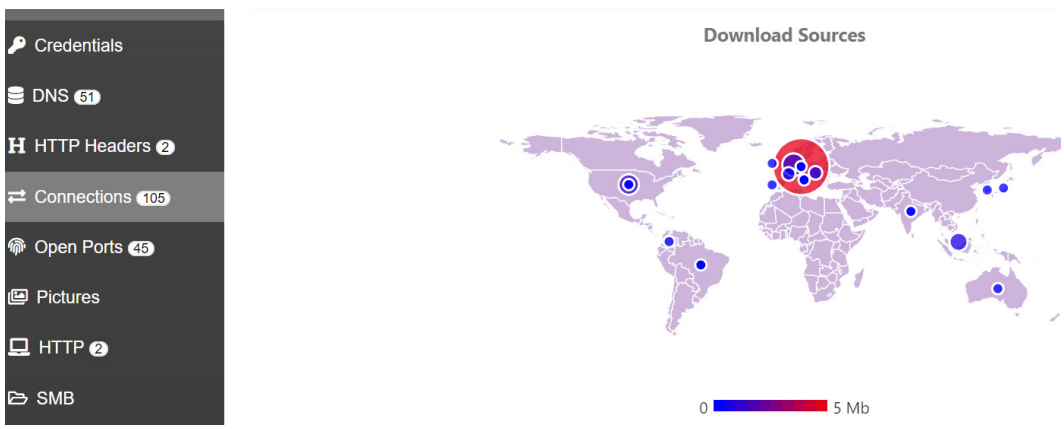


Figure 16.41 – A-Packets Connections tab communications map

5. Now, let's click on the **Open Ports** tab to view the **45** findings discovered by A-Packets, as seen here:

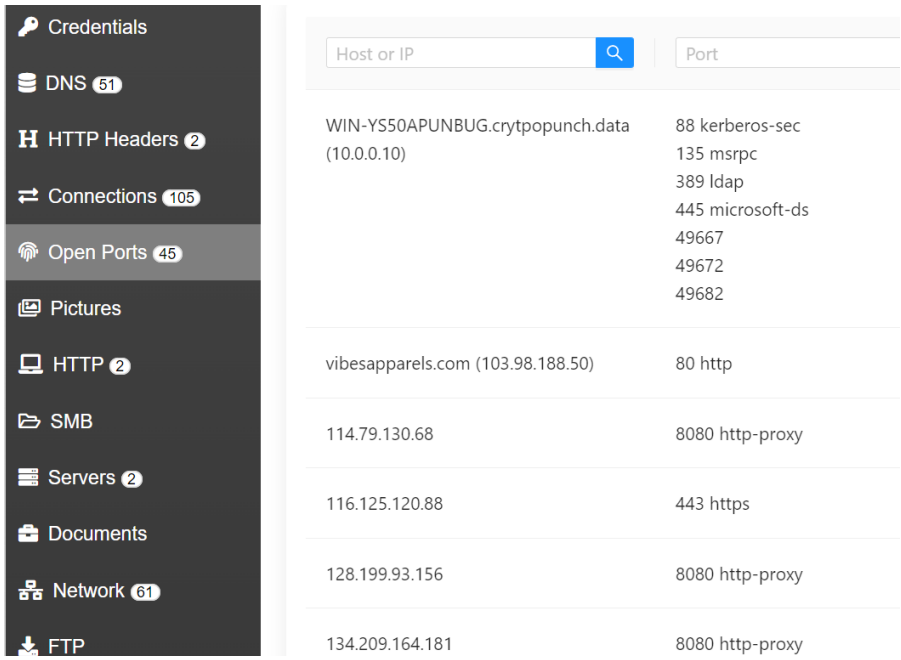


Figure 16.42 – A-Packets Open Ports tab findings

In the preceding screenshot, we can see open ports for various IP addresses involved in this communication.

6. Let's move on to the **Servers** tab to view server communication findings:

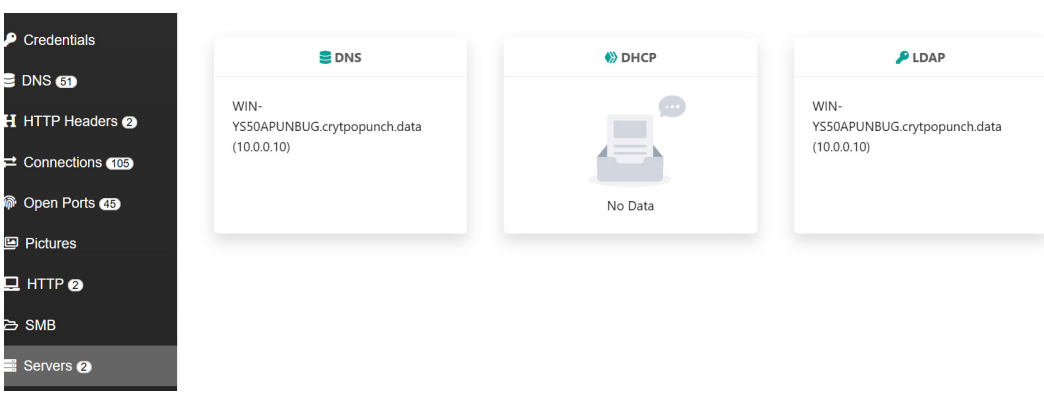


Figure 16.43 – A-Packets Servers tab

7. In the **Servers** tab, we can see DNS information. Let's take a deeper dive into the DNS findings by clicking on the **DNS** tab within the left menu:

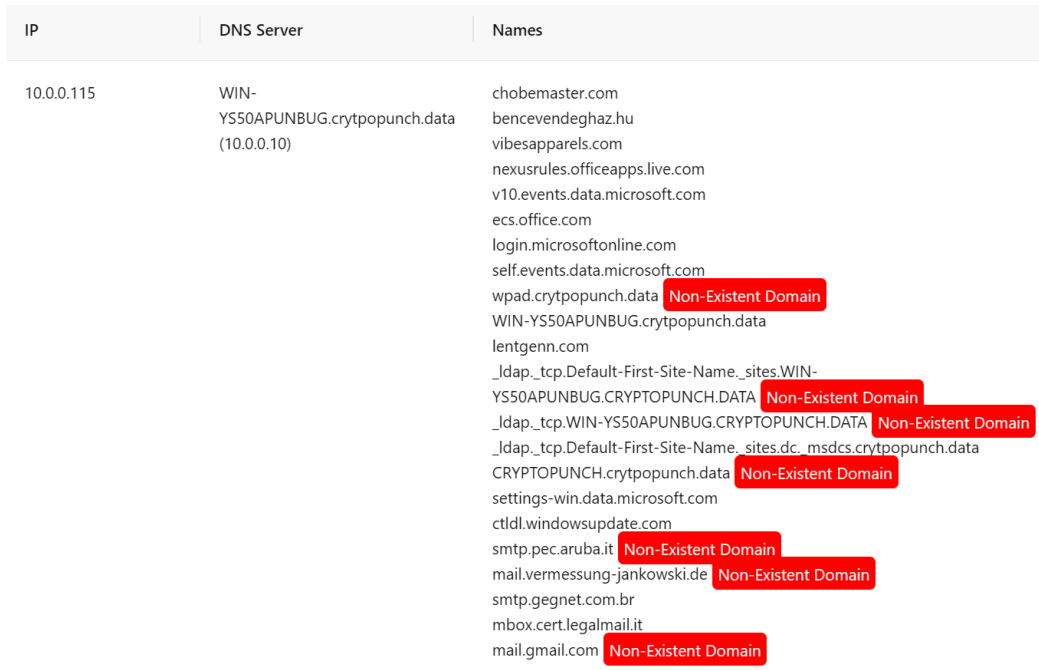


Figure 16.44 – A-Packets DNS tab findings

Viewing the DNS findings in the preceding screenshot, we can see that many non-existent domains populate the list of DNS entries, which may indicate suspicious web activity and communications and should be investigated further.

As you can see, the `apackets.com` website gives us a very detailed analysis of packet captures, and although an online NFAT, it should be part of your network DFIR arsenal. Let's move on to a brief section on reporting and presentation to wrap things up.

Reporting and presentation

Everyone may have their own style when creating and presenting reports. However, the general format is one of a typical analysis report in PDF format. Headings are typically structured as follows:

- Title
- Table of contents
- Investigator and case details
- Scope of works

- Executive summary
- Technical findings
- Appendices

Although again a personal preference, it is customary to include all findings, documentation, and screenshots for full disclosure. Handing over all media containing acquired the evidence, documentation, and findings to the client is also a very common practice to ensure confidentiality.

Summary

Congratulations! You made it to the end of this book, or rather, the DFIR journey with me, as this was our last chapter together! You're now capable of creating your own PCAP files using Wireshark and you can also choose a tool or, as I do, use all tools from this chapter to gather as much information as possible when performing network forensics using Wireshark, NetworkMiner, PcapXray, PacketTotal, and A-Packets.

I hope you enjoyed this book and find it useful in your DFIR investigations. Should you have any questions or need advice, feel free to add me on LinkedIn at `https://tt.linkedin.com/in/shiva-parasram-87a66564`.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

packtpub.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

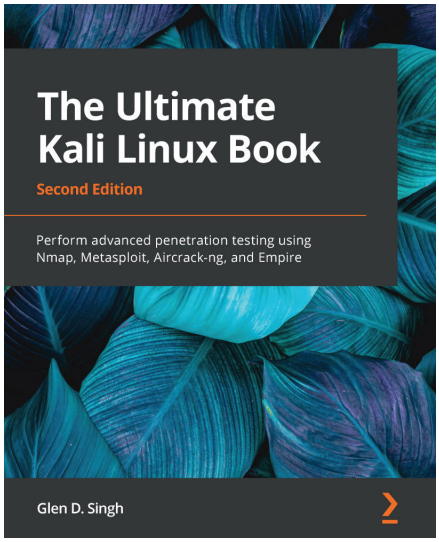
- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at packtpub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packtpub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:

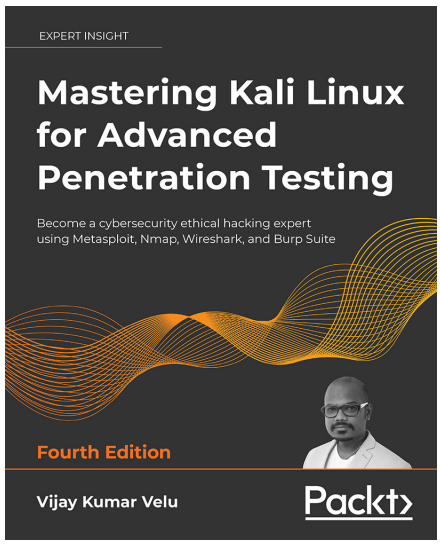


The Ultimate Kali Linux Book - Second Edition

Glen D. Singh

ISBN: 978-1-80181-893-3

- Explore the fundamentals of ethical hacking
- Understand how to install and configure Kali Linux
- Perform asset and network discovery techniques
- Focus on how to perform vulnerability assessments
- Exploit the trust in Active Directory domain services
- Perform advanced exploitation with Command and Control (C2) techniques
- Implement advanced wireless hacking techniques
- Become well-versed with exploiting vulnerable web applications



Mastering Kali Linux for Advanced Penetration Testing - Fourth Edition

Vijay Kumar Velu

ISBN: 978-1-80181-977-0

- Exploit networks using wired/wireless networks, cloud infrastructure, and web services
- Learn embedded peripheral device, Bluetooth, RFID, and IoT hacking techniques
- Master the art of bypassing traditional antivirus and endpoint detection and response (EDR) tools
- Test for data system exploits using Metasploit, PowerShell Empire, and CrackMapExec
- Perform cloud security vulnerability assessment and exploitation of security misconfigurations
- Use bettercap and Wireshark for network sniffing
- Implement complex attacks with Metasploit, Burp Suite, and OWASP ZAP

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share Your Thoughts

Now you've finished *Digital Forensics with Kali Linux*, we'd love to hear your thoughts! If you purchased the book from Amazon, please click here to go straight to the Amazon review page for this book and share your feedback or leave a review on the site that you purchased it from.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below

`https://packt.link/free-ebook/9781837635153`

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly