Week-1

(By PRIYANK KUMAR)

Theoretical Knowledge: Understanding Security Assessment

1. Understanding Security Assessment

Objective

The main objective of a security assessment is to **evaluate the security posture of a system, network, or application** by identifying vulnerabilities, weaknesses, and misconfigurations **without relying on paid or proprietary tools**.
This approach is especially useful for students, beginners, small organizations, and learning environments where **open-source tools and frameworks** are preferred.
Security assessment helps organizations:

- Understand their **current security level**
- Reduce the risk of cyberattacks
- Protect sensitive data
- Meet industry and regulatory requirements

Vulnerability Scanning Tools

- **Nmap** – Scanning
- **OpenVAS** – Assessment
- **Nikto** – Webscan

Reporting / Documentation

- **Google Docs** – Reporting
- **Google Sheets** – Scoring
- **Slack** – Logging

2. What is Security Assessment?

A **Security Assessment** is a systematic process of:

- Identifying security weaknesses
- Evaluating risks
- Measuring how well security controls are implemented

It answers important questions such as:

- What vulnerabilities exist in the system?
- How serious are these vulnerabilities?
- Can an attacker exploit them?
- Are security policies and standards being followed?

Security assessments are usually based on **recognized frameworks and guidelines**, such as:

Security Frameworks (Example: NIST)

Week-1

(By PRIYANK KUMAR)

NIST (National Institute of Standards and Technology) provides structured guidelines like:

- **NIST Cybersecurity Framework (CSF)**
- **NIST SP 800 series**

These frameworks help in:

- Identifying assets
- Protecting systems
- Detecting threats
- Responding to incidents
- Recovering from attacks

Using frameworks ensures that security assessment is **organized, repeatable, and industry-aligned**.

3. Types of Security Testing

Security assessment is performed through different types of testing. Each type focuses on a specific aspect of security.

A. Vulnerability Assessment

A **Vulnerability Assessment** is the process of:

- Scanning systems to find **known security vulnerabilities**
- Identifying outdated software, weak configurations, and missing patches

It focuses on **finding problems**, not exploiting them.

Key Characteristics

- Automated scanning
- Non-intrusive (safe for production systems)
- Risk-based reporting
- Regular and repeatable

Common Open-Source Tool:

OpenVAS

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanner that:

- Scans networks, servers, and applications
- Uses a large vulnerability database
- Detects known CVEs (Common Vulnerabilities and Exposures)

What OpenVAS Can Detect

- Missing security patches

- Weak encryption
- Misconfigured services
- Outdated software versions
- Open ports and services

Output of Vulnerability Assessment

- List of vulnerabilities
- Severity levels (Low, Medium, High, Critical)
- CVE references
- Basic remediation suggestions

Limitations

- Does not prove exploitability
- May produce false positives
- Requires human validation

B. Penetration Testing

Penetration Testing (Pentesting) is an advanced security testing method where:

- Real-world attacks are **simulated**
- Vulnerabilities are actively exploited
- The goal is to test **how far an attacker can go**

Unlike vulnerability assessment, penetration testing focuses on **exploitation and impact**.

Penetration Testing Methodology

1. **Reconnaissance** – Information gathering
2. **Scanning** – Identify open ports and services
3. **Exploitation** – Use vulnerabilities to gain access
4. **Privilege Escalation** – Gain higher permissions
5. **Post-Exploitation** – Maintain access and assess impact
6. **Reporting** – Document findings and risks

Kali Linux Tools for Penetration Testing

1. Nmap (Network Mapper)

- Used for network scanning
- Identifies open ports and running services
- Detects OS and service versions

Example Use:

- Discover attack surface

- Identify entry points

2. Metasploit Framework

- Exploitation framework
- Contains ready-made exploits
- Allows testing real attack scenarios

Metasploit helps in:

- Verifying if a vulnerability is exploitable
- Understanding real-world risks
- Demonstrating impact to management

Benefits of Penetration Testing

- Shows real security risks
- Helps prioritize vulnerabilities
- Tests incident response readiness
- Improves overall defense strategy

Risks

- Can disrupt services if not controlled
- Must be done with authorization
- Requires skilled testers

C. Compliance Testing

Compliance Testing ensures that systems:

- Follow security standards
- Meet regulatory requirements
- Align with best practices

It focuses on **policy and configuration validation**, not attacks.

Examples of Security Standards

- **CIS Benchmarks** (Center for Internet Security)
- **ISO/IEC 27001**
- **NIST guidelines**
- **PCI-DSS** (for payment systems)

CIS Benchmarks

CIS Benchmarks provide:

- Secure configuration guidelines

- Step-by-step checklists
- Best practices for OS, servers, databases, and applications

Examples:

- Disable unnecessary services
- Enforce strong password policies
- Enable logging and auditing
- Apply least privilege principle

How Compliance Testing Works

- Manual or automated checklist verification
- Compare system settings with benchmark standards
- Identify deviations and gaps

Outcome of Compliance Testing

- Compliance score or status
- List of non-compliant controls
- Recommendations for improvement
- Audit-ready documentation

4. Summary

Security assessment is a **core cybersecurity activity** that helps organizations understand and improve their security posture using **cost-effective, open-source tools**.

Type	Purpose	Tools	Focus
Vulnerability Assessment	Identify known weaknesses	OpenVAS	Detection
Penetration Testing	Exploit vulnerabilities	Kali Linux, Metasploit, Nmap	Real-world attack simulation
Compliance Testing	Meet security standards	CIS Benchmarks, NIST	Policy & configuration

Week-1 (By PRIYANK KUMAR)

2. VAPT Methodology (Vulnerability Assessment & Penetration Testing)

Objective

The objective of **VAPT Methodology** is to follow a **structured, repeatable, and ethical approach** to identify, validate, and report security vulnerabilities in systems, networks, and applications.
Using a defined methodology ensures that testing is **systematic**, **legal**, and **aligned with industry best practices**, rather than random or tool-based hacking.

1. What is VAPT?

VAPT (Vulnerability Assessment and Penetration Testing) is a combined security testing approach:

- **Vulnerability Assessment (VA):** Identifies known weaknesses.
- **Penetration Testing (PT):** Actively exploits those weaknesses to assess real-world risk.

A methodology provides:

- Clear testing phases
- Defined scope and rules
- Proper documentation
- Professional and audit-ready outcomes

2. Importance of Using a Methodology

Following a VAPT methodology:

- Prevents accidental damage to systems
- Ensures complete coverage of assets
- Maintains legal and ethical boundaries
- Produces reliable and repeatable results

- Helps in compliance and audits

Commonly aligned frameworks include:

- OWASP Testing Guide
- PTES (Penetration Testing Execution Standard)
- NIST SP 800-115

3. Phases of VAPT Methodology

A standard VAPT methodology is divided into **four main phases**:

Phase 1: Planning

Purpose

The planning phase defines **what will be tested, how it will be tested, and under what rules**.
This is the most critical phase because mistakes here can lead to **legal issues or incomplete testing**.

Key Activities

- Define scope (IP ranges, domains, applications)
- Identify testing type (black-box, grey-box, white-box)
- Get written authorization
- Identify stakeholders and timelines
- Decide reporting format

Tool Example: Dradis CE

Dradis Community Edition (CE) is an open-source collaboration and documentation tool used during planning.

How Dradis Helps:

- Centralizes scope information
- Manages findings and notes
- Maintains communication between team members
- Prepares structured reports

Week-1

(By PRIYANK KUMAR)

Outcome of Planning Phase

- Clear scope definition
- Approved test plan
- Reduced risk of accidental damage
- Legal authorization for testing

Phase 2: Discovery (Information Gathering & Scanning)

Purpose

The discovery phase focuses on **understanding the target environment** by identifying:

- Live systems
- Open ports
- Running services
- Web application vulnerabilities

This phase builds the **attack surface**.

A. Network Discovery – Nmap

Nmap (Network Mapper) is used for:

- Host discovery
- Port scanning
- Service and version detection
- OS fingerprinting

Why Nmap is Important

- Reveals entry points
- Helps identify outdated services
- Provides data for exploitation planning

B. Web Application Discovery – OWASP ZAP

OWASP ZAP (Zed Attack Proxy) is an open-source web application security scanner.

OWASP ZAP Can Detect:

- SQL Injection
- Cross-Site Scripting (XSS)

- Broken authentication
- Security misconfigurations
- Sensitive data exposure

Types of Scanning

- Passive scanning (safe, no attack)
- Active scanning (sends attack payloads)

Outcome of Discovery Phase

- Network map
- List of vulnerabilities
- Identified attack vectors
- Risk prioritization input

Phase 3: Attack (Exploitation)

Purpose

The attack phase validates whether identified vulnerabilities are **actually exploitable** and determines the **real business impact**.

Tool Example: Metasploit Framework

Metasploit Framework is a penetration testing platform used to:

- Exploit known vulnerabilities
- Gain system access
- Perform post-exploitation analysis

Activities in Attack Phase

- Select appropriate exploit modules
- Launch controlled attacks
- Verify access (shell, meterpreter)
- Perform privilege escalation (if permitted)
- Capture evidence (screenshots, logs)

Why Exploitation Matters

- Confirms true risk (not just theoretical)
- Helps eliminate false positives
- Demonstrates real-world attacker capability

Week-1

(By PRIYANK KUMAR)

Ethical Considerations

- Exploitation must stay within scope
- No data destruction
- Minimal service disruption

Outcome of Attack Phase

- Confirmed vulnerabilities
- Proof of exploitation
- Impact analysis
- Evidence for reporting

Phase 4: Reporting

Purpose

Reporting converts technical findings into **clear, actionable insights** for both technical teams and management.

Reporting Tool Example: Pentest-Tools Templates

Pentest-Tools provides structured report templates that include:

- Executive summary
- Risk ratings
- Vulnerability descriptions
- Proof of concept
- Remediation recommendations

Key Elements of a Good VAPT Report

- Clear vulnerability explanation
- Severity classification (CVSS)
- Business impact
- Reproducible steps
- Mitigation guidance

Audience of the Report

- Management (high-level risk overview)

- Technical teams (fix implementation)
- Auditors and compliance teams

Outcome of Reporting Phase

- Professional security report
- Remediation roadmap
- Audit-ready documentation

4. How to Learn VAPT Methodology

OWASP Web Security Testing Framework (WSTG)

The **OWASP WSTG** is a practical guide that:

- Provides structured testing categories
- Covers authentication, authorization, input validation, session management
- Maps vulnerabilities to real test cases

Benefits of Learning OWASP WSTG

- Industry-recognized standard
- Step-by-step testing approach
- Improves web application security skills
- Aligns with real-world pentesting jobs

5. Summary Table

Phase	Purpose	Tools Used
Planning	Define scope & rules	Dradis CE
Discovery	Identify attack surface	Nmap, OWASP ZAP
Attack	Exploit vulnerabilities	Metasploit Framework
Reporting	Document findings	Pentest-Tools Templates
Learning	Skill development	OWASP WSTG

Week-1 (By PRIYANK KUMAR)

3. Security Standards & Compliance

Objective

The objective of **Security Standards and Compliance** is to ensure that organizations **protect sensitive data, manage security risks, and follow legal and regulatory requirements**.

By aligning systems and processes with recognized standards, organizations can:

- Reduce cybersecurity risks
- Avoid legal penalties
- Build trust with users and customers
- Demonstrate accountability and governance

1. What Are Security Standards & Compliance?

Security Standards

Security standards are **formal guidelines and best practices** that define:

- How information should be protected
- What security controls must be implemented
- How risks should be managed

They act as a **blueprint for building and maintaining secure systems**.

Compliance

Compliance means **following and proving adherence** to:

- Laws
- Regulations
- Industry standards

Non-compliance can lead to:

- Heavy fines
- Legal action
- Loss of reputation

- Business disruption

2. Why Security Standards Are Important

Security standards help organizations:

- Create a structured security program
- Standardize security practices across teams
- Improve incident response readiness
- Ensure continuous improvement
- Support audits and certifications

They shift security from being **reactive** to **proactive**.

3. Major Security Standards

A. GDPR (General Data Protection Regulation)

GDPR is a data protection regulation enforced by the **European Union (EU)**.
It applies to **any organization worldwide** that processes personal data of EU citizens.

Key Objectives of GDPR

- Protect personal data
- Give users control over their information
- Ensure transparency in data processing

Key GDPR Principles

1. Lawfulness, Fairness, and Transparency
2. Purpose Limitation
3. Data Minimization
4. Accuracy
5. Storage Limitation
6. Integrity and Confidentiality
7. Accountability

Security Requirements Under GDPR

- Encryption of personal data
- Access control and authentication
- Logging and monitoring
- Incident and breach notification within **72 hours**
- Regular security assessments

Penalties

- Fines up to **€20 million or 4% of global annual turnover**

Relation to Cybersecurity

GDPR directly links data protection with:

- Vulnerability management
- Secure application development
- Incident response planning

B. HIPAA (Health Insurance Portability and Accountability Act)

Overview

HIPAA is a US regulation focused on protecting **healthcare data** known as **PHI (Protected Health Information)**.

Key Objectives

- Protect patient privacy
- Ensure secure handling of medical data
- Prevent unauthorized access

HIPAA Security Rule

The HIPAA Security Rule requires three types of safeguards:

1. Administrative Safeguards

- Risk analysis
- Security training
- Incident response procedures

2. Physical Safeguards

- Secure access to facilities
- Device and media controls

3. Technical Safeguards

- Access control
- Audit logs
- Transmission security (encryption)

Cybersecurity Focus

HIPAA compliance requires:

- Secure networks
- Vulnerability scanning
- Access monitoring
- Strong authentication mechanisms

Penalties

- Civil and criminal penalties
- Fines per violation category

C. ISO/IEC 27001

ISO/IEC 27001 is an international standard for establishing an Information Security Management System (ISMS).
Unlike GDPR or HIPAA, ISO 27001 is:

- Voluntary
- Certification-based
- Applicable to all industries

Core Objective

To systematically manage:

- Confidentiality
- Integrity
- Availability (CIA Triad)

Key Components of ISO 27001

1. Risk assessment and risk treatment
2. Security policies and procedures
3. Asset management
4. Access control
5. Incident management
6. Business continuity
7. Continuous improvement (PDCA cycle)

Annex A Controls

ISO 27001 includes **Annex A**, which lists:

- Organizational controls
- Technical controls
- Physical controls

Examples:

- Least privilege
- Secure logging
- Network security
- Supplier security

Why ISO 27001 Is Important

- Provides structured governance
- Builds customer trust
- Helps meet other regulatory requirements
- Demonstrates security maturity

4. Relationship Between Standards and Compliance

Standard	Focus Area	Mandatory
GDPR	Personal data protection	Yes
HIPAA	Healthcare data	Yes (US)
ISO 27001	Information security management	No (certification-based)

5. How to Learn Security Standards Using OWASP Top 10

What Is OWASP Top 10?

The **OWASP Top 10** is a globally recognized list of the **most critical web application security risks**.

It helps security professionals prioritize vulnerabilities that:

- Have high impact
- Are commonly exploited
- Affect compliance directly

Why OWASP Top 10 Is Important for Compliance

Most compliance failures happen due to:

- Poor authentication
- Data exposure
- Injection flaws
- Security misconfigurations

These issues are directly covered in the **OWASP Top 10**.

Examples of OWASP Top 10 Mapping to Compliance

Week-1

(By PRIYANK KUMAR)

OWASP Risk	Compliance Impact
Broken Access Control	GDPR & HIPAA violation
Cryptographic Failures	Data breach penalties
Injection Attacks	Data integrity compromise
Security Misconfiguration	Audit failures
Identification & Authentication Failures	Unauthorized data access

OWASP Top 10 – Web Application Security Risks

1. A01: Broken Access Control
2. A02: Cryptographic Failures
3. A03: Injection
4. A04: Insecure Design
5. A05: Security Misconfiguration
6. A06: Vulnerable and Outdated Components
7. A07: Identification and Authentication Failures
8. A08: Software and Data Integrity Failures
9. A09: Security Logging and Monitoring Failures
10. A10: Server-Side Request Forgery (SSRF)

Learning Strategy

1. Study each OWASP Top 10 category
2. Understand real-world attack scenarios
3. Practice detection using OWASP ZAP
4. Implement secure coding practices
5. Map vulnerabilities to compliance controls

6. Practical Compliance Learning Approach

- Use **OWASP Top 10** as a baseline checklist
- Perform vulnerability assessments
- Align findings with:
    - GDPR technical safeguards
    - HIPAA security rule
    - ISO 27001 Annex A controls
- Document remediation steps

7. Summary

Security standards and compliance are not just legal requirements; they form the **foundation of modern cybersecurity governance**.

- **GDPR** protects personal data
- **HIPAA** secures healthcare information
- **ISO 27001** provides a complete security management framework
- **OWASP Top 10** helps prioritize and implement practical security controls

4. Risk Assessment Basics

Objective

The objective of **Risk Assessment** is to **prioritize vulnerabilities** so that security teams can focus on **what matters most** instead of fixing issues randomly.
Since time, budget, and resources are limited, risk assessment helps answer:

- Which vulnerability is **most dangerous**?
- Which issue should be fixed **first**?
- What is the **business impact** if exploited?

Risk assessment converts **technical vulnerabilities into business risks**.

1. What is Risk Assessment?

Definition

Risk assessment is the process of:

- Identifying vulnerabilities
- Evaluating their severity and exploitability
- Measuring potential impact
- Assigning a risk level (High / Medium / Low)

A common formula used in cybersecurity:

Risk = Likelihood × Impact

2. Why Risk Assessment is Important

Without risk assessment:

- Teams waste time fixing low-impact issues
- Critical vulnerabilities remain unpatched
- Management cannot understand technical findings

With risk assessment:

- Critical risks are fixed first

- Better decision-making
- Improved compliance and audit readiness
- Clear communication with management

3. CVSS (Common Vulnerability Scoring System)

What is CVSS?

CVSS is an industry-standard scoring system used to **numerically rate the severity of vulnerabilities** on a scale of **0.0 to 10.0**.

It is maintained by **FIRST** and used globally by:

- NVD (National Vulnerability Database)
- Security vendors
- SOC and VAPT teams

CVSS Score Levels

Score Range	Severity
0.0	None
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 – 10.0	Critical

4. CVSS Calculator (NVD) – Free Tool

What is NVD CVSS Calculator?

The **NVD CVSS Calculator** is a **free, web-based tool** used to calculate CVSS scores based on vulnerability characteristics.

Installation / Access Tips (Free)

✅ **No installation required**

You can access it directly via browser:

- Search: **"NVD CVSS Calculator v3.1"**
- Works on Chrome, Firefox, Edge
- No login required
- 100% free

How CVSS Calculation Works

CVSS is divided into **three metric groups**:

A. Base Metrics (Core Severity)

These never change and describe the vulnerability itself:

- **Attack Vector (AV)** – Network, Adjacent, Local, Physical
- **Attack Complexity (AC)** – Low / High
- **Privileges Required (PR)** – None / Low / High
- **User Interaction (UI)** – Required / None
- **Scope (S)** – Changed / Unchanged
- **Confidentiality (C)** – None / Low / High
- **Integrity (I)** – None / Low / High
- **Availability (A)** – None / Low / High

B. Temporal Metrics (Optional)

- Exploit maturity
- Remediation level
- Report confidence

C. Environmental Metrics (Organization-Specific)

- Business impact
- Asset importance
- Customized severity

Why CVSS is Important

- Standardized severity scoring
- Accepted globally
- Used in audits and reports
- Helps prioritize vulnerabilities objectively

Limitations of CVSS

- Does not consider business context fully
- Same score may have different impact in different organizations
- Needs risk matrix for better prioritization

5. Risk Matrix

A **Risk Matrix** visually categorizes risks based on:

- **Likelihood** (how easy it is to exploit)
- **Impact** (damage to business)

It converts CVSS scores into **business-friendly risk levels**.

Week-1

(By PRIYANK KUMAR)

Typical Risk Matrix

Impact ↓ / Likelihood →	Low	Medium	High
Low Impact	Low	Low	Medium
Medium Impact	Low	Medium	High
High Impact	Medium	High	Critical

6. Risk Categorization

Risk Level	Meaning
High	Immediate fix required
Medium	Fix in planned timeline
Low	Accept or monitor

7. Tools for Risk Matrix (Free)

A. Google Sheets (Free & Online)

Why Google Sheets?

- Free
- Cloud-based
- Easy sharing
- No installation required

How to Use

1. Create a new spreadsheet
2. Add columns:
    - Vulnerability Name
    - CVSS Score
    - Likelihood
    - Impact
    - Risk Level
3. Use conditional formatting:
    - Red → High
    - Yellow → Medium
    - Green → Low

Week-1

(By PRIYANK KUMAR)

B. Microsoft Excel (Offline Option)

Installation Tips (Free Options)

- Excel Online (free with Microsoft account)
- LibreOffice Calc (100% free alternative)

LibreOffice Calc (Free Alternative)

- Open-source
- Offline
- No license cost
- Works like Excel

8. Practical Learning Approach (Step-by-Step)

1. Scan system using OpenVAS / Nmap
2. Identify vulnerabilities
3. Look up CVE on NVD
4. Calculate CVSS score using NVD Calculator
5. Add vulnerability to Risk Matrix
6. Assign High / Medium / Low
7. Recommend remediation

9. Example (Real-World)

- Vulnerability: SQL Injection
- CVSS Score: 9.8
- Likelihood: High
- Impact: High
- **Final Risk: Critical**

10. Summary

- **CVSS Calculator** provides technical severity scoring
- **Risk Matrix** translates technical data into business risk
- Free tools like **NVD, Google Sheets, LibreOffice** are enough
- Risk assessment ensures **smart prioritization**

5. Common Vulnerabilities

Objective

The objective of studying **common vulnerabilities** is to **identify, understand, and safely exploit security flaws in controlled lab environments**.
This helps security professionals recognize real-world attack patterns and improve defensive measures.

1. What Are Common Vulnerabilities?

Definition

Common vulnerabilities are **frequently occurring security weaknesses** in:

- Networks
- Systems
- Applications

These vulnerabilities usually arise due to:

- Misconfigurations
- Poor secure coding practices
- Lack of patching
- Weak authentication and authorization

Understanding these flaws is the **foundation of Vulnerability Assessment and Penetration Testing (VAPT)**.

2. Network Vulnerabilities

What Are Network Vulnerabilities?

Network vulnerabilities are weaknesses in network services, protocols, or configurations that can be exploited by attackers.

A. Misconfigurations

Examples

- Default credentials enabled
- Unnecessary services running
- Weak firewall rules
- Improper access controls

Impact

- Unauthorized access
- Lateral movement
- Data leakage

B. Open Ports

Open ports indicate **services listening for incoming connections**.
While some are required, unnecessary open ports increase the **attack surface**.

Tool: Nmap (Free & Open Source)

Nmap is the most widely used tool for identifying:

- Live hosts
- Open ports
- Running services
- OS and version details

What Nmap Reveals

- FTP, SSH, HTTP, SMB services
- Outdated software versions
- Potential exploitation points

Why Network Vulnerabilities Matter

- Open ports allow entry points
- Misconfigured services can be exploited remotely
- Often used as the **first step** in an attack

3. Web Application Vulnerabilities

Web vulnerabilities target **application logic and user input handling**.

A. SQL Injection (SQLi)

SQL Injection occurs when **untrusted input is directly included in database queries**, allowing attackers to:

- Read sensitive data
- Modify database contents
- Bypass authentication
- Execute administrative operations

Root Causes

- Lack of input validation
- Dynamic SQL queries
- Missing parameterized queries

Impact

- Database compromise
- Data breach
- Compliance violations (GDPR, HIPAA)

B. Cross-Site Scripting (XSS)

What is XSS?

XSS allows attackers to **inject malicious scripts into web pages**, which are then executed in the victim's browser.

Types of XSS

- Stored XSS
- Reflected XSS
- DOM-based XSS

Impact

- Session hijacking
- Credential theft
- Defacement
- Malware delivery

4. Practice Platform: OWASP Juice Shop

What is OWASP Juice Shop?

OWASP Juice Shop is an **intentionally vulnerable web application** designed for learning and practicing web security.

Why Use Juice Shop?

- Covers OWASP Top 10 vulnerabilities
- Safe, legal learning environment
- Realistic attack scenarios
- Free and open source

What You Can Practice

- SQL Injection
- XSS
- Broken authentication
- Security misconfigurations
- Business logic flaws

Installation Tips (Free)

Docker (Recommended)

- Requires Docker Desktop (free)

5. Learning Through Vulnerable Labs

Hands-on labs are essential for understanding **how vulnerabilities are discovered and exploited**.

A. Metasploitable

Metasploitable is an intentionally vulnerable Linux virtual machine created by Rapid7.

Purpose

- Practice network and system exploitation
- Learn service-based attacks
- Understand post-exploitation techniques

Common Vulnerabilities in Metasploitable

- Weak FTP services
- Vulnerable SMB
- Outdated web servers
- Default credentials

Installation Tips (Free)

- Use VirtualBox or VMware Player (free)
- Import Metasploitable VM
- Run in host-only or NAT network

Week-1

(By PRIYANK KUMAR)

B. VulnHub

What is VulnHub?

VulnHub is a platform providing **vulnerable virtual machines** for practicing penetration testing skills.

Why VulnHub is Useful

- Real-world style challenges
- Multiple difficulty levels
- Community walkthroughs
- Covers web, network, and system attacks

Skills Gained

- Enumeration
- Exploitation
- Privilege escalation
- Report writing

Installation Tips

- Download VM image
- Import into VirtualBox/VMware
- Isolated lab setup only

6. Safe Lab Practices

- Always practice on **intentionally vulnerable systems**
- Never scan public or unauthorized networks
- Use host-only or internal VM networks
- Take VM snapshots before attacks

7. Learning Roadmap (Beginner to Intermediate)

1. Learn Nmap basics
2. Scan Metasploitable
3. Identify open ports
4. Practice exploitation using Metasploit
5. Practice SQLi and XSS on Juice Shop
6. Solve VulnHub machines
7. Document findings

8. Summary Table

Category	Vulnerability	Practice Tool
Network	Open ports, misconfiguration	Nmap, Metasploitable
Web	SQLi, XSS	OWASP Juice Shop
System	Privilege escalation	VulnHub
Learning	Safe exploitation	Virtual Labs

6. Documentation Fundamentals

Objective

The objective of **documentation fundamentals** is to **properly record, organize, and present security findings** in a clear and professional way.
Good documentation helps convert **technical testing results** into **useful information** that management, developers, and security teams can understand and act upon.
In cybersecurity, **testing is incomplete without documentation**.

1. Why Documentation is Important in Cybersecurity

Documentation is important because:

- Not everyone understands technical commands
- Management needs risk-based explanations
- Developers need clear steps to fix issues
- Auditors need written proof
- Future teams need reference material

Without proper documentation:

- Findings are forgotten
- Fixes are delayed
- Compliance fails
- Knowledge is lost

2. What is Security Documentation?

Security documentation is the process of:

- Writing down what was tested
- Recording vulnerabilities found
- Explaining how vulnerabilities were identified
- Providing evidence (screenshots, logs)

- Suggesting remediation steps

Documentation acts as a **bridge** between:

- Technical security teams
- Non-technical stakeholders

3. Types of Documentation in VAPT

1. Notes during testing
2. Final security report
3. Remediation and follow-up documentation

Each stage requires different tools.

4. Tools for Documentation

A. Dradis CE (Community Edition)

Dradis CE is a **free and open-source reporting and collaboration tool** used by penetration testers and security teams.

It allows teams to:

- Collect findings in one place
- Collaborate during assessments
- Generate professional reports

Why Dradis is Useful

- Centralized documentation
- Reduces duplicate work
- Easy reporting
- Team collaboration

What You Can Document in Dradis

- Scope details
- Vulnerability findings
- CVSS scores
- Evidence screenshots
- Remediation recommendations

Simple Example

Instead of keeping notes in many files, Dradis stores everything in **one project**.

Week-1

(By PRIYANK KUMAR)

Installation Tips (Free)

- Runs on Linux, Windows (via Docker), and macOS
- Dradis CE is free forever
- Suitable for individuals and teams

Who Uses Dradis?

- Penetration testers
- VAPT teams
- Security consultants
- Students learning reporting

B. CherryTree (Note-Taking Tool)

CherryTree is a **free hierarchical note-taking application** that helps organize technical findings.

It works like a **tree structure**:

- Main topic
- Subtopics
- Detailed notes inside

Why CherryTree is Popular in Pentesting

- Keeps notes organized
- Supports code snippets
- Supports screenshots
- Easy navigation

How CherryTree Helps

During testing, you may run:

- Nmap scans
- Exploits
- Payloads

CherryTree lets you:

- Save commands
- Store outputs
- Add screenshots
- Write explanations

Example Structure

- Target Information

  - IP Address
  - Open Ports
- Vulnerabilities
  - SQL Injection
  - XSS

Installation Tips (Free)

- Available for Linux, Windows, macOS
- Lightweight and fast
- No internet required

Best Use Case

- Personal pentesting notes
- Learning labs (VulnHub, Metasploitable)
- Exam and interview preparation

C. Using Your Own Standard Reporting Tool

You do **not need special tools** to document security findings.

Commonly Used Standard Tools

- Microsoft Word
- Google Docs
- LibreOffice Writer
- Markdown editors
- Notion

Why This is Acceptable

- Reporting format matters more than tools
- Clarity and structure are key
- Many companies use custom templates

Basic Report Sections

1. Title Page
2. Executive Summary
3. Scope
4. Methodology
5. Findings
6. Risk Ratings
7. Recommendations

8. Conclusion

5. How to Learn Documentation

Using Free Templates from GitHub

GitHub has many **free VAPT and pentest report templates** created by professionals.

Why Use Templates?

- Saves time
- Teaches professional structure
- Reduces mistakes
- Industry-ready format

What Templates Usually Include

- Pre-written sections
- Risk rating tables
- Vulnerability descriptions
- Remediation examples

GitHub examples of free report templates

1. Pentesting Report LaTeX Template

Use it: Write professional reports with structure, CVSS badges, summaries, technical sections, and easy formatting

👉 **Template link:** https://github.com/profi248/pentest-report GitHub

⭐ This is a **full LaTeX-based template** that you can edit on your computer.
You can use tools like **Overleaf (online LaTeX editor)** or install **LuaLaTeX** to compile it into a finished PDF.
It has:

- Automated CVSS scoring boxes
- Summary table placeholders
- Ready vulnerability sections

👉 Excellent for **professional-level reports**. GitHub

2. VAPT Report Example (DVWA Lab)

Use it: See a *real documented VAPT report*, with practical sections, CVSS scores, findings and remediation

👉 **Report project:** https://github.com/Ninadjos/DVWA-VAPT-Report GitHub

Week-1

(By PRIYANK KUMAR)

⭐ This repository includes:

- Full assessment of DVWA (Damn Vulnerable Web App)
- Markdown and Microsoft Word report
- Vulnerability-by-vulnerability breakdown
- Screenshots and evidence in folders
- CVSS scores & severity tables

👉 Very useful to **learn structure and real content organisation**. GitHub

3. Simple Web Pentesting Sample Report

Use it: A beginner-friendly filled-in Word/PDF sample report

👉 **Sample report:** https://github.com/h0tPlug1n/Web-Penetration-Testing-Report-Sample GitHub

⭐ Contains:

- A sample test report in **.docx and .pdf**
- Sections like **Introduction, Scope, Summary, Risk Classification, PoCs**
- Ready parts to edit for your own work

👉 Ideal for beginners who need a **ready layout** to copy and fill. GitHub

✅ **4. Pentesting Report Template (Microsoft Word / Markdown)**

Use it: A basic, editable template you can start with right away

👉 Another free template collection: https://github.com/mtk911/pentest-report-template awesome.ecosyste.ms

⭐ Includes:

- Simple **pentest report structure**
- Uses **DREAD risk model**
- Word and Markdown formats
- Easy to customize for your own findings

👉 Great for **students or beginners who prefer Word/Markdown**. awesome.ecosyste.ms

⚡ **Bonus Resource**

📌 **Free Pentest Templates Collection**

👉 Browse many **download-ready templates (Word/LaTeX/Markdown)** here: https://pentestreports.com/templates pentestreports.com

⭐ Includes formats like:

- Markdown templates for exam reports (OSCP style)

Week-1

(By PRIYANK KUMAR)

- Word templates suitable for VAPT
- Example structures you can reuse

  👉 Good if you want **multiple formats** and choose what you like.

  pentestreports.com

✅ **Tips for Using These Templates**

🎯 **Download & customize**

- Clone or download the GitHub repo
- Replace sample details with your own lab findings
- Add screenshots, CVSS scores, risk ratings

What to include in your report

1. **Executive Summary** – high-level overview for management
2. **Scope & Objectives** – what was tested
3. **Methodology** – tools and approach
4. **Findings** – list of vulnerabilities
5. **CVSS & Risk Matrix** – priority scores
6. **Evidence / PoC** – screenshots, commands
7. **Remediation** – how to fix each issue
8. **Conclusion & Recommendations**

Tools that help with templates

- **Google Docs / Sheets** – for editing and tables
- **Word / LibreOffice** – for offline editing
- **Markdown editors** – for GitHub/Dev reports

Learning Tip

After downloading a template:

1. Try editing it with your own lab results (e.g., OWASP Juice Shop findings)
2. Add your screenshots and CVSS scores
3. Share with a teacher, mentor, or security community for feedback

How to Use Them

1. Download template
2. Replace sample content
3. Add your findings
4. Customize based on project

Week-1

(By PRIYANK KUMAR)

6. Documentation Workflow

1. **During Testing**
    - Use CherryTree to save notes
    - Capture screenshots
    - Write commands and outputs
2. **After Testing**
    - Move findings to Dradis or report template
    - Assign severity (CVSS)
    - Add remediation steps
3. **Final Stage**
    - Generate final report
    - Review for clarity
    - Share with stakeholders

7. Common Documentation Mistakes to Avoid

- Using too much technical jargon
- Missing screenshots or evidence
- No clear remediation steps
- Poor structure
- Inconsistent severity ratings

8. Benefits of Good Documentation

- Professional image
- Easy audits
- Faster remediation
- Knowledge reuse
- Career growth

9. Summary Table

Tool	Purpose	Best For
Dradis CE	Collaborative reporting	Team-based VAPT
CherryTree	Technical note-taking	Individual testing
Word / Docs	Final reports	Management & audits

Week-1

(By PRIYANK KUMAR)

Tool	Purpose	Best For
GitHub Templates	Learning & structure	Beginners

Practical

Objective

The objective of this experiment is to install **Kali Linux operating system** using **VMware Workstation** in a virtualized environment for penetration testing and cybersecurity learning purposes.

3. System Requirements

3.1 Hardware Requirements

- Processor: Intel / AMD (64-bit)
- RAM: Minimum 4 GB (Recommended 8 GB)
- Disk Space: Minimum 30 GB
- Virtualization: Enabled in BIOS

3.2 Software Requirements

- Host OS: Windows 10 / Windows 11
- VMware Workstation Player
- Kali Linux ISO / VMware Image

4. Tools Used

- **VMware Workstation Player**
- **Kali Linux (64-bit)**
- Internet Browser (for downloading files)

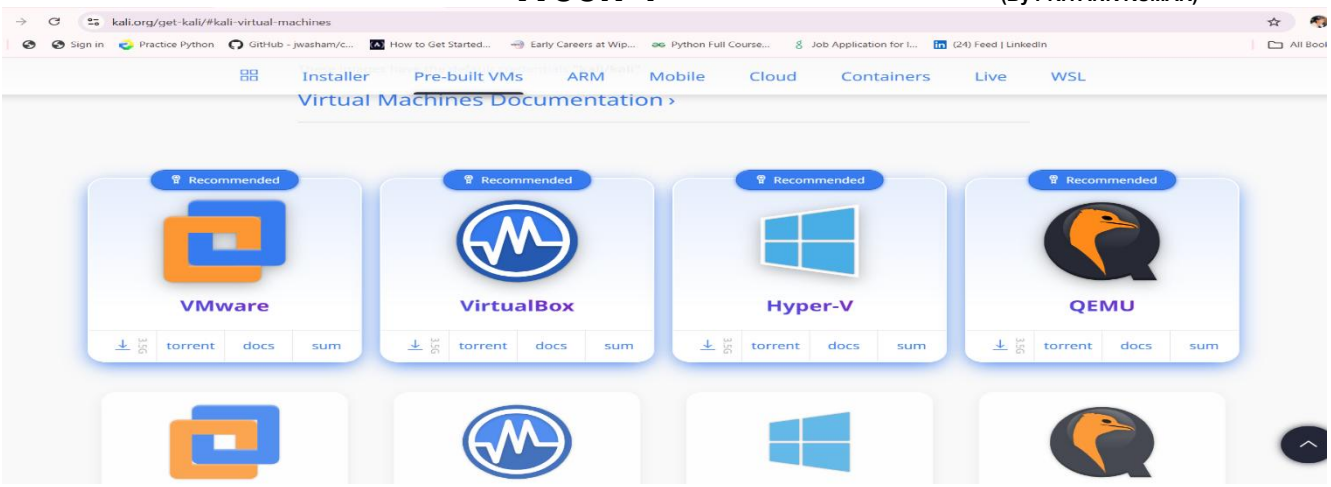
5. Installation Procedure

Step 1: Download VMware Workstation

1. Open a web browser
2. Download **VMware Workstation Player**
3. Install VMware using default settings

Week-1

(By PRIYANK KUMAR)



Step 2: Download Kali Linux

1. Visit the Kali Linux official website
2. Download **Kali Linux VMware Image**
3. Extract the downloaded .zip file

Step 3: Open Kali Linux in VMware

1. Open **VMware Workstation**
2. Click **Open a Virtual Machine**
3. Select the Kali Linux .vmx file
4. Click **Open**

Step 4: Configure Virtual Machine

- Memory: 2–4 GB RAM
- Processor: 2 cores
- Network Adapter: NAT / Bridged
- Display: Default

Step 5: Start Kali Linux

1. Click **Power On This Virtual Machine**
2. Kali Linux boot screen appears
3. Login using default credentials

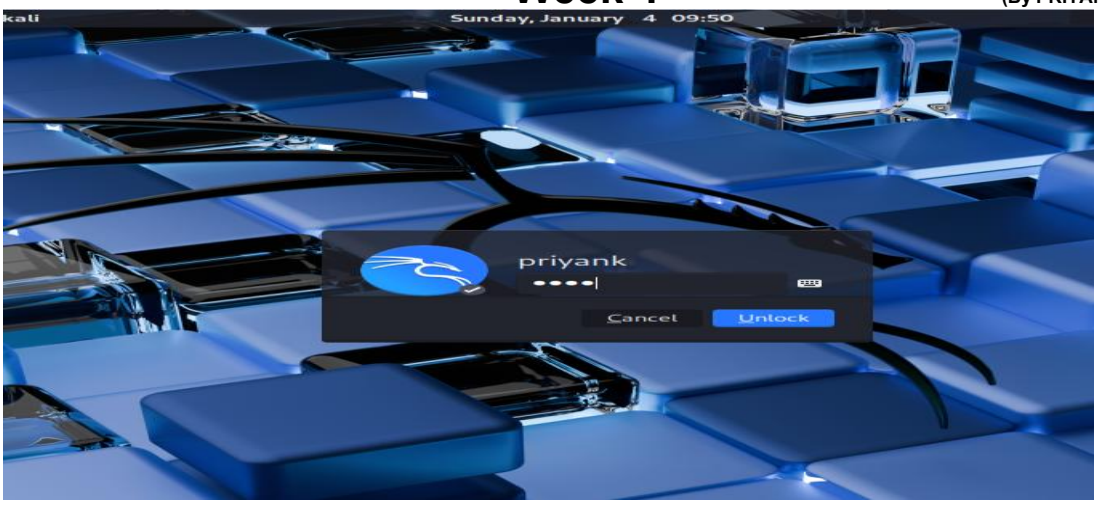
Default Login:

Username: kali

Password: kali

Week-1

(By PRIYANK KUMAR)



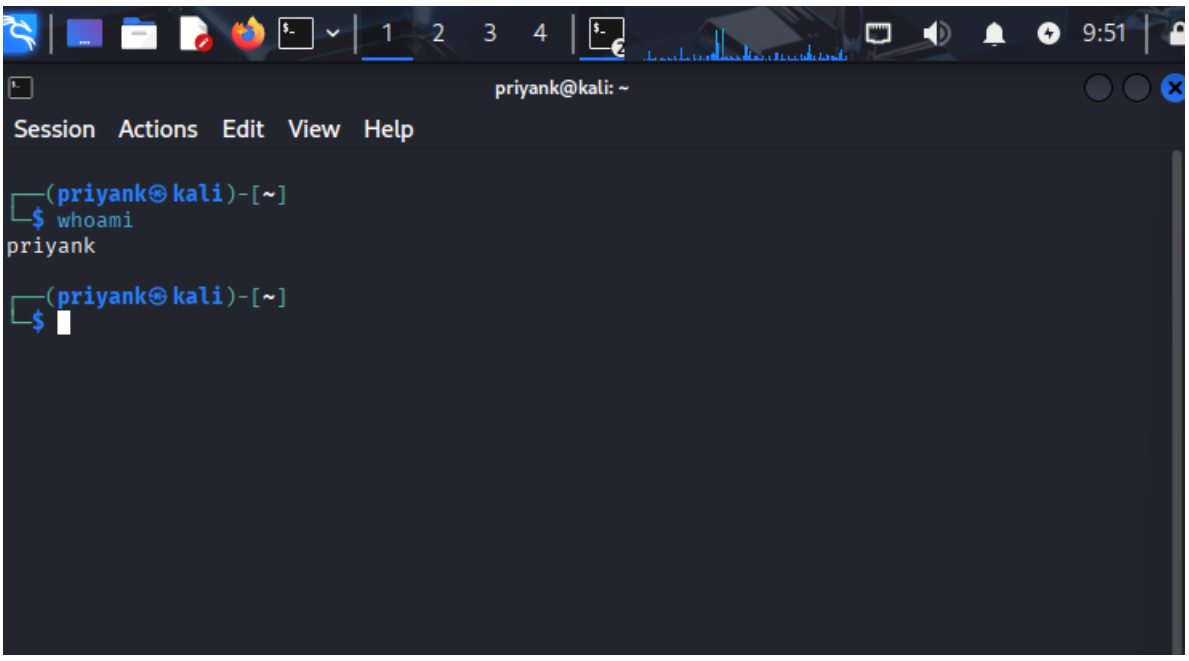
Screenshot: Kali Linux desktop screen

6. Verification of Installation

To verify installation, open terminal and run:

whoami

This confirms that Kali Linux is successfully installed and running.



Screenshot: Terminal output

7. Result

Kali Linux was successfully installed and configured on **VMware Workstation**. The system is ready for penetration testing and security experiments.

8. Conclusion

Virtual installation of Kali Linux using VMware provides a safe and controlled

environment for learning cybersecurity concepts. This setup avoids risks to the host system while allowing full access to Kali Linux tools.

Download Metasploitable 3

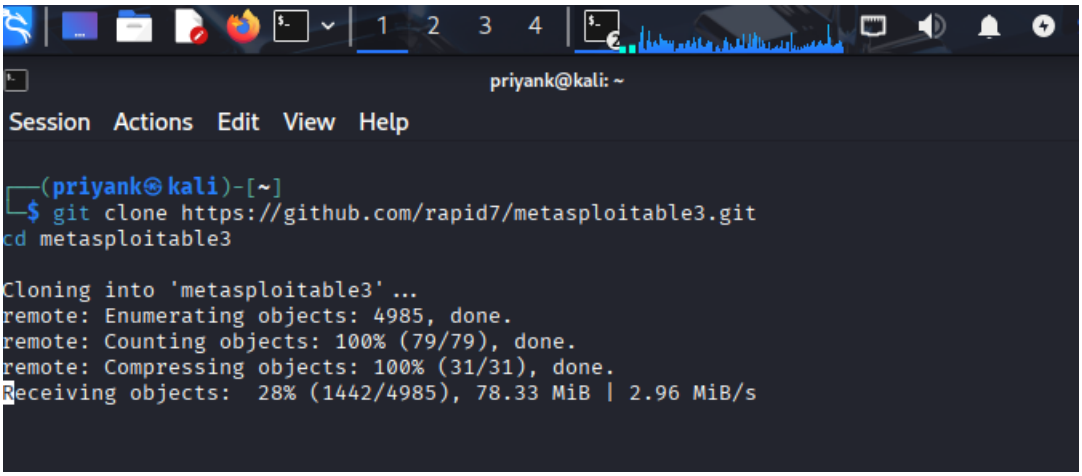
Step 1: Install Requirements (on Host or Kali)

- Install:
  - VirtualBox
  - Vagrant
  - Git

Step 2: Download Metasploitable 3

git clone https://github.com/rapid7/metasploitable3.git

cd metasploitable3



1.3 Configure Networking (Very Important)

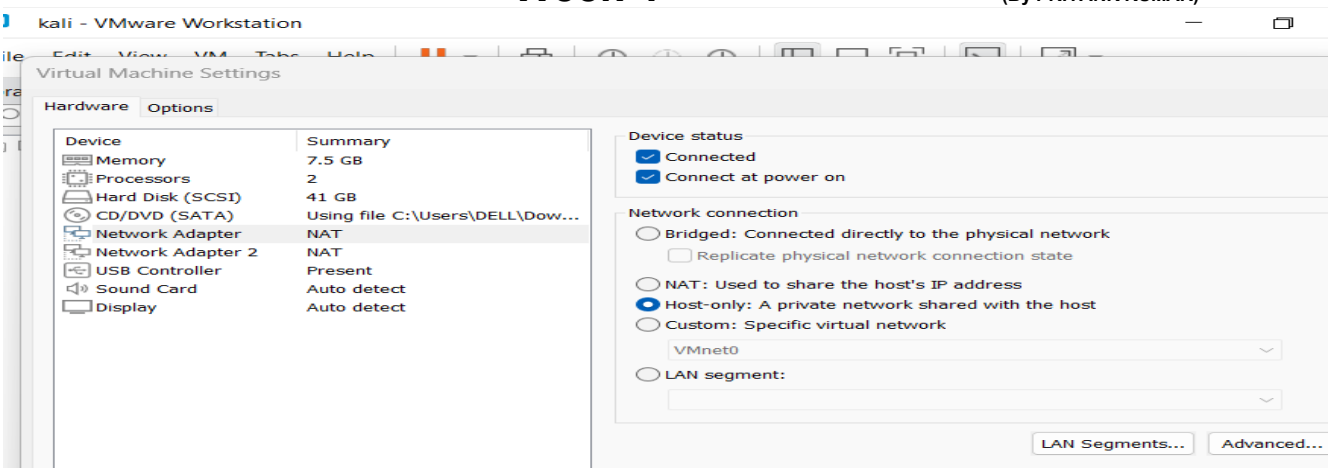
For **both Kali and Metasploitable**:

1. Open **VirtualBox**
2. Go to **Settings → Network**
3. Set:
4. Adapter 1 → Host-Only Adapter

✅ This ensures they can talk **only to each other**

Week-1

(By PRIYANK KUMAR)



Find IP Addresses

On Kali:

ip a

```
┌──(priyank㉿kali)-[~/metasploitable3]
└─$ ip a
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defaul
len 1000
    link/ether 00:0c:29:3f:82:18 brd ff:ff:ff:ff:ff:ff
: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defaul
len 1000
    link/ether 00:0c:29:3f:82:22 brd ff:ff:ff:ff:ff:ff
    inet 192.168.79.131/24 brd 192.168.79.255 scope global dynamic noprefixroute eth1
       valid_lft 1233sec preferred_lft 1233sec
    inet6 fe80::20c:29ff:fe3f:8222/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(priyank㉿kali)-[~/metasploitable3]
```

On Metasploitable:

ip a

```
sfadmin@metasploitable:~$ ip a
: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:a2:50:3e brd ff:ff:ff:ff:ff:ff
    inet 192.168.217.129/24 brd 192.168.217.255 scope global eth0
    inet6 fe80::20c:29ff:fea2:503e/64 scope link
       valid_lft forever preferred_lft forever
: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:a2:50:48 brd ff:ff:ff:ff:ff:ff
sfadmin@metasploitable:~$ _
```

Note down Metasploitable IP

Week-1

(By PRIYANK KUMAR)

Example:

192.168.217.129

What is OpenVAS?

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanning tool used to:

- Detect known vulnerabilities
- Analyze system misconfigurations
- Generate detailed security reports

Why OpenVAS?

- Free & open source
- Regular vulnerability feed updates
- Web-based GUI
- Industry-recognized tool

3️ Objective of the Project

The objectives of this assessment are:

- To scan a target system for vulnerabilities
- To identify security risks and severity levels
- To generate a vulnerability assessment report
- To recommend mitigation steps

4️ System & Environment Details

Operating System:

- Kali Linux

OpenVAS Version:

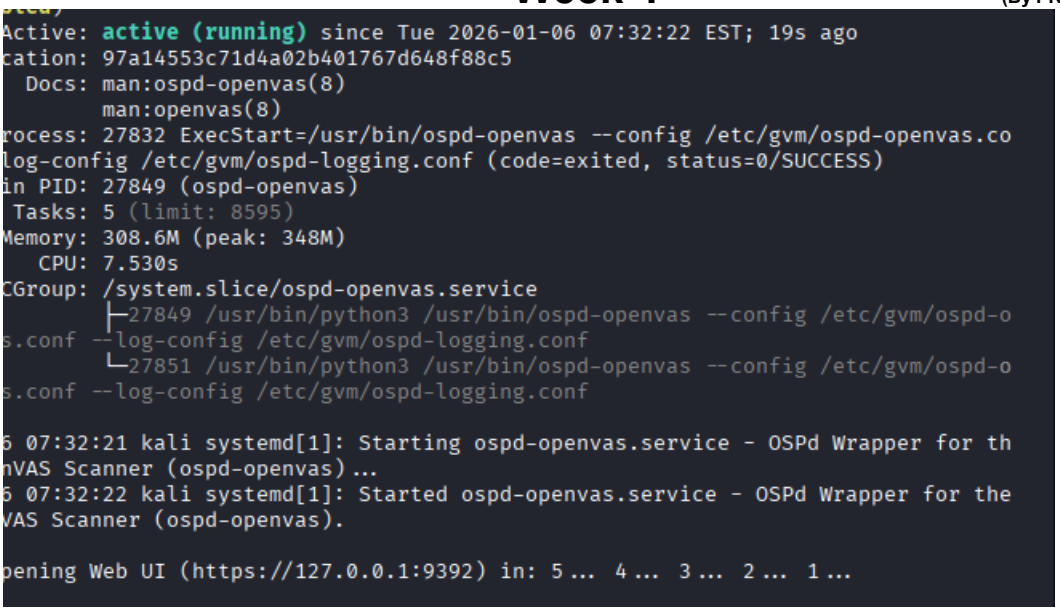
gvm-check-setup

Step 1: Start GVM Services

sudo gvm-start

Week-1

(By PRIYANK KUMAR)



GVM services running message

Step 2: Access OpenVAS Web Interface

Open browser and go to:

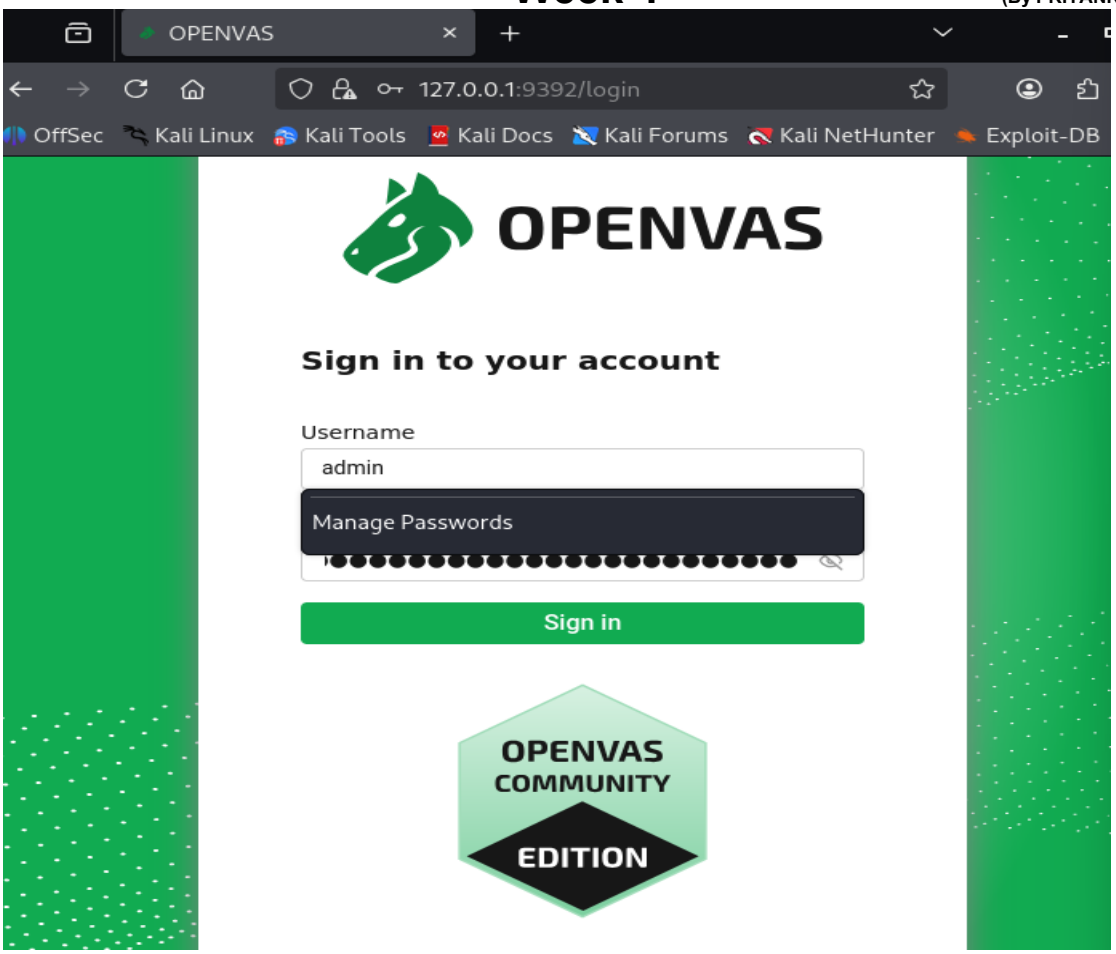
https://127.0.0.1:9392

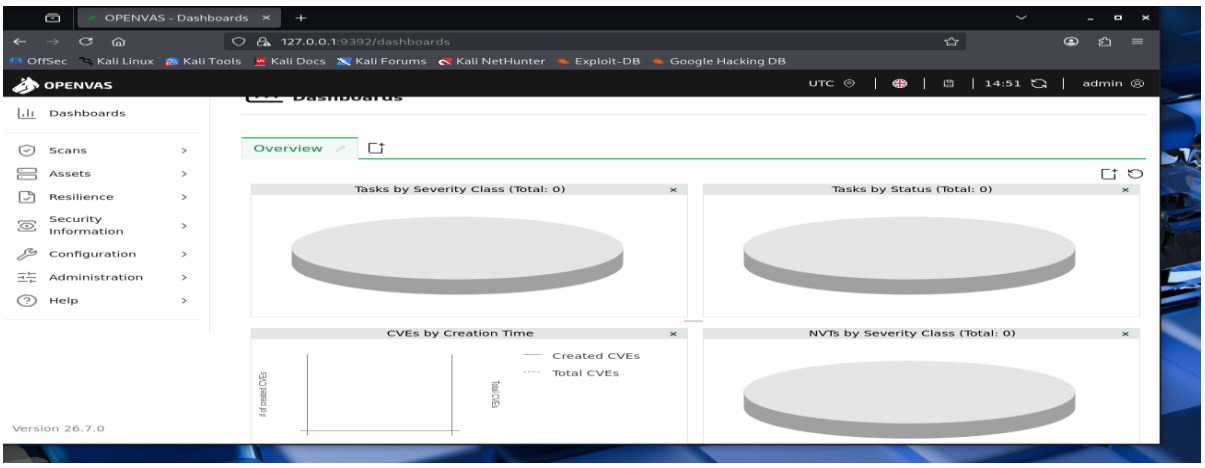
⚠️ Accept self-signed certificate warning:

Advanced → Accept the Risk and Continue

Week-1

(By PRIYANK KUMAR)





OpenVAS Dashboard

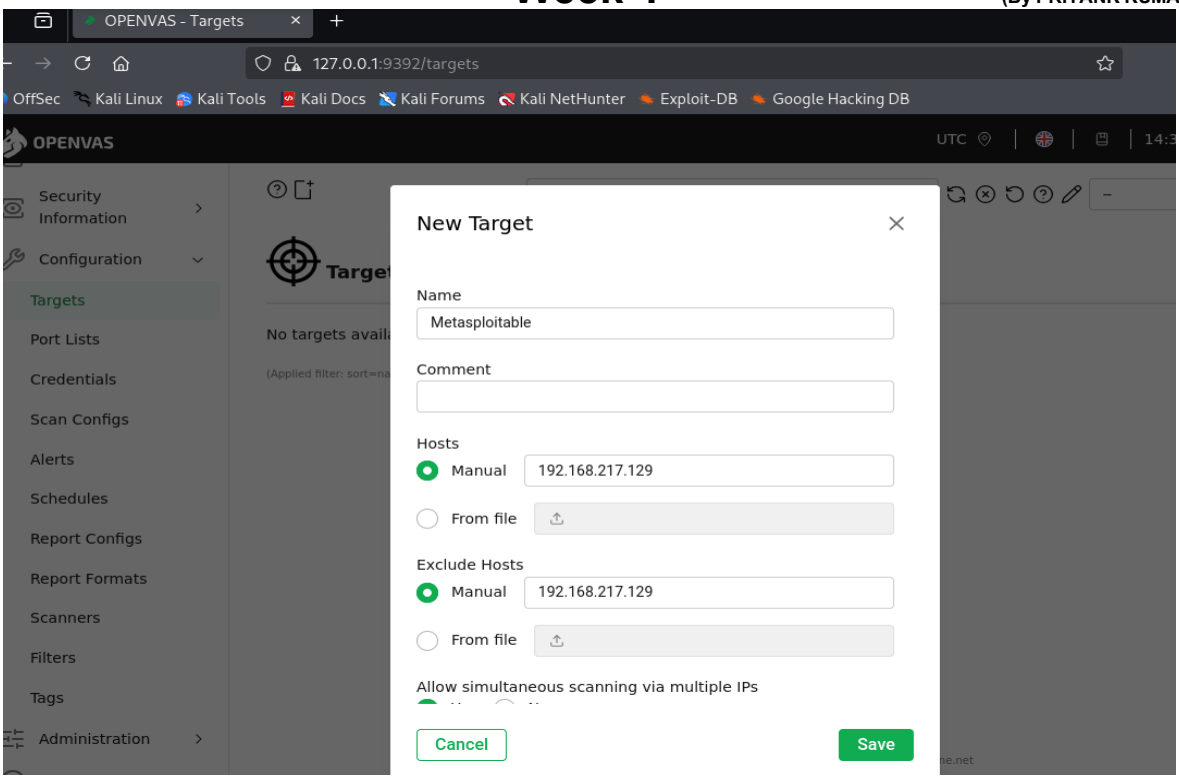
Step 3: Create a Target

Path:

Configuration → Targets → New Target

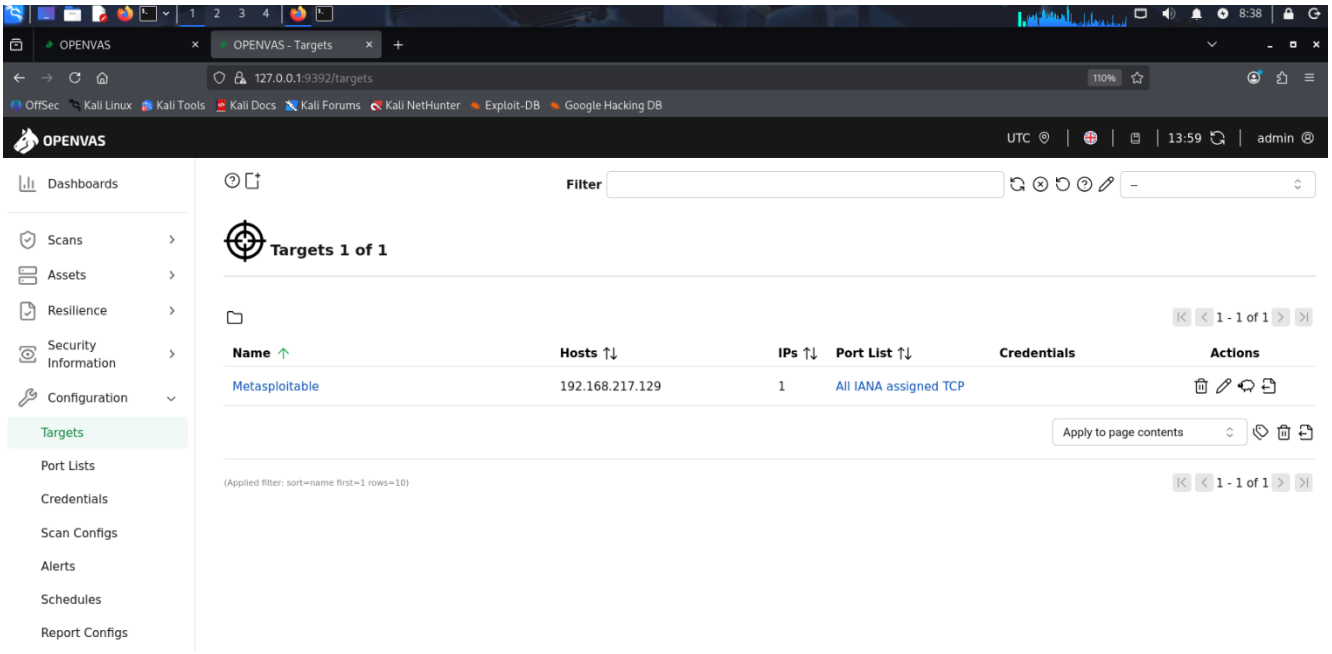
Week-1

(By PRIYANK KUMAR)



Example Target Details:

- Name: Metasploitable
- IP Address: 192.168.217.129
- Port List: All IANA TCP
- 97d4ca65-aece-43e1-80a9-1d2feb87aa4a



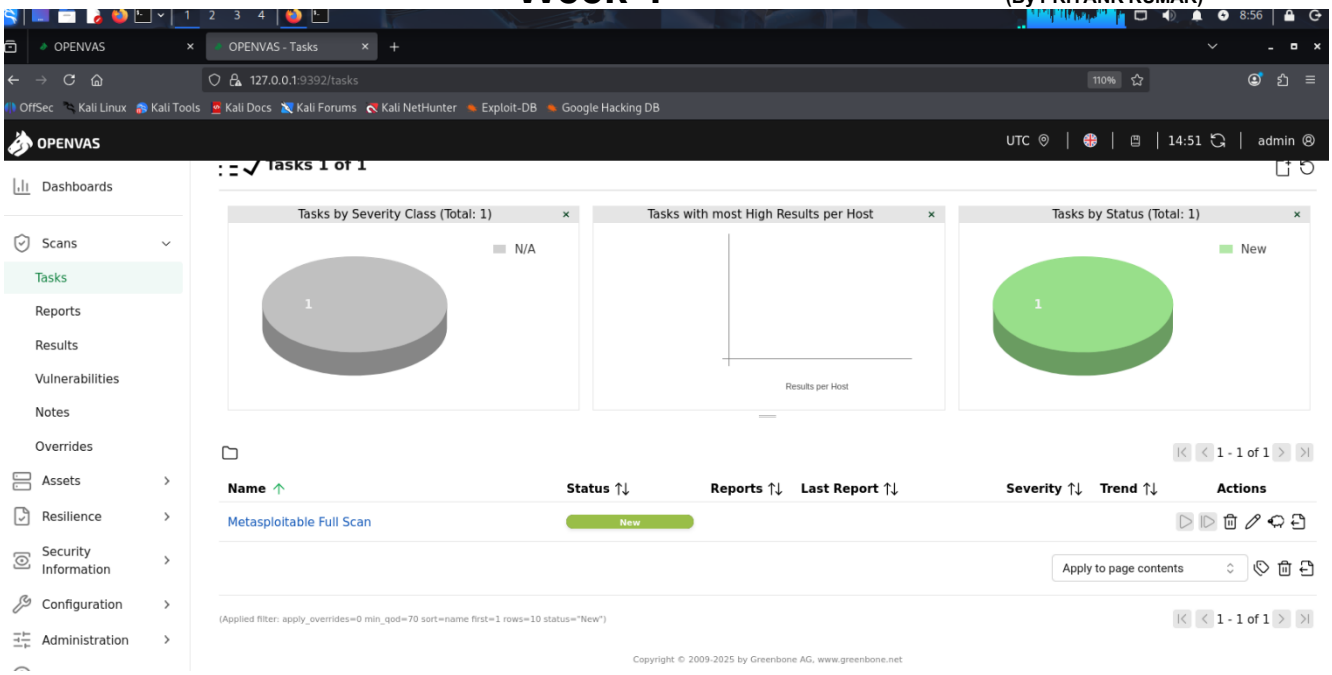
Start the Scan

1. In **Scans → Tasks**
2. Click the ▶ **Start** button next to your task

Scan time: **10–30 minutes**

Week-1

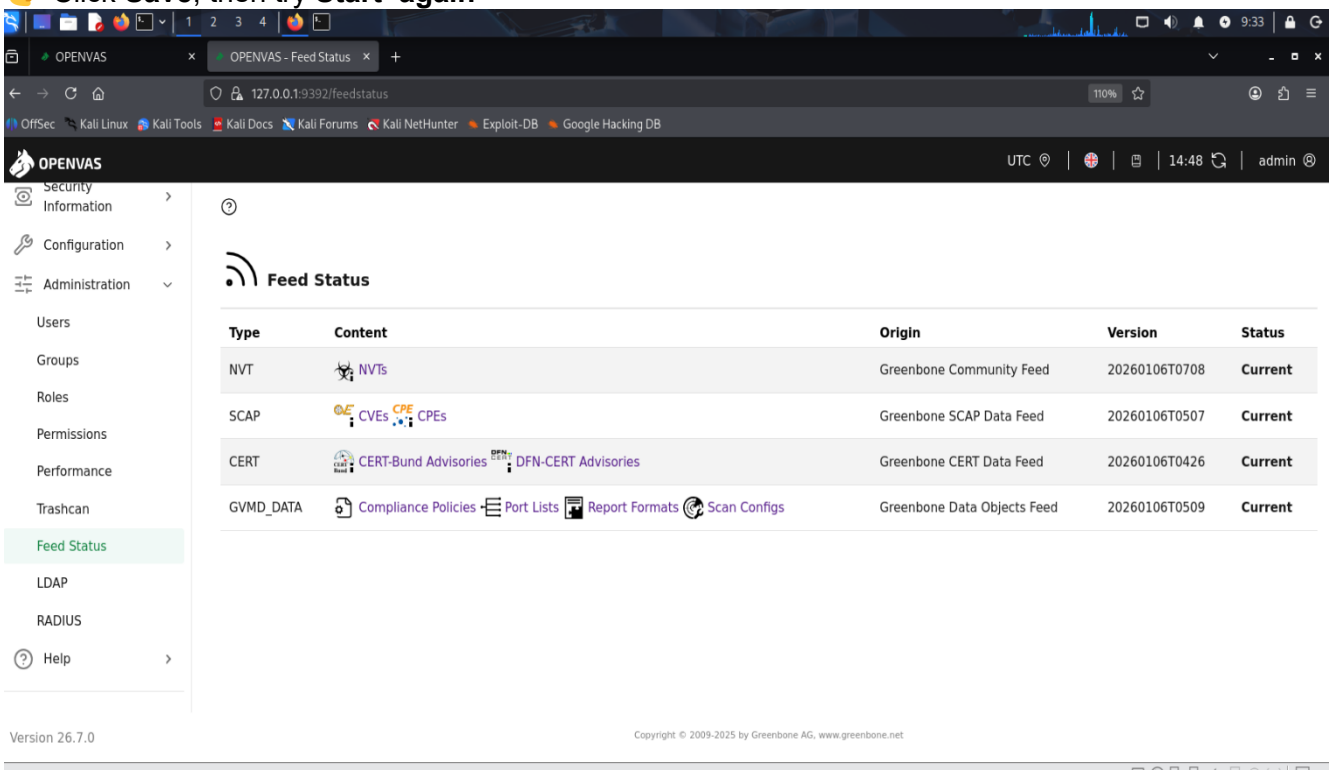
(By PRIYANK KUMAR)



Go to **Scans → Tasks → Edit Task**
Verify EXACTLY:

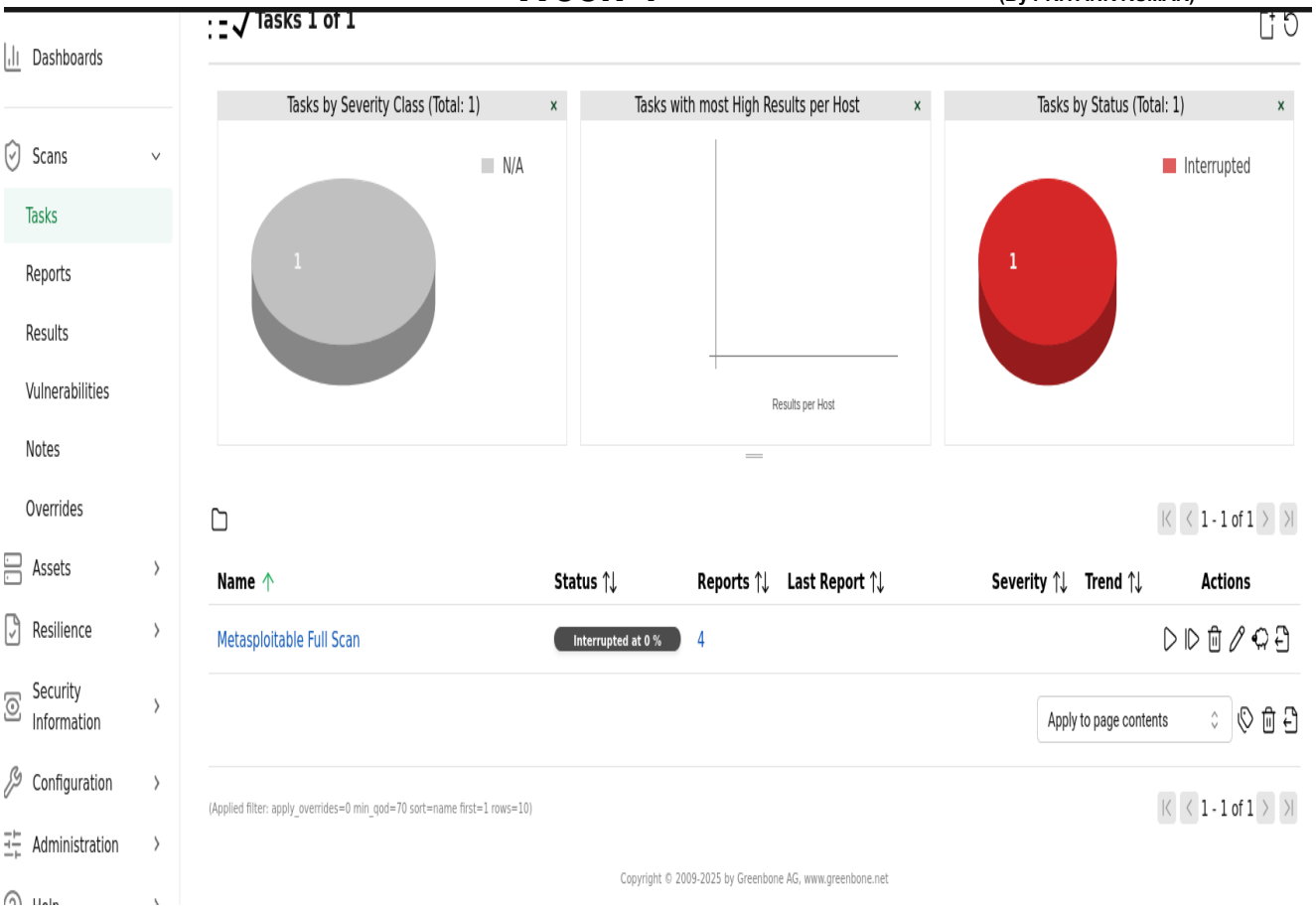
Field	**Must Be**
Scan Target	Your target
Scan Config	Full and Fast
Scanner	OpenVAS Default
Status	New / Stopped

⚠️ If **Scanner = None** → Start button does nothing
👉 Click **Save**, then try **Start again**



Week-1

(By PRIYANK KUMAR)



Analyzing Nikto Scan Results

nikto -h http://192.168.217.129

Objective

To identify **web server vulnerabilities**, **misconfigurations**, and **information disclosure issues** on the target system.

STEP 1: Record Basic Scan Information

From the Nikto output, write:

- **Target IP:** 192.168.217.129
- **Port Scanned:** 80 (HTTP)
- **Web Server:** Apache/2.2.8 (Ubuntu)
- **Backend Technology:** PHP/5.2.4
- **Scan Tool:** Nikto v2.5.0
- **Scan Duration:** ~48 seconds

STEP 2: Identify and Classify Vulnerabilities

Now, **convert raw Nikto output into findings**.

2.1 Outdated Software (High Risk)

Finding:

Week-1

(By PRIYANK KUMAR)

Apache/2.2.8 appears to be outdated
PHP/5.2.4 detected
Why it's a problem:

- Apache 2.2.x is **End of Life**
- PHP 5.2.x has **multiple known CVEs**

Impact:

- Remote Code Execution
- Privilege escalation

Severity: High
"The web server is running outdated Apache and PHP versions which are vulnerable to multiple known exploits."

2.2 Missing Security Headers (Medium Risk)
Findings:

- X-Frame-Options header missing
- X-Content-Type-Options header missing

Why it's a problem:

- Allows **Clickjacking**
- MIME-type sniffing attacks

Severity: Medium
"Missing HTTP security headers increase the risk of client-side attacks such as clickjacking."

2.3 Dangerous HTTP Methods Enabled (High Risk)
Finding:
HTTP TRACE method is active
Why it's a problem:

- Vulnerable to **Cross-Site Tracing (XST)**

Severity: High
"The TRACE HTTP method is enabled, which can be exploited for Cross-Site Tracing attacks."

2.4 Directory Listing Enabled (Medium Risk)
Findings:

- /doc/
- /test/
- /icons/

Why it's a problem:

- Attackers can browse internal files
- Information disclosure

Severity: Medium
Write:
"Directory indexing is enabled, allowing attackers to view internal files and directory structures."

2.5 Sensitive Files Exposed (High Risk)
Findings:

- /phpinfo.php
- #wp-config.php#
- /phpMyAdmin/

Why it's dangerous:

- phpinfo.php → system details
- wp-config.php → database credentials
- phpMyAdmin → database access

Severity: High
"Sensitive configuration and administrative files were accessible, leading to possible credential exposure."

Week-1

(By PRIYANK KUMAR)

Likelihood \ Impact Low Medium High
High Medium High **Critical**

- phpMyAdmin exposure → **High Likelihood + High Impact = Critical**
- Missing headers → **Medium**
- Directory listing → **Medium**

Document Findings

Tools Used

- Microsoft Excel / Google Sheets (for vulnerability tracking)
- Screenshots (Nikto scan results and CVSS calculator)

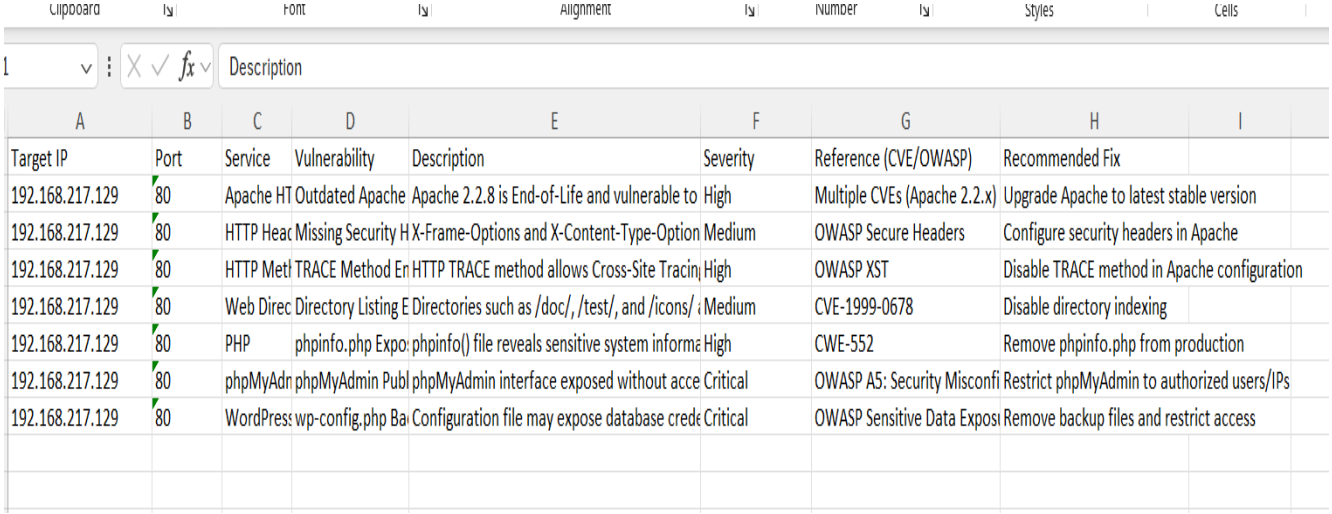
Procedure

1. The target system was scanned using Nikto and OpenVAS.
2. Identified vulnerabilities were documented in a spreadsheet.
3. Screenshots were captured as evidence of scan results.
4. Each vulnerability was recorded with relevant technical details.

Recorded Information Format

Table: Vulnerability Documentation

IP Address	Port	Service	Vulnerability Description	Reference	Severity
192.168.217.129	80	Apache HTTP	Outdated Apache version detected	CVE / OWASP	High
192.168.217.129	80	PHP	phpinfo.php file exposed	CWE-552	High
192.168.217.129	80	phpMyAdmin	phpMyAdmin publicly accessible	OWASP A5	Critical
192.168.217.129	80	HTTP	HTTP TRACE method enabled	OWASP XST	High
192.168.217.129	80	Apache	Directory listing enabled	CVE-1999-0678	Medium



Clipboard | Font | Alignment | Number | Styles | Cells

Description

	A	B	C	D	E	F	G	H	I
	Target IP	Port	Service	Vulnerability	Description	Severity	Reference (CVE/OWASP)	Recommended Fix	
	192.168.217.129	80	Apache HT	Outdated Apache	Apache 2.2.8 is End-of-Life and vulnerable to	High	Multiple CVEs (Apache 2.2.x)	Upgrade Apache to latest stable version	
	192.168.217.129	80	HTTP Heac	Missing Security H	X-Frame-Options and X-Content-Type-Option	Medium	OWASP Secure Headers	Configure security headers in Apache	
	192.168.217.129	80	HTTP Metl	TRACE Method En	HTTP TRACE method allows Cross-Site Tracin	High	OWASP XST	Disable TRACE method in Apache configuration	
	192.168.217.129	80	Web Direc	Directory Listing E	Directories such as /doc/, /test/, and /icons/	Medium	CVE-1999-0678	Disable directory indexing	
	192.168.217.129	80	PHP	phpinfo.php Expo	phpinfo() file reveals sensitive system informa	High	CWE-552	Remove phpinfo.php from production	
	192.168.217.129	80	phpMyAdn	phpMyAdmin Publ	phpMyAdmin interface exposed without acce	Critical	OWASP A5: Security Misconfi	Restrict phpMyAdmin to authorized users/IPs	
	192.168.217.129	80	WordPress	wp-config.php Ba	Configuration file may expose database cred	Critical	OWASP Sensitive Data Expos	Remove backup files and restrict access	

Evidence Collected

- Screenshot of Nikto scan output
- Screenshot of identified vulnerabilities
- Screenshot of spreadsheet entries

Observation

"The scan revealed multiple vulnerabilities mainly related to outdated software, insecure configurations, and exposure of sensitive files."

4. Practice Risk Assessment

Objective

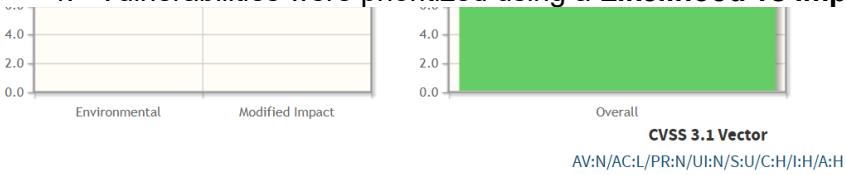
Week-1

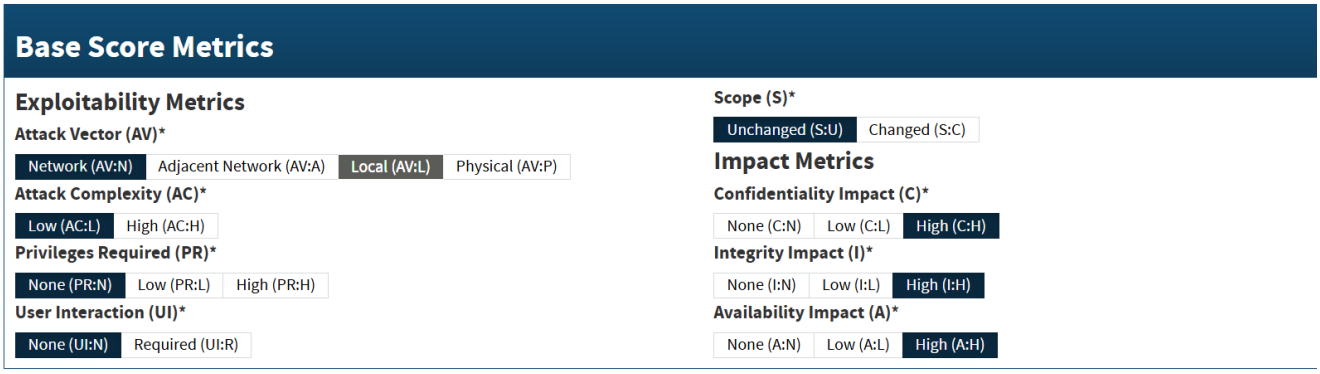
(By PRIYANK KUMAR)

To evaluate the **severity and risk level** of identified vulnerabilities using **CVSS scoring** and a **3×3 risk matrix**.

Risk Assessment Procedure

1. Each vulnerability was analyzed using the **CVSS v3.1 Calculator**.
2. Base metrics such as Attack Vector, Privileges Required, and Impact were selected.
3. CVSS scores were generated and recorded.
4. Vulnerabilities were prioritized using a **Likelihood vs Impact (3×3) risk matrix**.





CVSS Scoring Format

Vulnerability	CVSS Score	Severity
Outdated Apache	7.5	High
phpinfo.php exposed	6.5	Medium
phpMyAdmin exposed	9.0	Critical
Missing security headers	5.0	Medium
Directory listing	4.3	Low

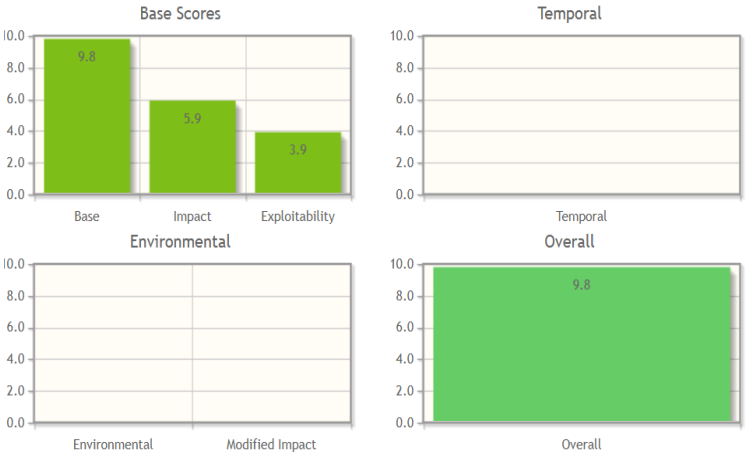
Week-1

(By PRIYANK KUMAR)

CVSS Version 3.0 CVSS Version 3.1

Common Vulnerability Scoring System Calculator

This page shows the components of a CVSS assessment and allows you to refine the resulting CVSS score with additional or different metric values. Please read the CVSS standards guide to fully understand how to assess vulnerabilities using CVSS and to interpret the resulting scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



CVSS Base Score: 9.8
Impact Subscore: 5.9
Exploitability Subscore: 3.9
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 9.8

Show Equations

3×3 Risk Matrix

Likelihood \ Impact	**Low**	**Medium**	**High**
Low	Low	Low	Medium
Medium	Low	Medium	High
High	Medium	High	Critical

Risk Prioritization

- **Critical Risk:** phpMyAdmin exposure
- **High Risk:** Outdated Apache, HTTP TRACE enabled
- **Medium Risk:** Missing security headers, directory listing

Conclusion

“Based on CVSS scores and risk matrix evaluation, vulnerabilities with high likelihood and high impact were prioritized for immediate remediation.”