Unit II
Cyber Security Threats and Vulnerabilities

1. Overview of Security threats and Vulnerability:
- **1.1 Vulnerability and Threats**
- **1.2 Types of attacks on confidentiality**
- **1.3 Types of attacks Integrity and Availability**
- **1.4 Types of Malware and Threats**
- **(Spyware, Virus and Warms, Trojan and backdoors)**

2. Web attack:
- **2.1 Browser Attacks**
- **2.2 Web Attacks Targeting Users**
- **2.3 Obtaining User or Website Data**
- **2.4 Email Attacks**

3. Network Vulnerabilities: Overview of vulnerability scanning
4. Open Port / Service Identification, Banner /Version Check
5. Traffic Probe, Vulnerability Probe
6. Vulnerability scanning using Nmap, OpenVAS
7. Metasploit
8. Networks Vulnerability Scanning using Netcat, Socat
9. Network Sniffers and Injection tools

In cybersecurity, the terms **threats** and **vulnerabilities** are essential because they help us understand **how attackers can harm systems, networks, and data**.

- A **vulnerability** is a weakness.
- A **threat** is something that can exploit that weakness.

Both together determine how secure or insecure a system is.

1.1 Vulnerability and Threats

1.1.1 What is a Vulnerability in Cybersecurity?

A **vulnerability** is a flaw, weakness, or gap in a computer system, network, software, or device that a cyber attacker can misuse.
It is similar to a **crack in the wall of a house**—if the crack is not fixed, someone might use it to break in.

Common Examples of Vulnerabilities

- **Outdated software**
  When software or operating systems are not updated (patched), known security issues remain open.
  Attackers can use these weaknesses to install malware or gain access.
- **Weak passwords**
  Simple or predictable passwords (like *123456*, *password*, or birth dates) can be easily cracked using tools or guesswork.
- **Unprotected Wi-Fi networks**
  If Wi-Fi is not secured with strong encryption (e.g., WPA2/WPA3), attackers nearby can connect to the network, monitor traffic, or access devices.

1.1.2 What is a Cybersecurity Threat?

A cybersecurity threat is anything that has the potential to damage, disrupt, or gain unauthorised access to a computer system, network, or digital data.
It is similar to a danger that can harm your online safety or privacy.
A threat becomes serious when it targets a vulnerability in the system.

Common Examples of Cybersecurity Threats

- **Hackers**
  Individuals or groups who try to break into systems to steal information, disrupt services, or cause damage.
- **Malware**
  Malicious software such as viruses, worms, Trojans, and spyware, designed to infect systems, steal data, or take control of devices.
- **Phishing**
  Fake emails, messages, or websites created to trick people into sharing personal information like passwords, banking details, or credit card numbers.
- **Ransomware**
  A harmful program that locks or encrypts files and then demands money (ransom) to restore access.

1.2 Types of Attacks on Confidentiality in Cybersecurity

Confidentiality means ensuring that information is kept **secret**, **private**, and accessible only to authorised people.
Attacks on confidentiality aim to **steal, read, or intercept sensitive data** without permission.

Below are some common types of confidentiality attacks:

1. Eavesdropping (Sniffing)

This occurs when an attacker secretly listens to or captures your communication—such as emails, chat messages, or online activities.
The attacker may steal sensitive data like:

- passwords
- bank information
- credit card numbers
- personal messages

Eavesdropping is usually done using tools that capture network traffic.

2. Man-in-the-Middle Attack (MITM)

In this attack, the attacker secretly positions themselves **between two communicating parties** (for example, your device and a website).

The attacker can:

- intercept information
- read private messages
- alter the communication
- steal login credentials

Neither party realises that someone else is controlling the communication.

3. Phishing

Phishing is a method used to trick people into revealing confidential information.
Key points:
i. Attackers send **fake** emails or messages pretending to be trusted organisations such as banks, government agencies, or social media sites.

ii. These messages often look genuine but contain links that lead to **fake websites** designed to steal your login details, banking information, or credit card numbers.

iii. Victims unknowingly enter their information into these fake websites, allowing attackers to steal their data.

4. Social Engineering

Social engineering involves **manipulating or deceiving** people into giving away confidential information.

Key points:
i. Attackers may pretend to be someone the victim knows or trusts (e.g., a colleague, bank officer, or IT support).

ii. They use psychological tricks—such as urgency, fear, or authority—to convince people to share sensitive information or perform unsafe actions.

iii. People who are unaware of security risks are more likely to fall victim to social engineering attacks.

1.3 Types of Attacks on Integrity and Availability in Cybersecurity

Integrity

Integrity means ensuring that information remains accurate, complete, and unaltered. Attacks on integrity aim to modify, damage, or manipulate data. Common types include:

Data Corruption:
Attackers intentionally alter or damage data, making it incorrect or unreliable. This can lead to wrong decisions or improper functioning of systems.

Man-in-the-Middle (MITM) Attack:
An attacker secretly intercepts communication between two parties and changes the information being exchanged, without either party knowing.

Replay Attacks:
In this attack, an attacker captures data being sent over the network and sends it again later to deceive the system. This can result in fraudulent transactions or unauthorised access.

Availability

Availability ensures that information, systems, and services are accessible whenever needed. Attacks on availability aim to disrupt or block access to data or services.

Common types include:

Denial of Service (DoS) Attack:
The attacker overwhelms a system or website with excessive traffic, causing it to slow down or crash, making it unavailable to legitimate users.

Distributed Denial of Service (DDoS) Attack:
A stronger form of DoS attack where multiple compromised devices (botnets) are used to flood the target with traffic, causing severe disruption.

Resource Exhaustion:
The attacker consumes system resources such as CPU, memory, or storage, causing the system to slow down significantly or stop functioning.

1.4 Types of Malware and Threats in Cybersecurity

Malware (short for malicious software) refers to any harmful program designed to damage a computer, steal information, or take control of a system. Common forms include viruses, worms, Trojans, spyware, and backdoors.

Below are some major types:

1. Spyware

Definition:
Spyware secretly monitors a user's activities on a device and collects sensitive information such as passwords, browsing behaviour, and credit card details without permission.

How it works:
It runs silently in the background, making it hard to detect. Some spyware can even record keystrokes (known as keylogging).

2. Viruses and Worms

• Virus

Definition:
A virus is a malicious program that attaches itself to clean files and spreads from one file or system to another.
Effects:
It can corrupt or delete files, slow down the system, or cause various malfunctions.
How it spreads:
Viruses often spread through infected email attachments, downloads from untrusted sources, or removable media.

• Worm

Definition:
A worm is similar to a virus but can spread automatically across computers and networks without user action.
Effects:
Worms exploit system vulnerabilities and can cause large-scale damage or slow down entire networks.

3. Trojan Horse

Definition:
A Trojan is malware disguised as a legitimate program (e.g., a game, utility tool, or software update). Once installed, it allows attackers to access or control the system.
How it spreads and harms:

Trojans do not replicate like viruses or worms, but they can steal data, install additional malware, or enable remote control of the victim's computer.

4. Backdoors

Secret Entry:

A backdoor is a hidden method that allows unauthorised access to a system. It may be created by malware or by exploiting weaknesses in software.

Ongoing Control:

Once inside, attackers can control the system, steal data, or launch further attacks. Backdoors remain hidden, making them difficult to detect and remove.

2. Web attack:

Here's a simple explanation of different web attacks:

2.1 Browser Attacks:

i. Browser attacks target the web browser you use to access websites.
ii. These attacks try to exploit weaknesses in the browser to steal information, install malware, or redirect you to fake websites.
iii. For example, a hacker could trick you into clicking on a malicious link, which could lead to your personal data being stolen.

2.2 Web Attacks Targeting Users:

i. These attacks focus on users who visit websites.
ii. Hackers may try to trick users into revealing personal information, such as login details or credit card numbers.
iii. Common methods include phishing (fake websites or emails that look real) or social engineering (tricking users into giving sensitive information).

2.3 Obtaining User or Website Data:

i. In these attacks, hackers aim to steal data from users or websites.
ii. They might do this by hacking into a website's database or using malware to capture personal information entered by users.
iii. This could include usernames, passwords, or payment details. The stolen data can be sold or used for fraud.

2.4 Email Attacks:

i. Email attacks are when hackers send fake or harmful emails to trick people.
ii. These emails can look like they come from trusted sources, such as banks or companies.
iii. The goal might be to get the person to open a malicious attachment or click a link that leads to a fake website.
iv. These emails can steal personal information or install malware on your device.

In all these cases, the goal of the hacker is to steal sensitive data, infect devices with malware, or trick people into giving away their information. Always be careful about clicking links, downloading attachments, or sharing personal information online.

3. Network Vulnerabilities: Overview of Vulnerability Scanning

Network vulnerabilities are weaknesses in a computer network that hackers can exploit (taking advantage of a vulnerability or weakness in a system) to break into systems. Vulnerability scanning is a process that checks a network to find these weaknesses. Special tools look for outdated software, misconfigured settings, and other problems that might allow hackers to access the network. By finding and fixing these vulnerabilities, we can protect the network from attacks.

Here are 5 common vulnerability scanning techniques/tools used to find weaknesses in computer networks and systems:

a. Network Scanning

Network scanning helps identify devices, open ports, and services running on a network. This technique is used to see which parts of a network are vulnerable to attacks. Tools used for network scanning often detect open ports and services that could be exploited by hackers.

• Tools: Nmap, Netcat

b. Port Scanning

Port scanning checks which ports on a system are open or closed. Open ports are potential entry points for hackers to exploit. By scanning ports, administrators can detect unused or unnecessary open ports that could be a security risk.

• Tools: Nmap, Zenmap

c. Vulnerability Scanning

Vulnerability scanning checks systems for known weaknesses or flaws that could allow hackers to gain unauthorized access. These scanners compare the system against a database of known vulnerabilities and provide reports on what needs fixing.

• Tools: OpenVAS, Nessus, Qualys

d. Web Application Scanning

Web application scanning looks for vulnerabilities in websites and online applications. This includes checking for issues like SQL injection, cross-site scripting (XSS), or misconfigurations that could be exploited to steal data or compromise the application.

• Tools: Burp Suite, OWASP ZAP

e. Database Scanning

Database scanning checks for security weaknesses in a database, such as outdated software, weak passwords, or permissions that allow unauthorized users to access sensitive data. It helps ensure that databases are properly protected.

- Tools: SQLmap, AppScan

4. Open Port / Service Identification and Banner / Version Checking

Open Port / Service Identification and Banner / Version Checking are important steps in network security and diagnostics. They help determine which services are running on a device or server and whether they pose any security risks.

1. Open Port / Service Identification

- Ports act like doors on a computer or server that allow data to enter or leave the system. Each port is assigned a specific number.
- Service Identification involves determining which services are running on these open ports.
    - For example:
        - Port 80 → HTTP (web server)
        - Port 443 → HTTPS (secure web traffic)
- By identifying open ports and their associated services, administrators can understand the device's functions and detect potential security risks.

2. Banner / Version Check

- A banner is a message displayed by a service when a connection is established. It often contains information about the service or software.
- Version Checking identifies the exact version of the service running (similar to software version numbers on a phone).
- Determining the version is crucial, as older versions may contain known vulnerabilities that attackers can exploit.
- Banner and version information therefore help verify whether services are updated and secure.

Why It Is Important

- Security:
  Identifying open ports, running services, and their versions helps detect security weaknesses. It ensures that outdated or vulnerable services are patched or disabled.
- Troubleshooting:

Network administrators can verify whether correct services are running on expected ports, making it easier to diagnose system or network issues.

Summary

These steps enable cybersecurity professionals to discover open "doors" (ports), identify the services running on them, check their versions, and assess whether they are secure or outdated.

5. Traffic Probe and Vulnerability Probe

In cybersecurity, a probe refers to examining or inspecting something closely to gather information or understand it better. Traffic probes and vulnerability probes are two essential tools used to analyse networks and identify potential risks.

1. Traffic Probe

- A Traffic Probe is a tool or process that monitors and analyses data flowing through a network.
- What It Does:
    - It captures and studies network traffic, such as the websites accessed, files transferred, and communication patterns between devices.
- Why It Is Useful:
    - It helps identify normal and abnormal traffic patterns.
    - It can detect unusual activity, such as suspicious connections, data leaks, or potential cyber-attacks.
    - It assists in ensuring that the network is functioning efficiently and securely.
- Example:
    - A traffic probe may notice a large volume of data being sent to an unknown IP address, which could indicate a security breach or unauthorised data exfiltration.

2. Vulnerability Probe

- A Vulnerability Probe is a tool designed to scan a system or network for weaknesses that attackers could exploit.
- What It Does:
    - It looks for known security issues in software, configurations, or network components.
    - These issues may include software bugs, outdated applications, misconfigurations, or unnecessary open ports.
- Why It Is Useful:
    - It allows organisations to identify and fix vulnerabilities before

cybercriminals can exploit them.
    - It supports proactive security by helping maintain a strong, secure system.
- Example:
    - A vulnerability probe might detect that a server is running an outdated version of web server software with known security flaws and recommend updating it.

Summary

- Traffic Probe: Monitors and analyses network traffic to ensure operations are normal, secure, and efficient.
- Vulnerability Probe: Scans systems and networks for weaknesses, enabling organisations to address them before they are exploited by attackers.

6. Vulnerability Scanning Using Nmap and OpenVAS

6.1 What is Nmap?

Nmap (Network Mapper) is a widely used network scanning tool that helps identify devices on a network, the services they are running, and any potential security issues. It is commonly used by system administrators, penetration testers, and cybersecurity professionals.

How Nmap Works

Nmap operates by sending specially crafted network packets to a target device or network and then analysing the responses. From these responses, Nmap can determine:

- Which ports are open
- Which services are running
- Possible vulnerabilities
- System and network behaviour patterns

Here is a simplified description of the process:

1. Scanning a Target
    - You provide Nmap with the IP address of a device or an entire network.
    - Nmap sends probes (special messages) to those devices.
2. Response Analysis
    - Nmap listens for the replies.
    - These responses reveal which ports are open and what services are active on them.
3. Information Gathering
    - Nmap compiles the collected data and presents a summary of its findings.
    - This helps identify potential risks or unusual configurations.

Common Fields in Nmap Output

When an Nmap scan is performed, the output usually includes several important fields:

1. IP Address
    - The unique numerical address assigned to each device on the network, similar to a house's street address.
2. Host Status
    - Indicates whether the device is online (up) or offline (down).
3. Open Ports
    - These are like "doors" into a system that allow communication.
    - Nmap identifies which ports are open and may also reveal the service running on them.
        - Example: Port 80 is commonly used for HTTP (web traffic).
4. Service / Protocol
    - Specifies the service or protocol operating on a particular port, such as:
        - HTTP – web browsing
        - FTP – file transfer
        - SSH – secure remote login
5. Operating System Detection
    - Based on the characteristics of its responses, Nmap can often estimate whether the device is running Windows, Linux, or another operating system.

Example:
If you want to scan a website or device to see what ports are open, you would type something like:

```
└─$ nmap amazon.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-02 22:57 EST
Nmap scan report for amazon.com (54.239.28.85)
Host is up (0.23s latency).
Other addresses for amazon.com (not scanned): 52.94.236.248 205.251.242.103
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 14.95 seconds
```

- The IP address of the device is 192.168.1.1.
- Two ports (80 and 443) are open.
- Port 80 is used for HTTP (web browsing), and port 443 is used for HTTPS (secure web browsing).

Key Points:

• Nmap helps find devices on a network and check if they are secure.

• It works by sending probes and looking at responses to find open ports and running services.

• Nmap is often used for security testing to find weaknesses in a network.

In simple terms, Nmap is like a detective that helps you learn more about devices on a network and whether they are secure.

6.2 OpenVAS – Open Vulnerability Assessment System

What is OpenVAS?

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanning tool used to identify security weaknesses in computers, servers, and networks. It helps organisations check whether their systems are secure against hackers and other cyber threats.

In simple terms, OpenVAS works like a security health check-up for systems and networks.

How Does OpenVAS Work?

OpenVAS scans a target system to find known vulnerabilities such as outdated software, open ports, and misconfigurations.

Steps involved in OpenVAS scanning:

1. Target Selection:
   The user selects the target to be scanned, such as a server, website, or entire network.
2. Scanning for Vulnerabilities:
   OpenVAS sends probes to the target to detect known security weaknesses like:
   - Outdated software
   - Open or unprotected ports
   - Incorrect system configurations
3. Analysing Results:
   After scanning, OpenVAS generates a detailed report showing the vulnerabilities found and their severity.
4. Fixing the Issues:
   Based on the report, corrective actions such as software updates or closing unused ports can be taken.

Common Fields in an OpenVAS Report

After the scan, OpenVAS provides a report containing the following key fields:

1. Vulnerability Name:
   Name of the detected issue (e.g., Outdated Web Server Software, Open SSH Port).

2. Severity:
   Indicates how dangerous the vulnerability is:

    - Low: Minor issue, fix when possible
    - Medium: Moderate risk
    - High: Serious issue, needs quick action
    - Critical: Very dangerous, immediate fix required
3. Description:
   Explains the vulnerability and why it is a security risk.
4. Solution / Recommendation:
   Suggested steps to fix the vulnerability, such as upgrading software or disabling a service.
5. Affected Systems:
   Lists the systems or devices impacted by the vulnerability.
6. Port:
   Specifies the network port where the vulnerability is found (if applicable).

Example of an OpenVAS Report

- Vulnerability: Outdated Apache Web Server (CVE-2021-12345)
- Severity: High
- Description: Apache Web Server version 2.4.10 is outdated and contains a known vulnerability that may allow arbitrary code execution.
- Solution: Upgrade Apache Web Server to version 2.4.51 or higher.
- Port: 80
- Affected System: www.example.com (192.168.1.10)

Explanation:

- The report identifies an outdated Apache version
- Severity is High, meaning it must be fixed soon
- It clearly explains the problem and provides a solution
- It also shows the affected system and port number

- OpenVAS is used to **identify security vulnerabilities** in systems and networks
- It scans targets for **known weaknesses**
- The generated report includes **severity, description, and solutions**
- It helps organisations **fix risks before attackers exploit them**

OpenVAS is an essential vulnerability scanning tool in cybersecurity. By identifying weaknesses early, it helps improve system security and prevents potential cyberattacks.
In simple words, **OpenVAS helps find security problems before they turn into serious threats**.

7. Metasploit

Metasploit is a tool used by security experts to test and identify vulnerabilities in computer systems and networks. It helps in checking how secure a system is by simulating real-world cyberattacks.

You can think of Metasploit as a “practice hacker” tool. It shows how an attacker might break into a system so that security professionals can fix the weaknesses before real hackers exploit them.

How Metasploit works:

1. Simulate Attacks:
   Metasploit imitates hacker attacks by using known vulnerabilities in software to try gaining access to systems.

2. Test Defences:
   It checks whether security measures like firewalls, intrusion detection systems, and antivirus software can block these attacks.

3. Learn and Improve Security:
   By identifying weak points, Metasploit helps IT teams strengthen security and protect systems better.

Metasploit can be used by ethical hackers and security professionals for legal security testing. However, if misused, it can also be used by attackers.
Its main purpose is to improve cybersecurity by finding risks before real hackers do.

8. Networks Vulnerability Scanning using Netcat, Socat

Netcat (nc) is a powerful networking tool that allows users to read and write data across network connections. It is often called the "Swiss Army knife" of networking because it can perform many tasks such as creating network connections, transferring files, and doing basic network diagnostics.

1. Netcat (nc)

Netcat is commonly used for **basic vulnerability scanning**, especially for **checking open ports** on a system or network.
Open ports indicate running services, and these services may contain vulnerabilities that attackers can exploit.

Using Netcat for Scanning

- **Scan a single port:**

```
nc -zv 192.168.1.1 80
```

Explanation:

- `-z` → Scan mode (no data sent)
- `-v` → Verbose output
- `192.168.1.1` → Target IP address
- `80` → Port number (HTTP)
- **Scan multiple ports:**

```
nc -zv 192.168.1.1 1-1000
```

This command scans ports **1 to 1000** on the target system.

If a port is open, it means a service is running. If that service has security flaws, it may be vulnerable to attacks.

Therefore, **port scanning is the first step in vulnerability assessment**.

2. Socat (Socket CAT)

Socat is an advanced networking tool similar to Netcat but with **more powerful features**. It supports multiple protocols and is used for **redirection, tunnelling, and complex connection handling**.

Using Socat for Testing Connections

Although Socat is not mainly designed for vulnerability scanning, it is useful for **testing and interacting with services**.

- **Example of connecting to a service:**
  ```
  socat TCP4:192.168.1.1:80 -
  ```

This command connects to port 80 of the target system and allows interaction with the service (e.g., sending HTTP requests and viewing responses).

Socat is useful for:

- Checking whether a service is active
- Testing service responses
- Tunnelling traffic during security testing

Summary

- **Netcat:**
  - Used for checking **open ports**
  - Helpful for **basic vulnerability scanning**
  - Simple and fast tool
- **Socat:**
  - More advanced than Netcat
  - Used for **complex network testing**, redirection, and tunnelling
  - Helps in interacting deeply with network services

9. Network Sniffers and Injection tools

Network Sniffers and Injection Tools

Network sniffers and injection tools play an important role in network security testing. They help security professionals monitor network traffic and simulate attacks to identify weaknesses before attackers can exploit them.

1. Network Sniffers

A network sniffer is a tool used to capture and analyse data packets travelling over a network. It works like an invisible eavesdropper, listening to all data being sent and received on a network and recording it for analysis.

Purpose of Network Sniffers

- Understand network behaviour
- Detect unauthorised or suspicious activity
- Identify security issues such as unencrypted sensitive data

What Network Sniffers Can Capture

- Emails
- Website visits
- Login credentials
- File transfers

If sensitive information (like passwords or credit card numbers) is transmitted without encryption, it becomes a serious security risk that sniffers can reveal.

Common Network Sniffing Tools

- Wireshark:
  A popular graphical tool that captures and analyses network packets in detail.
- tcpdump:
  A command-line tool used to capture and display network traffic in text format.

Example: Using Wireshark

1. Open Wireshark
2. Select a network interface (Wi-Fi or Ethernet)
3. Start packet capture
4. Analyse captured packets for:
    - Unencrypted data
    - Unusual traffic patterns
    - Suspicious activities

2. Injection Tools

Injection tools are used to send malicious or unexpected input into systems to test how they respond. These tools simulate attacks to find vulnerabilities in web applications, databases, and servers.

Types of Injection Attacks

- SQL Injection:
  Malicious SQL queries are inserted into input fields to access, modify, or delete database data.

- Command Injection:
  Attackers inject system commands that the server executes, potentially allowing full system control.

- XSS (Cross-Site Scripting):
  Malicious scripts are injected into web pages and executed in users' browsers to steal data or spread malware.

Common Injection Tools

- SQLmap:
  Automates detection and exploitation of SQL injection vulnerabilities.

- OWASP ZAP (Zed Attack Proxy):
  Identifies web application vulnerabilities including SQL injection and XSS.
- Burp Suite:
  A comprehensive web security testing tool used by professionals.

Example: SQL Injection Testing

Security testers can use SQLmap by providing a potentially vulnerable URL. The tool injects SQL payloads to check whether the database can be accessed improperly.

Summary

- Network Sniffers:
  - Capture and analyse network traffic
  - Help detect unencrypted data and suspicious activity
  - Examples: Wireshark, tcpdump
- Injection Tools:
  - Simulate attacks by injecting malicious input
  - Help identify vulnerabilities like SQL injection and XSS
  - Examples: SQLmap, Burp Suite, OWASP ZAP

Conclusion

Both network sniffers and injection tools are essential in cybersecurity testing. Sniffers allow inspection of network traffic, while injection tools help simulate real-world attacks. Together, they help organisations identify and fix security weaknesses, strengthening overall system security.

Difference among Nmap, Netcap, Socat, OpenVAS, Burp Suite, SQLmap

Feature/Tool	Nmap	Netcat	Socat	OpenVAS	Burp Suite	SQLmap
Purpose	Network scanning & mapping	Simple network communication	Advanced network communication & tunneling	Vulnerability scanning (web apps)	Web application security testing	Automated SQL injection testing
Primary Use	Discover open ports, services, and vulnerabilities	Establishing connections, testing ports, data transfer	Redirecting and tunneling network traffic	Scan for vulnerabilities in networked systems	Identify security flaws in web apps	Detect and exploit SQL injection flaws
Type of Tool	Network scanner	Networking utility	Networking tool (for advanced use)	Vulnerability scanner (open-source)	Web vulnerability scanner	Exploit tool for SQL injection
Focus	Port scanning and network mapping	Simple communication (TCP/UDP)	Complex communication, data redirection	Web app vulnerabilities, security flaws	Cross-site scripting, SQL injections, more	SQL injection on web applications
Main Features	Port scanning, OS fingerprinting, service detection	Simple connection testing, port listening	Network redirection, VPN tunneling	Vulnerability scanning, security auditing	Intercepting, scanning web traffic, brute-forcing	Automated SQL injection and database exploitation
Ease of Use	Command-line, can be complex for beginners	Simple command-line, easy to use	Command-line, more complex than Netcat	GUI and command-line, setup required	GUI with various tools, more complex	Command-line, automated processes
Use in Ethical Hacking	Yes, used for penetration testing to map networks and find vulnerabilities	Yes, used for testing connections and setting up communication channels	Yes, used for advanced penetration testing, setting up communication channels	Yes, used to perform vulnerability assessments on networks	Yes, used to test and fix vulnerabilities in web apps	Yes, used to test SQL injection vulnerabilities in web apps