Cyber Security

DU # 2101CS632

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

UNIT - 4

VULNERABILITY AND SCANNING TOOLS

Anindya Sinha

Cyber Security Analyst

Samatrix Consulting Private Limited

anindya.sinha@samatrix.io

9952061704

WHAT ARE VULNERABILITY AND SCANNING TOOLS?

Vulnerability scanning tools are software applications designed to identify weaknesses in computer systems, networks, applications, and other IT infrastructure that could be exploited by attackers. These tools work by actively probing systems for known vulnerabilities, misconfigurations, and security weaknesses. They can also assess the overall security posture of an organization's IT assets.

Darshan UNIVERSITY

How do these tools work?

- **Discovery:** The tool scans the network to identify all active devices, services, and applications.
- **Enumeration:** Once the active devices are identified, the tool further probes them to gather information about their configuration, software versions, and potential vulnerabilities.
- **Vulnerability Assessment:** Using a database of known vulnerabilities, the tool compares the information gathered during enumeration against its database to identify potential security issues.
- **Reporting:** Finally, the tool generates a report outlining the discovered vulnerabilities, their severity levels, and recommendations for remediation.

Different types of Vulnerability Scanning Tools

- **Network Vulnerability Scanners:** These tools scan network devices such as routers, switches, and servers for vulnerabilities. Examples include Nessus, OpenVAS, and Nexpose.

- **Web Application Scanners:** These tools are specialized in identifying vulnerabilities in web applications such as SQL injection, cross-site scripting (XSS), and insecure authentication mechanisms. Examples include Acunetix, OWASP ZAP, and Burp Suite.

- **Database Scanners:** These tools focus on identifying vulnerabilities in database systems, such as weak authentication mechanisms, misconfigurations, and SQL injection vulnerabilities. Examples include DbProtect and SQLmap.

- **Wireless Network Scanners:** These tools are designed to identify vulnerabilities in wireless networks, including weaknesses in encryption protocols and unauthorized access points. Examples include Aircrack-ng and kismet.

Different types of Vulnerability Scanning Tools

- **Endpoint Security Scanners:** These tools scan individual endpoints (computers, laptops, mobile devices) for vulnerabilities such as outdated software, missing security patches, and malware infections. Examples include QualysGuard and McAfee Vulnerability Manager.

- **Cloud Security Scanners:** These tools assess the security posture of cloud-based infrastructure and services, identifying misconfigurations, access control issues, and vulnerabilities specific to cloud environments. Examples include CloudMapper and AWS Inspector.

Different types of Vulnerability Scanning Tools

- **Endpoint Security Scanners:** These tools scan individual endpoints (computers, laptops, mobile devices) for vulnerabilities such as outdated software, missing security patches, and malware infections. Examples include QualysGuard and McAfee Vulnerability Manager.

- **Cloud Security Scanners:** These tools assess the security posture of cloud-based infrastructure and services, identifying misconfigurations, access control issues, and vulnerabilities specific to cloud environments. Examples include CloudMapper and AWS Inspector.

It's important to note that while **vulnerability scanning tools are valuable for identifying security weaknesses, they are just one component of a comprehensive cybersecurity strategy.** Regular scanning, along with patch management, secure configuration practices, and employee training, are all essential for maintaining a strong security posture.

Working of Vulnerability Scanner Tools - Example NESSUS

Nessus is known for its comprehensive scanning capabilities and extensive database of known vulnerabilities.

- **Discovery:** The scanning process begins with Nessus attempting to discover all devices connected to the network. It sends out various probes and requests to identify active IP addresses and open ports. For example, it might send out ICMP echo requests (ping) to determine which IP addresses are responsive, and then it might perform port scans using TCP SYN or UDP probes to identify open ports on each device.

- **Enumeration:** Once Nessus has identified active devices and open ports, it begins gathering more detailed information about each device and service. It might send requests for information using protocols like SNMP (Simple Network Management Protocol) for network devices or SSH (Secure Shell) for Unix-based systems. For example, it might query a server to determine its operating system, installed software packages, and running services along with their versions.

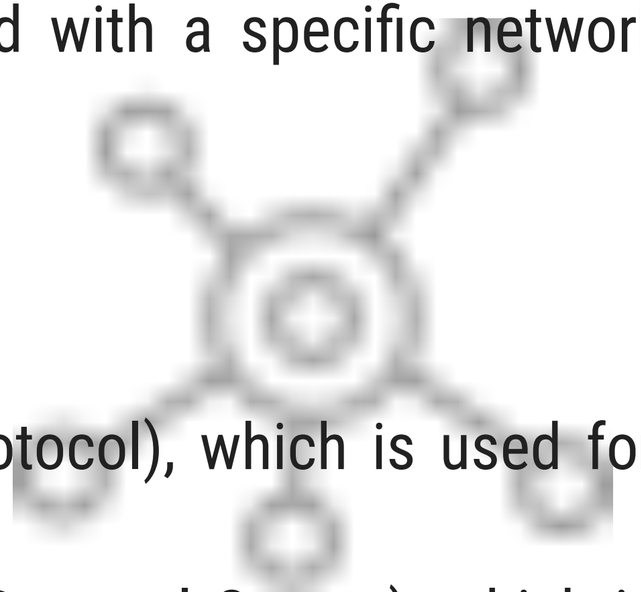
Working of Vulnerability Scanner Tools - Example NESSUS

- **Vulnerability Assessment:** Using the information gathered during enumeration, Nessus compares it against its database of known vulnerabilities. This database contains signatures and checks for a wide range of vulnerabilities, including those related to operating systems, network services, and applications. For example, if Nessus discovers that a server is running an outdated version of Apache web server software, it will check its vulnerability database for known exploits or vulnerabilities associated with that version of Apache.

- **Reporting:** Once the scanning process is complete, Nessus generates a detailed report outlining the vulnerabilities discovered, their severity levels, and recommendations for remediation. The report might include information such as CVE (Common Vulnerabilities and Exposures) identifiers, descriptions of the vulnerabilities, and suggested actions to mitigate or fix them. This report helps system administrators prioritize and address the most critical security issues first.

PORTS

Ports are virtual endpoints used by network protocols to enable communication between devices over a network. They facilitate the transfer of data between applications or services running on different devices. Each port on a device is associated with a specific network service or application



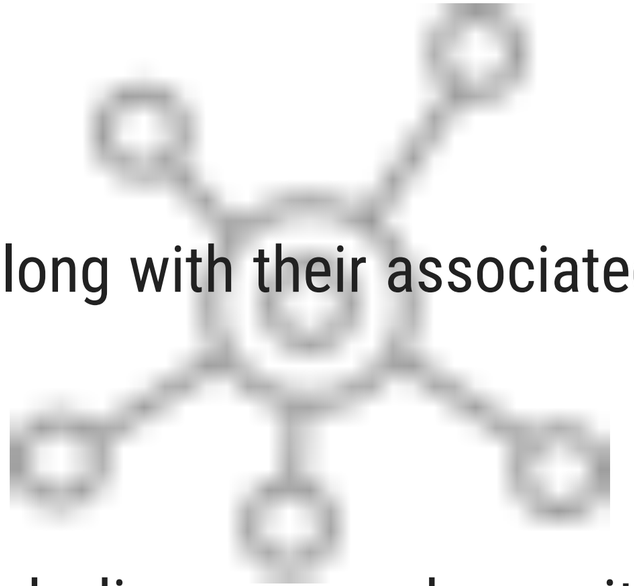
Here are some examples of important ports:

Port 80: This is the default port for HTTP (Hypertext Transfer Protocol), which is used for accessing websites and web services.
Port 443: This is the default port for HTTPS (Hypertext Transfer Protocol Secure), which is HTTP encrypted with SSL/TLS. It is used for secure communication over the web.
Port 22: This is the default port for SSH (Secure Shell), a cryptographic network protocol used for secure remote access to systems and for secure file transfer.
Port 25: This is the default port for SMTP (Simple Mail Transfer Protocol), which is used for sending email messages between servers.
Port 21: This is the default port for FTP (File Transfer Protocol), which is used for transferring

PORTS - How to find open ports

- **Using netstat command:**

On Unix-like systems (Linux, macOS), you can use the netstat command to display network connections, routing tables, and a list of open ports. For example:



netstat -tuln

This command will list all listening (-l) TCP (-t) and UDP (-u) ports along with their associated process identifiers (PIDs).

- **Using nmap:**

Nmap (Network Mapper) is a powerful open-source tool for network discovery and security auditing. It can be used to scan a system for open ports, detect services running on those ports, and gather other information about the network.

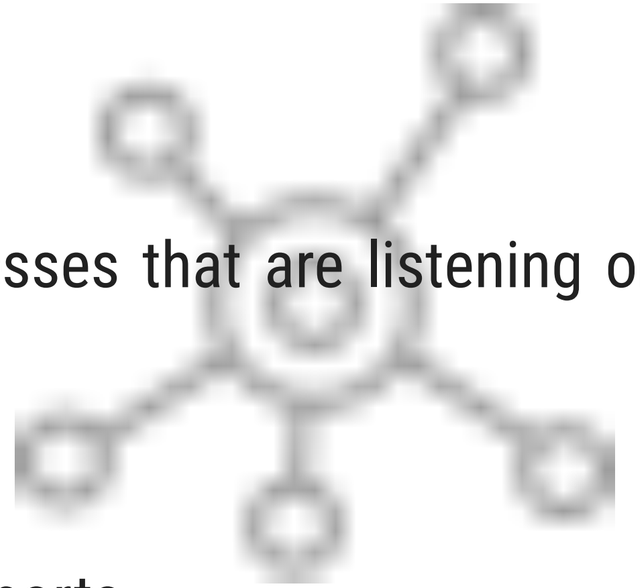
nmap -p- <target>

This command will scan all 65,535 TCP ports on the target system and report which ones are

PORTS - How to find open ports

- **Using lsof command:**

On Unix-like systems, you can also use the lsof command (list open files) to list all open ports and associated processes.



sudo lsof -i -P -n | grep LISTEN

This command will display all listening ports along with the processes that are listening on them.

Using PowerShell:

On Windows systems, you can use PowerShell cmdlets to find open ports.

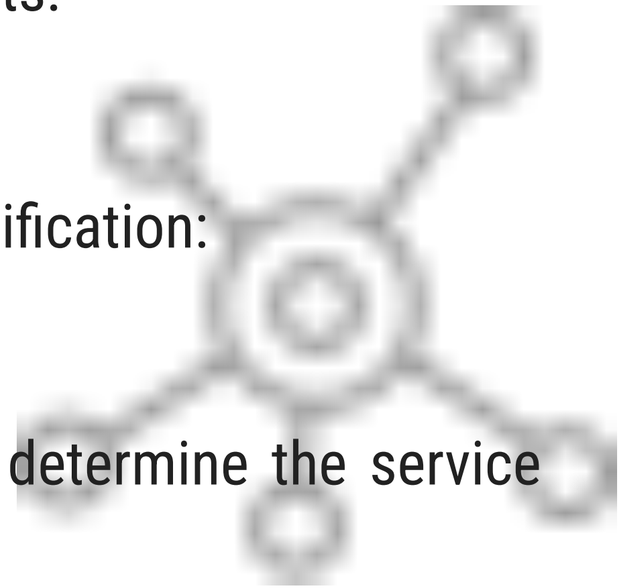
Get-NetTCPConnection | Where-Object {$_.State -eq 'Listen'}

This PowerShell command will display all TCP ports that are in the listening state.

Darshan UNIVERSITY

Service Identification

Service identification of ports involves determining which network services or applications are running on specific ports. This process is essential for understanding the functionality of a system, identifying potential security risks, and performing network audits.



Using Nmap:

You can use the following command with Nmap to perform service identification:

nmap -sV <target>

This command will scan the target system (<target>) and attempt to determine the service version (-sV) running.

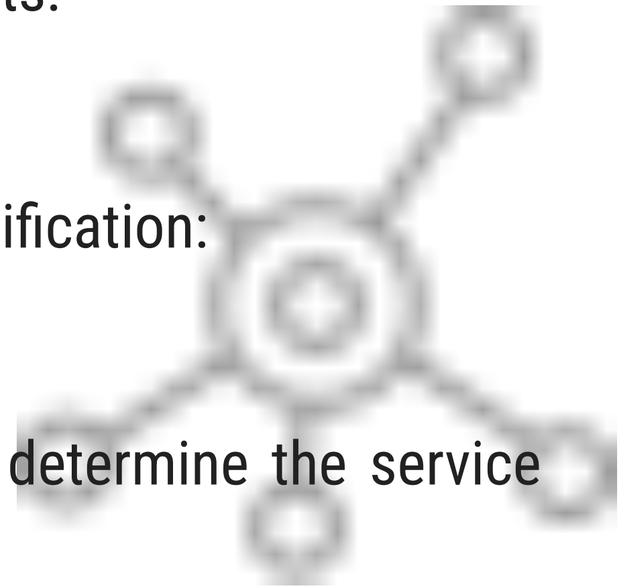
Manual Inspection:

You can manually inspect open ports by attempting to connect to them using appropriate client software.

For example, if you encounter an open port 80, you can try accessing it using a web browser to see if it serves HTTP content.

Banner Grabbing

Service identification of ports involves determining which network services or applications are running on specific ports. This process is essential for understanding the functionality of a system, identifying potential security risks, and performing network audits.



Using Nmap:

You can use the following command with Nmap to perform service identification:

nmap -sV <target>

This command will scan the target system (<target>) and attempt to determine the service version (-sV) running.

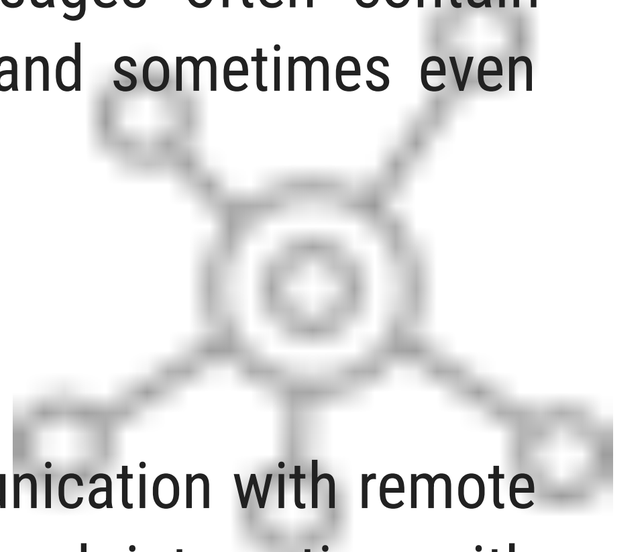
Manual Inspection:

You can manually inspect open ports by attempting to connect to them using appropriate client software.

For example, if you encounter an open port 80, you can try accessing it using a web browser to see if it serves HTTP content.

Banner Grabbing

Banner grabbing is a technique used in network reconnaissance to gather information about network services by capturing and analyzing the banner messages sent by those services when a connection is established. These banner messages often contain valuable information such as the service name, version number, and sometimes even additional details about the software running on the server.

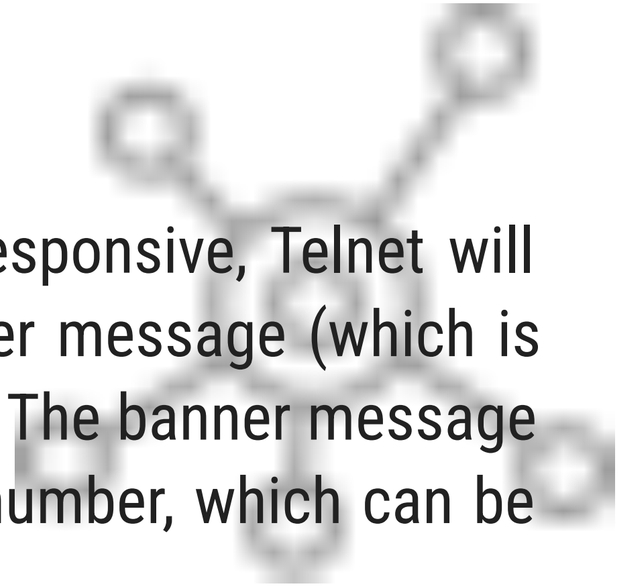


Using Telnet:

Telnet is a command-line tool that can establish text-based communication with remote systems over the network. It's commonly used for debugging and interacting with network services. When connecting to a service with Telnet, the service may send a banner message that provides information about itself.

Banner Grabbing - Telnet example

Let's say you want to perform banner grabbing on a web server running on port 80 of a target system with the IP address 192.168.1.100.
Here's how you can do it using the Telnet command:



telnet 192.168.1.100 80

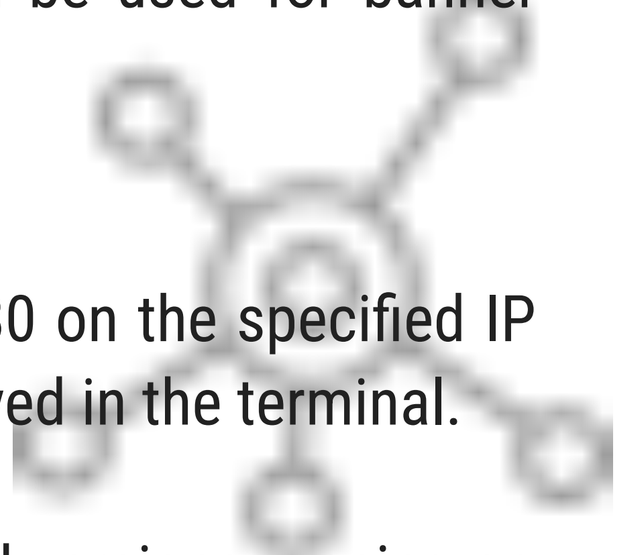
After executing this command, if the web server is running and responsive, Telnet will establish a connection to port 80. If the web server sends a banner message (which is common for HTTP servers), you will see it displayed in the terminal. The banner message may contain information such as the server software and version number, which can be useful for further analysis.
For example, if the web server is running Apache, you might see a banner message like:

Trying 192.168.1.100...
Connected to 192.168.1.100.
Escape character is '^]'.
HTTP/1.1 400 Bad Request
Content-Type: text/html

Banner Grabbing - Telnet example

Using Netcat (nc):

Netcat, also known as nc, is another command-line tool that can be used for banner grabbing. You can use it in a similar way to Telnet:



nc 192.168.1.100 80

This command will attempt to establish a TCP connection to port 80 on the specified IP address. If the web server sends a banner message, it will be displayed in the terminal.

Banner grabbing can provide valuable insights into the software and versions running on a target system, which is useful for security assessments, troubleshooting, and reconnaissance activities. However, it's essential to use this technique responsibly and ethically, respecting the privacy and security of the target systems.

Version Check

Version checking typically refers to the process of verifying the versions of software, firmware, or other components within a system to ensure they are up-to-date and free from known vulnerabilities or weaknesses. This practice is crucial for maintaining the security of systems and networks because outdated software often contains known security flaws that attackers can exploit.

Inventory: First, organizations need to maintain an inventory of all the software and hardware components within their systems. This includes operating systems, applications, libraries, firmware, and any other software dependencies. Inventory management systems like Lansweeper, PDQ Inventory, or Snipe-IT maintain a database of installed software and their versions across the organization.

Version Identification: Once the inventory is established, security teams can identify the versions of each component. This involves determining the specific version numbers or build identifiers associated with installed software.Vulnerability scanning tools like Nessus, OpenVAS, and Qualys can automatically scan networks and systems to identify

Version Check

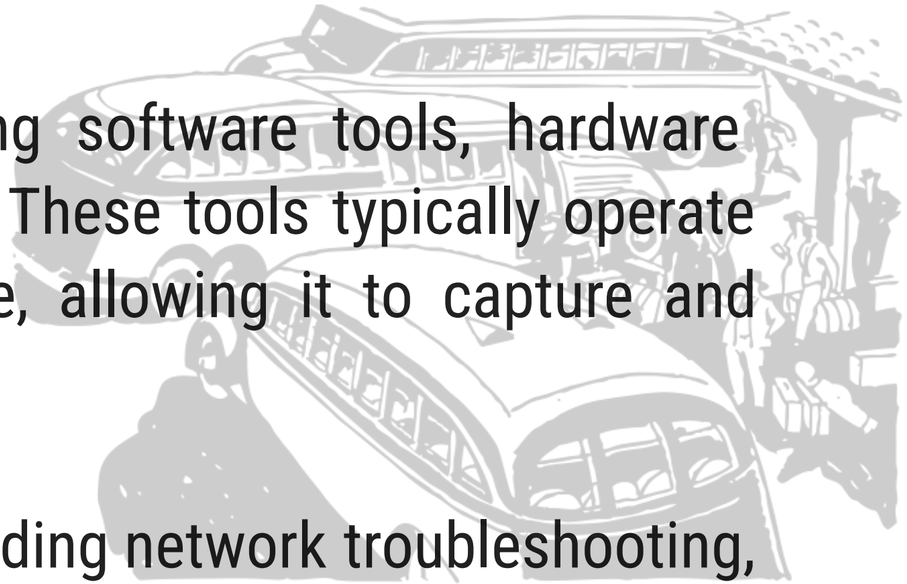
Vulnerability Assessment: Security teams then cross-reference the version information against known databases of vulnerabilities and security advisories. This could include databases like the National Vulnerability Database (NVD) or vendor-specific security bulletins.

Patch Management: If vulnerabilities are identified in any of the components, organizations must take appropriate action to remediate them. This often involves applying patches or updates provided by the software vendors or manufacturers. Patch management processes help ensure that systems are kept up-to-date with the latest security fixes. Patch management solutions such as Microsoft WSUS (Windows Server Update Services), SCCM (System Center Configuration Manager), or third-party tools like Ivanti or ManageEngine help organizations deploy patches and updates to software and operating systems.

Version checking is a critical aspect of maintaining cybersecurity hygiene and ensuring the security of systems and networks against known vulnerabilities. It helps

Traffic Probe

A traffic probe is a tool or mechanism used to monitor and analyze network traffic. It inspects data packets as they pass through a network interface, capturing information such as the source and destination addresses, protocols used, packet size, and content.

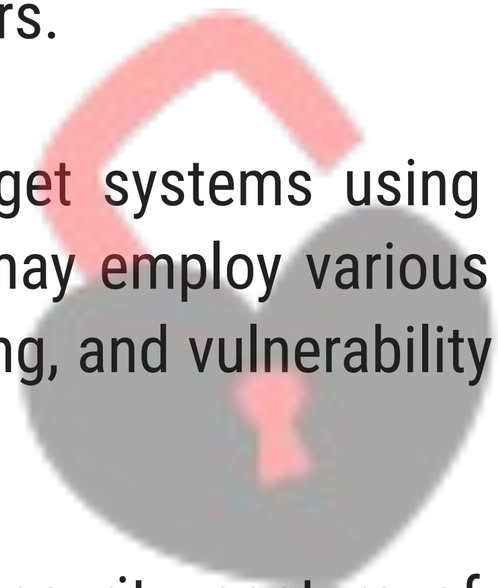


How it's done: Traffic probes can be implemented using software tools, hardware appliances, or specialized network monitoring equipment. These tools typically operate by placing the network interface into promiscuous mode, allowing it to capture and analyze all traffic passing through the network segment.

Why it's done: Traffic probing serves several purposes, including network troubleshooting, performance monitoring, security analysis, and compliance auditing. By monitoring network traffic, organizations can identify anomalies, detect potential security threats, and gain insights into the overall health and efficiency of their network infrastructure.

Vulnerability Probe

A vulnerability probe is a tool or process used to identify weaknesses or vulnerabilities in software, systems, or networks. It scans target systems for known vulnerabilities, misconfigurations, or security flaws that could be exploited by attackers.



How it's done: Vulnerability probes typically work by scanning target systems using predefined signatures, patterns, or exploit techniques. These tools may employ various methods, such as port scanning, service enumeration, version checking, and vulnerability fingerprinting, to identify potential weaknesses.

Why it's done: Vulnerability probing is conducted to assess the security posture of systems and networks, identify potential entry points for attackers, and prioritize remediation efforts. By discovering and addressing vulnerabilities proactively, organizations can reduce the risk of security breaches, data compromises, and service disruptions.

Vulnerability Examples

Vulnerabilities refer to weaknesses or flaws in software, hardware, or configurations that could be exploited by attackers to compromise the confidentiality, integrity, or availability of systems and data. Examples of vulnerabilities include:

Buffer Overflow: A programming error that occurs when a program attempts to write data beyond the boundaries of a memory buffer, potentially leading to arbitrary code execution.

SQL Injection: A technique used to exploit vulnerabilities in web applications by inserting malicious SQL queries into input fields, allowing attackers to manipulate databases or execute unauthorized commands.

Cross-Site Scripting (XSS): A vulnerability found in web applications that allows attackers to inject malicious scripts into web pages viewed by other users, leading to unauthorized access, data theft, or session hijacking.

Remote Code Execution (RCE): A vulnerability that allows attackers to execute arbitrary code on a target system, often leading to complete compromise of the system and unauthorized access to sensitive data.

Network Vulnerability Scanner - Netcat

Netcat, often abbreviated as nc, is a versatile networking utility commonly used for reading from and writing to network connections using TCP or UDP protocols. It's a Swiss Army knife tool for network troubleshooting, testing, and data transfer.

Why Netcat is Used:

Network Debugging: Netcat can be used to diagnose network connectivity issues, test network services, and troubleshoot network configurations.
Port Scanning: Netcat can perform basic port scanning to check for open ports on a remote host, helping identify potential entry points for attackers.
File Transfer: Netcat can transfer files between systems over a network connection, either interactively or using pipes.
Remote Shell Access: Netcat can be used to establish a simple command-line interface between two systems, allowing remote shell access.
Banner Grabbing: Netcat can retrieve banner information from network services, providing details about the software and version running on a remote server.

Understanding Ports and Services Tool - Nmap

Functionality: Nmap is a powerful open-source network scanning tool used for discovering hosts and services on a computer network, thus creating a "map" of the network's topology. It employs various techniques to gather information about target hosts, such as port scanning, version detection, OS fingerprinting, and service enumeration.

How it Works: Nmap sends specially crafted packets to target hosts and analyzes their responses to determine which ports are open, what services are running on those ports, and sometimes even what operating system the target is using. It supports different scan types, including TCP connect scans, SYN scans, UDP scans, and more, allowing users to tailor their scans based on their specific needs.

Use Cases: Nmap is widely used by network administrators, security professionals, and ethical hackers for network inventory, vulnerability assessment, security auditing, and penetration testing.

Network Sniffers and Injection Tools - Tcpdump

Functionality: Tcpdump is a command-line packet analyzer tool for capturing and analyzing network traffic in real-time. It allows users to inspect individual packets, filter traffic based on various criteria, and save captured packets to a file for offline analysis.

How it Works: Tcpdump operates by capturing packets from a network interface in promiscuous mode, meaning it captures all packets traversing the interface, not just those intended for the host system. It then parses and displays packet contents, including headers and payload data, according to user-defined filters and display options.

Use Cases: Tcpdump is commonly used for network troubleshooting, performance monitoring, security analysis, and protocol debugging. It helps network administrators and security analysts gain insights into network traffic patterns, identify anomalies, and investigate security incidents.

Network Sniffers and Injection Tools - Wireshark

Functionality: Wireshark is a graphical network protocol analyzer that provides a comprehensive view of network traffic in real-time. It offers a user-friendly interface for capturing, analyzing, and dissecting packets from various network interfaces and protocols.

How it Works: Wireshark operates similarly to Tcpdump by capturing packets from network interfaces in promiscuous mode. However, Wireshark presents captured packets in a more user-friendly and visually appealing manner, allowing users to drill down into packet details, apply complex filters, and perform advanced protocol analysis.

Use Cases: Wireshark is widely used for network troubleshooting, protocol analysis, network forensics, and security monitoring. It helps network administrators, security analysts, and developers diagnose network issues, detect anomalies, and analyze protocol behavior.

Network Sniffers and Injection Tools - Wireshark

Functionality: Wireshark is a graphical network protocol analyzer that provides a comprehensive view of network traffic in real-time. It offers a user-friendly interface for capturing, analyzing, and dissecting packets from various network interfaces and protocols.

How it Works: Wireshark operates similarly to Tcpdump by capturing packets from network interfaces in promiscuous mode. However, Wireshark presents captured packets in a more user-friendly and visually appealing manner, allowing users to drill down into packet details, apply complex filters, and perform advanced protocol analysis.

Use Cases: Wireshark is widely used for network troubleshooting, protocol analysis, network forensics, and security monitoring. It helps network administrators, security analysts, and developers diagnose network issues, detect anomalies, and analyze protocol behavior.

Scanning for Web Vulnerabilities - Nikto

Functionality: Nikto is an open-source web server scanner that performs comprehensive tests against web servers for known vulnerabilities and misconfigurations. It scans web servers for over 6700 potentially dangerous files/CGIs, outdated server software, and various security issues.

How it Works: Nikto sends HTTP requests to the target web server and analyzes the responses to identify potential vulnerabilities and security weaknesses. It checks for common issues such as outdated software versions, insecure server configurations, and known vulnerabilities in web applications and server software.

Use Cases: Nikto is commonly used by security professionals, penetration testers, and system administrators to perform security assessments of web servers. It helps identify and remediate security vulnerabilities before they can be exploited by attackers.

Scanning for Web Vulnerabilities - Nikto - Use Commands

Basic scan against a target:

nikto -h <target>

Scan a target and output results to a file:

nikto -h <target> -o <output_file>

Scan a target over SSL:

nikto -h <target> -ssl

Scanning for Web Vulnerabilities - W3AF (Web Application Attack and Audit Framework)

Functionality: w3af is an open-source web application security testing framework that helps identify and exploit security vulnerabilities in web applications. It supports both black-box and white-box testing methodologies and includes a wide range of plugins for vulnerability scanning, exploitation, and reporting.

How it Works: w3af analyzes web applications by sending crafted HTTP requests and analyzing the responses for security vulnerabilities. It supports various scanning techniques, including SQL injection, cross-site scripting (XSS), directory traversal, and more.

Use Cases: w3af is used by security professionals, ethical hackers, and developers to assess the security posture of web applications. It helps identify vulnerabilities in web applications and provides recommendations for remediation.

Scanning for Web Vulnerabilities - W3AF - Use Commands

Launch w3af's console UI:

w3af_console

Perform a vulnerability scan against a target:

w3af_console -s <target>

HTTP Utilities - CURL

Functionality: curl is a command-line tool and library for transferring data with URLs. It supports various protocols, including HTTP, HTTPS, FTP, FTPS, SCP, SFTP, LDAP, and more. curl can be used to send and receive data from web servers, download files, and perform other network-related tasks.

How it Works: curl operates by sending HTTP requests to web servers and receiving responses, which can be saved to files or displayed on the console. It supports various options and parameters for customizing requests, handling authentication, and performing data transfers.

Use Cases: curl is commonly used by developers, system administrators, and security professionals for various tasks, including testing web services, downloading files from remote servers, and interacting with APIs.

HTTP Utilities - CURL - Use Commands

Fetch a webpage:

curl <url>

Download a file:

curl -O <url>

Send a POST request with data:

curl -X POST -d 'param1=value1&param2=value2' <url>

HTTP Utilities - STUNNEL

Functionality: stunnel is an open-source SSL/TLS encryption wrapper that provides secure encrypted connections for network services. It acts as a proxy between clients and servers, encrypting data transmitted over unsecured network connections.

How it Works: stunnel establishes SSL/TLS-encrypted connections between clients and servers by wrapping plaintext traffic with SSL/TLS encryption. It can be configured to encrypt traffic for various network services, including SMTP, POP3, IMAP, HTTP, and more.

Use Cases: stunnel is used to secure network communications between clients and servers by encrypting data transmitted over unsecured connections. It's commonly used to add SSL/TLS encryption to legacy network services that do not natively support encryption, such as SMTP, POP3, and IMAP. Additionally, it can be used to secure connections to web servers, databases, and other network services.

HTTP Utilities - STUNNEL - Use Commands

Create a basic SSL tunnel for a service:

stunnel -c -d <local_port> -r <remote_host>:<remote_port>

Use stunnel as a client:

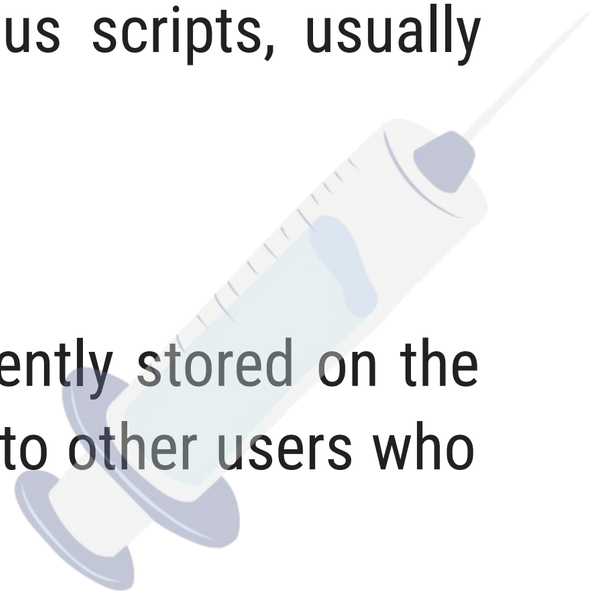
stunnel -c -r <remote_host>:<remote_port> -d <local_port>

Use stunnel as a server:

stunnel -p <cert.pem> -d <local_port> -r <remote_host>:<remote_port>

OWASP TOP 10 - Cross Site Scripting (XSS) Vulnerability

Cross-Site Scripting (XSS):
XSS is a vulnerability that occurs when an attacker injects malicious scripts, usually JavaScript, into web pages viewed by other users.



Stored XSS:
In Stored XSS, the attacker injects malicious scripts that are permanently stored on the target server, such as in a database. These scripts are then displayed to other users who visit the affected page.

Example:
Imagine a web application has a comment section where users can leave comments. If the application doesn't properly sanitize user input, an attacker could submit a comment containing malicious JavaScript code, which gets stored in the database. When other users view the comments, the malicious code executes in their browsers, potentially stealing their session cookies or redirecting them to phishing sites.

OWASP TOP 10 - Cross Site Scripting (XSS) Vulnerability

Reflected XSS:
Reflected XSS occurs when the injected script is reflected off the web server, such as in an error message or search result. The malicious script is sent to the server, which then reflects it back in the response.

Example:
Consider a search page where users can enter keywords to search for products. If the search term is not properly sanitized and reflected back in the search results page, an attacker could craft a URL containing malicious script and trick a user into clicking it. When the user clicks the link, the script executes in their browser within the context of the vulnerable page.

OWASP TOP 10 - Cross Site Scripting (XSS) Vulnerability

DOM-based XSS:
DOM-based XSS arises when the client-side JavaScript code reads data from the DOM (Document Object Model) and then dynamically updates the page's content in a way that inadvertently executes malicious code.

Example:
Suppose a website uses JavaScript to display personalized messages to users based on their user ID in the URL. If the website fails to properly sanitize the user ID parameter, an attacker could craft a URL with a malicious script as the user ID. When the page loads and the JavaScript retrieves the user ID from the URL to display the personalized message, the malicious script executes in the user's browser.

OWASP TOP 10 - Insecure Deserialisation

Insecure Deserialization:
Insecure deserialization is a vulnerability that occurs when an application deserializes data from an untrusted source without proper validation, leading to various security risks.

Example:
Imagine an application that allows users to upload serialized objects, such as user settings or session data. If the application deserializes this data without proper validation, an attacker could upload a maliciously crafted serialized object. When the application deserializes this object, it could execute arbitrary code, leading to severe consequences such as remote code execution, privilege escalation, or denial of service.

For instance, in a Java application, insecure deserialization vulnerabilities can arise when using frameworks like Apache Commons Collections. Attackers can exploit these vulnerabilities by crafting malicious serialized objects to execute arbitrary code when deserialized by the application.