Corvit System Multan

Report No: 3rd week 2nd Day

Report Title: Network Scanning and Enumeration

Submitted By: Muhammad Adnan

Submitted To: Muhammad Bilal

Date: 6/08/2024

Operating System Detection

Definition:

*Operating system detection is the process of determining which operating system (OS) a computer or device is using. This is important for ensuring compatibility with software, performing troubleshooting, or configuring system settings***.**

1- **Nmap:**
   *Nmap (Network Mapper) is a powerful tool used for network discovery and security auditing. One of its features is OS detection, which allows you to determine the operating system of a target device on a network. This can be useful for network administrators, security professionals, or anyone wanting to understand the systems they are interacting with***.**

2- **Passive OS Fingerprinting:**
   Passive OS Fingerprinting is a method of identifying the operating system of a networked device without actively sending packets or interacting with it directly. Instead, it involves observing and

*analyzing the network traffic that the target device generates or responds to. This technique is less intrusive than active fingerprinting and can be used to gather information without alerting the target or affecting its normal operation***.**

Tools and Techniques

- Nmap:

  Nmap (Network Mapper) is a powerful tool used for network discovery and security auditing. One of its features is OS detection, which allows you to determine the operating system of a target device on a network.

- **pOf:**

  Passive OS Fingerprinting (POF) *is a technique used to identify the operating system of a networked device without actively interacting with it. Instead of sending probes or requests to the device, POF involves monitoring and analyzing the traffic that the device generates or receives. This method is particularly useful for network administrators, security professionals, and researchers who need to gather information discreetly.*

- **Xprobe2:**

Xprobe2 is a network security tool used for passive OS fingerprinting, designed to identify the operating system of a remote host by analyzing the responses to crafted packets. Unlike active OS fingerprinting tools that send probes to the target, Xprobe2 uses a database of known OS signatures to match observed network behaviors to specific operating systems.

Key Features of Xprobe2

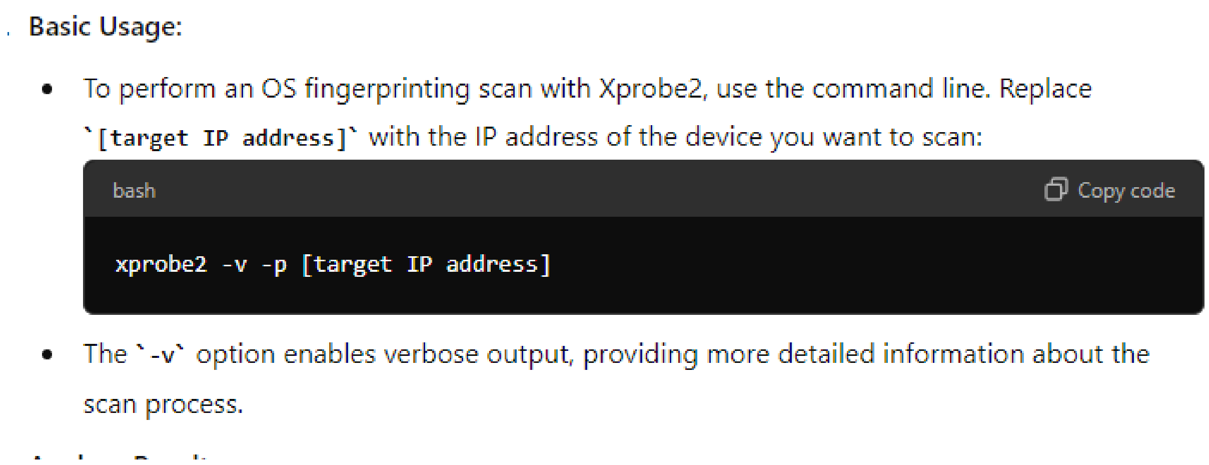
1.Passive OS Fingerprinting:

Technique: *Xprobe2 relies on observing how different operating systems respond to specific network packets. It doesn't need to interact with the target actively but can infer the OS based on responses or the absence of expected responses.*

Database: *It uses a pre-defined database of known OS fingerprint patterns to compare against the observed network behaviors.*

2.OS Detection:

Xprobe2 identifies the operating system of the target by sending a series of specially crafted packets and analyzing the *responses. It uses these responses to match against its database of operating system signatures.*



Basic Usage:

- To perform an OS fingerprinting scan with Xprobe2, use the command line. Replace `[target IP address]` with the IP address of the device you want to scan:

```bash
xprobe2 -v -p [target IP address]
```

- The `-v` option enables verbose output, providing more detailed information about the scan process.

Automated Scanning Workflows

Definition:

Automated scanning workflows involve the use of tools and processes to systematically assess and analyze systems, networks, or applications for vulnerabilities, compliance, or performance issues. These workflows help ensure that scans are consistent, comprehensive, and efficient, reducing the need for manual intervention and increasing the accuracy of findings

Examples

1- Continuous Integration (CI) Scans:

Continuous Integration (CI) Scans *refer to the practice of integrating automated scanning tools into the CI/CD (Continuous Integration/Continuous Deployment) pipeline to identify and address security vulnerabilities, code quality issues, and other problems in the software development lifecycle. The goal is to catch issues early and ensure that code is secure, stable, and ready for deployment.*

GitLab CI:

GitLab CI/CD (Continuous Integration/Continuous Deployment) is a powerful feature within GitLab that automates the process of testing, building, and deploying code. It integrates closely with GitLab's version control system, allowing developers to streamline their workflows and ensure that code changes are continuously tested and deployed.

GitLab CI/CD Configuration File:

.gitlab-ci.yml : *This YAML file defines the pipeline, including stages, jobs, and their dependencies. It resides in the root of the repository.*

Benefits of GitLab CI/CD

- ***Automation:*** *Automates repetitive tasks, reducing manual errors and speeding up the development process.*
- ***Integration:*** *Seamlessly integrates with GitLab's version control, issue tracking, and other features.*
- ***Scalability:*** *Supports complex workflows and can scale with your development needs.*
- ***Visibility:*** *Provides clear visibility into the pipeline's status and history, helping teams identify and resolve issues quickly.*

Scheduled Network Scans:

*Scheduled Network Scans involve setting up automated scans to periodically assess the security and performance of a network. These scans can identify vulnerabilities, monitor network health, and ensure compliance with security policies. By scheduling scans, you ensure that your network is regularly checked for issues without requiring manual initiation each time***.**

Tools and Techniques:

- ### Nessus

  Nessus is a robust tool for vulnerability management and security assessment, offering a range of features to help organizations *identify and address security issues effectively. By incorporating Nessus into your security practices, you can enhance your ability to protect your systems and data from potential threats*

OpenVAS:

OpenVAS (Open Vulnerability Assessment System) is an open-source vulnerability scanning tool designed to assess the security of networked systems and applications. It is part of the Greenbone Vulnerability Management (GVM) suite and provides a comprehensive solution for detecting vulnerabilities, misconfigurations, and security issues.

Scanning For Web Application

Definition:

Scanning a web application is an essential part of ensuring its security and performance. There are various approaches and tools available for scanning web applications, depending on what you're looking to assess—whether it's vulnerabilities, performance issues, or compliance with standards. Here's a high-level overview of how you might approach web application scanning.

Examples

1. **SQL Injection Scan:**
   SQL Injection (SQLi) is a common and serious vulnerability where an attacker can manipulate SQL queries to gain unauthorized

access to a database, execute arbitrary SQL commands, or retrieve sensitive data. Scanning for SQL Injection involves testing the web application to identify potential vulnerabilities. Here's how you can perform a SQL Injection scan.

Steps to Perform a SQL Injection Scan:

- *Identify which parts of the application you will test (e.g., login forms, search fields).*
- *Determine the type of SQL Injection attacks you are testing for (e.g., classic SQLi, blind SQLi, time-based SQLi).*

***Select Tools**:*

- ***Automated Scanners**: Use tools specifically designed for SQL Injection testing.*
  - ***OWASP ZAP**: Open-source security scanner that can detect SQL Injection vulnerabilities.*
  - ***Burp Suite**: A popular tool with SQL Injection scanning features.*
  - ***SQLmap**: A powerful tool that automates the process of detecting and exploiting SQL Injection flaws.*

Manual Testing:

Input Testing: Manually inject SQL payloads into input fields to see if you can manipulate the SQL queries. Common payloads include:

  - `' OR '1'='1`
  - `' UNION SELECT NULL, NULL, NULL--`
  - `1=1—`

2. XSS Detection:

Cross-Site Scripting (XSS) is a security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can be used to steal cookies, perform actions on behalf of users, or manipulate content on the page. Detecting XSS involves identifying areas of the application where untrusted data is being output without proper sanitization or escaping.

Select Tools:

- **Automated Scanners:**
  - **OWASP ZAP**: Detects XSS vulnerabilities by injecting payloads and analyzing responses.
  - **Burp Suite**: Includes XSS detection features as part of its scanner.
  - **Acunetix**: A commercial tool that can find XSS vulnerabilities.

- **Manual Testing Tools:**
  - **XSSer**: An open-source tool for detecting XSS vulnerabilities.
  - **XSStrike**: A tool designed for advanced XSS scanning and payload generation.

Threat Intelligence Integration

Definition:

Threat intelligence integration means adding useful information about potential cyber threats into a company's security systems. This helps the company spot and handle threats more quickly and effectively, keeping their systems safer from attacks.

Examples

Security Information and Event Management (SIEM):

- *Examples: Splunk, IBM QRadar, ArcSight.*
- *Function: Collects and analyzes security data from across the organization to detect and respond to threats, often incorporating threat intelligence feeds.*

Firewall Rules Update:

Updating firewall rules involves changing the settings on a firewall to better control which network traffic is allowed or blocked. This process ensures the firewall effectively protects the network by blocking unwanted access and allowing legitimate communication based on the latest security needs.

Techniques and Tools:

Theat Intelligence Platforms (TIPs):

- **Examples**: *Recorded Future, ThreatConnect, Anomali.*
- **Function**: *Aggregate and analyze threat data from multiple sources, providing actionable insights and alerts*.

Security Information and Event Management (SIEM):

- **Examples**: Splunk, IBM QRadar, ArcSight.
- ***Function****: Collects and analyzes security data from across the organization to detect and respond to threats, often incorporating threat intelligence feeds*

Firewall:

*Updating firewall rules involves changing the settings on a firewall to better control which network traffic is allowed or blocked. This process ensures the firewall effectively protects the network by blocking unwanted access and allowing legitimate communication based on the latest security needs***.**

Wireless Network Scanning

Definition:

Updating firewall rules involves changing the settings on a firewall to better control which network traffic is allowed or blocked. This process ensures the firewall effectively protects the network by blocking unwanted access and allowing legitimate communication based on the latest security needs.

Examples:

1.Wi-Fi Security Assessment:

Wi-Fi security assessment involves evaluating the security of a wireless network to identify and address vulnerabilities.

2.Rogue Access Point Detection:

Rogue access point detection involves identifying unauthorized or potentially malicious wireless access points within a network. These rogue access points can be set up by attackers to intercept traffic or gain unauthorized access to the network.

Techniques and Tools:

1.Aircrack-ng:

Aircrack-ng is a suite of tools used for network security testing, particularly focused on wireless networks. It's commonly used for assessing the strength of Wi-Fi security and for performing attacks to identify vulnerabilities.

2.Kismet:

Kismet is an open-source wireless network detector, sniffer, and intrusion detection system. It is used for monitoring and analyzing wireless networks to gather detailed information about them.

3.Wireshark:

Wireshark is a widely used open-source network protocol analyzer that allows users to capture and examine the data traveling through a network. It is highly regarded for its ability to provide detailed insights into network traffic and troubleshoot network issues.

Automating Scans with Scripts

Definition:

Automating scans with scripts involves using custom scripts to perform security assessments or network scans on a scheduled or triggered basis. This process helps streamline repetitive tasks, ensures consistency, and can improve overall efficiency in identifying vulnerabilities or issues. Here's a brief overview of how to automate scans with scripts.

Examples:

1.Bash Scripts for Nmap:

Bash scripts can be used to automate Nmap scans, making it easier to perform regular network assessments or targeted scans.

2.Python Script for Vulnerability Scanning:

Creating a Python script for vulnerability scanning typically involves integrating with vulnerability scanning tools or libraries. One popular tool for this purpose is Nmap, and Python can interact with Nmap via libraries such as python-nmap. Below is a basic example of a Python script that uses the python-nmap library to perform a vulnerability scan by executing an Nmap scan and processing the results.

Techniques and Tools:

1.Bash/Shell Scripting:

Bash (Bourne Again Shell) and shell scripting are essential for automating tasks and managing system operations on Unix-like systems. Shell scripts can perform a wide range of tasks, from simple file operations to complex system management and security scanning. Here's an overview and some basic examples of what you can do with Bash shell scripting:

-Python:

Using python libraries such as 'subprocess' to automate scanning tools.

PowerShell:

PowerShell is a powerful scripting language and command-line shell developed by Microsoft, primarily used for automating administrative tasks and managing system configurations on Windows, though it has cross-platform support for Linux and macOS as well. It integrates with *the .NET framework and provides a rich set of cmdlets (built-in commands) for performing various tasks.*

Scanning Best Practices:

Definition:

When performing scanning for security assessments or network management, following best practices ensures that the process is effective, minimizes disruption, and provides reliable results.

Scanning Ethics and Legal Considerations

Definition:

When performing scanning activities, whether for network security, vulnerability assessment, or compliance, it's crucial to be aware of both ethical and legal considerations to ensure that your actions are responsible and lawful. Here's a comprehensive overview of these considerations.

Ethical Considerations:

Ethical and legal considerations in scanning involve obtaining proper authorization, respecting privacy, minimizing impact, avoiding exploitation, ensuring compliance with laws, and maintaining professionalism. By adhering to these principles, you can perform scanning activities in a responsible and legally compliant manner, contributing positively to security and operational efficiency while avoiding legal and ethical pitfalls.

Legal Guidelines:

Follow ethical guidelines provided by organizations such as (ISC)2 or ISACA

Port-Scanning Analysis

Definition:

Port scanning is a technique used to identify open ports and services on a networked system. Analyzing the results of port scans is crucial for understanding network security, identifying vulnerabilities, and managing network configurations. Here's a detailed guide on how to analyze port-scanning results effectively.

Vulnerability Prioritization:

Vulnerability prioritization is the process of evaluating and ranking security vulnerabilities to determine which ones should be addressed first. This is crucial for effective risk management, as it helps organizations focus resources on the most critical issues that could potentially cause significant harm

Remediation Planning:

Remediation planning is the process of developing and implementing strategies to address identified vulnerabilities and security weaknesses in an organization's systems or networks. Effective remediation planning helps mitigate risks, improve security posture, and ensure compliance with security policies and regulations.

Reporting Tools:

- Nessus or Qualys

Threats Modeling:

- Incorporate Threats modeling techniques to access the potential impact of identified vulnerabilities.

Scanning for Insider Threats

Definition:

Scanning for insider threats involves monitoring and analyzing activities within an organization to detect and mitigate potential security risks posed by individuals with internal access. Unlike external threats, insider threats come from people within the organization, making detection more challenging but crucial.

Examples

1.User Behaviour Analytics (UBA):

User Behavior Analytics (UBA) is a security approach that focuses on analyzing and understanding user behavior patterns within an organization to identify potential security threats. UBA tools and techniques help detect anomalies that may indicate malicious activities, insider threats, or compromised accounts. Here's a detailed look at UBA.

2.Acess Monitoring:

Access Monitoring is the process of keeping track of who is using your systems, apps, and data to make sure that everything is secure and used correctly.

Techniques and Tools:

1.Splunk User Behavior Analytics:

Splunk User Behavior Analytics (UBA) is a feature within the Splunk platform designed to analyze and detect unusual user behaviors that could indicate potential security threats. It leverages machine learning and advanced analytics to help organizations identify and respond to anomalies and risks effectively.

2.Varonis:

Varonis is a cybersecurity platform designed to protect sensitive data, detect insider threats, and manage data security through comprehensive monitoring and analytics. It focuses on data governance, user behavior analytics, and threat detection to help organizations secure their information and respond to potential risks.