Performing Passive Reconnaissance

Reconnaissance is the initial step in a cyber attack where an attacker gathers information about the target. There are two types of reconnaissance: active and passive. Active reconnaissance involves sending probes to the target network or systems, while passive reconnaissance does not interact directly with the target, using third-party databases and eavesdropping on network traffic instead.

Common active reconnaissance methods include host, network, user, group, network share, web page, application, and service enumeration, as well as packet crafting. Passive reconnaissance methods include domain enumeration, packet inspection, open-source intelligence (OSINT), Recon-ng, and eavesdropping.

Performing active reconnaissance typically starts with a small amount of information, and then gather more while scanning, eventually moving on to different types of scans and gathering additional information. Some techniques used by attackers include DNS lookups, identification of technical and administrative contacts, social media scraping, and inspecting cryptographic flaws in SSL certificates. Certificate transparency is another tool attackers can use to gather information about an organization's subdomains and systems.

Security breaches can directly impact a company's reputation. Attackers use various methods to gather information, including password dumps, file metadata, strategic search engine analysis, website archiving, and public source code repositories. Tools like h8mail and WhatBreach exploit breached data repositories, while ExifTool reveals metadata in files. Advanced search engine operators can uncover sensitive information, and website archiving allows for a historical view of websites. Open-source intelligence (OSINT) gathering involves collecting and analyzing publicly available information, with Recon-ng being a powerful framework for this purpose. Shodan scans the internet for vulnerable hosts and other exposed systems.

Performing Active Reconnaissance

Performing active reconnaissance involves enumeration, which is the process of gathering information about a target during a penetration test. The first step is to identify the target's internet-facing hosts, followed by a port scan to enumerate the services running on those hosts. Nmap is a popular tool for such scans, including SYN scans, TCP connect scans, UDP scans, and TCP FIN scans.

A SYN scan sends a TCP SYN packet to the target port and analyzes the response to determine if the service is listening. TCP connect scans use the operating system's networking mechanism to establish a full TCP connection, which may trigger alarms on intrusion detection systems. UDP scans are useful for enumerating services like DNS, SNMP, or DHCP, which use UDP for communication. TCP FIN scans send a FIN packet to the target port, and if no response is received, the port is considered open.

Host discovery scans help determine if a host is online and responding on a network. Nmap also provides six timing templates (-T 0-5) to dictate the aggressiveness of a scan, ranging from very slow for IDS evasion to very aggressive, which may overwhelm targets or miss open ports.

Enumeration techniques used in the information-gathering include:

- **Host Enumeration**: Performed internally and externally, it involves scanning the IP addresses of a target using tools like Nmap or Masscan.
- **User Enumeration**: Collects a list of valid users to crack credentials by manipulating the Server Message Block (SMB) protocol on a Windows network.
- **Group Enumeration**: Helps determine authorization roles in the target environment by enumerating SMB groups using the Nmap NSE script **smb-enum-groups**.
- **Network Share Enumeration**: Identifies systems sharing files, folders, and printers on a network using the Nmap **smb-enum-shares** NSE script.
- **Web Page Enumeration/Web Application Enumeration**: Maps the attack surface of a web application using Nmap's **http-enum** NSE script and other tools like Nikto.
- **Service Enumeration**: Identifies services running on a remote system, primarily through Nmap's port scanning functionality.
- **Enumeration via Packet Crafting**: Scapy, a Python-based framework, can be used to perform network reconnaissance through packet generation.

Additionally, packet inspection and eavesdropping can be performed using tools like Wireshark, tshark, and tcpdump, aiding in passive reconnaissance during penetration testing.

Understanding the Art of Performing Vulnerability Scans

Vulnerability scanning is the process of identifying weaknesses in a system by probing services to determine if they are vulnerable. Vulnerability scanners use different methods, but typically follow a four-step process: discovery, software/version identification, vulnerability correlation, and report generation. However, these reports may contain false positives, so validation is crucial.

There are various types of vulnerability scans including:

- unauthenticated (scanner operates without credentials)
- authenticated (scanner uses root-level access credentials)
- discovery (scanner identifies the attack surface of a target)
- full (scanner enables all scanning options)
- stealth (scanner minimizes noise to avoid detection)
- passive (scanner monitors and analyzes network traffic)
- compliance (scanner checks for adherence to industry regulations).

Each type of scan has its own strengths and limitations. For example, unauthenticated scans only show exposed network services, while authenticated scans provide more comprehensive information. Stealth scans are useful for production environments, but may not detect all vulnerabilities. Compliance scans address specific industry requirements, but can be challenging due to varying interpretations of regulations.

Challenges to consider when running a vulnerability scan on a network or device include:

- **Best Time to Run a Scan**: Scans on production networks should be done carefully to minimize impact on users and servers, typically during early hours when network usage is low.
- **Determining Protocols in Use**: Identify whether the target device uses TCP, UDP, or both, so vulnerabilities in both services are assessed.
- **Network Topology**: Scans should be performed as close to the target as possible to avoid impacting devices along the path and affecting scan results.
- **Bandwidth Limitations**: Scanner settings may need to be adjusted for lower-bandwidth situations to prevent bandwidth consumption issues.
- **Query Throttling**: Slowing down scanner traffic can help manage bandwidth limitations. This can be achieved by reducing attack threads or the scope of plugins/attacks.
- **Fragile Systems/Nontraditional Assets**: Vulnerability scanners may need to adjust scanning options for fragile systems, such as printers or IoT devices, to avoid crashing them. Alternatively, these devices may be exempted from scans, but this could reduce overall security.

Understanding How to Analyze Vulnerability Scan Results

Running a vulnerability scan is the easy part of identifying potential threats; the main challenge lies in analyzing the results. Vulnerability scanning tools can produce false positives, which need to be eliminated to accurately identify actual vulnerabilities. Reducing false positives is particularly important when providing a report for a paid penetration testing assignment.

Eliminating false positives involves validating version information and investigating the details of the vulnerability. Each vulnerability maps to items in the Common Vulnerabilities and Exposures (CVE) list, which should be examined to better understand the criteria.

Various organizations and resources, such as US-CERT, the CERT Division of Carnegie Mellon University, NIST, JPCERT, CAPEC, CVE, CWE, and CVSS, provide helpful information for further investigation of vulnerabilities. When dealing with a vulnerability, it is important to determine its priority by assessing its severity, the number of affected systems, and other factors.

Overall, properly analyzing vulnerability scan results involves a detailed examination of the tool's findings and prioritizing vulnerabilities for mitigation based on their severity and potential impact.