

Cisco Ethical Hacker — Your Original Notes

Converted: preserving your exact content

module 1 planning and scoping a pentest assessment (comparing and contrasting governance , risk and compliance concepts | explaining the importance of scoping and organizational or customer requirements | demonstrating and ethical hacking mindset by maintaining professionalism and integrity)

2.4.1 What Did I Learn in this Module?

Comparing and Contrasting Governance, Risk, and Compliance Concepts (

The planning and preparation phase is crucial in any penetration testing engagement and involves scoping the project properly. This includes understanding the target audience, rules of engagement, communication channels, available resources, budget, technical constraints, and any disclaimers. In addition, it is essential to be familiar with various regulatory compliance considerations, including PCI DSS, HIPAA, FedRAMP, and GDPR. Most of these regulations require third-party penetration testing to verify compliance and assess the security posture of the organization. It is important to be familiar with these regulations and their checklists for a successful compliance-based assessment.

Regulations in the Financial Sector

The financial sector is responsible for safeguarding customer information and maintaining the critical infrastructure of financial services. Regulations applicable to the financial sector include the Gramm-Leach-Bliley Act (GLBA), the Federal Financial Institutions Examination Council (FFIEC), the Federal Deposit Insurance Corporation (FDIC) Safeguards Act, and the New York Department of Financial Services Cybersecurity Regulation (NY DFS Cybersecurity Regulation). GLBA applies to all financial services organizations, including non-traditional financial institutions such as check-cashing businesses, payday lenders, and technology vendors providing loans to clients. Compliance with some regulations, such as GLBA and NY DFS Cybersecurity Regulation, is mandatory. The regulations mandate financial institutions to undergo periodic penetration testing and vulnerability assessments in their infrastructure. The Federal Trade Commission (FTC) is responsible for enforcing GLBA as it pertains to financial firms not covered by federal banking agencies, the Securities and Exchange Commission (SEC), the Commodity Futures Trading Commission (CFTC), and state insurance authorities.

Regulations in the Healthcare Sector

The HIPAA Security Rule, published in 2003, requires technical and nontechnical safeguards to protect electronic health information. Since then, several legislations have modified and expanded the scope and requirements of the Security Rule, including the HITECH Act, the Breach Notification Rule, and the Omnibus Rule. The Security Rule applies to covered entities and business associates, including healthcare providers, health plans, healthcare clearinghouses, and certain business associates. HHS has published additional cybersecurity guidance to help healthcare professionals defend against security vulnerabilities, ransomware, and modern cybersecurity threats. HHS has also provided guidance material for the HIPAA Security Rule at their website.

Payment Card Industry Data Security Standard (PCI DSS)

The Payment Card Industry Security Standards Council (PCI SSC) was formed by major payment card brands (Visa, MasterCard, Discover, and American Express) to develop the Payment Card Industry Data Security Standard (PCI DSS) in order to protect cardholders against misuse of their personal information and to minimize payment card channel losses. PCI DSS must be adopted by any organization that transmits, processes, or stores payment card data or that directly or indirectly affects the security of cardholder data. Any organization that leverages a third party to manage cardholder data has the full responsibility of ensuring that this third party is compliant with PCI DSS. The Luhn algorithm is used to validate credit card numbers and other identification numbers. The PCI SSC website provides guidance on the requirements for penetration testing.

Key Technical Elements in Regulations You Should Consider

Several key technical elements are mandated by most regulations, including data isolation, password management, and key management. Data isolation involves creating a separate network for systems involved in payment card processing to ensure they are completely isolated. Password management strategies must meet specific implementation standards and should include the use of strong passwords and multifactor authentication. Key management is also critical and involves the proper management and protection of cryptographic keys to ensure the security of information protected by cryptography. Policies and standards for key management should include assigned responsibilities, the nature of information to be protected, the classes of threats, and the cryptographic protection mechanisms to be used.

Legal Considerations

Important legal concepts that are relevant to performing a penetration test include Service-level Agreements (SLAs), confidentiality agreements, statements of work (SOWs), master service agreements (MSAs), and non-disclosure agreements (NDAs). The section also emphasizes the importance of contracts in a penetration testing engagement and the need for clarity, specificity, and legal advice. Finally, it suggests adding disclaimers to pre-engagement documentation and final reports to address the limitations of penetration testing and to avoid potential legal liabilities.

)

Explaining the Importance of Scoping and Organizational or Customer Requirements (

The rules of engagement document outlines the conditions under which the testing will be performed and includes details such as the testing timeline, location of testing, time window of testing, preferred method of communication, security controls that could potentially prevent testing, IP addresses or networks from which testing will originate, and types of allowed or disallowed tests. Gantt charts and work breakdown structures can be used to document the timeline of the testing.

Scoping is one of the most important elements of the pre-engagement tasks and includes documentation of the systems, networks, and applications to be tested, as well as any specific requirements needed for the test. There are different types of API documentation and additional support resources that might be available for the penetration tester. The engagement scope must include physical location, DNS fully qualified domain names, and external and internal target identification.

It is important to validate the scope of a penetration testing engagement and understand the target audience for the report. The topic provides a list of questions that can help discover different characteristics of the target audience, such as their need for the report, their position within the organization, and their responsibility and authority to make decisions based on the findings. It is also important to maintain open lines of communication with clients and stakeholders. The topic provides a list of questions to consider when communicating with stakeholders. Also, there may be questions about budget and return on investment that may arise from both the client and the tester sides in penetration testing. Make sure that the client understands that penetration testing is a point-in-time assessment and that clear and achievable mitigation strategies, impact analysis, and remediation timelines must be discussed with all stakeholders.

Unknown-environment testing involves giving the tester only limited information, such as domain names and IP addresses, to simulate the perspective of an external attacker. The tester does not have prior knowledge of the target's organization and infrastructure, and the network support personnel of the target may not be informed about the test. This allows for a realistic assessment of the security posture.

Known-environment testing, on the other hand, provides the tester with a significant amount of information about the organization and its infrastructure, including network diagrams, IP addresses, configurations, and user credentials. In some cases, the tester may also be provided with the source code of the target application. The goal of this type of testing is to identify as many security holes as possible.

The scope of the test and the amount of time and money spent on it depend on various factors, such as the company's specific concerns and the level of sophistication and capabilities of potential attackers. While known-environment testing can be useful for identifying specific vulnerabilities, unknown-environment testing is often a good choice because it provides a more realistic assessment of the network's security posture.

)

Demonstrating an Ethical Hacking Mindset by Maintaining Professionalism and Integrity (

This topic discusses several key considerations for ethical hackers or penetration testers to demonstrate professionalism and integrity. These include undergoing background checks, adhering to the specific scope of engagement, identifying criminal activity and reporting it immediately, limiting tool usage, respecting invasiveness based on scope, maintaining confidentiality of data and information, and understanding the risks involved. Additionally, the topic emphasizes the importance of risk management and risk tolerance in cybersecurity governance programs. All parties involved should make informed decisions and manage risk while keeping organizational objectives in mind.

)

3.5.1 What Did I Learn in this Module?

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 system, while passive reconnaissance does not interact directly with the target, using third-party databases and snooping on network traffic instead.

Testing

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 savedropping.

Scanning

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 lookup, 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 eystame

Vulnerability Scans

Security breaches can directly impact a company's reputation. Attackers use various methods to gather information, including password dump file metadata, strategic search engine analysis, website archiving, and public source code repositories. Tools like himeil and WhatBreach exploit breached data repositories, while ExdfTool 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. Shoden 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 can help determine if a host is online and responding on a network. Nmap also provides six timing templates (-10-5) 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-anum-groups.

Network's 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 big-anum 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: Soapy, 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.

)

4.6.1 What Did I Learn in this Module?

Protecting for an Approach and Impersonation

Social engineering involves influence, interrogation, and impersonation to gain information from victims without directly asking for it. Interrogators use open-ended and closed-ended questions to understand a victim's values, viewpoints, and goals. They also pay close attention to a victim's body language and speech patterns to gather more information.

Phishing, or impersonation, is when an attacker poses as someone else to gain access to information. Examples include impersonating a delivery person or an IT support worker. Pharming is a type of impersonation attack where a victim is redirected from a valid website to a malicious one to extract confidential information or install malware. This can be done by altering the host file on a victim's system, through DNS poisoning, or by exploiting a

vulnerability in a DNS server. Melvertising is another similar attack, which involves incorporating malicious ads on trusted websites, leading users to sites hosting malware.

)

Social Engineering Attacks (

Social engineering attacks exploit human vulnerabilities to gain access to sensitive information or systems. These attacks can take many forms, including email phishing, spear phishing, whaling, vishing, SMS phishing, USB drop attacks, and watering hole attacks. Email phishing involves sending a fraudulent email that appears legitimate, while spear phishing is a targeted phishing attack. Whaling targets high-profile business executives, while vishing is a phishing attack carried out over the phone. SMS phishing uses text messages to trick victims into divulging sensitive information. USB drop attacks involve leaving USB sticks in strategic locations to encourage victims to plug them in. Watering hole attacks involve injecting malicious code into a website to redirect users to an attacker-controlled site. Organizations can protect themselves from these attacks by developing policies, updating security applications, scanning websites for malware, and educating users

)

Physical Attacks (

Types of physical attacks in the context of penetration testing or red teaming include:

Tailgating and Piggybacking: Unauthorized persons gain access to restricted areas by tagging along with authorized personnel. Access control vestibules and multifactor authentication can help prevent these attacks.

Dumpster Diving: Scavenging through garbage and recycling containers for private information. Organizations should securely store and dispose of sensitive documents to protect against this threat.

Shoulder Surfing: Obtaining confidential data by looking over a victim's shoulder or using cameras or telescopes. User awareness, training, and screen filters can help prevent these attacks.

Badge Cloning: Attackers use specialized software and hardware, or social engineering techniques, to duplicate or impersonate access badges.

)

Social Engineering Tools(

Tools used in social engineering attacks include:

The Social-Engineer Toolkit (SET): Developed by David Kennedy, SET is a tool for launching social engineering attacks and integrating with third-party tools like Metasploit. It comes pre-installed in Kali Linux and Parrot Security but can also be installed on other Linux distributions and macOS. This topic outlines a step-by-step process for creating a spear phishing email using SET, including launching the toolkit, selecting attack vectors, creating a file format payload, generating a PDF with embedded EXE, setting up a reverse TCP shell, crafting the email, and setting up a listener for a reverse TCP connection.

Browser Exploitation Framework (BeEF): This tool manipulates users by leveraging XSS vulnerabilities. It starts a web service on port 3000. Attackers can use the web console to manipulate victims of XSS attacks. BeEF can be used to perform various attacks, including social engineering attacks like sending fake notifications to a victim's browser

Call Spoofing Tools: These tools allow attackers to change the caller ID information displayed on a phone. Examples include SpoofApp, SpoofCard (which can also change voice, record calls, generate background noises, and send calls to spam), and Asterisk (a VoIP management tool that can impersonate caller ID).

)

Methods of Influence (

Tools used in social engineering attacks include:

The Social-Engineer Toolkit (SET): Developed by David Kennedy, SET is a tool for launching social engineering attacks and integrating with third-party tools like Metasploit. It comes pre-installed in Kali Linux and Parrot Security but can also be installed on other Linux distributions and macOS. This topic outlines a step-by-step process for creating a spear phishing email using SET, including launching the toolkit, selecting attack vectors, creating a file format payload, generating a PDF with embedded EXE, setting up a reverse TCP shell, crafting the email, and setting up a listener for a reverse TCP connection.

Browser Exploitation Framework (BeEF): This tool manipulates users by leveraging XSS vulnerabilities. It starts a web service on port 3000. Attackers can use the web console to manipulate victims of XSS attacks. BeEF can be used to perform various attacks, including social engineering attacks like sending fake notifications to a victim's browser

Call Spoofing Tools: These tools allow attackers to change the caller ID information displayed on a phone. Examples include SpoofApp, SpoofCard (which can also change voice, record calls, generate background noises, and send calls to spam), and Asterisk (a VoIP management tool that can impersonate caller ID).

)

5.3.1 What Did I Learn in this Module?

Exploiting Network-Based Vulnerabilities (

This topic discusses a variety of network-based vulnerabilities and exploits.

Nat108 Name Service and LLMNR

Network Basic Input/Output System (NetBIOS) and Link-Local Multicast Name Resolution (LLM) are protocols primarily used by Microsoft Windows for host identification. LLMNR is based on the Domain Name System (DNS) protocol format. NetBIOS provides three services: Name Service (NetBIOS-NS), Datagram Service (NetBIOS-DOM), and Session Service (NetBIOS-SSN). These operations use specific TCP and UDP ports for communication. Windows workgroups are LAN peer-to-peer networks, while domain-based implementations are client-to-server networks supporting numerous hosts across multiple subnets.

Historically, there have been many vulnerabilities in NetBIOS, SMB, and LLMNR. A common LAN vulnerability involves an attacker spoofing an authoritative source for name resolution, poisoning the LLMNR service, and obtaining the victim's username and NTLMv2 hash. Tools like NBNSpoof, Metasploit, and Responder can be used to conduct these attacks. Pupy, an open-source Python-based cross-platform remote administration tool, is also popular among penetration testers and attackers.

Shi Exploits

The topic discusses SME's history of vulnerabilities and highlights the notorious EternalBlue exploit. Labeled by the Shadow Brokers, this exploit has been used in ransomware like WannaCry and Nyatta. Metasploit is one tool that has ported the EternalBlue exploit. Example 5-2 demonstrates how to use the exploit in Metasploit, requiring the configuration of RHOST and LHOST IP addresses. Once executed, Metasploit launches a Meterpreter session for further system control and compromise. Enumeration is an essential aspect of penetration testing, and tools like Nmap and Enum4linux can gather information on vulnerable SME systems, which can then be exploited using Metasploit.

DNS Cache Poisoning

DNS cache poisoning is an attack in which threat actors manipulate the DNS resolver cache by injecting corrupted data. This forces the DNS server to send the wrong IP address to the victim, redirecting them to the attacker's system. The process involves the following steps:

1. The attacker corrupts the DNS server cache to impersonate a website.
2. After the attack, the DNS server resolves the website to the attacker's IP address instead of the correct
3. The victim requests the IP address of the domain from the DNS server.
4. The DNS server replies with the attacker's IP address.
5. The victim sends an HTTP GET request to the attacker's system, and the attacker impersonates the domain.

DNS cache poisoning attacks may also use social engineering tactics to trick victims into downloading malware or entering sensitive data into spoofed forms and applications.

SNMP Exploits

SNMP is a protocol used to manage network devices, with each device containing an SNMP agent that connects to an SNMP server. Administrators can use SNMP to obtain information, change configurations, and perform other tasks. There are multiple versions, with SNMPv2c and SNMPv3 being the most popular. SNMPv2c uses community strings as passwords, while SNMPv3 is more secure with usernames and passwords. However, both versions are susceptible to attacks if weak or default credentials are used.

The Nmap scanner, along with its NSE scripts, can be used to gather information from SNMP-enabled devices and brute-force weak credentials. Additionally, the snmp-check tool can be utilized to perform an SNMP walk for gathering device information.

SMTP Exploits

Insecure SMTP servers can be exploited to send spam and conduct phishing and other email-based attacks. SMTP open relay is an email server configuration that can be abused for such purposes. Nmap provides an NSE script to test for open relay configurations. Useful SMTP commands, such as HELO, EHLO, and VRFY, can be used to evaluate an email server's security. The smtp-user-enum tool in Kali Linux automates information gathering. Disabling VRFY and EXPN commands on modern email servers and using firewalls to block SMTP connections with these commands can improve security. Known SMTP server exploits can be found using the searchsploit command.

FTP Exploits

FTP servers are often abused by attackers to steal information, as the legacy FTP protocol lacks encryption and integrity validation. To enhance security, it is recommended to use FTPS or SFTP. These protocols use encryption, but some implementations have weak encryption ciphers like Blowfish and DES. It is advised to use stronger algorithms such as AES. SFTP and FTPS servers also use hashing algorithms for verifying file transmission integrity. Best practices include disabling weak hashing protocols like MD5 or SHA-1 and using stronger algorithms in the SHA-2 family.

FTP servers might enable anonymous user authentication, which can be exploited by attackers. To mitigate this, disable anonymous login in the server configuration file. Additional best practices include using strong passwords and multifactor authentication, implementing file and folder security, encrypting files stored on the server, locking down administration accounts, keeping server software up-to-date, using FIPS 140-2 validated encryption ciphers, storing backup databases on separate servers, and requiring re-authentication for inactive sessions.

Pass-the-Hash Attacks

Pass-the-hash attacks exploit the storage of password hashes in Windows Security Accounts Manager (SAM) and use collected password hashes from compromised systems to log in to other systems without knowing the actual password. This bypasses the usual password-entry and conversion process.

Kerberos and LDAP-Based Attacks

Kerberos is an authentication protocol used by Windows and many applications and operating systems. Active Directory uses LDAP as an access protocol, which supports Kerberos authentication. Common attacks include Kerberos golden ticket and silver ticket attacks, where attackers manipulate Kerberos tickets based on available hashes. Unconstrained Kerberos delegation is another weakness, which allows applications to reuse end-user credentials to access resources hosted on different servers.

Kerberoasting

Kerberoasting is an attack that extracts service account credential hashes from Active Directory for offline cracking. It exploits weak encryption implementations and improper password practices.

On-Path Attacks

On-path attacks involve an attacker intercepting communication between two devices or individuals to steal or manipulate data. These can happen at Layer 2 or Layer 3. ARP spoofing, MAC spoofing, and manipulating Spanning Tree Protocol (STP) are examples of on-path attacks. To secure infrastructure, follow Layer 2 security best practices such as selecting an unused VLAN, configuring switch ports as access ports, limiting the number of MAC addresses learned on a port, controlling Spanning Tree, turning off Cisco Discovery Protocol (CDP) on untrusted ports, shutting down all ports on a new switch, using Root Guard, implementing 802.1X when possible, and deploying access control lists (ACL).

In downgrade attacks, attackers force a system to use a weaker encryption protocol or hashing algorithm that is susceptible to vulnerabilities. The Padding Oracle on Downgraded Legacy Encryption (POODLE) vulnerability in OpenSSL is an example of a downgrade attack. To prevent such attacks, removing backward compatibility is often the only solution.

Route Manipulation Attacks

One common route manipulation attack is BGP hijacking. In this attack, a threat actor configures or compromises an edge router to announce unauthorized prefixes. This can redirect the victim's traffic to the attacker if the malicious route is more specific or shorter than the legitimate one. Attackers sometimes use unused prefixes to avoid attention from the legitimate user or organization.

DOS and DDoS Attacks

DOS and DDoS attacks aim to overwhelm a target with an excessive amount of traffic or exploit vulnerabilities to crash the system. There are three categories of DoS attacks: direct, reflected, and amplification.

Direct DoS Attacks: The attacker sends packets directly to the victim, flooding their connection bandwidth or depleting their system resources. SYN flood attacks are an example of direct DoS attacks.

Reflected DoS and DDoS Attacks: Attackers send spoofed packets to sources that appear to be from the victim, making the sources unwitting participants in the attack. UDP is often used as the transport mechanism in these attacks due to the lack of a three-way handshake.

Amplification DDoS Attacks: A type of reflected DoS attack where the response traffic is much larger than the initial packets sent by the attacker. An example is sending DNS queries to an open resolver, which replies with larger

responses,

flooding the victim's machine.

Network Access Control (NAC) Bypass

NAC interrogates endpoints before joining a wired or wireless network, enforcing policies like checking for security software, operating system versions, and patching. Attackers can bypass NAC by spoofing authorized MAC addresses, enabling them to connect to the network.

VLAN Hopping

VLAN hopping is a method of gaining access to traffic on other VLANs that would normally be inaccessible. Two primary methods of VLAN hopping exist: switch spoofing and double tagging. Switch spoofing involves imitating a trunking switch by sending the respective VLAN tag and trunking protocols. Double tagging adds two VLAN tags to a frame, with most switches removing only the outer tag, enabling the attacker to access the victim's VLAN.

DHCP Starvation Attacks and Rogue DHCP Servers

DHCP starvation attacks involve broadcasting numerous fake DHCP REQUEST messages with spoofed MAC addresses depleting available IP addresses in the DHCP server scope. With no available IP addresses, the attacker can set up a rogue DHCP server and respond to new DHCP requests, intercepting traffic from network hosts.

)

Exploiting Wireless Vulnerabilities

This topic covers a variety of wireless vulnerabilities and exploits

Rogue Access Points, Evil Twin, and Disassociation attacks involve an attacker installing a rogue AP or impersonating a legitimate one to gain unauthorized access to the network. To defend against such attacks, use packet filtering, cryptographic protocols, and spoofing detection features

Preferred Network List (PNL) attacks involve attackers listening to client requests and impersonating the wireless network to intercept communication. Wireless signal jamming and interference involve attackers causing disruption of denial of service (DoS) on wireless networks. War driving and war flying involve attackers searching for wireless networks while driving or flying in order to exploit them.

Deauthentication Vector (V) attacks involve exploiting vulnerabilities in older protocols like WEP. Attacks against WEP are possible due to weak encryption methods, while attacks against WPA and WPA2 involve capturing the four-way handshake and brute forcing the PSK. WPA2 addresses many of these vulnerabilities, but is not completely immune to attacks such as de-authentication attacks, downgrade attacks, and DoS conditions. Wi-Fi Protected Setup (WPS) PIN attacks involve brute-forcing the PIN used to provision the wireless device. Tools like Reaver can be used to execute WPS attacks.

KARMA (Karma attacks radio machines automatically) is an on-path attack where a rogue access point (AP) intercepts wireless traffic from radio machines like mobile devices and laptops.

Fragmentation attacks target WEP-configured devices to acquire 1500 bytes of pseudo-random generation algorithm (PRGA) seeds, allowing attackers to generate packets for wireless injection attacks.

Credential harvesting involves obtaining or compromising user credentials through methods like phishing attacks, or by impersonating a wireless AP or a captive portal.

Manjacking sends unsolicited messages to victims via Bluetooth, while Bluesnarfing accesses unauthorized information from a Bluetooth-enabled device.

Bluetooth Low Energy (BLE) are typically targeted at Internet of Things (IoT) devices that use BLE for communication. These devices are susceptible to on-path attacks, with attackers modifying BLE messages or launching DoS attacks.

Radio-Frequency Identification (RFID) technology is used to identify and track tags holding electronically stored information. Common attacks include silently stealing RFID information, cloning RFID tags, implanting skimmers, and performing NFC amplification attacks.

Password spraying is a type of credential attack where attackers brute-force logins using a list of usernames with default

passwords.

Exploit chaining are sophisticated attacks that leverage multiple vulnerabilities, where an attacker chains exploits against known or zero-day vulnerabilities to compromise systems and steal, modify, or corrupt data.

)

6.13.1 What Did I Learn in this Module?

Overview of Web Application-Based Attacks for Security Professionals and the OWASP Top 10

How to Build Your Own Web Application Lab

Understanding Business Logic Flaws (

Business logic flaws are pique vulnerabilities in an application's normal operations that are difficult to detect with standard tools. Their exploitation can lead to serious consequences, but can be prevented with robust data validation and threat modeling. OWASP and MITRE provide guidance and detailed information on these flaws. MITRE has assigned Common Weakness Enumeration (CWE) ID 840 to business logic errors. Examples include unverified ownership, authentication bypasses, weak password recovery mechanisms, incorrect ownership assignment, and flaws that apply to improper resource management: and insufficient enforcement of unique actions and workflows.

)

Understanding Injection-Based Vulnerability(

This topic discussed the following Injection-based vulnerabilities:

SQL Injection Vulnerabilities

SQL Injection vulnerabilities pose a serious risk as they allow an attacker to view, insert, delete, or modify records in a database. The attack occurs when an attacker injects SQL commands into input fields of a web application or a URL to execute predefined

SQL commands

SQL statements are used in various ways, such as obtaining, updating, deleting, and inserting data into a database, as well as creating and modifying databases, tables, and indexes. SQL injections can be accomplished using user-supplied strings or numeric input.

There are three types of SQL Injection attacks:

In-band SQL Injection, where data is obtained via the same channel used to inject the SQL code.

Out-of-band SQL Injection, where the attacker retrieves data using a different channel.

Blind SQL Injection, where the attacker doesn't make the application display or transfer any data, but deduces the information by sending specific statements and observing the application and database's behavior.

There are several techniques to exploit SQL Injection vulnerabilities, such as using the Union operator to combine queries, Boolean to verify conditions, error-based technique to generate an error to refine the attack, out-of-band technique to obtain records from the database via a different channel, and time delay to use database commands to delay answers. A

Identifying SQL Injection vulnerabilities involves understanding when the application interacts with a database, making a list of input fields whose values could be used in a SQL query, and adding a single quote or a semicolon to the field or parameter in a web form.

Prevention of SQL injections involves using immutable queries like static queries, parameterized queries, and stored procedures that don't generate dynamic SQL. Immutable queries don't contain data that could be interpreted, which prevents injection attacks

Command Injection Vulnerabilities

Command injection is an attack where a attacker executes unauthorized commands on a system through a weak application. This often happens when user-supplied data isn't properly validated. Despite becoming less common due to improved defense in modern web applications, it remains a threat.

LDAP Injection Vulnerabilities

Lightweight Directory Access Protocol (LDAP) injection vulnerabilities occur when user input isn't properly validated before its use in LDAP statements, leading to potential unauthorized access or data manipulation. Like SQL Injection, attackers exploit these flaws to gain valuable data or further infiltrate systems. The two main types of LDAP Injection attacks are authentication bypass, which sidesteps credential checks, and information disclosure, which uses crafted LDAP packets to reveal organizational resources for reconnaissance.

)

Exploiting Authentication-Based Vulnerability (

This topic provides an in-depth look at various methods through which attackers can exploit authentication-based vulnerabilities in a system.

Credential Brute Forcing: This is an attack that involves trying multiple combinations of usernames and passwords until the correct one is found.

Session Hijacking: This involves intercepting a user's web session, which can be achieved by stealing the user's session ID, enabling the attacker to impersonate the user. Various methods exist for maintaining session state, such as cookies, URL parameters, and URL arguments on GET requests.

Redirecting: Redirecting is an attack where an attacker can exploit unvalidated redirects and forwards in a web application. This could result in users being sent to a malicious website or the attacker accessing restricted areas of the application.

Exploiting Default Credentials: This vulnerability arises when default usernames and passwords provided by device manufacturers aren't changed. Attackers can easily find these default credentials online and gain unauthorized access to systems.

Exploiting Weak Credentials: This involves attackers taking advantage of weak or easily guessable usernames and passwords to gain access to a system.

Exploiting Kerberos: Kerberos, a network authentication protocol, can also be exploited. In one notable example, an attacker can carry out a Kerberos golden ticket attack by manipulating Kerberos tickets based on available password hashes. Another vulnerability involves the misuse of Kerberos delegation, allowing an application to reuse user credentials to access resources hosted on another server.

These vulnerabilities highlight the need for robust security measures, including the use of HTTPS encryption for all web sessions, enforcing strong user credentials, validating redirects, and carefully managing session IDs
)

Exploiting Authorization-Based Vulnerabilities()

Understanding Cross-Site Scripting (XSS) Vulnerabilities()

Understanding Cross-Site Request Forgery (CSRF/XSRF) and Server-Side Request Forgery Attacks()

Understanding Clickjacking()

Exploiting Security Misconfigurations()

Exploiting File Inclusion Vulnerabilities()

Exploiting Insecure Code Practices()

7.3.1 What Did I Learn In this Module?

Researching Attacks and Performing Attacks on Cloud Technologies

Numerous organizations are transitioning to the cloud or employing hybrid models for their applications. This move usually entails a shift from capital expenditure (CapEx) to operating expenditure (OpEx). Cloud computing security, which includes protection against data theft, exfiltration, and deletion, is vital. The National Institute of Standards and Technology (NIST) established a standard set of definitions for cloud computing aspects in its SP 800-145 publication.

Benefits of using cloud-based services include distributed storage, scalability, resource pooling, access from any location, measured service, and automated management. Essential characteristics of cloud computing, as defined by NIST, are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

There are different models for cloud deployment: public cloud, which is available for public use; private cloud, used solely by a client organization; community cloud, shared among several organizations; and hybrid cloud, which includes a mix of two or more clouds and on-prem services.

Cloud computing services are categorized into three basic models: Infrastructure as a Service (IaaS), where users rent infrastructure; Platform as a Service (PaaS), which provides everything except the applications; and Software as a Service (SaaS), which provides a complete packaged solution, usually accessed via a web portal or front end.

Potential attacks against cloud technologies include the following:

Credential Harvesting This is the act of gathering valid user credentials, often using tactics like phishing and creating fake websites that mirror legitimate services. Attackers use these methods to trick users into revealing their login details, even sometimes bypassing multi-factor authentication. For example, the Social-Engineer Toolkit (SET) is one tool used by attackers to create a fake website for credential harvesting. As the use of cloud services expands, credential harvesting now targets both cloud and non-cloud services, emphasizing the need for strong security measures and user awareness.

Privilege Escalation This is exploiting system vulnerabilities to gain unauthorized access or privileges within a system. It comes in two forms: vertical, where a lower-privileged user gains higher-level access, and horizontal, where a user accesses content reserved for other users. Regular system updates, use of access control systems, and user vigilance, such as locking computers when unattended, are key to preventing such breaches.

Account Takeover - This is a security breach where an attacker gains unauthorized access to a user or application account, subsequently utilizing it to infiltrate more accounts and information. This can occur in a cloud environment and has distinct impacts compared to an on-premises attack, especially in terms of detection, damage assessment, and remediation strategies. Detection methods include monitoring user login locations and failed login attempts, identifying lateral phishing emails, detecting malicious connections via OAuth, SAML, or OpenID Connect, and observing abnormal file sharing and downloading behavior. Note that while location-based detection can provide clues about a possible breach, attackers may bypass these restrictions using VPNs.

Metadata Breach Attacks Traditionally, software developers used hard-coded credentials to access services, which can be insecure. To mitigate this, cloud providers like AWS introduced metadata services, which offer temporary access credentials to services like AWS S3 buckets. These services also store user data for setting up new virtual machines, such as AWS EC2 Instances. However, these metadata services are prime targets for attackers who can gain valid AWS credentials and sensitive information from user startup scripts. Tools like nimbostratus can help identify vulnerabilities that could lead to metadata service attacks.

Attacks Against Misconfigured Cloud Assets - These include the following:

IAM Implementations Solutions used for managing user and application authentication and authorization.

manipulated in IaaS or PaaS environments, it could be devastating for the organization using the cloud applications. **Federation Misconfigurations** Federated authentication allows for the association of a user's identity across various identity management systems. However, developers often misconfigure the protocols used (e.g., SAML, OAuth, OpenID), which attackers can exploit by replaying or modifying messages, thereby gaining unauthorized access.

Object Storage: Insecure permission configurations for cloud object storage services, like AWS S3 buckets, can lead

to data breaches

Containerization Technologies Attacks on container-based deployments (like Docker, Rocket, LXC, and containerd) have resulted in significant data breaches. Attackers can use stolen credentials or known vulnerabilities to compromise cloud-based applications. They can also create malicious containers and post them on Docker Hub, leading to supply chain attacks

Cloud Malware Injection Attacks - Cloud deployments can fall prey to malware injection attacks, where a rogue application is inserted into various cloud environments. Once operational, this malware enables the attacker to execute additional malicious activities like data manipulation and theft.

Side-Channel Attacks - Side-channel attacks exploit information from a system's implementation, such as timing, power consumption, electromagnetic leakage, and sound, to compromise the system and extract sensitive data, which is typically credentials, cryptographic keys, or other sensitive information.

This topic concluded with a brief discussion of software development kits (SDKs) and cloud development kits (CDKs). SDKs and CDKs are collections of tools aiding in application creation and cloud deployment respectively, with SDKs providing resources like compilers and debuggers, and CDKs, like AWS CDK, helping utilize cloud resources with familiar programming languages.

Explaining Common Attacks and Vulnerabilities Against Specialized Systems

This topic provides an in-depth look at various methods through which attackers can exploit authentication-based vulnerabilities in a system.

Credential Brute Forcing: This is an attack that involves trying multiple combinations of usernames and passwords until the correct one is found.

Session Hijacking: This involves intercepting a user's web session, which can be achieved by stealing the user's session ID, enabling the attacker to impersonate the user. Various methods exist for maintaining session state, such as cookies, URL parameters, and URL arguments on GET requests.

Redirecting: Redirecting is an attack where an attacker can exploit unvalidated redirects and forwards in a web application. This could result in users being sent to a malicious website or the attacker accessing restricted areas of the application.

Exploiting Default Credentials: This vulnerability arises when default usernames and passwords provided by device manufacturers aren't changed. Attackers can easily find these default credentials online and gain unauthorized access to systems.

Exploiting Weak Credentials: This involves attackers taking advantage of weak or easily guessable usernames and passwords to gain access to a system.

Exploiting Kerberos: Kerberos, a network authentication protocol, can also be exploited. In one notable example, an attacker can carry out a Kerberos golden ticket attack by manipulating Kerberos tickets based on available password hashes. Another vulnerability involves the misuse of Kerberos delegation, allowing an application to reuse user credentials to access resources hosted on another server.

These vulnerabilities highlight the need for robust security measures, including the use of HTTPS encryption for all web sessions, enforcing strong user credentials, validating redirects, and carefully managing session IDs

7.3.1 What Did I Learn In this Module?

Researching Attacks and Performing Attacks on Cloud Technologies

Numerous organizations are transitioning to the cloud or employing hybrid models for their applications. This move usually entails a shift from capital expenditure (CapEx) to operating expenditure (OpEx). Cloud computing security, which includes protection against data theft, exfiltration, and deletion, is vital. The National Institute of Standards and Technology (NIST) established a standard set of definitions for cloud computing aspects in its SP 800-145 publication.

Benefits of using cloud-based services include distributed storage, scalability, resource pooling, access from any location, measured service, and automated management. Essential characteristics of cloud computing, as defined by NIST, are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

There are different models for cloud deployment: public cloud, which is available for public use; private cloud, used solely by a client organization; community cloud, shared among several organizations; and hybrid cloud, which includes a mix of two or more clouds and on-prem services.

Cloud computing services are categorized into three basic models: Infrastructure as a Service (IaaS), where users rent infrastructure; Platform as a Service (PaaS), which provides everything except the applications; and Software as a Service (SaaS), which provides a complete packaged solution, usually accessed via a web portal or front end.

Potential attacks against cloud technologies include the following:

Credential Harvesting This is the act of gathering valid user credentials, often using tactics like phishing and creating fake websites that mirror legitimate services. Attackers use these methods to trick users into revealing their login details, even sometimes bypassing multi-factor authentication. For example, the Social-Engineer Toolkit (SET) is one tool used by attackers to create a fake website for credential harvesting. As the use of cloud services expands, credential harvesting now targets both cloud and non-cloud services, emphasizing the need for strong security measures and user awareness.

Privilege Escalation This is exploiting system vulnerabilities to gain unauthorized access or privileges within a system. It comes in two forms: vertical, where a lower-privileged user gains higher-level access, and horizontal, where a user accesses content reserved for other users. Regular system updates, use of access control systems, and user vigilance, such as locking computers when unattended, are key to preventing such breaches.

Account Takeover - This is a security breach where an attacker gains unauthorized access to a user or application account, subsequently utilizing it to infiltrate more accounts and information. This can occur in a cloud environment and has distinct impacts compared to an on-premises attack, especially in terms of detection, damage assessment, and remediation strategies. Detection methods include monitoring user login locations and failed login attempts, identifying lateral phishing emails, detecting malicious connections via OAuth, SAML, or OpenID Connect, and observing abnormal file sharing and downloading behavior. Note that while location-based detection can provide clues about a possible breach, attackers may bypass these restrictions using VPNs.

Metadata Breach Attacks Traditionally, software developers used hard-coded credentials to access services, which can be insecure. To mitigate this, cloud providers like AWS introduced managed services, which offer temporary access credentials to services like AWS S3 buckets. These services also store user data for setting up new virtual machines, such as AWS EC2 Instances. However, these metadata services are prime targets for attackers who can gain valid AWS credentials and sensitive information from user startup scripts. Tools like nimbostratus can help identify vulnerabilities that could lead to metadata service attacks.

Attacks Against Misconfigured Cloud Assets - These include the following:

IAM Implementations Solutions used for managing user and application authentication and authorization.

manipulated in IaaS or PaaS environments, it could be devastating for the organization using the cloud applications. **Federation Misconfigurations** Federated authentication allows for the association of a user's identity across various identity management systems. However, developers often misconfigure the protocols used (e.g., SAML, OAuth, OpenID), which attackers can exploit by replaying or modifying messages, thereby gaining unauthorized access.

Object Storage: Insecure permission configurations for cloud object storage services, like AWS S3 buckets, can lead

to data breaches

Containerization Technologies Attacks on container-based deployments (like Docker, Rocket, LXC, and containerd) have resulted in significant data breaches. Attackers can use stolen credentials or known vulnerabilities to compromise cloud-based applications. They can also create malicious containers and post them on Docker Hub, leading to supply chain attacks

Cloud Malware Injection Attacks - Cloud deployments can fall prey to malware injection attacks, where a rogue application is inserted into various cloud environments. Once operational, this malware enables the attacker to execute additional malicious activities like data manipulation and theft.

Side-Channel Attacks -Side-channel attacks exploit information from a system's implementation, such as timing, power consumption, electromagnetic leakage, and sound, to compromise the system and extract sensitive data, which is typically credentials, cryptographic keys, or other sensitive information.

This topic concluded with a brief discussion of software development kits (SDKs) and cloud development kits (CDKs), SDKs and CDKs are collections of tools aiding in application creation and cloud deployment respectively, with SDKs providing resources like compilers and debuggers, and CDKs, AWS CDK, helping utilize cloud resources with familiar programming languages.

Explaining Common Attacks and Vulnerabilities Against Specialized Systems

This topic covered the following attacks and vulnerabilities against specialized systems:

Attacking Mobile Devices

Methods attackers use to compromise mobile devices include reverse engineering, sandbox analysis, spamming, and exploiting

the most prevalent known vulnerabilities.

In reverse engineering, attackers analyze the compiled mobile app to extract information about its source code and manipulate

the mobile device, Sandbox analysis involves an attacker examining the sandbox implementation in a mobile device to potentially bypass the access control mechanisms implemented by the operating system. Spamming involves unsolicited messages, often carrying links that redirect users to malicious sites to steal sensitive information or install malware.

Several prevalent vulnerabilities affecting mobile devices include the following:

Insecure storage refers to developers not successfully using secure storage APIs, allowing an attacker to exploit these

vulnerabilities

Passcode vulnerabilities and biometric integrations could lead to sensitive data exposure if not implemented securely

Certificate pinning is a method where attackers force a mobile app to store a server certificate or public key, which could be exploited.

Usage of known vulnerable components and over-reach of permissions by applications could also lead to attacks.

Execution of activities using root should not be allowed on mobile applications

Business logic vulnerabilities are also a concern where an attacker can manipulate legitimate transactions and flows of an application to cause harm.

Several tools like Burp Suite, Drozer, Needle, Mobile Security Framework (MobSF), Postman, Ettercap, Frida, Objection, Android SOK tools, AptX, and APK Studio can be used to test the security posture of mobile devices and perform security research.

Attacking Internet of Things (IoT) Devices

The Internet of Things (IoT) encompasses a wide range of devices and systems across various industries.

Designing and securing IoT systems is complex due to integration challenges, scalability issues, and the need for diverse skills, IoT platforms must integrate different devices, work with legacy technologies, and handle multiple vendors. Existing security efforts often focus on specific components rather than the entire system.

Analyzing IoT protocols is crucial for reconnaissance and exploitation. Common network protocols in IoT include Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRaWAN, Insteon, Modbus, and Siemens 57comm. For example, Bluetooth Low

Energy (BLE) is used in various IoT devices but may have vulnerabilities, such as misconfigurations and lack of encryption, which can be exploited by attackers.

Securing IoT implementations requires special considerations. IoT devices often have limited computing resources, making encryption support challenging. Denial-of-Service (DoS) attacks, data corruption, and data exfiltration are major concerns in IoT security.

Common vulnerabilities in IoT implementations include insecure defaults, plaintext communication, hard-coded configurations, and the use of outdated firmware and insecure components. Many IoT devices have default credentials and insecure

configurations exposed on the internet. Encryption is often not implemented properly, leading to data leakage. Outdated software and hardware pose risks, and IoT devices often lack secure update mechanisms.

Data Storage System Vulnerabilities

The complexity and security vulnerabilities of Internet of Things (IoT) architectures range from endpoint devices, through intermediary devices (such as routers and switches, known as the "fog" layer), to cloud computing platforms. Misconfigurations in IoT systems can expose these devices to cyberattacks and data theft. Common issues include the use of default or hardcoded credentials, inappropriate exposure of systems to the internet, lack of user input sanitization, software and injection vulnerabilities, and revealing too much information in error messages and debug outputs. These weaknesses can lead to a range of attacks, from denial of service to unauthorized code execution.

Management Interface Vulnerabilities

IoT implementations are often vulnerable due to issues with management interfaces like the Intelligent Platform Management Interface (IPMI), a suite of specifications utilized by IoT systems for management and monitoring independently of the host system's core functions. This is facilitated by the IPM's baseboard management controller (BMC) and other satellite controllers, which connect through the Intelligent Platform Management Bus/Bridge (IPMB). However, due to its direct hardware access, a compromised BMC can lead to significant security breaches, including system monitoring, rebooting, or unwanted software installation, equating to physical access to the underlying system.

Exploiting Virtual Machines

VMs are meant to be isolated systems, each running unique applications and operating systems, and are managed by a hypervisor. There are two types of hypervisors: Type 1, which runs directly on the system, with examples being VMware ESXi and Microsoft Hyper-V; and Type 2, which runs atop another OS, like VirtualBox and VMware Player. Despite their design, VMs have been vulnerable to various threats, including:

VM escape vulnerabilities This allows attackers to break out from one VM and access data from another VM or even the hypervisor itself. Hypervisor vulnerabilities, including hyperjacking. Hyperjacking can let an attacker seize control of the hypervisor, often

through the installation of a stealthy, malicious hypervisor.

VM repository vulnerabilities. Threat actors can exploit these to compromise systems and applications by uploading malicious VMs to public and private VM repositories, like VMware Marketplace or AWS Marketplace. Once deployed by unsuspecting users, these VMs can lead to systems, applications, and data manipulations.

Vulnerabilities Related to Containerized Workloads

The evolution of computing has led to serverless architectures from traditional bare-metal servers, through VMS and containers. These advancements also present potential vulnerabilities in application and open-source software, which are often overlooked in containers like Docker, Rocket, and containerd. Security measures need to be applied at various layers, including the container image, software inside the container, the host operating system, the interaction between containers and the host, and in the runtime environment and orchestration platform, such as Kubernetes.

Running containers with root privileges could lead to full compromise, hence securing container images is crucial. The CIS Benchmarks provide guidance on securing Docker containers and Kubernetes deployments. There are also various tools: Anchore's Grype, Clair, Dagda, kube-bench, kube-hunter, and Falco that can help detect vulnerabilities in Docker images, assess Kubernetes deployments, and maintain the overall security posture.

Threat actors have been known to insert malicious code into Docker images on Docker Hub, illustrating the potential for "supply chain" attacks, indicating the importance of robust security measures in the containerized environment.

Understanding the Different Use Cases of Penetration Testing Tools and Analyzing Exploited Code

This section covers the tools that are most commonly used in penetration testing engagements.

Penetration Testing-Focused Linux Distributions

Kali Linux: One of the most well-known penetration testing Linux distributions, Kali Linux is based on Debian GNU/Linux and has evolved from earlier distributions like WHopplX, WHAX, and BackTrack. It comes preloaded with hundreds of tools for penetration testing, and the community continuously contributes new ones. Kali Linux is accessible through a Live

image on a CD/DVD/USB/PXE, providing a bare-metal installation. **Parrot OS:** This Debian-based Linux distribution focuses on penetration testing, digital forensics, and privacy protection.

BlackArch Linux: BlackArch Linux is packed with more than 1900 security penetration testing tools. This distribution can be downloaded from its official website, where its documentation is also available.

Tools for Passive Reconnaissance

Eslookup, Host, and Dig: These are DNS-based tools used for passive reconnaissance. They allow you to gather Information about a domain, such as IP addresses associated with it.

Whois: The Whois utility is used to query the Whois database, which contains information about domain registrations. It provides details such as the domain owner, registration date, and contact information. However, due to GDPR restrictions, the amount of information available through Whois has been limited.

FOCA (Fingerprinting Organization with Collected Archives): FOCA is a tool designed to find metadata and hidden information in various types of documents, including websites, Microsoft Office files, PDFs, and more. It can be used to extract information such as EXIF data from graphics files and analyze URLs.

ExifTool: ExifTool is a popular tool for extracting EXIF (Exchangeable Image File Format) information from images. It can reveal details about the device used to capture the image, such as the camera model, date and time, GPS coordinates, and more.

thelarvester, the Harvester is a versatile tool used for DNS enumeration. It can query multiple data sources, including search engines like Google and Bing, social media platforms like Twitter and LinkedIn, PGP servers, and more. It helps gather information about a target domain, such as subdomains and associated email addresses.

Shodan: Shodan is a powerful search engine that scans and indexes devices connected to the Internet. It allows users to search for specific devices, services, or vulnerabilities. Shodan can help identify exposed and potentially vulnerable systems, such as misconfigured IoT devices or infrastructure devices.

Maltego: Maltego is a widely used tool for passive reconnaissance that gathers information from public records. It supports various integrations with third-party sources and offers different versions, including a free community edition. Maltego can be used to find information about individuals, companies, organizations, and more, presenting the results in a hierarchical and organized manner.

Recon-ng: Recon-ng is a menu-based tool specifically designed for automating OSINT information gathering. It comes with a wide range of modules that allow users to perform detailed searches on public records, files, DNS records, and other sources. Recon-ng supports querying third-party tools and sources like Shodan, social media platforms, and search engines.

Censys: Censys provides information about devices and networks on the internet. Censys offers both a web interface and an API and provides free access with limitations on the number of queries.

Tools for Active Reconnaissance

Nmap and Zenmap: Nmap is a comprehensive tool for active reconnaissance, providing various scanning options to enumerate hosts and discover open ports. Zenmap is a graphical user interface (GUI) tool that enhances the usability of Nmap and offers features like network topology visualization.

Enumlinux: Enumlinux is specifically designed for enumerating SMB (Server Message Block) shares and vulnerable Samba implementations. It helps identify users and enumerate available SMB shares on a target host.

Common Tools for Vulnerability Scanning

These tools provide various capabilities for vulnerability scanning, web application security testing, and detecting common security flaws. They are widely used by security professionals and penetration testers to assess the security of systems and applications

OpenVAS: OpenVAS is an open-source vulnerability scanner that allows detailed vulnerability scanning of hosts and networks. It offers various services and tools and can be used for scanning and identifying vulnerabilities. It can be scheduled and configured to perform scans using different methods and interfaces.

Messus: Nessus is a vulnerability scanner that enables continuous monitoring and compliance analysis. It provides features for scanning and detecting vulnerabilities, and it supports integrations with other security products.

■■■■■■■■: ■■■■pose, created by Rapid7, is a popular vulnerability scanner used by professional penetration testers. It offers features for vulnerability scanning and can integrate with other security tools.

Qualys: Qualys is a security company that provides a cloud-based vulnerability management and monitoring service. It offers continuous monitoring, vulnerability management, and compliance checking. Qualys interacts with different types of scanners and agents to provide comprehensive security assessments

SQLmap: SQLmap is a tool used for automating the detection and exploitation of SQL injection vulnerabilities in web applications. It helps enumerate vulnerable applications and can exploit SQL Injection techniques

Hikto: Nikdo is an open-source web vulnerability scanner that allows scanning for common web vulnerabilities. It can be used to detect security flaws in web applications and servers.

OWASP ZAP: OWASP Zed Attack Proxy (ZAP) is a widely used free security tool that provides web vulnerability scanning capabilities. It can also be used as a web proxy and a fuzzer. ZAP offers an API for automation and is actively maintained by a large community of contributors

w3af: w3af is an open-source web application vulnerability scanner. It allows scanning for vulnerabilities in web applications and offers various plugins for different types of vulnerability testing.

DirBuster: DirBuster is a tool designed to perform brute-force directory and filename discovery on web application servers. It is an inactive project, and its functionality has been integrated and enhanced in OWASP ZAP as an add-on.

Common Tools for Credential Attacks

These tools provide different capabilities for password cracking, credential guessing, and generating wordlists, catering to

various security testing and offensive purposes.

John the Ripper: John the Ripper is a popular tool for offline password cracking. It supports various cracking modes and can crack passwords using search patterns or wordlists. It can handle different ciphertext formats, including DES variants, MD5, and Blowfish. John the Ripper can be used to extract passwords from various sources, such as password files and Kerberos AFS.

Cain: Cain (or Cain and Abel) is a tool used for password recovery on Windows-based systems. It can perform packet captures, crack encrypted passwords using dictionary and brute-force attacks, and employ other techniques to recover user credentials.

Hashcat: Hashcat is a password-cracking tool that is particularly popular among penetration testers. It utilizes graphical processing units (GPUs) to accelerate the cracking process. It supports various algorithms and provides flexibility in using wordlists and different attack modes.

Hydra: Hydra is a tool for guessing and cracking credentials by attempting username/password combinations against target servers such as web servers, FTP servers, SSH servers, and file servers. It supports both dictionary and brute-force attacks and can be used to automate credential cracking.

RainbowCrack: RainbowCrack is a tool that automates password cracking using precomputed tables known as rainbow tables. Rainbow tables accelerate the cracking process by providing a way to reverse cryptographic hash functions and derive passwords from hashed values.

Medusa and Ncrack: Medusa and Ncrack are similar tools to Hydra, used for performing brute-force credential attacks against systems. Medusa can be installed on Debian-based Linux systems, while Ncrack can be downloaded from the official Nmap website. Both tools support various protocols and can perform dictionary and brute-force attacks.

CeWL: COWL is a tool used to create wordlists by crawling websites. It retrieves words from the target website, allowing users to generate custom wordlists for password cracking or other purposes.

Mimikatz: Mimikatz is a versatile tool used by penetration testers, attackers, and even malware for retrieving password hashes from memory. It is commonly used as a post-exploitation tool and can be downloaded from GitHub. Mimikatz is also integrated into Metasploit as a Meterpreter script.

Patator: Patator is a tool designed for brute-force attacks on various types of credentials, such as SNMPv3 usernames and VPN passwords. It offers multiple modules and can be used to automate credential attacks

Common Tools for Persistence

PowerSploit: A collection of PowerShell modules that can be used for post-exploitation and other phases of an assessment.

Empire: A PowerShell-based open-source post-exploitation framework that includes a PowerShell Windows agent and a

Python Linux agent.

Common Tools for Evasion

Vall: Vell is a framework that works in conjunction with Metasploit to bypass antivirus checks and other security controls. It offers evasion techniques and can generate payloads that are less likely to be detected by antivirus software. Vell is available for download from GitHub and provides detailed documentation on its website.

Tor: Tor is a free tool that enables users to browse the web anonymously by routing their IP traffic through a network of Tor relays. It utilizes "onion routing" to encrypt and route data through multiple relays, making it difficult to trace the user's location. Tor is commonly used for privacy purposes and can help evade security monitoring and controls.

Proxychains: Proxychains is a tool that forces specified applications to use Tor or other proxy types for TCP connections. It can be used to redirect network traffic through proxies and enhance evasion techniques.

Proxychains is available for download from GitHub.

Encryption: Encryption plays a vital role in security and privacy, but it can also pose challenges in Incident response and forensics. While encryption protects sensitive information, it can be used by threat actors to evade detection and obfuscate their activities. Security products can Intercept and Inspect encrypted traffic, and other logs and metadata can be leveraged for Investigation purposes.

Encapsulation and Tunneling Using DNS: Threat actors have exploited nontraditional techniques like DNS tunneling to exfiltrate data from corporate networks. DNS tunneling involves using DNS protocols to send unauthorized data, such as stolen credit card information, intellectual property, or confidential documents. Several tools have been developed to perform DNS tunneling, enabling cybercriminals to bypass security monitoring and controls.

Exploitation Frameworks

Metasploit: Metasploit is a widely-used exploitation framework created by H.D. Moore and now owned by Rapid7. It offers a community (free) edition and a professional edition. Metasploit has a robust architecture, written in Ruby, and comes pre-installed in Kali Linux. It provides various modules for exploits, auxiliary tasks, encoders, payloads, and more. The Metasploit console (msfconsole) is used to interact with the framework, and it supports a PostgreSQL database for indexing and accelerating tasks.

BeEF: BeEF is an exploitation framework specifically designed for web application testing. It exploits browser vulnerabilities and interacts with web browsers to launch directed command modules. BeEF allows for targeting multiple browsers in different security contexts, enabling security professionals to deploy specific attack vectors and modules in real-time. It has an extensive library of command modules and supports the development of custom modules.

Common Decompilation, Disassembly, and Debugging Tools

These debugger tools provide capabilities for debugging, analyzing, and reverse engineering software and binaries.

GNU Project Debugger (GDB): A popular debugger used for troubleshooting and finding bugs in software. Supports multiple programming languages.

Windows Debugger (WinDbg): Used for analyzing kernel and user-mode code in Windows, crash dump analysis, and

CPU register analysis.

OllyDbg: Debugger for analyzing Windows 32-bit applications, commonly used in penetration testing and reverse engineering.

edib Debugger: Cross-platform debugger supporting ARM32, x86, and x86-64 architectures, included in Kali Linux.

Ghidra: A free and open-source reverse engineering tool developed by the NSA, providing powerful decompilation and analysis capabilities for multiple architectures.

IDA: Commercial disassembler, debugger, and decompiler widely used for analyzing binary files and reverse engineering. Objdump: Linux program for displaying information about object files, commonly used for quick checks and disassembly of binaries.

Common Tools for Forensics

These tools aid forensic investigators in analyzing digital evidence, recovering data, and extracting valuable information for

Investigations.

Autopsy: Open-source digital forensics platform with a graphical interface for analyzing digital evidence. The Sleuth Kit: Collection of command-line tools for disk image and Me system analysis.

Volatility: Memory forensics framework for analyzing volatile memory in a system.

EnCase: Commercial digital forensics tool with features like disk imaging, Me recovery, and email analysis. FTK

(Forensic Toolkit): Commercial digital forensice tool for disk imaging, file analysis, and data carving. Wireshark: Network protocol analyzer for network forensics and capturing network traffic.

Cellebribs UFED: Mobile forensic tool for extracting and analyzing data from mobile devices.

X-Ways Forensics: Comprehensive forensic tool with features for disk imaging, file analysis, and registry analysis.

Common Tools for Software Assurance

These tools aid in ensuring software quality and security by detecting bugs, vulnerabilities, and potential issues.

SpotiBugs: Formerly known as FindBug, SpotBugs is a static analysis tool for Java applications that helps identify bugs and potential issues in Java code.

Findsebugs: Findsscbugs is a Java-specific tool that focuses on finding security-related bugs in Java applications. I Integrates well with continuous Integration systems like Jenkins and SonarQube.

SonarQube: SonarQube is a comprehensive tool for identifying vulnerabilities and quality issues in code. It supports continuous integration and DevOps environments.

Fuzzers and Fuzz Testing: Fuzz testing is a technique used to identify software errors and security vulnerabilities by injecting random or malformed data. Fuzzers are the tools used for fuzz testing. Here are some examples:

Peach: Peach is a popular fuzzer that offers both a free (open-source) version called Peach Fuzzer Community Edition and a commercial version.

Mutiny Fuzzing Framework: Developed by Cisco, the Mutiny Fuzzing Framework is an open-source fuzzer that replays packet capture files (pcaps) through a mutational fuzzer.

American Fuzzy Lop (AFL): AFL in a widely used fuzzer that incorporates compile-time Instrumentation and genetic algorithms to enhance fuzzing test cases' functional coverage.

Wiraless Tools

Winte2: A Python program to test wireless networks.

Rogue access points: You can easily create rogue access points by using open-source tools such as hostapd.

EAPHammer: This tool can be used to perform evil twin attacks.

mdk4: This tool is used to perform fuzzing, IDS evasions, and other wireless attacks.

Spooftooph: This tool is used to spoof and clone Bluetooth devices.

Reaver: This tool is used to perform brute-force attacks against WI-FI Protected Setup (WPS) Implementations.

Wireless Geographic Logging Engine (WIGLE): This is a war driving tool.

Fern WI-FI Cracker: This tool is used to perform different attacks against wireless networks, including cracking WEP, WPA,

and WPS keys.

Steganography Tools

OpenStego: You can download this steganography tool from <https://www.openstego.com>.

snow: Text-based steganography tool.

Sonic Visualiser: This tool can be used to analyze embedded information in music or audio recordinga.

Coagule: This program can be used to make sound from an image.

TinEye: A reverse image search website.

metagoofit: This tool can be used to extract metadata information from documents and images.

Cloud Tools :

ScoutSuite: Tools can be used to reveal vulnerabilities in AWS, Azure, Google Cloud Platform, and other cloud platforms.

CloudBrute: A cloud enumeration tool.

Pacu: A tamework for AWS exploitation.

Cloud Custodian: A cloud security, governance, and management tool.