Project Name: Theoretical Knowledge (VAPT)

Client/System: VirtualBox (Kali Linux, Metasploit)

Start Date: 19/08/2025

End Date: 22/08/2025

Author Name: Hilary H Joachim

1. Vulnerability Scanning Techniques.

While working on this topic, I learned that vulnerability scanning is one of the most important steps in assessing the security posture of a system. It helps identify weaknesses that attackers could exploit, and I explored different approaches to achieve this.

Core Concepts I Learned

- **Scan Types**:
  I discovered that **network scans** (using tools like Nmap) allow me to detect open ports and services such as SSH, HTTP, or SQL, which gives a baseline understanding of what is exposed. On the other hand, **application scans** (e.g using Nikto) are focused on web applications and can reveal issues like outdated libraries, misconfigurations, or injection flaws. I also learned the difference between **authenticated** and **unauthenticated scans**: authenticated scans provide deeper insights by logging into the system with valid credentials, whereas unauthenticated scans simulate an external attacker's perspective with no access.

- **Vulnerability Scoring**:
  I studied the **CVSS v4.0** system, which assigns a severity score from 0–10. For instance, a Remote Code Execution (RCE) vulnerability with a score of **8.8** is considered *High*. A real example is the **Apache Struts (CVE-2017-5638)** vulnerability, which was marked as *Critical* and famously exploited in the Equifax breach.

- **False Positives**:
  I understood that scanners sometimes report risks that are not actually exploitable. For example, an open port might show as vulnerable, but manual verification is

always needed to confirm the result. This highlighted to me the importance of not relying solely on automated tools.

Key Takeaways
From this study, I realized the importance of:

- Configuring scans properly and validating results for accurate risk assessment.
- Using **CVSS scoring** to prioritize vulnerabilities that pose the highest risk.
- Reducing **false positives** by cross-checking findings with multiple tools and manual checks.

Learning Resources
To deepen my understanding, I reviewed the **OWASP Testing Guide** for web scanning methodologies, and **NIST SP 800-115**, which provides standardized scanning practices. I also analyzed the **WannaCry ransomware case**, where attackers exploited SMB vulnerabilities rated as *Critical* in CVSS, showing how dangerous unpatched systems can be.

2. Penetration Testing Techniques

While studying penetration testing, I learned that it goes beyond just identifying vulnerabilities it focuses on actively exploiting them to understand the real world risks to a system. This makes it a more comprehensive way of assessing security than scanning alone, since it demonstrates how an attacker could move through different stages of an attack.

Core Concepts I Learned

Phases of Penetration Testing:
I explored the structured phases of a penetration test. Reconnaissance (for example, using OSINT tools like Shodan) allows me to gather information about the target. Scanning (e.g. using Nmap or Nessus) helps in identifying open ports, services, and potential vulnerabilities. Exploitation is the stage where I attempt to leverage those weaknesses, often with tools like Metasploit, to gain access. Post-exploitation, on the other hand, focuses on maintaining access, escalating privileges, and collecting evidence. Finally, the reporting phase is essential to document findings and suggest fixes.

Testing Methodologies:
I also studied recognized frameworks such as the Penetration Testing Execution Standard (PTES) and OWASP Web Security Testing Guide (WSTG). PTES provides a clear structure for scoping and executing pentests, while OWASP WSTG is more web-application focused.

Learning these methodologies helped me understand how to follow a professional and ethical process.

Ethics and Scope:
One important takeaway was the need to always conduct penetration testing within a defined scope and with explicit client authorization. This ensures that the test remains legal, controlled, and aligned with the client's needs.

Key Takeaways
From this study, I realized that penetration testing is not just about "hacking" but about following a professional, ethical, and structured process. My main lessons include:

- Always defining the scope and objectives before testing.
- Following clear methodologies such as PTES or OWASP WSTG.
- Demonstrating real-world risks by moving beyond scans into controlled exploitation.
- Documenting findings clearly to support remediation.

Learning Resources
To deepen my understanding, I studied the PTES documentation, which describes each penetration testing phase in detail. I also reviewed the OWASP WSTG to learn methodologies specific to web applications. Additionally, I looked at SANS penetration testing case studies, which provided real-world insights into how professional pentesters conduct assessments.

3. Exploit Development Basics

While studying exploit development, I learned that it focuses on turning software vulnerabilities into practical attacks. Unlike scanning or pentesting alone, exploit development requires a deeper understanding of how systems work internally, including memory, processes, and mitigations. This knowledge helps me see how attackers craft exploits and how defenders can design protections.

Core Concepts I Learned

Vulnerability Types:
I explored common categories of vulnerabilities that can lead to exploitation. For example, buffer overflows occur when programs fail to check input sizes, allowing attackers to overwrite memory and hijack execution flow. I also learned about SQL injection and Cross-Site Scripting (XSS), which are more application-layer exploits targeting input validation flaws.