



## VAPT TASK - 2

### A) Theoretical Knowledge

#### 1. Vulnerability Scanning Techniques

##### The Types of Vulnerability Scans

1. **Internal Scanning:** Internal scanning looks at the internal network and systems from within the organization's network perimeter. It aims to identify vulnerabilities present in devices, servers, and applications accessible from within the network. This type of scan is crucial for identifying potential threats and weaknesses that may exist inside the organization's boundaries.
2. **External Scanning:** External scanning is the scanning the organization's external-facing systems and assets from outside the organizational perimeter. This scan helps identify vulnerabilities that attackers could exploit from the internet or other external networks. It's essential for understanding the security posture visible to potential attackers.
3. **Authenticated Scanning:** Authenticated scanning involves conducting scans using valid credentials to access systems and applications. Through authentication, the scanner can access deeper levels of information, including configuration settings and installed software versions. This type of scan provides more accurate results as it can identify vulnerabilities specific to the authenticated user's access level.
4. **Unauthenticated Scanning:** Instead of using credentials, unauthenticated scanning relies on alternative techniques such as port scanning or vulnerability fingerprinting. While it provides a broader overview of potential vulnerabilities, unauthenticated scanning may not find weaknesses that require authenticated access to accurately detect.
5. **Assessment Scanning:** Assessment scanning evaluates the security posture of an organization's systems and networks comprehensively. It encompasses various scanning techniques, including vulnerability scanning, configuration auditing, and compliance checks. The goal is to assess the overall security of the organization's IT infrastructure and identify areas for improvement.
6. **Discovery Scanning:** Discovery scanning focuses on identifying and mapping all devices, servers, and assets connected to the network. It helps organizations understand their network topology, including the presence of any unauthorized or unmanaged devices. Discovery scanning lays the foundation for subsequent vulnerability assessments and security management efforts.
7. **Compliance Scanning:** Compliance scanning involves evaluating systems and networks against predefined security standards or regulatory requirements. It ensures that the organization's IT infrastructure aligns with industry best practices and regulatory mandates and mitigates the risk associated with non-compliance.

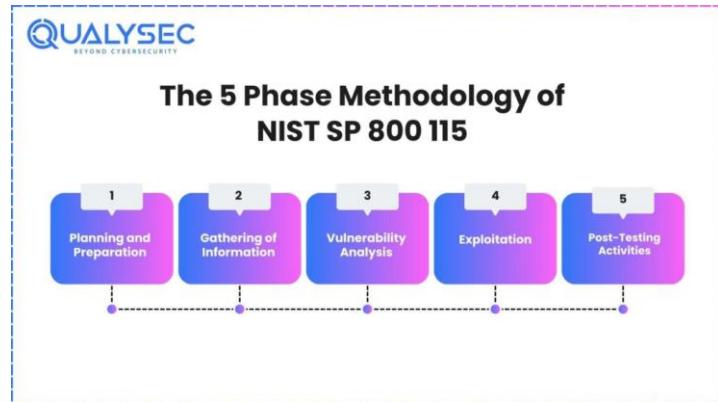


- 
8. **Host-Based Scanning:** Host-based scanning scans individual devices, such as servers, workstations, or endpoints, for vulnerabilities and security misconfigurations. It focuses on the specific characteristics of each host, including installed software, running services, and system configurations. Host-based scanning is crucial for identifying vulnerabilities that may be unique to a particular device or operating system.
  9. **Network Scanning:** Network scanning reviews the entire network infrastructure to identify active hosts, open ports, and potential security vulnerabilities. It helps organizations understand the layout of their network and identify potential entry points for attackers. Network scanning is an essential component of both network security management and risk assessment.
  10. **Web Application Scanning:** Web application scanning assesses the security of web applications and services hosted on web servers. It identifies vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure authentication mechanisms. Web application scanning helps organizations secure their web applications and protect against common attack vectors targeting web servers.
  11. **Port Scanning:** Port scanning involves scanning a network to discover open ports on devices and services. It helps identify active services running on networked devices and assess potential entry points for attackers. Port scanning is a fundamental technique used in vulnerability assessment and network security monitoring.
  12. **Database Scanning:** Database scanning investigates the security of databases and database servers for vulnerabilities and misconfigurations. It helps identify weaknesses such as weak authentication mechanisms, outdated software versions, and insecure database configurations. Database scanning is critical for protecting sensitive data stored in databases from unauthorized access, modification, compromise, and exploitation.
  13. **Source Code Vulnerability Scanning:** Source code vulnerability scanning analyzes the source code of applications and software for security vulnerabilities and coding errors. It helps identify issues such as buffer overflows, injection flaws, and insecure coding practices. Source code vulnerability scanning is an essential part of secure software development practices and helps developers identify and remediate security vulnerabilities early in the development lifecycle.
  14. **Cloud Vulnerability Scanning:** Cloud vulnerability scanning involves assessing the security of cloud-based infrastructure, platforms, and services for vulnerabilities and misconfigurations. It helps identify issues such as exposed data storage, insecure cloud configurations, and vulnerable cloud-based applications. Cloud vulnerability scanning is crucial for ensuring the security of cloud deployments and mitigating risks associated with cloud-based services. <sup>[1]</sup>



## NIST SP 800-115 for scanning methods

NIST SP 800-115 provides guidelines for security testing, including vulnerability scanning, by defining a phased approach: Planning, Discovery, Attack/Testing, and Reporting, emphasizing systematic identification and exploitation of weaknesses to assess security controls. It details techniques like network scanning (port, vulnerability, wireless), log analysis, and configuration review, acting as a framework to develop tailored assessments, not a strict toolset.



## Analysis of WannaCry case

**WannaCry Ransomware** is a high-profile ransomware attack that rapidly spread through computer networks around the world in May 2017. The attack targeted a vulnerability in old Windows versions, for which a patch had been released by Windows more than two months before WannaCry spread across the world. The attack was highly effective because it spread across devices by exploiting the Windows Server Message Block (SMB) protocol, which enables Windows machines to communicate with each other on a network. The attack was spread using **EternalBlue**, a zero-day vulnerability in devices that use an old version of SMB. It was first discovered by the U.S. National Security Agency (NSA) before being obtained by **hacking group Shadow Brokers**, which published the exploit within a post on blogging site Medium in April 2017.

The WannaCry attack was formed of several components, which included:

### 1. An application that encrypts and decrypts data

The initial WannaCry dropper contains an application that enables an attacker to encrypt and decrypt data.

The encryption component is known as Wana Decrypt0r 2.0, and within it was a password-protected ZIP file.

### 2. Files containing encryption keys

Within that ZIP file were several individual files containing configuration information that helped the hacker launch their attack. It also included encryption keys that enabled them to unlock data.

### 3. A copy of Tor

The ZIP file also contained a copy of the Tor network, which is an open-source web browser that aims to protect and hide users' data, locations, and online activity through anonymous browsing.

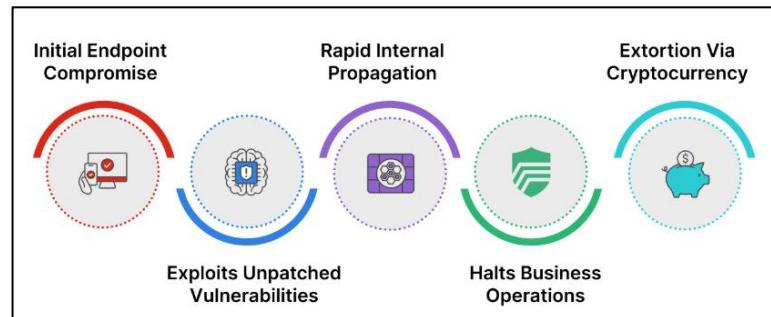


Figure 1: WannaCry Chain



WannaCry attackers typically issued a demand for victims to pay either \$300 or \$600 in bitcoin within a week of their device being attacked. However, victims were advised not to settle the ransom. In most cases, attackers did not decrypt data, and it was suspected they were not technically capable of doing so. The WannaCry ransomware attack works by using a dropper known as DoublePulsar, a software program that extracts embedded application components, to attack an infected computer. WannaCry attempts to access a Uniform Resource Locator (URL) that is hard-coded into the attack, and when accessed, shuts WannaCry down, which became known as its "kill switch." WannaCry then searches for important files on the device, which are typically Microsoft Office documents, MPEG Audio Layer 3 (MP3) files, or Matroska Multimedia Container (MKV) files. It encrypts the files, making them unavailable to the user, and displays a ransom demand for the user to pay to decrypt the files.

## 2. Penetration Testing Techniques

Explore PTES for phase details.

The **Penetration Testing Execution Standard (PTES)** is an internationally recognized framework designed to standardize the way penetration tests are planned, executed, and reported. Developed in 2009 by a group of industry experts, it provides a comprehensive end-to-end process to ensure consistent quality and actionable results for both technical teams and business stakeholders.

### The 7 Phases of PTES

The standard divides a penetration test into seven distinct stages:

1. **Pre-Engagement Interactions:** Defines the project's **scope**, goals, and **Rules of Engagement (RoE)**. It ensures both the client and tester agree on what is "off-limits" and establishes legal authorization.
2. **Intelligence Gathering:** Collects information about the target using **OSINT** (Open Source Intelligence), footprinting, and reconnaissance to identify potential entry points.
3. **Threat Modeling:** Analyzes gathered data to identify high-value assets and model likely attack scenarios based on real-world threat actors.
4. **Vulnerability Analysis:** Discovers security flaws through passive (traffic monitoring) and active (port scanning, application probing) methods.
5. **Exploitation:** The actual "attack" phase where testers bypass security controls to gain access, focusing on stealth, speed, and depth.
6. **Post-Exploitation:** Assesses the value of the compromised system, demonstrating how an attacker could maintain persistence or pivot to other network segments.
7. **Reporting:** Delivers a structured report containing an **Executive Summary** for management and a detailed **Technical Report** with remediation steps.

### The Five Phases of the OWASP Web Security Testing Framework

- 1) *Before Development Begins:* This initial phase involves establishing the overall security policy for the application and the organization. Key activities include defining security requirements and identifying the scope of testing.
- 2) *During Definition and Design:* In this phase, security considerations are integrated into the application's design. This typically involves performing threat modeling to identify potential attack vectors and designing security controls to mitigate identified risks.



- 3) *During Development:* This phase focuses on the actual implementation of security measures during coding. Activities include source code reviews to find vulnerabilities and ensuring that secure coding practices are followed.
- 4) *During Deployment:* Security testing during deployment involves ensuring the application and its infrastructure are securely configured before going live. This includes configuration and deployment management testing, as well as final penetration testing to validate the security posture in the production environment.
- 5) *During Maintenance and Operations:* This final phase emphasizes continuous security monitoring and assessment after the application is operational. It involves ongoing testing, incident response planning, and regular updates to maintain the application's security over time.

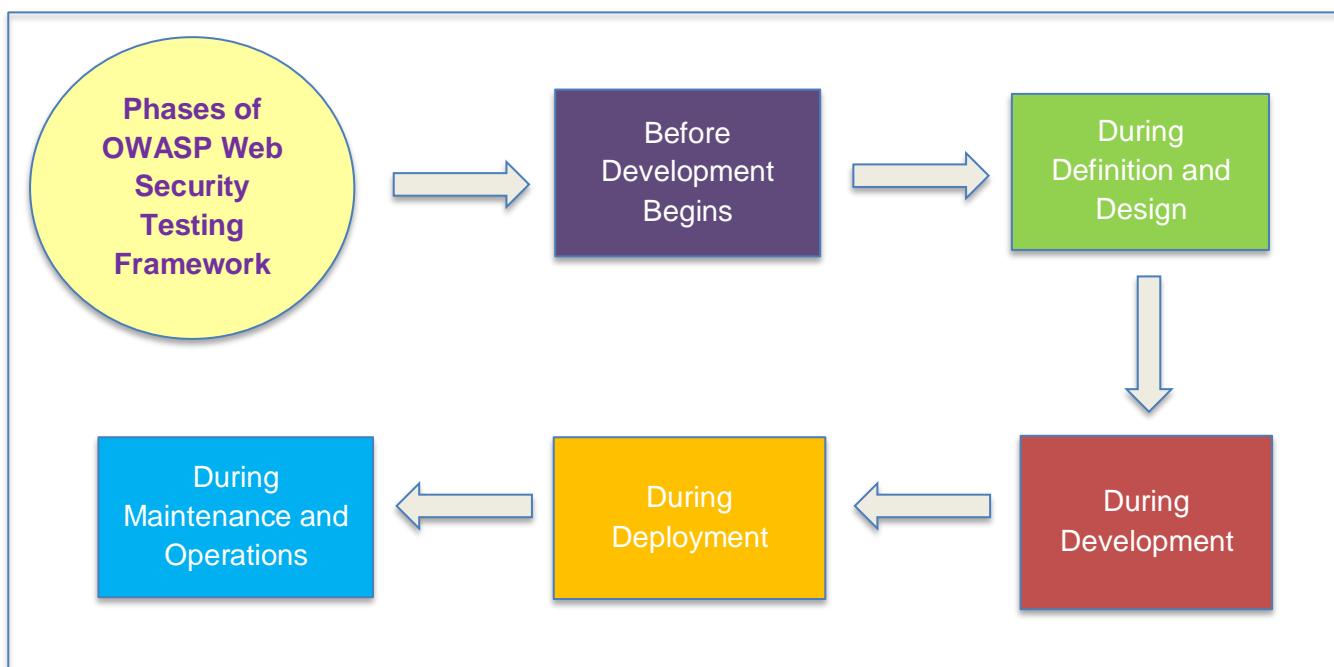


Figure 3: Phases of OWASP WSTG

SANS Institute provides a wealth of penetration testing case studies through its research library, white papers, and specialized training courses. These resources illustrate real-world applications of methodologies like **PTES** and the **SANS 6-Step Incident Response** framework.



### 3. Exploit Development Basics

#### 1. Buffer Overflow (Binary Vulnerability)

A **Buffer Overflow** occurs when a program writes more data to a fixed-length block of memory (a buffer) than it can hold. The extra data "overflows" into adjacent memory space, potentially overwriting the program's control flow.

#### 2. SQL Injection (Database Vulnerability)

As you saw with your sqlmap task, **SQL Injection (SQLi)** occurs when untrusted user input is concatenated directly into a database query.

#### 3. XSS: Cross-Site Scripting (Client-Side Vulnerability)

**Cross-Site Scripting (XSS)** involves injecting malicious JavaScript into a web page viewed by other users. The web application takes user input and displays it back on the page without "cleaning" or "escaping" it.

#### 1. ASLR (Address Space Layout Randomization)

ASLR is a memory protection transition that makes it harder for an attacker to predict where specific functions or libraries are located in RAM.

**How it works:** Every time a program or the OS starts, ASLR "shuffles" the memory addresses.

#### 2. WAF (Web Application Firewall)

A WAF is a specific type of firewall that sits in front of web applications (like DVWA) and monitors/filters HTTP traffic.

#### 3. Patching

Patching is the process of updating software to fix security vulnerabilities that have been discovered after the software was released.

### Capstone Project: Full VAPT Cycle

**DVWA (Damn Vulnerable Web Application)** is a PHP/MySQL web application that is intentionally designed to be vulnerable. It is used as a legal environment for security professionals to test their skills and tools, and for web developers to learn how to secure their code.

**sqlmap** is an open-source, Python-based **penetration testing tool** that automates the process of detecting and exploiting SQL injection (SQLi) vulnerabilities in web applications. It is widely used by both security professionals (for auditing and defense) and malicious actors (for attacks) to access, manage, and potentially take over back-end database servers.

## References

- [1] <https://www.impactmybiz.com/blog/different-vulnerability-scans/>
- [2] [https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP\\_Testing\\_Guide\\_v4.pdf](https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf)
- [3] <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-115.pdf>
- [4] <https://qualysec.com/nist-sp-800-115-penetration-testing/>
- [5] <https://www.fortinet.com/resources/cyberglossary/wannacry-ransomware-attack>
- [6] [https://youtu.be/BdF3i4ZJ3\\_k?si=IkzqrWPdosfLKsna](https://youtu.be/BdF3i4ZJ3_k?si=IkzqrWPdosfLKsna)