LITERATURE REVIEW

CYBER ATTACK LIFE CYCLE AND VAPT

PROJECT PHASE 1

AMBADI KURUP S G

CSA 2025

ICTAK MARCH

29/7/2025

INDEX

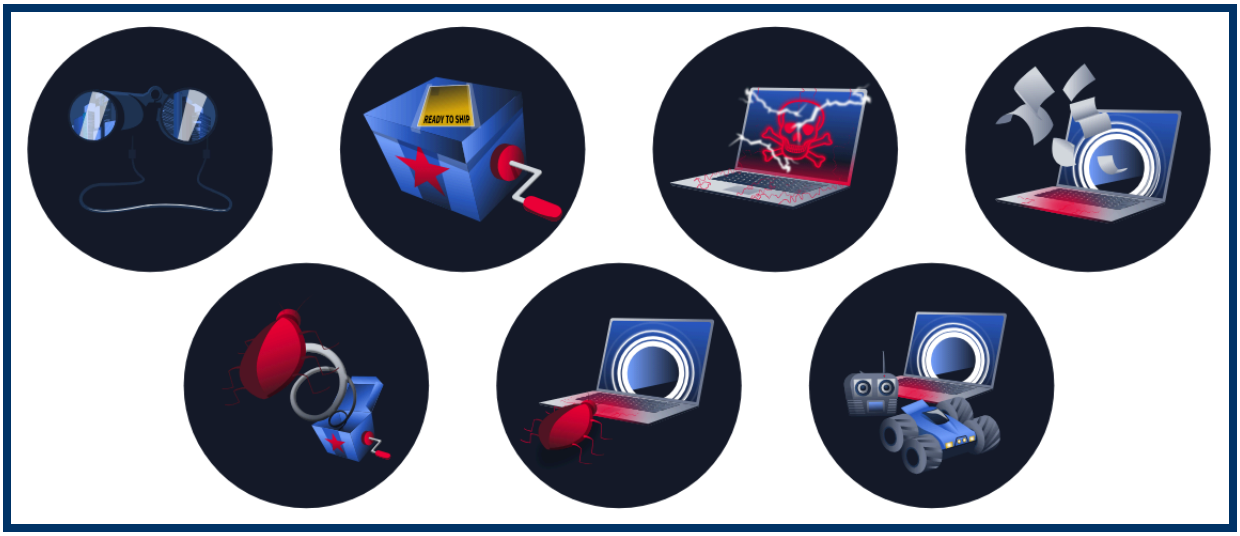
INTRODUCTION

In the age of digital dependency, cybersecurity has evolved from a background IT function into a critical pillar of organizational resilience. As businesses migrate to cloud platforms, deploy smart devices, and embrace automation, the digital attack surface has expanded exponentially. With this expansion comes a rise in sophisticated cyber threats, ranging from ransomware and data breaches to state-sponsored espionage.

Cybersecurity encompasses the frameworks, technologies, and practices designed to protect digital systems, networks, and data from unauthorized access, damage, or disruption. Beyond just firewalls and antivirus software, modern cybersecurity relies on **proactive threat detection**, **behavioral analytics**, and **structured incident response protocols.**

The purpose of this initial phase is to conduct a comprehensive review of existing literature on the Cyber Attack Lifecycle and VAPT (Vulnerability Assessment and Penetration Testing). This will help establish foundational knowledge, differentiate between theoretical attack models and practical assessment techniques, and identify how these frameworks are adapted across different environments like cloud systems, IoT devices, and traditional web applications. It also sets the stage for upcoming phases by pinpointing knowledge gaps and framing a structured approach to testing and mitigation.

CYBER ATTACK LIFE CYCLE



The **Cyber Attack Lifecycle** represents the **sequential stages followed by threat actors** to infiltrate, exploit, and achieve objectives within a target environment. Understanding this lifecycle is essential for organizations to anticipate attacks, implement layered defenses, and detect adversarial movements early in the kill chain.

Three famous frameworks for the Cyber Attack Lifecycle are:

1. **Lockheed Martin Cyber Kill Chain**
2. **MITRE ATT&CK Framework**
3. **NIST Cybersecurity Framework (CSF)**

The different stages of the lifecycle are:

- **Reconnaissance (Information Gathering)**

The attacker collects data on the target—network architecture, employee details, exposed systems, and software stack—using tools like Nmap, WHOIS, and Google Dorking. This can be passive (open-source info) or active (port scanning).

- **Weaponization**

Based on the gathered intelligence, the attacker creates or customizes an exploit—often embedding malicious code (e.g., malware or scripts) into a benign-looking document or application.

- **Delivery**

The crafted payload is transmitted to the victim via channels such as phishing emails, USB drives, malicious links, or infected websites.

- **Exploitation**

Once delivered, the attacker leverages a vulnerability (e.g., buffer overflow, XSS) to execute code and gain a foothold in the system.

- **Installation**

A backdoor, trojan, or remote access tool (RAT) is installed to establish persistent access to the compromised machine.

- **Command and Control (C2)**

The attacker establishes communication with the victim system to issue commands, move laterally, or exfiltrate data. Techniques like DNS tunneling, HTTP beacons, or custom protocols are often used here.

- **Actions on Objectives**

This final phase involves fulfilling the attack's motive—data theft, espionage, sabotage, financial fraud, or system disruption.

CONCLUSION

The Cyber Attack Kill Chain offers a structured and strategic view of how cyber threats evolve from initial reconnaissance to final exploitation. By dissecting each stage of an attacker's operation, organizations can better anticipate, detect, and disrupt malicious activities. This lifecycle model not only aids in understanding the mindset of adversaries but also empowers defenders to implement **proactive security controls** at every phase. As cyber threats continue to evolve in complexity, leveraging frameworks like the Kill Chain remains essential for building resilient and adaptive cybersecurity defenses.

VAPT(Vulnerability Assessment & Penetration Testing)

VAPT is a combination of security testing processes used for identifying,evaluating and safely exploiting weaknesses within a digital system. We can divide VAT into two parts as Vulnerability Assessment and Penetration Testing. Vulnerability Assessment and Penetration Testing are two different processes but they serve or complement each other.

Vulnerability Assessment: It is a process of scanning, identifying and reporting vulnerabilities in a digital environment. The main objective is to find any vulnerabilities or weak points in the assigned target. Tools commonly used for Vulnerability Assessment include Nessus, OpenVas, Qualys and Nikto.

Penetration Testing : Penetration Testing is performed by ethical hackers or penetration testers after finding the vulnerabilities or weak points using the vulnerability assessment. The main objective is to exploit the vulnerabilities and assess the impact they can have on the targeted system. The tools used for this purpose are Meatsploit, Burpsuite, Hydra, SQLmap.

Difference Between Cyber Attack Life Cycle and VAPT

The Cyber Attack Life Cycle is a blueprint showing the stages of a cyber attack. It shows the approach of a malicious hacker from gathering information to gaining access, stealing data, or causing damage. It's basically a theoretical model used to understand how the attackers operate.

On the other hand, Vulnerability Assessment and penetration Testing is a legal and controlled process carried out either by cyber security professionals or ethical hackers to identify and exploit the vulnerabilities in the system. It is basically a preventive measure to strengthen security.

Environment To Conduct VAPT

VAPT can be conducted in multiple Environments and is not limited to just websites or servers. Following are the different environments where VAPT can be conducted.

▪Web Applications

Web applications are frequently targeted due to their public accessibility. VAPT in this environment focuses on identifying issues like SQL injection, XSS, and authentication flaws. Tools like Burp Suite and OWASP ZAP are commonly used.

▪Mobile Applications

Mobile apps often handle sensitive data, making them attractive to attackers. Testing involves analyzing insecure storage, API vulnerabilities, and reverse engineering the app. Tools such as MobSF and Frida assist in uncovering these flaws.

▪Cloud Environments

Cloud platforms introduce risks through misconfigurations and exposed services. VAPT focuses on permissions, storage security, and API protection. Tools like ScoutSuite and Prowler are commonly used in cloud assessments.

▪Network Infrastructure

This includes routers, switches, and internal networks that keep systems connected. VAPT identifies open ports, outdated firmware, and insecure protocols. Tools like Nmap and Wireshark help map and analyze the network.

▪IOT Devices

IoT devices are often shipped with weak security features. VAPT checks for hardcoded credentials, insecure communication, and outdated firmware. Tools like Binwalk and Ghidra are used to inspect firmware and device behavior.

▪Operating Systems and Endpoints

Systems running Windows, Linux, or macOS are tested for local exploits and misconfigurations. VAPT looks for privilege escalation paths and patching gaps. Metasploit and privilege-checker tools are commonly used here.

▪APIs and Microservices

APIs are essential for app communication but often poorly secured. Testing looks for broken authentication, data leaks, and abuse of functions. Tools like Postman and Burp are used for crafting and analyzing API requests.

Implementation Of VAPT in Different Environments

VAPT is tailored based on the environment being tested. For **web apps**, it involves scanning inputs and simulating attacks like SQL injection or XSS. In **mobile apps**, testers reverse engineer the app and inspect data storage and API usage.

In **cloud platforms**, the focus is on misconfigurations, exposed services, and access control using tools like ScoutSuite. **Networks** are tested by scanning for open ports, outdated devices, and weak protocols. For **IoT devices**, firmware is analyzed and hardware interfaces are tested for insecure defaults.

Operating systems are assessed for privilege escalation paths, weak configurations, and missing patches. **Enterprise apps and databases** are tested for injection flaws, access control, and sensitive data leaks. Finally, **APIs** are probed using custom requests to find broken authentication, poor validation, or excessive data exposure.

Similarities and Differences in VAPT phases across different Environments.

Similarities

▪**Planning** is essentia**l** in every environment, where the scope, targets, and permissions are clearly defined.

▪**Scanning and information gathering** always occur, whether it's ports on a server, endpoints in an API, or firmware in an IoT device.

▪**Reporting** is a mandatory final phase where findings, risk levels, and mitigation strategies are documented and presented to stakeholders.

Differences

▪In **web applications**, testing is input-focused, involving form fields, URL parameters, and session cookies using tools like Burp Suite or OWASP ZAP.
▪In **IoT devices**, the approach is hardware-centric—testers may open the device, extract firmware, and analyze communications over UART/JTAG interfaces.
▪**Mobile apps** require static and dynamic analysis, focusing on APK/IPA decompilation, insecure storage, and runtime behavior.

▪**Cloud environments** demand configuration audits, privilege testing (IAM roles), and checks for exposed resources like S3 buckets.
▪**APIs** are tested by crafting various requests (GET, POST, PUT) and observing how the service handles input, rate limits, and authorization.

In brief VAPT process structure remains constant, but the tools and the methods used in testing depend on the environment.

Conclusion

In an era where technology is evolving stronger and faster, it is necessary to have knowledge about cyber attacks and cyber threats. Knowledge of cyber attack lifecycle and VAPT is essential for defending the attack from malicious hackers. Cyber attack life cycle gives an insight into how the attacker approaches the target system and stages of attack. It is mapping how the attacker thinks in each stage. VAPT, on the other hand, tests the target systems or assigned targets to discover the vulnerabilities before the attackers do and patch those vulnerabilities.

These frameworks help in spotting security issues early, planning better responses, and reducing risks. However, VAPT might miss some flaws if the testing scope is limited, and the Cyber Attack Lifecycle may not fully capture how real attacks happen, since attackers don't always follow a fixed pattern.

Overall, both models have pros and cons, but they work best when used together. Using the Cyber Attack Lifecycle alongside targeted VAPT helps organizations improve their security and stay ahead of threats.

Annexure A – Phase 2: Reconnaissance Phase Report Template

Introduction

▪Objective of the Phase

▪Scope of Reconnaissance

Work Allocation Table

▪Member Name

▪Assigned Tasks

▪Public IP Addresses used by each member accessing the target

Target Environment Overview

▪Description of Target Environment/Application

▪Network Topology (if applicable)

Information Gathering Methodology

▪Tools & Techniques Used

▪Passive vs Active Recon

Collected Information Details

▪DNS Information

▪Network Information (Open Ports, Services)

▪Application Information and Versions

▪Other Relevant Details (e.g., Users, Technologies, etc.)

Summary of Findings

Challenges Faced

Conclusion

Appendices

▪Raw Data / Screenshots / Logs (if any)

Annexure B – Phase 3: Vulnerability Assessment Report Template

Introduction

▪Objective of Vulnerability Assessment

▪Scope and Constraints

▪Reminder to avoid exploitation/damage

Work Allocation Table

▪Member Name

▪Assigned Tasks

▪Public IP Addresses used by each member accessing the target

Methodology

▪Tools and Techniques Used (from Phase 1 + Phase 2 findings)

▪Vulnerability Identification Process

Vulnerabilities Identified

▪Vulnerability ID/Number

▪Vulnerability Name

▪Description

▪Affected Component(s)

▪Detection Method/Tool

▪Evidence (screenshots/logs)

▪Severity Level (e.g., High, Medium, Low)

Summary of Vulnerabilities

Recommendations / Mitigation Suggestions

Challenges Faced

Conclusion

Appendices

- Detailed Scan Reports
- Logs/Screenshots

Annexure C – Phase 4: Penetration Testing Report Template

Introduction

▪Objective of the Penetration Test

▪Scope and Limitations

Work Allocation Table

▪Member Name

▪Assigned Tasks

▪Public IP Addresses used by each member accessing the target

Reconfirmation of Recon and VA Findings

▪Summary & Validation of Phase 2 & 3 Results

Penetration Testing Methodology

▪Tools and Techniques

▪Rules of Engagement

Detailed Vulnerability Exploitation For each vulnerability:

▪Vulnerability ID/Name

▪Description

▪Steps to Reproduce (detailed with screenshots)

▪Impact of Exploit

▪Risk Level

▪Mitigation and Remediation Steps

Additional Security Issues Found

Overall Security Assessment

Challenges Faced

Conclusion and Recommendations

Appendices

- Proof of Concept Screenshots
- Logs and Evidence
- Supporting Documents