Vulnerability Assessment

&

Penetration Testing

Table of Contents

[Introduction: 6](#_Toc75257630)

[Comparison of Vulnerability Assessment and Penetration Testing: 7](#_Toc75257631)

[Vulnerability Assessment: 8](#_Toc75257632)

[Causes of Security Vulnerabilities: 8](#_Toc75257633)

[Network structure 8](#_Toc75257634)

[Unknown devices 9](#_Toc75257635)

[Account abuse 9](#_Toc75257636)

[Web configuration errors 9](#_Toc75257637)

[Security feature configurations 9](#_Toc75257638)

[Third-party applications 9](#_Toc75257639)

[Missing updates 9](#_Toc75257640)

[Importance of Vulnerability Assessment: 9](#_Toc75257641)

[Categories of Vulnerability Scans: 10](#_Toc75257642)

[Network-based scanners 10](#_Toc75257643)

[Host-based scanners 10](#_Toc75257644)

[Application scanners 10](#_Toc75257645)

[Wireless network scanners 11](#_Toc75257646)

[Database scanners 11](#_Toc75257647)

[Types of Vulnerability Scans 11](#_Toc75257648)

[Credentialed Scans Versus Non-Credentialed Scans 11](#_Toc75257649)

[*External Vulnerability Scans:* 11](#_Toc75257650)

[*Internal Vulnerability Scans:* 12](#_Toc75257651)

[*Environmental Scans:* 12](#_Toc75257652)

[Intrusive Versus Non-Intrusive Scans: 12](#_Toc75257653)

[Vulnerability Assessment Methodology Types: 12](#_Toc75257654)

[Black box network vulnerability testing: 12](#_Toc75257655)

[White box vulnerability testing: 12](#_Toc75257656)

[Gray box vulnerability assessments: 12](#_Toc75257657)

[Vulnerability Assessment Stages: 13](#_Toc75257658)

[Stage 1: Initial Assessment (Planning) 13](#_Toc75257659)

[Stage 2: System Baseline Definition 13](#_Toc75257660)

[Stage 3: Vulnerability Assessment and Risk Assessment 14](#_Toc75257661)

[Stage 4: Reporting and Remediation 15](#_Toc75257662)

[Vulnerability Assessment Process: 16](#_Toc75257663)

[Requirements: 16](#_Toc75257664)

[Hardware / VM: 16](#_Toc75257665)

[Software: 17](#_Toc75257666)

[Tools: 17](#_Toc75257667)

[Open Source Tools 17](#_Toc75257668)

[Paid Tools: 17](#_Toc75257669)

[Penetration Testing: 18](#_Toc75257670)

[What Should Be Tested 18](#_Toc75257671)

[Causes Of Vulnerabilities: 19](#_Toc75257672)

[Design and development errors: 19](#_Toc75257673)

[Poor system configuration: 19](#_Toc75257674)

[Human errors: 19](#_Toc75257675)

[Connectivity: 19](#_Toc75257676)

[Complexity: 19](#_Toc75257677)

[Passwords: 19](#_Toc75257678)

[User Input: 19](#_Toc75257679)

[Management: 19](#_Toc75257680)

[Lack of training to staff: 19](#_Toc75257681)

[Communication: 20](#_Toc75257682)

[Importance of Pen Testing: 20](#_Toc75257683)

[Different Types of Penetration Testing: 21](#_Toc75257684)

[Black Box, Gray Box, and White Box: 21](#_Toc75257685)

[Internal and External Penetration test: 21](#_Toc75257686)

[In-house and Third party Penetration test: 22](#_Toc75257687)

[Blind and Double-Blind Penetration test: 22](#_Toc75257688)

[Teaming: 22](#_Toc75257689)

[Red Team: 22](#_Toc75257690)

[Blue Team: 22](#_Toc75257691)

[Purple Team 22](#_Toc75257692)

[Different Categories of Pen Testing: 23](#_Toc75257693)

[Web Application Tests: 23](#_Toc75257694)

[Network Security Tests: 23](#_Toc75257695)

[Client-Side or Website & Wireless Network: 24](#_Toc75257696)

[Cloud Security Tests: 24](#_Toc75257697)

[IoT Security Tests: 25](#_Toc75257698)

[Physical Testing: 25](#_Toc75257699)

[Social Engineering: 25](#_Toc75257700)

[Penetration Testing Stages: 26](#_Toc75257701)

[1. Pre-engagement interactions: 26](#_Toc75257702)

[2. Reconnaissance: 26](#_Toc75257703)

[3. Threat modeling and vulnerability identification: 27](#_Toc75257704)

[4. Exploitation: 27](#_Toc75257705)

[5. Post exploitation: 27](#_Toc75257706)

[6. Reporting: 28](#_Toc75257707)

[Standardized Penetration Testing Methodologies: 28](#_Toc75257708)

[OSSTMM: 28](#_Toc75257709)

[OWASP: 29](#_Toc75257710)

[NIST: 29](#_Toc75257711)

[ISSAF: 30](#_Toc75257712)

[PTES: 30](#_Toc75257713)

[Penetration Testing Sample Test Cases 30](#_Toc75257714)

[Penetration Testing Process: 32](#_Toc75257715)

[Requirements: 32](#_Toc75257716)

[Hardware: 33](#_Toc75257717)

[Software: 33](#_Toc75257718)

[Tools: 33](#_Toc75257719)

[Secure Software Development Lifecycle (SSDLC): 34](#_Toc75257720)

[Importance of Secure SDLC: 34](#_Toc75257721)

[Secure Software Development Life Cycle Processes & VAPT: 34](#_Toc75257722)

Table of Figures

[Figure 1: Technical assurance by VA & PT 6](#_Toc75257723)

[Figure 2: Common security vulnerabilities 8](#_Toc75257724)

[Figure 3: Categories of vulnerability Scans 10](#_Toc75257725)

[Figure 4: Vulnerability assessment process 13](#_Toc75257726)

[Figure 5: Risk scenario 15](#_Toc75257727)

[Figure 6: Vulnerability Assessment Process 16](#_Toc75257728)

[Figure 7: Causes of vulnerability 19](#_Toc75257729)

[Figure 8: Types of pentesting 21](#_Toc75257730)

[Figure 9: 6 phases of pentesting 26](#_Toc75257731)

[Figure 10: Standardized Penetration Testing Methodologies 28](#_Toc75257732)

[Figure 11: Pentesting Process 32](#_Toc75257733)

[Figure 12: Secure SDLC 35](#_Toc75257734)

# Introduction:

A cybersecurity expert plays the same role in their organization as a soldier does on the battlefield or a sportsperson during a game. Every organization today is dependent on IT. The smallest of loopholes can cause a massive breach, resulting in loss of data, finances, and reputation. This is where a cybersecurity team steps in to perform vulnerability analysis and penetration testing to ensure these vulnerabilities are closed and cannot be penetrated till that day and monitor regularly.

In the modern world, awareness about vulnerability is not just limited to the IT and cybersecurity department. Cyber hygiene is essential for anyone whose work uses third-party apps and is connected to the internet 24×7. Employers should train their employees with a detailed understanding of ethical hacking to detect possible breaches on time.

The purpose of vulnerability assessment and penetration testing is to prevent the possibility of unauthorized access to systems. Vulnerability testing preserves the confidentiality, integrity, and availability of the system. The system refers to any computers, networks, network devices, software, web application, cloud computing, etc.

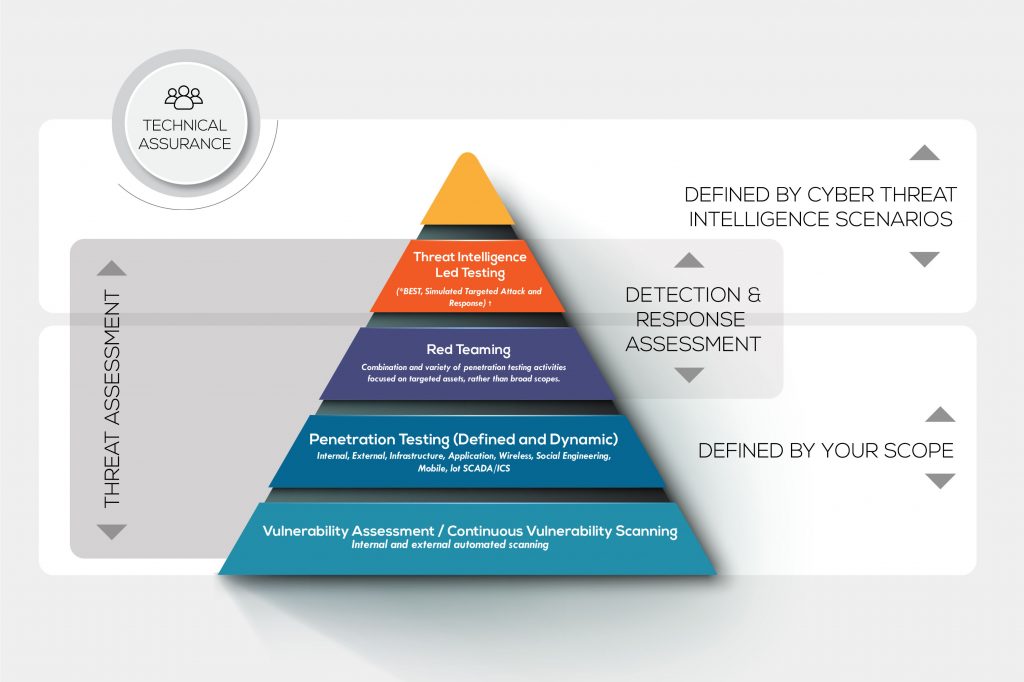


Figure : Technical assurance by VA & PT

## Comparison of Vulnerability Assessment and Penetration Testing:

|  |  |  |
| --- | --- | --- |
|  | **Vulnerability Assessment** | **Penetration Testing** |
| **Working** | Discover Vulnerabilities | Identify and Exploit Vulnerabilities |
| **Mechanism** | Discovery & Scanning | Simulation |
| **Focus** | Breadth over Depth | Depth over Breadth |
| **Coverage of Completeness** | High | Low |
| **Cost** | Low- Moderate | High |
| **Performed By** | In-house Staff or External parties | An attacker or Pen Tester |
| **Tester Knowledge** | High | Low |
| **How often to Run** | After each equipment is loaded | Quarterly (Depends on business) |
| **Result** | Provide Partial Details about Vulnerabilities | Provide Complete Details of Vulnerabilities |

# Vulnerability Assessment:

Vulnerability assessment—also called vulnerability analysis—is a process that identifies, quantifies and analyzes security weaknesses in IT infrastructure. The VA’s primary goal is to unearth any vulnerabilities that can compromise the organization’s overall security and operations. As such, the VA can help you minimize the probability of threats. It helps to determine

1. The hardware and software assets in an environment
2. The quantifiable value (criticality) of these assets
3. Identify the security vulnerabilities impacting the assets
4. Determine a quantifiable threat or risk score for each vulnerability
5. Mitigate the highest risk vulnerabilities from the most valuable assets

## Causes of Security Vulnerabilities:

There are countless ways bad actors could compromise a network and steal data. That said, there are common security vulnerabilities to watch out for. Not every network scanning tool will address all these concerns, but you should look for software to help you prioritize some or all of the following threats.

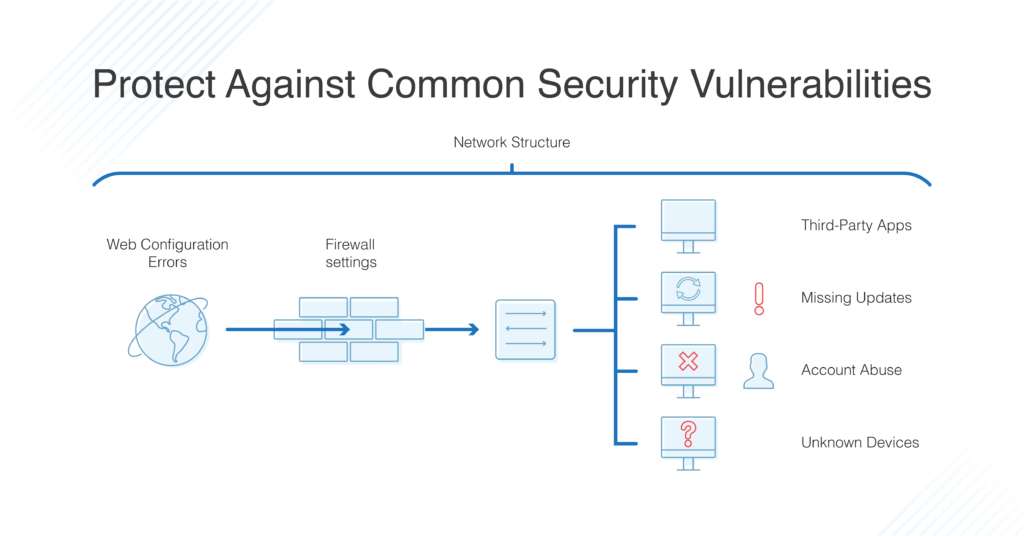


Figure : Common security vulnerabilities

Network structure – Too many business networks are essentially “open,” which means once an unauthorized user gains access, they have access to all parts of the network. This vulnerability can be prevented with better network segmentation and management of user group privileges.

Unknown devices – Unidentified or unmanaged assets on network are never good news. It’s important to make sure only approved devices have access to ports.

Account abuse – Unfortunately, insiders sometimes abuse their privileges, causing purposeful or inadvertent leaks of sensitive information, or the misconfiguration of programs, causing additional security holes. Furthermore, admins might allow default credentials, leave unused users or groups in the system, or assign incorrect privileges, all of which pose a security risk.

Web configuration errors – To ensure website application security, need to watch out for issues like distributed denial-of-service attacks, HTTP misconfigurations, expired SSL/TLS certificates, and insecure code.

Security feature configurations – Way of managing security settings and infrastructure could open risks. To avoid vulnerabilities, watch for firewall or OS misconfigurations.

Third-party applications – There’s a reason no one uses Java anymore. Too many third-party applications open security holes, whether because of how they’re built or how they’re downloaded and implemented. In addition to avoiding these applications, watch out for suspicious downloads, insecure remote desktop sharing software, and software nearing the end of its life.

Missing updates – One major cause of security issues on networks is basic errors in software and firmware configuration or cases where configuration levels become uneven across the network. Similarly, it’s all too easy to fall behind on updating and patching devices and programs, even if patches are available. Hackers can quickly exploit these gaps.

## Importance of Vulnerability Assessment:

Here are the benefits of VA.

1. Identify known security exposures before attackers find them. VA scans all the network components, verifying whether they have weaknesses that cybercriminals can use to attack the organization.
2. Create an inventory of all the devices on the network, including purpose and system information. This also includes vulnerabilities associated with a specific device.
3. Create an inventory of all devices in the enterprise to help with the planning of upgrades and future assessments.
4. Define the level of risk that exists on the network.
5. Establish a business risk/benefit curve and optimize security investments.
6. Proving to customers, prospects and other stakeholders that your systems are secure.
7. Evaluating the performance of third-party IT service providers.
8. Complying with industry and regulatory requirements.
9. Saving time and costs. Security breaches can hurt organizations on many fronts, creating limitations and liabilities that are costly. VA mitigates such risks, allowing the organization to save time and stop expensive litigations arising from data breaches.

## Categories of Vulnerability Scans:

The five categories of vulnerability scans are based on the kind of digital assets they can scan. They are network-based scanners, host-based scanners, application scanners, wireless network scanners, and database scanners.

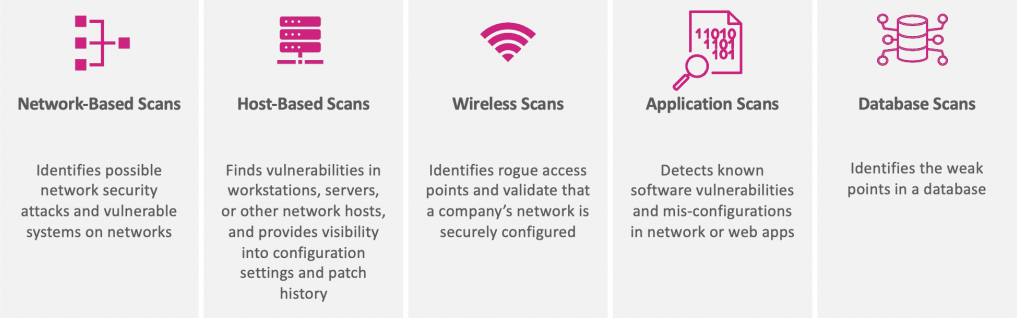


Figure : Categories of vulnerability Scans

### Network-based scanners

Use network-based scanners to discover unauthorized devices or unknown users on the network. These scanners allow network administrators to determine whether obscure perimeter loopholes such as unauthorized remote access exist on the network. Network-based scanners do not have direct access to the file system. As such, they cannot undertake low-level security checks.

### Host-based scanners

As the name suggests, a host-based scanner resides on every host on the monitored network. It locates and identifies vulnerabilities on workstations, servers or other network hosts, providing greater visibility in your assets’ configuration settings.

### Application scanners

Application scanners find vulnerabilities in websites. Their operation mode is similar to those of search engines—they “crawl” through websites by sending a range of probes to each web page on a website to look out for weaknesses.

### Wireless network scanners

Wireless network scanners—also called wireless protocol analyzers—are tools that can be used to discover open wireless networks in your environment. Organizations that prohibit wireless networks can use these wireless network scanners to detect any unauthorized Wi-Fi networks.

### Database scanners

Use database scanners to identify the vulnerabilities in database. Database scanners can help thwart malicious hacks like SQL injection attacks.

## Types of Vulnerability Scans

Some of vulnerability scanning tools are comprehensive in their coverage, able to perform multiple types of scans across heterogeneous environments that include on-prem, Unix, Linux, Windows, cloud, off-site, and onsite. Other scanning tools serve particular niches, so it’s always critical to thoroughly explore your use cases before investing in a scanner.

Different types of vulnerability scans, which each have their place, depending on use cases.

### Credentialed Scans Versus Non-Credentialed Scans

Credentialed and non-Credentialed scans (also respectively referred to as authenticated and non-authenticated scans) are the two main categories of vulnerability scanning.

Non-credentialed scans, as the name suggests, do not require credentials and do not get trusted access to the systems they are scanning. While they provide an outsider’s eye view of an environment, they tend to miss most vulnerabilities within a target environment. So, while they can provide some valuable insights to a potential attacker as well as to a security professional trying to gauge risk from the outside, non-credentialed scans give a very incomplete picture of vulnerability exposure.

Credentialed scans require logging in with a given set of credentials. These authenticated scans are conducted with a trusted user’s eye view of the environment. Credentialed scans uncover many vulnerabilities that traditional (non-credentialed) scans might overlook. Because credentialed scans require privileged credentials to gain access for scanning, organizations should look to integrate an automated privileged password management tool with the vulnerability scanning tool, to ensure this process is streamlined and secure (such as by ensuring scan credentials do not grow stale).

### External Vulnerability Scans:

These scans target the areas of IT ecosystem that are exposed to the internet, or are otherwise not restricted to internal users or systems. They can include websites, ports, services, networks, systems, and applications that need to be accessed by external users or customers.

### Internal Vulnerability Scans:

These scan and target internal corporate network. They can identify vulnerabilities that leave organization susceptible to damage once a cyberattacker or piece of malware makes it to the inside. These scans allow to harden and protect applications and systems that are not typically exposed by external scans.

### Environmental Scans:

These scans are based on the environment that organization’s technology operates in. Specialized scans are available for multiple different technology deployments, including cloud-based, IoT devices, mobile devices, websites, and more.

Intrusive Versus Non-Intrusive Scans:   
Non-intrusive scans simply identify a vulnerability and report on it so issue can be fixed. Intrusive scans attempt to exploit a vulnerability when it is found. This can highlight the likely risk and impact of a vulnerability, but may also disrupt your operational systems and processes, and cause issues for your employees and customers — so use intrusive scanning with caution.

## Vulnerability Assessment Methodology Types:

There are three different types of methodologies that network security specialists can employ when conducting an assessment. They include the following:

### Black box network vulnerability testing:

In this method, security team attempts to infiltrate your own cyber defenses from the outside just as a hacker might. Without having any administrative privileges or account passwords, the team attempts to exploit public IP addresses, firewalls and anything located in your demilitarized zone (DMZ) with that goal in mind.

### White box vulnerability testing:

The opposite side of the coin, white box testing involves team being given all of the privileges that authorized users have in order to conduct a thorough analysis of the entire network, including file servers and databases. Their job is to scan the entire internal environment for vulnerabilities and to use tools to assess the security of the stored information and machine configuration.

### Gray box vulnerability assessments:

This incorporate some of both white and black box methods. This type of analysis is done if security team experts receive certain intelligence about a network such as user login details but do not have full access to the entire environment.

## Vulnerability Assessment Stages:

Vulnerability assessment process starts with the approval process. IT (VA) team will define the objective, goals, scopes, timeline, and budget and get the approval from the authority to do vulnerability assessment with high-level details and methods.

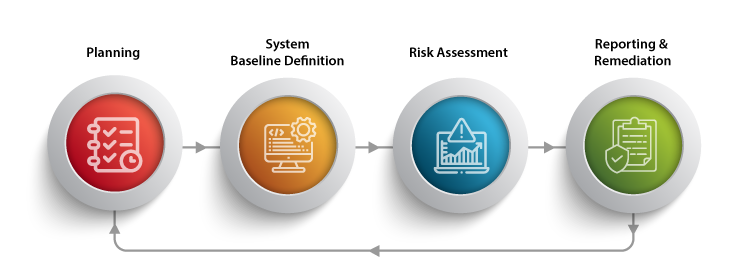


Figure : Vulnerability assessment process

### Stage 1: Initial Assessment (Planning)

1. Identify assets and know the worth of the devices that are part of the network.
2. Identify the risks and critical value device. It should also include a security analysis vulnerability scanner.
3. Analyze if the device is accessible to everyone or limited to the authorized users and administrators alone.

The information collected from the above three steps can be used to find solutions and predict:

1. The level and impact of risk.
2. Risk tolerance level
3. Risk mitigation practices and policies for each device
4. Residual risk treatment
5. Analysis of the impact of the said risk(s) on the business.
6. Proposing a risk strategy.
7. Simulating a cyberattack scenario.
8. Possible mitigation of risks for each device or service.
9. Setting up practices and policies for risk mitigation in each device.

### Stage 2: System Baseline Definition

1. Documentation of installed systems on UPAY network, their capabilities, and the users who have access.
2. Document all the services, processes, and open ports of those devices.
3. Scan device or use threat intelligence and a vulnerability database to detect vulnerabilities and remove false positives.

Furthermore, it would help if acquainted with certified drivers and software. Create a record of public data and vulnerabilities regarding the device program, vendor, version, and other significant details. This way, installing any low-quality product with high risk can be avoided.

### Stage 3: Vulnerability Assessment and Risk Assessment

Prior to starting the vulnerability scan, look for any compliance requirements based on company’s posture and business, and know the best time and date to perform the scan. It’s important to recognize the client industry context and determine if the scan can be performed all at once or if a segmentation is needed. An important step is to re-define and get the approval of the policy for the vulnerability scan to be performed.

For the best results, use related tools and plug-ins on the vulnerability assessment platform, such as:

1. Best scan (i.e., popular ports)
2. CMS web scan (Joomla, WordPress, Drupal, general CMS, etc.)
3. Quick scan
4. Most common ports best scan (i.e., 65,535 ports)
5. Firewall scan
6. Stealth scan
7. Aggressive scan
8. Full scan, exploits and distributed denial-of-service (DDoS) attacks
9. MITRE ATT&CK
10. User behavior analysis
11. List of vulnerabilities from OWASP, CVE, NVD
12. Open Web Application Security Project (OWASP) Top 10 Scan, OWASP Checks
13. Payment Card Industry Data Security Standard (PCI DSS) preparation for web applications

This stage helps in identifying the root cause of the vulnerabilities. From here, prioritize such vulnerabilities according to the threat level.

For example, if the source of vulnerability is an obsolete version of an open-source library, need to upgrade it.

A certified ethical hacker or other specialized security analyst will conduct the risk assessment by allocating a severity score. They will rank the vulnerability based on the following factors:

1. Severity of an attack.
2. Systems affected during that attack.
3. What data is at risk?
4. The potential business function(s) at risk.
5. The possible harm the vulnerability may trigger.

A comprehensive vulnerability assessment process guides in understanding, and deciding on the optimal course of action in these four common scenarios

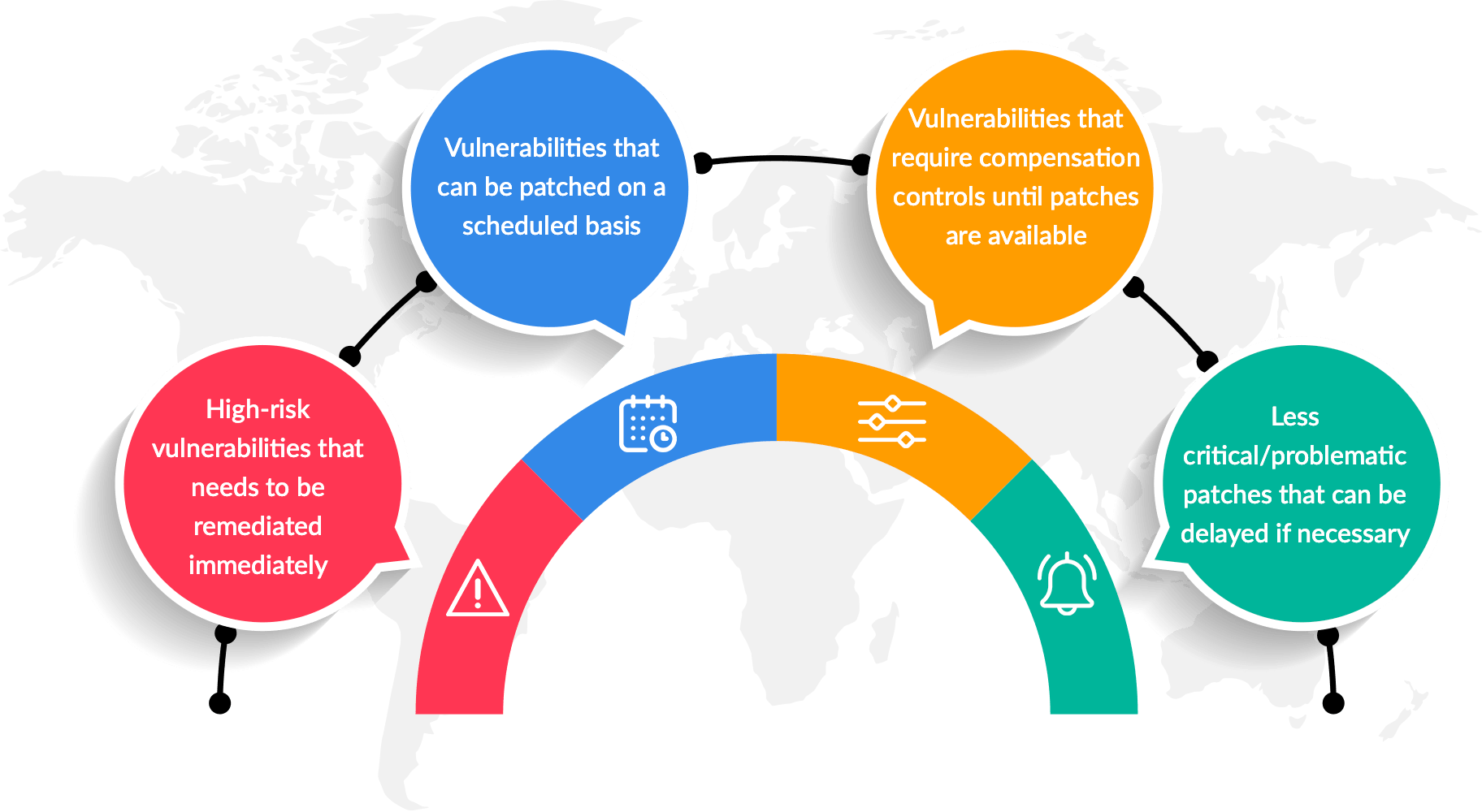


Figure : Risk scenario

### Stage 4: Reporting and Remediation

This stage is the most important because it aims to close all the security gaps. It is usually a team effort by the security staff, IT professionals, developers, and operations team. It also requires involvement from the incidence response team. Their presence ascertains the most effective response and remediation strategy that is mapped out in a vulnerability assessment report. This report should include:

1. The name and date of vulnerability
2. The score, based on Common Vulnerabilities and Exposures (CVE) databases
3. A detailed description of the vulnerability
4. Details regarding the affected systems
5. Details regarding the process to correct the vulnerability
6. A proof of concept (PoC) of the vulnerability for the system (if possible)
7. A blank field for the owner of the vulnerability, the time it took to correct, the next revision and countermeasures between the final solution
8. Introduction of new techniques for risk mitigation.
9. Identifying the potential gap between the results and the system baseline.
10. Implementing measures to mitigate potential vulnerabilities.
11. Solutions are reported based on the original assessment objectives.
12. Conclusions are drawn according to the data collected during vulnerability assessment and are organized to assure the findings’ assessment.

## Vulnerability Assessment Process:

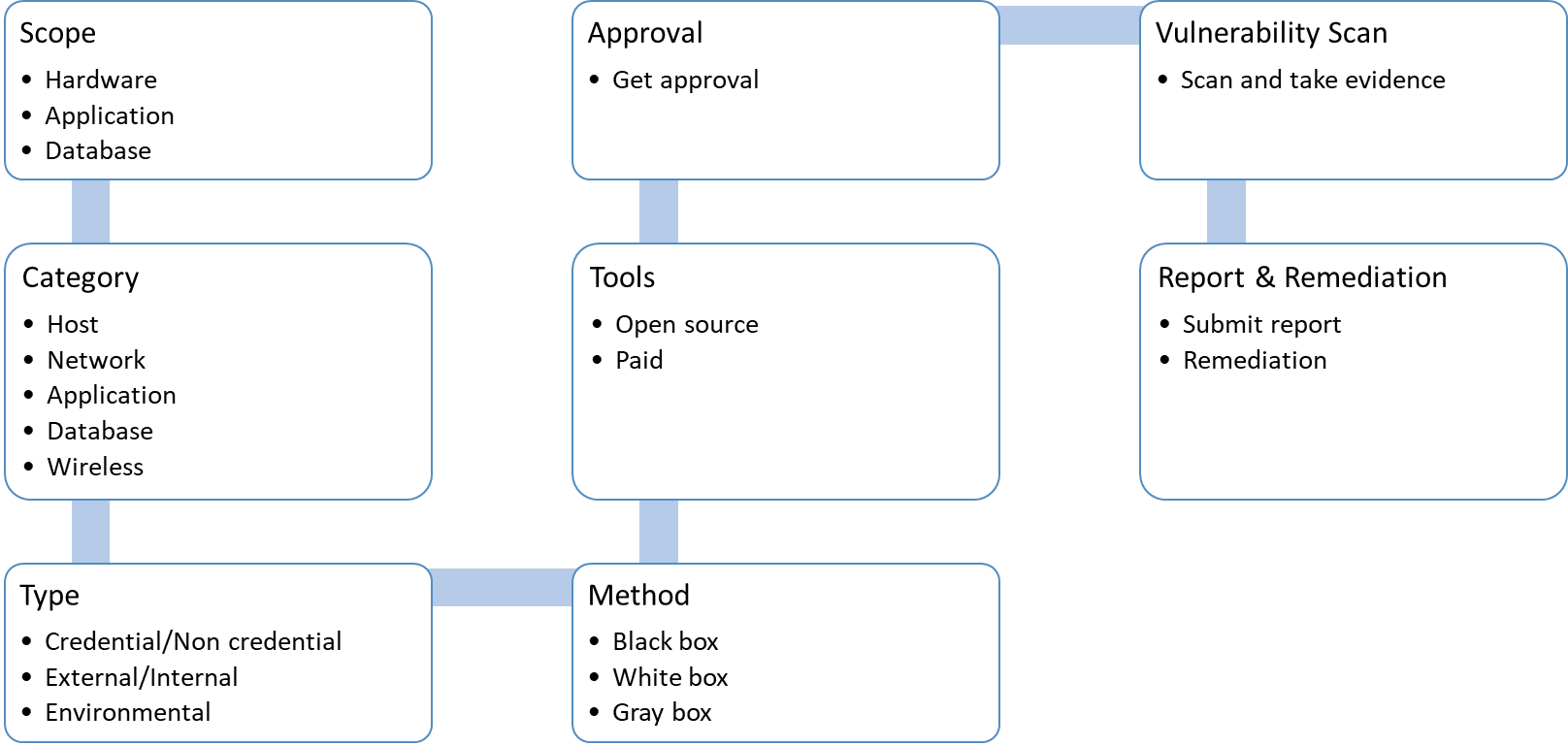


Figure : Vulnerability Assessment Process

# Requirements:

Following hardware, software and tolls are required to do VA

## Hardware / VM:

For VA Tools:

* + - Qty : 6
    - Core : 4
    - RAM : 32 GB
    - HDD : 500 GB

For Log Analysis:

* + - Qty : 4
    - Core : 32
    - RAM : 128 GB
    - HDD : 10 TB

## Software:

OS : Centos Stream

Kali Linux

## Tools:

List of tools required to do vulnerability assessment

### Open Source Tools

1. OpenVAS, maintained by Greenbone Networks
2. Nexpose Community
3. Metasploit Framework
4. Retina CS Community, from BeyondTrust
5. Burp Suite Community Edition, from PortSwigger
6. Nikto, sponsored by Netsparker
7. OWASP Zed Attack Proxy (ZAP)
8. Clair
9. Moloch
10. Powerfuzzer
11. W3AF
12. Nmap
13. OSSIM
14. Wazuh
15. Quality Community Edition
16. Aircrack
17. Grabber
18. Vega

### Paid Tools:

1. Nessus Professional
2. Appknox
3. Burp Suite
4. Tripwire IP360
5. Netsparker
6. Acunetix
7. Intruder
8. AppTrana
9. IBM Security QRadar

# Penetration Testing:

A penetration test, also known as a pen test, is a simulated cyber attack against computer system to check for exploitable vulnerabilities. In the context of web application security, penetration testing is commonly used to augment a web application firewall (WAF).

Pen testing can involve the attempted breaching of any number of application systems, (e.g., application protocol interfaces (APIs), frontend/backend servers) to uncover vulnerabilities, such as unsanitized inputs that are susceptible to code injection attacks.

Insights provided by the penetration test can be used to fine-tune your WAF security policies and patch detected vulnerabilities.

Penetration testing should be performed on a regular basis to ensure more consistent IT and network security management. A pen-tester will reveal how newly discovered threats or emerging vulnerabilities may potentially be assailed by attackers. In addition to regularly scheduled analysis and assessments required by regulatory mandates, tests should also be run whenever:

1. Network infrastructure or applications are added
2. Upgrades to infrastructure or applications are done
3. Security patches are applied
4. End user policies are modified
5. New office locations are established

## What Should Be Tested

1. Software (Operating system, services, application)
2. Hardware
3. Network
4. Processes
5. End-user behavior

## Causes Of Vulnerabilities:



Figure : Causes of vulnerability

Design and development errors: There can be flaws in the design of hardware and software. These bugs can put your business-critical data at risk of exposure.

Poor system configuration: This is another cause of vulnerability. If the system is poorly configured, then it can introduce loopholes through which attackers can enter into the system & steal the information.

Human errors: Human factors like improper disposal of documents, leaving the documents unattended, coding errors, insider threats, sharing passwords over phishing sites, etc. can lead to security breaches.

Connectivity: If the system is connected to an unsecured network (open connections) then it comes in the reach of hackers.

Complexity: The security vulnerability rises in proportion to the complexity of a system. The more features a system has, the more chances of the system being attacked.

Passwords: Passwords are used to prevent unauthorized access. They should be strong enough that no one can guess your password. Passwords should not be shared with anyone at any cost and passwords should be changed periodically. In spite of these instructions, at times people reveal their passwords to others, write them down somewhere and keep easy passwords that can be guessed.

User Input: You must have heard of SQL injection, buffer overflows, etc. The data received electronically through these methods can be used to attack the receiving system.

Management: Security is hard & expensive to manage. Sometimes organizations lack behind in proper risk management and hence vulnerability gets induced in the system.

Lack of training to staff: This leads to human errors and other vulnerabilities.

Communication: Channels like mobile networks, internet, telephone opens up security theft scope.

## Importance of Pen Testing:

Penetration testing is required to protect them from all forms of weaknesses. It is vital to implement penetration testing strategies for these reasons:

1. Weaknesses in the architecture are identified and fixed before a hacker can find and exploit them; thus, causing a business loss or unavailability of services.
2. Uncover new threats by potential attackers and vulnerabilities that would otherwise have remained undetected.
3. Aids security team to learn how to tackle any break-in from malicious actors.
4. Verify the effectiveness of specific tools.
5. Recognize real-time vulnerabilities within systems and web applications.
6. Develop strategies to tackle the flows detected in the infrastructure, application, or process.
7. Optimize security response time.
8. Assist developers in making fewer errors.
9. Provides insight into which channels in your application or organization are at most risk and determines the tools to be used or protocols to be followed.
10. Need to comply with various standards and compliance procedures. A penetration test will ensure that the gaps are fixed in time to meet compliance.

## Different Types of Penetration Testing:

Types of penetration testing can be categorized on the basis of either, the knowledge of the target or the position of the penetration tester. There are a few other parameters to the categorization of penetration.

### Black Box, Gray Box, and White Box:

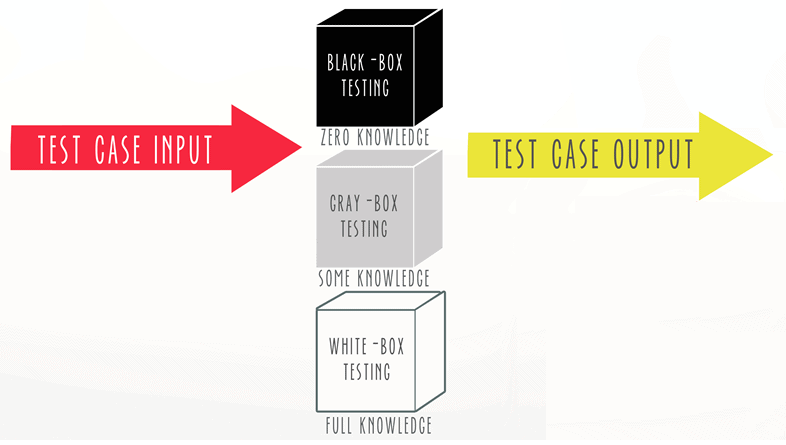


Figure : Types of pentesting

When the penetration tester is given the complete knowledge of the target, this is called a white box penetration test. The attacker has complete knowledge of the IP addresses, controls in place, code samples, etc. When the attacker has no knowledge of the target, this is referred to as a black box penetration test. Please note that the tester can still have all the information that is publically available about the target. When the tester is having partial information about the target, this is referred to as gray box penetration testing. In this case, the attacker is having some knowledge of the target like URLs, IP addresses, etc., but does not have complete knowledge or access. This is with respect to the knowledge.

### Internal and External Penetration test:

If the penetration test is conducted from outside the network, this is referred to as external penetration testing. If the attacker is present inside the network, simulation of this scenario is referred to as internal penetration testing. Since the attacker is an internal person, the knowledge about the system and the target will be abundant when compared to a test conducted from outside.

Common targets for external testing are:

1. Domain name servers (DNS).
2. Web applications.
3. Email servers.
4. Websites.
5. Web and application servers.
6. Firewalls.

### In-house and Third party Penetration test:

When the test is conducted by an in-house security team, it is another form of internal penetration testing. When hired a third-party organizations to conduct these tests, this is referred to as third-party penetration testing.

### Blind and Double-Blind Penetration test:

In a blind penetration test, the penetration tester is provided with no prior information but the organization name. The penetration tester will have to do all the homework, just like a legitimate attacker would do. This will surely take more time, but the results would be more close to the practical attacks. A double-blind test is like a blind test but the security professionals will not know when the testing will start. Only the senior management will have this information. This will test the processes, controls and the awareness of the security teams if and when a real attack occurs.

## Teaming:

The number of attacks are increasing and the amount of research and experience that’s required to get ahead of these attacks is expanding the gap between time of attack and time of discovery. That’s where teaming comes in. Teaming exercises simulate real-life attack scenarios--with one team attacking, and another defending.

### Red Team:

A red team is formed with the intention of identifying and assessing vulnerabilities, testing assumptions, viewing alternate options for attack, and revealing the limitations and security risks for an organization. This designated group tests the security posture of your organization to see how it will fare against real-time attacks before they actually happen. Because of their roles as the attackers, teaming exercises are sometimes also referred to as red-teaming.

### Blue Team:

The Blue Team is tasked with detecting adversaries and preventing them from breaking into the organization’s infrastructure. Blue teams can begin to prepare before an attack by evaluating the environment and hardening where needed. During the attack simulation, their goal is to identify breaches swiftly, limit the spread of infection by confining to the system it entered through, and successfully stop the attack. Some simulations may include the Blue Team planning or executing recovery measures.

### Purple Team

More recently, the idea of a purple team has become the latest buzzword in the cybersecurity world. While there is some confusion surrounding the usage and definition of the term, it’s best to focus on the ideal it is promoting. Ultimately, the concept of a purple team is the mindset of seeing and treating red and blue teams as symbiotic. It is not red teams versus blue teams, but rather one large team focusing on the one overarching goal: improving security. The key to becoming a purple team comes down to communication.

## Different Categories of Pen Testing:

While it's tempting to just request that at tester "test everything," this would most likely lead to pen testers only scratching the surface of a number of vulnerabilities, sacrificing gathering valuable intelligence gained by going more in-depth in fewer areas, with clear objectives in mind. In order to make sure pen tests can achieve these objectives and pinpoint weaknesses, there are various different categories of pen tests that focus on different areas of an IT infrastructure, including:

### Web Application Tests:

Web application penetration tests examine the overall security and potential risks of web applications, including coding errors, broken authentication or authorization, and injection vulnerabilities.

Typical subjects of these tests are:

1. Web applications.
2. APIs.
3. Connections.
4. Frameworks.
5. Mobile apps.

Common application vulnerabilities include:

1. Cross-site request forgery and scripting.
2. Injection flaws.
3. Poor session management.
4. Insecure direct object references.
5. Coding errors.
6. Broken authentication or authorization protocols.
7. Weak cryptography.

### Network Security Tests:

Network penetration testing aims to prevent malicious acts by finding weaknesses before the attackers do. Pen testers focus on network security testing by exploiting and uncovering vulnerabilities on different types of networks, associated devices like routers and switches, and network hosts. They aim to exploit flaws in these areas, like weak passwords or misconfigured assets, in order to gain access to critical systems or data.

This form of testing includes:

1. Firewall bypassing.
2. Evading a next-generation intrusion prevention system (NGIPS).
3. Router and proxy server testing.
4. IPS and DPS evasion.
5. Open port scanning.
6. SSH security attacks.
7. Testing policies that stop lateral movement.
8. Intercepting network traffic.
9. Discovering legacy devices and third-party appliances.

This type of testing includes both internal and external network exploitation. Common weak points network penetration discovers are:

1. Misconfigured assets.
2. Product-specific vulnerabilities.
3. Wireless network weaknesses.
4. Rogue services.
5. Weak password protocols.

### Client-Side or Website & Wireless Network:

This type of testing inspects wireless devices and infrastructures for vulnerabilities. A wireless pen test discovers insecure wireless network configurations and poor authentication checks.

This type of testing includes scanning of:

1. Web server misconfigurations.
2. Anti-malware and DDoS prevention strategies.
3. SQL injections.
4. MAC address spoofing.
5. Media players and content creation software.
6. Cross-site scripting.
7. Hotspots and access points.
8. Encryption protocols.

### Cloud Security Tests:

Security teams to work with cloud providers and third-party vendors to design and carry out cloud security testing for cloud-based systems and applications. Cloud pen testing validates the security of a cloud deployment, identifies overall risk and likelihood for each vulnerability, and recommends how to improve cloud environment.

### IoT Security Tests:

Pen testers take the nuances of different IoT devices into account by analyzing each component and the interaction between them. By using layered methodology, where each layer is analyzed, pen testers can spot weaknesses that may otherwise go unnoticed.

### Physical Testing:

Physical penetration tests attempt to gain physical access to business areas. This sort of testing ensures the integrity of:

1. RFID systems.
2. Cameras.
3. Door entry systems and keypads.
4. Employee and vendor behavior.
5. Motion and light sensors.

Hackers often combine physical tests with social engineering to create realistic attack conditions.

### Social Engineering:

Social engineering is a breach tactic, which involves using deception in order to gain access or information that will be used for malicious purposes. The most common example of this is seen in phishing scams. Pen testers use phishing tools and emails tailored to an organization to test defense mechanisms, detection and reaction capabilities, finding susceptible employees and security measures that need improvement.

Social engineering pen tests typically mimic real-world attacks such as:

1. Eavesdropping.
2. Tailgating.
3. Phishing attacks.
4. Spear-phishing.
5. Baiting.
6. Scareware.
7. Pretexting.

## Penetration Testing Stages:

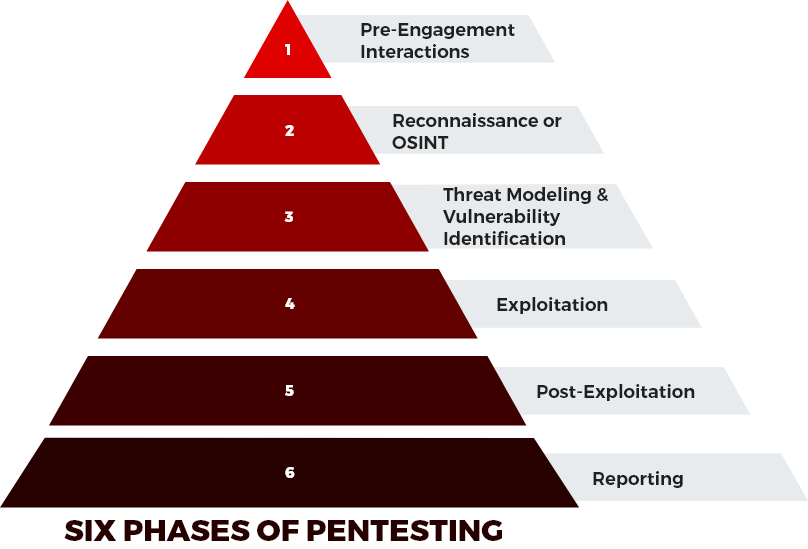


Figure : 6 phases of pentesting

### 1. Pre-engagement interactions:

In this phase, Pentester will define objective, goals, scopes, timeline, budget and get the approval from the authority to do pen testing with high-level details and methods. Here the penetration tester learns all they can about the target. The penetration tester works with employees to thoroughly understand risk posture, organizational culture, and consequently, the best penetration testing strategy to implement.

It is also known as the information-gathering phase. It is the stage where the pentester plans the testing exercise and aligns goals to specific pentesting results.

### 2. Reconnaissance:

Also called open-source intelligence (OSINT) gathering, reconnaissance involves using the information gathered to accumulate additional intelligence about the potential targets from publicly available sources. This stage is significant because, it allows the penetration tester to gather additional information that may have been previously overlooked.

* The penetration tester applies an extensive checklist for discovering open entry points and flaws within the organization.
* The OSINT Framework offers specific features for open information sources.
* The type of agreed pen testing will determine how the tester may gather various forms of Intel about UPAY to determine entry points and weaknesses in your environment.
* Some of the standard intelligence-gathering methodologies include Social Engineering, Search engine queries, Tailgating, Tax Records, Domain name searches/WHOIS lookups, or Internet Footprinting (e.g. email addresses, reverse DNS, usernames, packet sniffing, social networks, or Ping sweeps), etc.

The reconnaissance step includes the scan of:

1. All the hardware.
2. The local and wireless networks.
3. Firewalls.
4. Pertinent applications.
5. Websites.
6. Cloud-based systems.
7. Employee behavior and protocols.

### 3. Threat modeling and vulnerability identification:

Here, the pentester pinpoints targets and maps the attack vectors. Vulnerability scanners detect the security threats posed by the uncovered loopholes. Afterward, the tester will determine if the uncovered flaws are exploitable.

Penetration testers will map and identify an organization’s business assets and classify high-value assets such as customer data, employee data and technical data. The tester will also identify and classify internal threats (vendors, employees, or management) and external threats (Network traffic, ports, network protocols, or web applications).

### 4. Exploitation:

All the information is assembled and the pentester starts testing the exploits located within application, network, and data. This phase aims to understand precisely how attackers can break into UPAY environment and evade detection. The penetration tester can perform social engineering, web application attacks, physical attacks, network attacks, and memory-based attacks, among others, as exploit tactics.

### 5. Post exploitation:

Post exploitation processes involve risk analysis and recommendations. This phase of penetration testing aims to record the techniques exploited to gain access to an organization’s critical assets. The tester determines the significance of the compromised system and the significance of the collected data.

Afterward, the penetration tester makes recommendations based on these findings. The tester should also perform cleanups after the testing exercise. This can include removing any rootkits installed in the environment, eliminating any user accounts invented to connect to the breached system, deleting temporary files, scripts, etc.

### 6. Reporting:

The penetration tester gathers all the details of the exploitation and document the techniques exploited to gain access to UPAY’s critical asset. The ethical hacker prepares a detailed report covering all the activities in the previous five phases of penetration testing efforts. It includes how the vulnerabilities were detected and exploited. Apart from this, the report will also tell about the testing methodologies, outcomes, and recommendations for corrections.

## Standardized Penetration Testing Methodologies:

Companies typically rely on one of the five standardized penetration testing methods: OWASP, OSSTMM, ISSAF, PTES, and NIST.

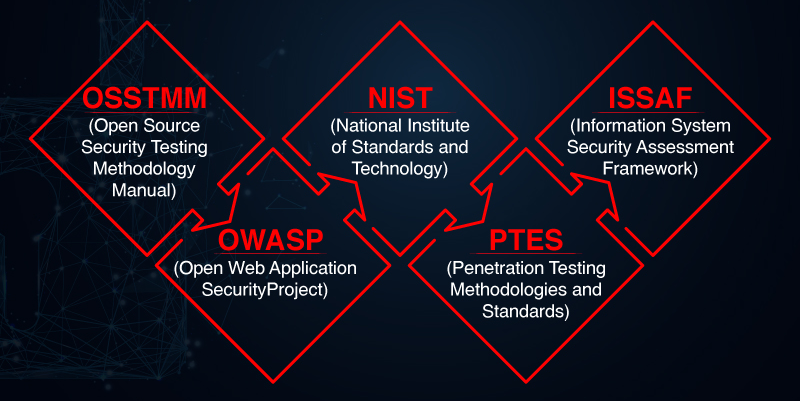


Figure : Standardized Penetration Testing Methodologies

### OSSTMM:

The OSSTMM (Open Source Security Testing Methodology Manual) is a recognized framework that details industry standards. The framework provides a scientific methodology for network penetration testing and vulnerability assessment. It is a comprehensive guide to the network development team and penetration testers to identify security vulnerabilities present in the network.

The OSSTMM methodology enables penetration testers to perform customized testing that fits the technological and specific needs of the organization. A customized assessment gives an overview of the network’s security, along with reliable solutions to make appropriate decisions to secure an organization’s network.

OSSTMM includes the following key sections:

1. Security Analysis
2. Operational Security Metrics
3. Trust Analysis
4. Work Flow
5. Human Security Testing
6. Physical Security Testing
7. Wireless Security Testing
8. Telecommunications Security Testing
9. Data Networks Security Testing
10. Compliance Regulations
11. Reporting with the STAR (Security Test Audit Report)

### OWASP:

The OWASP (Open Web Application Security Project) is a framework for identifying application vulnerabilities. This method allows a team to:

1. Recognize vulnerabilities within web and mobile applications.
2. Discover flaws within development practices.

The OWASP also enables testers to rate risks, which saves time and helps prioritize issues. This framework has a huge user community, so there is no shortage of OWASP articles, techniques, tools, and technologies.

The updated guide of OWASP provides over 66 controls to identify and assess vulnerabilities with numerous functionalities found in the latest applications today. However, it equips organizations with the resources to secure their applications and potential business losses. By leveraging the OWASP standard in security assessment, the penetration tester ensures almost nil vulnerabilities. Besides, it also enhances realistic recommendations to specific features and technologies in the applications.

### NIST:

The NIST (National Institute of Standards and Technology) varies information security manuals that differ from other information security manuals. In a way, NIST offers more specific guidelines intrinsic to penetration testing to improve the overall cybersecurity of an organization. Most American-based organizations and partners must comply with the regulatory compliance of the NIST framework. Moreover, the framework guarantees information security in industries like banking, communications, and energy. There is a probability of customizing the standards to meet their specific needs. Significantly, NIST contributes to security innovation in the American industries.

In order to comply with the NIST standards, organizations must conduct penetration testing on their applications and networks. However, organizations should follow pre-established guidelines. These guidelines ensure that the organizations fulfill their cybersecurity obligations and mitigate risks of possible cyberattacks.

### ISSAF:

The ISSAF (Information System Security Assessment Framework) is a specialized and structured approach to penetration testing. More importantly, the framework provides advanced methodologies that are personalized to the context. These standards allow a tester to plan and execute every step of the penetration testing process. Thus, it caters to all the requirements of the penetration testing process. As a penetration tester, if you are using different tools, then ISSAF is a crucial framework. For instance, it ties each step to a specific tool and thus reduces complexity.

ISSAF offers additional information concerning various attack vectors, as well as vulnerability outcome after exploitation. All this information allows testers to plan an advanced attack that guarantees a return on investment while securing systems from cyberattacks.

### PTES:

The PTES (Penetration Testing Methodologies and Standards) recommends a structured approach to a penetration test. On one side, the PTES guides you through the phases of penetration testing, beginning with communication, information gathering, and threat modeling phases. On the other hand, penetration testers acquaint themselves with the organization’s processes, which helps them identify the most vulnerable areas that are prone to attacks.

PTES provides guidelines to the testers for post-exploitation testing. If required, they can validate the successful fixing of previously identified vulnerabilities. The standard has seven phases that guarantee successful penetration testing with recommendations to rely on.

## Penetration Testing Sample Test Cases

1. In Pentest your goal is to find security holes in the system. Below are some generic test cases and not necessarily applicable to all applications.
2. Check if the web application is able to identify spam attacks on contact forms used on the website.
3. Proxy server – Check if network traffic is monitored by proxy appliances. The proxy server makes it difficult for hackers to get internal details of the network thus protecting the system from external attacks.
4. Spam email filters – Verify if incoming and outgoing email traffic is filtered and unsolicited emails are blocked.
5. Many email clients come with inbuilt spam filters that need to be configured as per your needs. These configuration rules can be applied to email headers, subject or body.
6. Firewall – Make sure the entire network or computers are protected with firewalls. A Firewall can be software or hardware to block unauthorized access to a system. A Firewall can prevent sending data outside the network without your permission.
7. Try to exploit all servers, desktop systems, printers, and network devices.
8. Verify that all usernames and passwords are encrypted and transferred over secure connections like https.
9. Verify information stored in website cookies. It should not be in a readable format.
10. Verify previously found vulnerabilities to check if the fix is working.
11. Verify if there is no open port in the network.
12. Verify all telephone devices.
13. Verify WIFI network security.
14. Verify all HTTP methods. PUT and Delete methods should not be enabled on a web server.
15. Verify if the password meets the required standards. The password should be at least 8 characters long containing at least one number and one special character.
16. Username should not be like “admin” or “administrator”.
17. The application login page should be locked upon a few unsuccessful login attempts.
18. Error messages should be generic and should not mention specific error details like “Invalid username” or “Invalid password”.
19. Verify if special characters, HTML tags, and scripts are handled properly as an input value.
20. Internal system details should not be revealed in any of the error or alert messages.
21. Custom error messages should be displayed to end-users in case of a web page crash.
22. Verify the use of registry entries. Sensitive information should not be kept in the registry.
23. All files must be scanned before uploading them to the server.
24. Sensitive data should not be passed in URLs while communicating with different internal modules of the web application.
25. There should not be any hardcoded username or password in the system.
26. Verify all input fields with long input string with and without spaces.
27. Verify if reset password functionality is secure.
28. Verify application for SQL Injection.
29. Verify application for Cross-Site Scripting.
30. Important input validations should be done at the server-side instead of JavaScript checks at the client-side.
31. Critical resources in the system should be available to authorized persons and services only.
32. All access logs should be maintained with proper access permissions.
33. Verify user session ends upon log off.
34. Verify that directory browsing is disabled on the server.
35. Verify that all applications and database versions are up to date.
36. Verify URL manipulation to check if a web application is not showing any unwanted information.
37. Verify memory leak and buffer overflow.
38. Verify if incoming network traffic is scanned to find Trojan attacks.
39. Verify if the system is safe from Brute Force Attacks – a trial and error method to find sensitive information like passwords.
40. Verify if the system or network is secured from DoS (denial-of-service) attacks. Hacker can target network or a single computer with continuous requests due to which resources on the target system gets overloaded resulting in the denial of service for legit requests.
41. Verify application for HTML script injection attacks.
42. Verify against COM & ActiveX attacks.
43. Verify against spoofing attacks. Spoofing can be of multiple types – IP address spoofing, Email ID spoofing,
44. ARP spoofing, Referrer spoofing, Caller ID spoofing, Poisoning of file-sharing networks, GPS spoofing.
45. Check for an uncontrolled format string attack – a security attack that can cause the application to crash or execute the harmful script on it.
46. Verify XML injection attack – used to alter the intended logic of the application.
47. Verify against canonicalization attacks.
48. Verify if the error pages are displaying any information that can be helpful for a hacker to enter into the system.
49. Verify if any critical data like the password is stored in secret files on the system.
50. Verify if the application is returning more data than it is required.

## Penetration Testing Process:

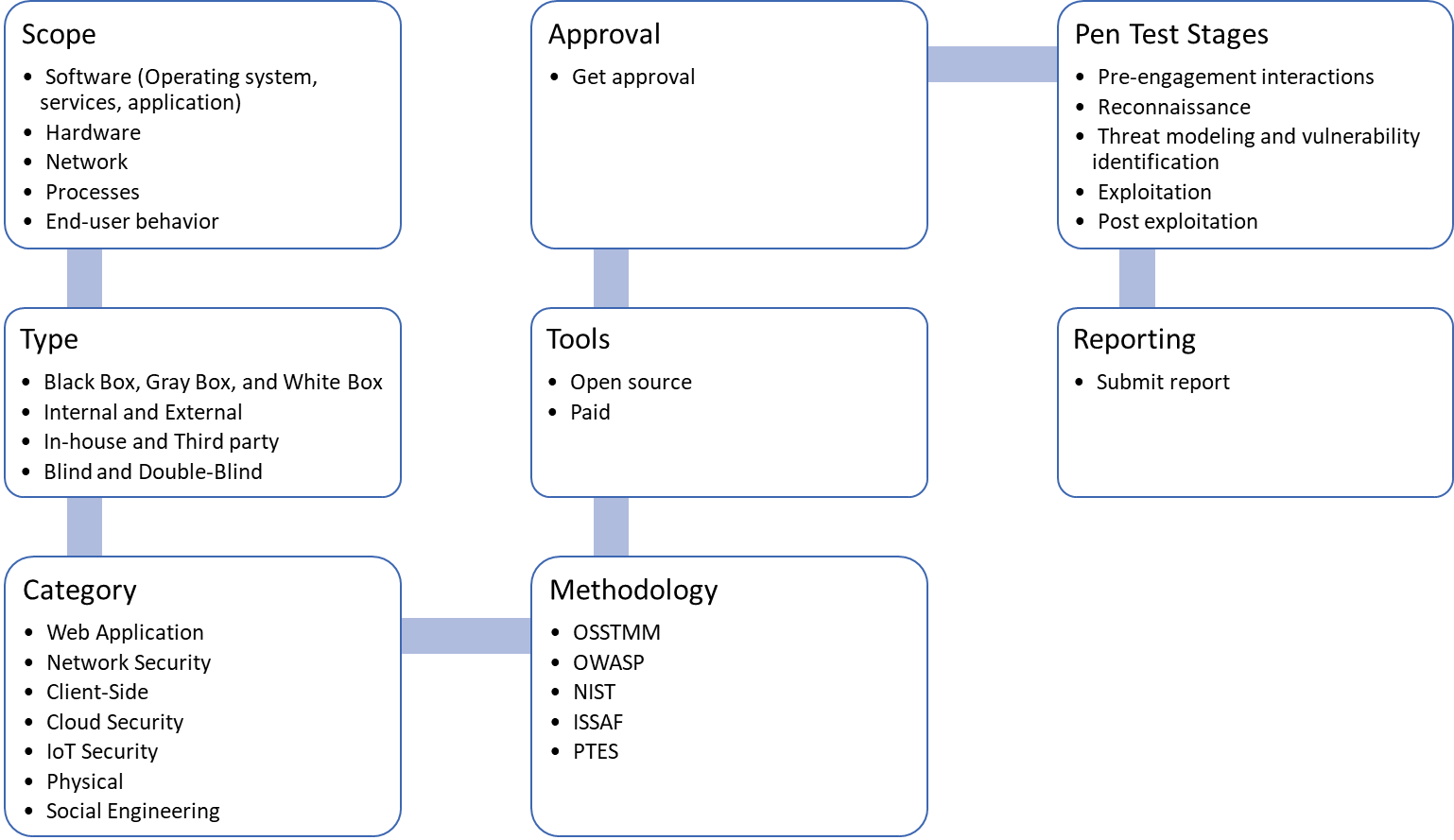


Figure : Pentesting Process

## Requirements:

Following hardware, software and tolls are required to do penetration testing.

### Hardware:

* + - Qty : 8
    - Core : 4
    - RAM : 16 GB
    - HDD : 500 GB

### Software:

### Tools:

# Secure Software Development Lifecycle (SSDLC):

Secure SDLC is a collection of best practices focused on adding security to the standard SDLC. Creating a secure SDLC process requires dedicated effort at each phase of the SDLC, from requirement gathering to deployment and maintenance. Secure SDLC requires a mind shift on the part of the development team, focusing on security at each phase of the project instead of just focusing on functionality.

With dedicated effort, security issues can be addressed in the SDLC pipeline well before deployment to production. This reduces the risk of finding security vulnerabilities in your app and works to minimize the impact when they are found.

Secure SDLC’s aim is not to completely eliminate traditional security checks, such as penetration tests, but rather to include security in the scope of developer responsibilities and empower them to build secure applications from the outset.

## Importance of Secure SDLC:

Secure SDLC is important because application security is important. The days of releasing a product into the wild and addressing bugs in subsequent patches are gone. Developers now need to be cognizant of potential security concerns at each step of the process. This requires integrating security into your SDLC in ways that were not needed before. As anyone can potentially gain access to your source code, you need to ensure that you are coding with potential vulnerabilities in mind. As such, having a robust and secure SDLC process is critical to ensuring your application is not subject to attacks by hackers and other nefarious users.

## Secure Software Development Life Cycle Processes & VAPT:

Secure software development life cycle processes incorporate security as a component of every phase of the SDLC. This is shown the in diagram below. The above process of vulnerability assessment and penetration testing must be done in SSDLC process as shown in the diagram.

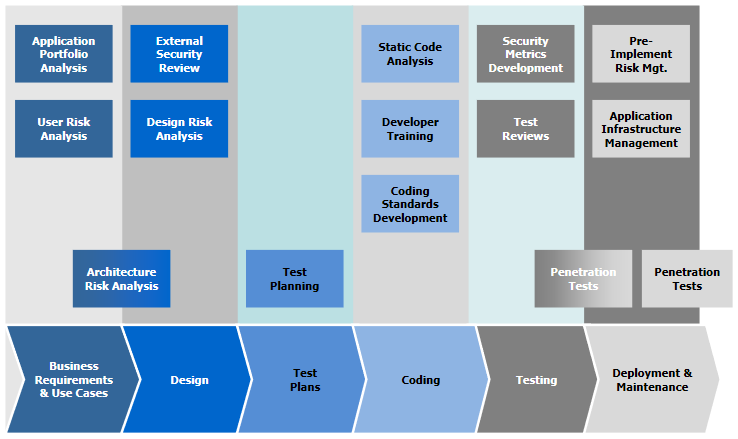


Figure : Secure SDLC