<https://www.greycampus.com/blog/information-security/penetration-testing-step-by-step-guide-stages-methods-and-application>

## What is Penetration Testing?

[Penetration testing is the art of finding vulnerabilities](https://www.greycampus.com/blog/information-security/five-deadliest-attacks-an-ethical-hacker-is-capable-of) and digging deep to find out how much a target can be compromised, in case of a legitimate attack. A penetration test will involve exploiting the network, servers, computers, firewalls, etc., to uncover vulnerabilities and highlight the practical risks involved with the identified vulnerabilities.

## Stages of Penetration Testing

Penetration testing can be broken down into multiple phases; this will vary depending on the organization and the type of test conducted– internal or external. Let’s discuss each phase:

### **1) Agreement phase:**

In this phase, there is a mutual agreement between the parties; the agreement covers high-level details- methods followed and the exploitation levels. The attacker cannot bring down the production server even if the testing has been done at non-peak hours. What if the attacker changes the data that has been contained in the database in production? This will unveil the vulnerabilities but at the cost of business. A non-disclosure agreement has to be signed between the parties before the test starts.

### **2) Planning and reconnaissance:**

In this phase, the attacker gathers as much information about the target as possible. The information can be IP addresses, domain details, mail servers, network topology, etc. An expert hacker will spend most of the time in this phase, this will help with further phases of the attack.

### **3) Scanning:**

This is the phase where the attacker will interact with the target with an aim to identify the vulnerabilities. An attacker will send probes to the target and records the response of the target to various inputs. This phase includes- scanning the network with various scanning tools, identification of open share drives, open FTP portals, services that are running, and much more. In case of a web application, the scanning part can be either dynamic or static. In static scanning, the application code is scanned by either a YTool or an expert application vulnerability analyst. The aim is to identify the vulnerable functions, libraries and logic implemented. In dynamic analysis, the tester will pass various inputs to the application and record the responses; various vulnerabilities like injection, cross-site scripting, remote code execution can be identified in this phase.

**Explore** [OWASP- Top 10 Vulnerabilities in web applications (updated for 2018)](https://www.greycampus.com/blog/information-security/owasp-top-vulnerabilities-in-web-applications)

### **4) Gaining Access:**

Once the vulnerabilities have been identified, the next step is to exploit the vulnerabilities with an aim to gain access to the target. The target can be a system, firewall, secured zone or server. Be aware that not all vulnerabilities will lead you to this stage. You need to identify the ones that are exploitable enough to provide you with access to the target.

### **5) Maintaining access:**

The next step is to ensure that the access is maintained; i.e., persistence. This is required to ensure that the access is maintained even if the system is rebooted, reset or modified. This kind of persistence is used by attackers who live in the system and gain knowledge about them over a period of time, and when the environment is suitable, they exploit.

### **6) Exploitation:**

This is the phase where the actual damage is done. An attacker will try to get the data, compromise the system, launch dos attacks, etc. Usually, this phase is controlled in penetration testing so as to ensure that the mayhem on the network is limited. This phase is modified in this way- a dummy flag is placed in the critical zone, may be in the database; the aim of the exploitation phase will be to get the flag. Revealing the contents of the flag will be enough to ensure practical exploitation of the network or data theft.

### **7) Evidence collection and report generation:**

Once the penetration test is complete, the final aim is to collect the evidence of the exploited vulnerabilities and report it to the executive management for review and action. Now, it is the management’s decision on how this risk has to be addressed. Whether they want to accept the risk, transfer it or ignore it (least likely option).

## Different Types and Methods of Penetration Testing

[Types of penetration testing](https://www.greycampus.com/opencampus/ethical-hacking/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:**

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.

### **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. Companies often hire 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.

### **1. Nessus**

Nessus is a network and web application [vulnerability scanner](https://www.greycampus.com/opencampus/ethical-hacking/hacking-methodology), it can perform different types of scans and help a penetration tester identify vulnerabilities. The attacker can then spend time in determining what can be exploited further. The free version of the tool is having some interesting features disabled. The full version is powerful and has a lot of features that will help during the scanning phase of the penetration test.

### **2. Dirbuster**

Dirbuster is a directory busting tool, this will help the attacker to find the directories that are present. The tool will take an input list and will help in testing their availability. This will allow for footprinting of the directory structure and find directories that will be difficult to find.

### **3. Metasploit**

Metasploit is an exploitation framework that has been packed with various capabilities. A skilled attacker can generate payloads, shellcodes, gain access, and perform privilege escalation attacks. The knowledge of python and ruby will be helpful since the framework uses them for most of the scripts.

### **4. Burp Suite**

This tool is specifically used for testing web applications. Let us assume that you have uncovered a test web application that is no longer used after production push. You can use this tool to dig deeper into the application and hunt vulnerabilities. The high severity vulnerabilities can be further exploited to move forward with the attack.

<https://www.softwaretestinghelp.com/getting-started-with-web-application-penetration-testing/>

#### Vulnerability Scanning or Pen Testing?

Vulnerability Scanning lets the user find out the known weaknesses in the application and defines methods to fix and improve the overall security of the application. It basically finds out if security patches are installed, whether the systems are properly configured to make attacks difficult.

Pen Tests mainly simulates real-time systems and helps the user find out if the system can be accessed by unauthorized users, if yes then what damage can be caused and to which data etc.

Hence, Vulnerability Scanning is a detective control method which suggests for ways to improve security program and ensure known weaknesses do not resurface, whereas pen test is a preventive control method which gives an overall view of the system’s existing security layer.

Though, both the methods have its importance, but it will depend on what really is expected as part of the testing.

As testers, it is imperative to be clear on the purpose of the testing before we jump into testing. If you are clear on the objective, you can very well define if you need to do a vulnerability scan or pen testing.

**Importance and the need for Web App Pen Testing:**

* Pentest Helps in identifying unknown vulnerabilities.
* Helps in checking the effectiveness of the overall security policies.
* Help in testing the components exposed publicly like firewalls, routers, and DNS.
* Lets user find out the most vulnerable route through which an attack can be made
* Helps in finding the loopholes which can lead to theft of sensitive data.

If you look at the current market demand, there has been a sharp increase in mobile usage, which is becoming a major potential for attacks. Accessing websites through mobiles are prone to more frequent attacks and hence compromising of data.

Penetration Testing thus becomes very important in ensuring we build a secure system which can be used by users without any worries of hacking or data loss.

### Web Penetration Testing Methodology

The methodology is nothing but a set of security industry guidelines on how the testing should be conducted. There are some well established and famous methodologies and standards which can be used for testing, but since each web application demands different types of test to be performed, testers can create their own methodologies by referring the standards available in the market.

Some of the Security Testing Methodologies and standards are –

* [**OWASP**](https://www.owasp.org/index.php/Main_Page) **(Open Web Application Security Project)**
* [**OSSTMM**](http://www.isecom.org/research/) **(Open Source Security Testing Methodology Manual)**
* **PTF** (Penetration Testing Framework)
* [**ISSAF**](http://cuchillac.net/archivos/pre_seguridad_pymes/2_hakeo_etico/lects/metodologia_oissg.pdf) **(Information Systems Security Assessment Framework)**
* [**PCI** **DSS**](https://www.pcisecuritystandards.org/pci_security/) (Payment Card Industry Data Security Standard)

**Test Scenarios:**

Listed below are some of the test scenarios which can be tested as part of **Web Application Penetration Testing (WAPT):**

1. Cross Site Scripting
2. SQL Injection
3. Broken authentication and session management
4. File Upload flaws
5. Caching Servers Attacks
6. Security Misconfigurations
7. Cross-Site Request Forgery
8. Password Cracking

Even though I have mentioned the list, testers should not blindly create their test methodology based on the above conventional standards.

Here’s an example to prove why I am saying so.

Consider you are asked to penetration test an eCommerce website, now give it a thought if all vulnerabilities of an eCommerce website can be identified using the conventional methods of OWASP like XSS, SQL injection etc.

The answer is a No because eCommerce works on a very different platform and technology when compared to other Websites. In order to make your pen testing for eCommerce website effective, testers should design a methodology involving flaws like Order Management, Coupon and Reward Management, Payment Gateway Integration and Content Management System Integration.

So, before you decide on the methodology, be very sure about what types of website are expected to be tested and which method will help in finding the maximum vulnerabilities.

### Types of Web Penetration Testing

Web applications can be penetration tested in 2 ways. Tests can be designed to simulate an inside or an outside attack.

**#1) Internal Penetration Testing –**

As the name suggests, the internal pen testing is done within the organization over the LAN, hence it includes testing web applications hosted on the intranet.

This helps in finding out if there could be vulnerabilities which exist within the corporate firewall.

We always believe attacks can happen only externally and many a time’s internal Pentest is overlooked or not given much importance.

Basically, it includes Malicious Employee Attacks by disgruntled employees or contractors who would have resigned but aware of the internal security policies and passwords, Social Engineering Attacks, Simulation of Phishing Attacks, and Attacks using User Privileges or misuse of an unlocked terminal.

Testing is mainly done by accessing the environment without proper credentials and identifying if an

**#2)** **External Penetration Testing –**

These are attacks done externally from outside the organization and include testing web applications hosted on the internet.

Testers behave like hackers who aren’t much aware of the internal system.

To simulate such attacks, testers are given the IP of the target system and not provided any other information. They are required to search and scan public web pages and find our information about target hosts and then compromise the found hosts.

Basically, it includes testing servers, firewalls, and IDS.

[**A1:2017-Injection**](https://www.owasp.org/index.php/Top_10-2017_A1-Injection)

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

[**A2:2017-Broken Authentication**](https://www.owasp.org/index.php/Top_10-2017_A2-Broken_Authentication)

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

[**A3:2017-Sensitive Data Exposure**](https://www.owasp.org/index.php/Top_10-2017_A3-Sensitive_Data_Exposure)

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

[**A4:2017-XML External Entities (XXE)**](https://www.owasp.org/index.php/Top_10-2017_A4-XML_External_Entities_(XXE))

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

[**A5:2017-Broken Access Control**](https://www.owasp.org/index.php/Top_10-2017_A5-Broken_Access_Control)

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

[**A6:2017-Security Misconfiguration**](https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration)

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

[**A7:2017-Cross-Site Scripting (XSS)**](https://www.owasp.org/index.php/Top_10-2017_A7-Cross-Site_Scripting_(XSS))

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

[**A8:2017-Insecure Deserialization**](https://www.owasp.org/index.php/Top_10-2017_A8-Insecure_Deserialization)

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

[**A9:2017-Using Components with Known Vulnerabilities**](https://www.owasp.org/index.php/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities)

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

[**A10:2017-Insufficient Logging&Monitoring**](https://www.owasp.org/index.php/Top_10-2017_A10-Insufficient_Logging%26Monitoring)

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Injection

Broken Authentication

Sensitive Data Exposure

XML External Entities (XXE)

Broken Access Control

Security Misconfiguration

Cross-Site Scripting (XSS)

Insecure Deserialization

Using Components with Known Vulnerabilities

Insufficient Logging&Monitoring