

**Red Team Testing**

**A complete guide of Penetration & Vulnerability testing**

4/5/2017

MagikMinds Software Services Pvt. Ltd.

Surendra Madugula

**Contents**

[**What is Red Team?** 3](#_Toc479349061)

[What is a red team exercise? 3](#_Toc479349062)

[What is a red team hacker? 3](#_Toc479349063)

[**Overview**: 4](#_Toc479349064)

[**Red Teaming approach**: 4](#_Toc479349065)

[**Red Team Methodology**: 5](#_Toc479349066)

[**An approach for Security Testing of Web Applications**: 7](#_Toc479349067)

[Security testing approach in detail: 7](#_Toc479349068)

[**Penetration testing**: 9](#_Toc479349069)

[**Why Penetration testing**? 9](#_Toc479349070)

[**Penetration Testing Types**: 9](#_Toc479349071)

[**Pen Testing Techniques**: 10](#_Toc479349072)

[1) Manual Penetration Test: 10](#_Toc479349073)

[Penetration testing sample test cases (Test scenarios): 10](#_Toc479349074)

[2) Automated penetration test tools: 12](#_Toc479349075)

[Examples of Free and Commercial Tools 12](#_Toc479349076)

[**Web Penetration Testing Methodology**: 12](#_Toc479349077)

[**Types of Web Penetration Testing**: 13](#_Toc479349078)

[**Web Pen Testing Approach**: 14](#_Toc479349079)

[1) Planning Phase (Before Testing): 14](#_Toc479349080)

[2) Attacks/Execution Phase (During Testing): 15](#_Toc479349081)

[3) Post Execution Phase (After Testing): 16](#_Toc479349082)

# **What is Red Team?**

**Red Teaming** is a process designed to detect network and system vulnerabilities and test security by taking an attacker-like approach to system/network/data access. This process is also called "ethical hacking" since its ultimate purpose is to enhance security.

## **What is a red team exercise?**

**A Red Team Exercise** is an all-out attempt to gain access to a system by any means necessary, and usually includes cyber penetration testing, physical breach, testing all phone lines for modem access, testing all wireless and RF systems present for potential wireless access, also testing employees through several scripted social engineering and phishing tests. These are real life exercises carried out by an elite small team of trained professionals that are hired to test the physical, cyber security, and social defenses of system.

## **What is a red team hacker?**

Penetration testers assess organization security, often unbeknownst to client staff. When used in a **hacking** context, a **red team** is a group of **white-hat hackers** that attack an organization's digital infrastructure as an attacker would to test the organization's defenses (often known as "**penetration testing**").

**White-hat hackers:**

A white hat hacker is a computer security specialist who breaks into protected systems and networks to test and asses their security. White hat hackers use their skills to improve security by exposing vulnerabilities before malicious hackers (known as black hat hackers) can detect and exploit them. Although the methods used are similar, if not identical, to those employed by malicious hackers, white hat hackers have permission to employ them against the organization that has hired them.

**Penetration testing:**

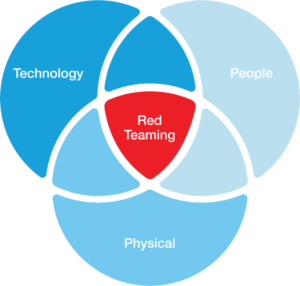
It’s the process to identify security **vulnerabilities** in an application by evaluating the system or network with various malicious techniques. Purpose of this test is to secure important data from outsiders like hackers who can have unauthorized access to system. Once vulnerability is identified it is used to exploit system to gain access to sensitive information.

**Vulnerabilities:**

The vulnerability is a terminology used to identify flaws in the system which can expose the system to security threats.

# **Overview:**

The objective of a red team test is to obtain a realistic level of risk and vulnerabilities against your **Technology**, **People** and **Physical/Facilities**.



1. **Technology** — Networks, applications, routers, switches, appliances, etc.
2. **People** — Staff, independent contractors, departments, business partners, etc.
3. **Physical** — Offices, warehouses, substations, data centers, buildings, etc.

Red teaming is a multi-blended attack involving several facets of social engineering, physical penetration testing, application penetration testing and network penetration testing, simultaneously. It’s aimed at revealing real-world opportunities for malicious insiders or bad actors to be able to compromise all aspects of your organization in such a way that allows for unauthorized virtual and/or physical access to sensitive information leading up to data breaches and full system/network compromise.

# **Red Teaming approach:**

Red Team Security’s web application penetration testing service utilizes a comprehensive, risk-based approach to manually identify critical application-centric vulnerabilities that exist on all in-scope applications.****

1. Information Gathering  
2. Threat Modeling  
3. Vulnerability Analysis  
4. Exploitation  
5. Post-Exploitation  
6. Reporting

**Information Gathering:**

Various methods including Google search are used to get target system data. One can also use web page source code analysis technique to get more info about the system, software and plugin versions. There are many free tools and services available in the market which can give you information like database or table names, DB versions, software versions, hardware used and various third party plugins used in the target system.

**Vulnerability Analysis:**

Based on the data collection one can find the security weakness in the target system. This helps penetration testers to launch attacks using identified entry points in the system.

**Actual Exploitation:**

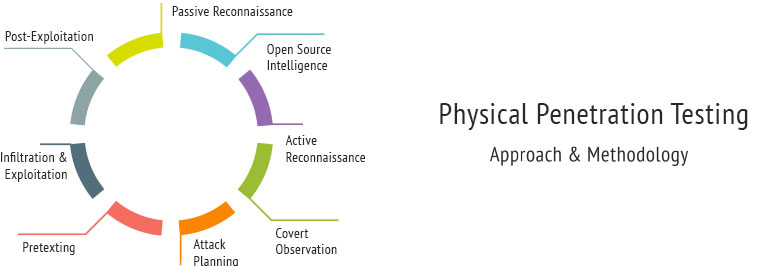
This is crucial step. It requires unique skills and techniques to launch attack on target system. Experienced penetration testers can use their skills to launch attack on the system.

**Result analysis and report preparation:**

After completion of penetration tests detailed reports are prepared for taking corrective actions. All identified vulnerabilities and recommended corrective methods are listed in these reports. You can customize vulnerability report format (HTML, XML, MS Word or PDF) as per your organization needs.

# **Red Team Methodology:**

Each and every web application penetration test is conducted consistently using globally accepted and industry standard frameworks. In order to ensure a sound and comprehensive penetration test, Red Team leverages industry standard frameworks as a foundation for carrying out penetration tests. At a minimum, the underlying framework is based on the NIST Special Publication 800 Series guidance and OSSTMM but goes beyond the initial framework itself.

****

**Passive Reconnaissance:**

The first phase in a physical penetration test is focused on collecting as much information as possible about the target. Passive Reconnaissance, aka Information Gathering, is one of the most critical steps of a physical pen test. This is done by using public tools, such as Google Earth. As a result, it is usually possible to learn a great deal about the target’s surroundings and environment.

**Open Source Intelligence:**

An important phase in a physical penetration test focuses on collecting information that is freely available. Open Source Intelligence Gathering can be quite telling about a target, its people and specifics about the environment. This is done by using of different set of public tools, such as social networks, job boards, etc. Through thorough analysis, it begins to paint a picture of the target and its primary operations.

**Active Reconnaissance:**

Active Reconnaissance in a physical penetration test generally involves gathering information offline. This often includes telephoning, emailing or otherwise directly querying target staff or vendors of the target for material not available or impossible to obtain through online means. The information obtained will be used to build a better plan as the process progresses.

**Covert Observation:**

Covert Observation is exactly what is sounds like. This often includes covert photography of the target up close to identify physical security controls and monitoring staff as they are coming and going.

**Attack Planning:**

By this time, the information collected in the previous phases are beginning to coalesce. Vulnerabilities, exit points, entrance points, cameras, guards, fences, company technology, staff members and other relevant information are used to begin planning an attack.

**Pretexting:**

Planning and intelligence gathered by various means by now have morphed into a plan of attack including. Pretexting involves setting the plan into action and ensuring the team’s equipment, transportation and personnel are synchronized and ready to execute.

**Infiltration, Exploitation & Post-Exploitation:**

During these phases, the team carries out the plan by exploiting vulnerabilities discovered using information and intelligence captured during the earlier phases of the assessment. Post-exploitation involves penetrating further into the environment and setting up to maintain a persistent backdoor.

# **An approach for Security Testing of Web Applications:**

Red Team testing approaches the security vulnerabilities of an application.

**Security testing** is the process that determines that **confidential data stays confidential** (i.e. it is not exposed to individuals/ entities for which it is not meant) and users can perform only those tasks that they are authorized to perform (e.g. a user should not be able to deny the functionality of the web site to other users, a user should not be able to change the functionality of the web application in an unintended way etc.).

**What is “Vulnerability”?**  
This is a weakness in the web application. The cause of such a “weakness” can be bugs in the application, an injection (SQL/ script code) or the presence of viruses.  
  
**What is “URL manipulation”?**  
Some web applications communicate additional information between the client (browser) and the server in the URL. Changing some information in the URL may sometimes lead to unintended behavior by the server.

**What is “SQL injection”?**  
This is the process of inserting SQL statements through the web application user interface into some query that is then executed by the server.

**What is “XSS (Cross Site Scripting)”?**  
When a user inserts HTML/ client-side script in the user interface of a web application and this insertion is visible to other users, it is called XSS.

**What is “Spoofing”?**  
The creation of hoax look-alike websites or emails is called spoofing.

# **Security testing approach in detail**:

To perform a useful security test of a web application, the security tester should have good knowledge of the HTTP protocol. It is important to have an understanding of how the client (browser) and the server communicate using HTTP. Additionally, the tester should at least know the basics of SQL injection and XSS. Hopefully, the number of security defects present in the web application will not be high. However, being able to accurately describe the security defects with all the required details to all concerned will help.

**1. Password cracking:**

The security testing on a web application can be kicked off by “password cracking”. In order to log in to the private areas of the application, one can either guess a username/ password or use some password cracker tool for the same. Lists of common usernames and passwords are available along with open source password crackers. If the web application does not enforce a complex password (e.g. with alphabets, number and special characters, with at least a required number of characters), it may not take very long to crack the username and password.

If username or password is stored in cookies without encrypting, attacker can use different methods to steal the cookies and then information stored in the cookies like username and password.

**2. URL manipulation through HTTP GET methods:**

The tester should check if the application passes important information in the query string. This happens when the application uses the HTTP GET method to pass information between the client and the server. The information is passed in parameters in the query string. The tester can modify a parameter value in the query string to check if the server accepts it.

Via HTTP GET request user information is passed to server for authentication or fetching data. Attacker can manipulate every input variable passed from this GET request to server in order to get the required information or to corrupt the data. In such conditions, any unusual behavior by application or web server is the doorway for the attacker to get into the application.

**3. SQL Injection:**

The next thing that should be checked is SQL injection. Entering a single quote (‘) in any textbox should be rejected by the application. Instead, if the tester encounters a database error, it means that the user input is inserted in some query which is then executed by the application. In such a case, the application is vulnerable to SQL injection.

SQL injection attacks are very critical as attacker can get vital information from server database. To check SQL injection entry points into your web application, find out code from your code base where direct MySQL queries are executed on database by accepting some user inputs.

If user input data is crafted in SQL queries to query the database, attacker can inject SQL statements or part of SQL statements as user inputs to extract vital information from database. Even if attacker is successful to crash the application, from the SQL query error shown on browser, attacker can get the information they are looking for. Special characters from user inputs should be handled/escaped properly in such cases.

**4. Cross Site Scripting (XSS):**

The tester should additionally check the web application for XSS (Cross site scripting). Any HTML e.g. <HTML> or any script e.g. <SCRIPT> should not be accepted by the application. If it is, the application can be prone to an attack by Cross Site Scripting.

Attacker can use this method to execute malicious script or URL on victim’s browser. Using cross-site scripting, attacker can use scripts like JavaScript to steal user cookies and information stored in the cookies.

# **Penetration testing:**

**What is Penetration Testing?**  
It’s the process to identify security vulnerabilities in an application by evaluating the system or network with various malicious techniques. Purpose of this test is to secure important data from outsiders like hackers who can have unauthorized access to system. Once vulnerability is identified it is used to exploit system to the gain access to sensitive information.

**Causes of vulnerabilities:**  
– Design and development errors  
– Poor system configuration  
– Human errors

# **Why Penetration testing?**

– Financial data must be secured while transferring between different systems  
– Many clients are asking for pen testing as part of the software release cycle  
– To secure user data  
– To find security vulnerabilities in an application

It’s very important for any organization to identify security issues present in internal network and computers. Using this information organization can plan defense against any hacking attempt. User privacy and data security are the biggest concerns nowadays. Imagine if any hacker manages to get user details of social networking site like Facebook. Organization can face legal issues due to a small loophole left in a software system. Hence big organizations are looking for PCI (Payment Card Industry) compliance certifications before doing any business with third party clients.

**What should be tested?**  
– Software  
– Hardware  
– Network  
– Process

# **Penetration Testing Types:**

**1) Social Engineering:**

Human errors are the main causes of security vulnerability. Security standards and policies should be followed by all staff members to avoid social engineering penetration attempt. Example of these standards include not to mention any sensitive information in email or phone communication. Security audits can be conducted to identify and correct process flaws.

**2) Application Security Testing:**

Using software methods one can verify if the system is exposed to security vulnerabilities.

**3) Physical Penetration Test:**

Strong physical security methods are applied to protect sensitive data. This is generally useful in military and government facilities. All physical network devices and access points are tested for possibilities of any security breach.

**Pen Testing Techniques:**  
1) Manual penetration test  
2) Using automated penetration test tools  
3) Combination of both manual and automated process  
The third process is more common to identify all kinds of vulnerabilities.

# **1) Manual Penetration Test:**

It’s difficult to find all vulnerabilities using automated tools. There are some vulnerabilities which can be identified by manual scan only. Penetration testers can perform better attacks on application based on their skills and knowledge of system being penetrated. The methods like social engineering can be done by humans only. Manual checking includes design, business logic as well as code verification.

**Penetration Test Process:**  
Let’s discuss the actual process followed by test agencies or penetration testers. Identifying vulnerabilities present in system is the first major step in this process. Corrective action is taken on these vulnerability and same penetration tests are repeated until system is negative to all those tests.

**We can categorize this process in following methods:**

**1.Data collection**

**2.Vulnerability assessment**

**3.Actual exploit**

**4.Result analysis and Report preparation**

**These are explained in detail at “Red team approaches” section.**

## **Penetration testing sample test cases (Test scenarios):**

Remember this is not functional testing. In Pentest, goal is to find security holes in the system. Below are some generic test cases and not necessarily applicable for all applications.

**1)** Check if web application can identify spam attacks on contact forms used in the website.  
**2)** Proxy server – Check if network traffic is monitored by proxy appliances. Proxy server make it difficult for hackers to get internal details of the network thus protecting the system from external attacks.  
**3)** Spam email filters – Verify if incoming and outgoing email traffic is filtered and unsolicited emails are blocked. Many email clients come with in-build spam filters which needs to be configured as per your

needs. These configuration rules can be applied on email headers, subject or body.  
**4)** Firewall – Make sure entire network or computers are protected with Firewall. Firewall can be a software or hardware to block unauthorized access to system. Firewall can prevent sending data outside the network without your permission.  
**5)** Try to exploit all servers, desktop systems, printers and network devices.  
**6)** Verify that all usernames and passwords are encrypted and transferred over secured connection like https.  
**7)** Verify information stored in [website cookies](http://www.softwaretestinghelp.com/website-cookie-testing-test-cases/). It should not be in readable format.  
**8)** Verify previously found vulnerabilities to check if the fix is working.  
**9)** Verify if there is no open port in network.  
**10)** Verify all telephone devices.  
**11)** Verify WIFI network security.  
**12)** Verify all HTTP methods. PUT and Delete methods should not be enabled on web server .  
**13)** Password should be at least 8 character long containing at least one number and one special character.  
**14)** Username should not be like “admin” or “administrator”.  
**15)** Application login page should be locked upon few unsuccessful login attempts.  
**16)** Error messages should be generic and should not mention specific error details like “Invalid username” or “Invalid password”.  
**17)** Verify if special characters, html tags and scripts are handled properly as an input value.  
**18)** Internal system details should not be revealed in any of the error or alert messages.  
**19)** Custom error messages should be displayed to end user in case of web page crash.  
**20)** Verify use of registry entries. Sensitive information should not be kept in registry.  
**21)** All files must be scanned before uploading to server.  
**22)** Sensitive data should not be passed in URLs while communicating with different internal modules of the web application.  
**23)** There should not be any hard coded username or password in the system.  
**24)** Verify all input fields with long input string with and without spaces.  
**25)** Verify if reset password functionality is secure.  
**26)** Verify application for[SQL Injection](http://www.softwaretestinghelp.com/sql-injection-%E2%80%93-how-to-test-application-for-sql-injection-attacks/).  
**27)** Verify application for [Cross Site Scripting](http://www.softwaretestinghelp.com/security-testing-of-web-applications/).  
**28)** Important input validations should be done at server side instead of JavaScript checks at client side.  
**29)** Critical resources in the system should be available to authorized persons and services only.  
**30)** All access logs should be maintained with proper access permissions.  
**31)** Verify user session ends upon log off.  
**32)** Verify that directory browsing is disabled on server.  
**33)** Verify that all applications and database versions are up to date.  
**34)** Verify url manipulation to check if web application is not showing any unwanted information.  
**35)** Verify memory leak and buffer overflow.  
**36)** Verify if incoming network traffic is scanned to find Trojan attacks.  
**37)** Verify if system is safe from Brute Force Attacks – a trial and error method to find sensitive information like passwords.  
**38)** Verify if system or network is secured from DoS (denial-of-service) attacks. Hacker can target network or single computer with continuous requests due to which resources on target system gets overloaded resulting in denial of service for legit requests.

# **2) Automated penetration test tools:**

Automated tools can be used to identify some standard vulnerability present in an application. Pentest tools scan code to check if there is malicious code present which can lead to potential security breach. Pentest tools can verify security loopholes present in the system like data encryption techniques and hard coded values like username and password.

**Criteria to select the best penetration tool:**  
– It should be easy to deploy, configure and use  
– It should scan your system easily  
– It should categorize vulnerabilities based on severity that needs immediate fix  
– It should be able to automate verification of vulnerabilities  
– It should re-verify exploits found previously  
– It should generate detailed vulnerability reports and logs

# **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 several 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 –

**1.**[**OWASP**](https://www.owasp.org/index.php/Main_Page) (Open Web Application Security Project)

**2.**[**OSSTMM**](http://www.isecom.org/research/) (Open Source Security Testing Methodology Manual)

**3.PTF** (Penetration Testing Framework)

**4.**[**ISSAF**](http://cuchillac.net/archivos/pre_seguridad_pymes/2_hakeo_etico/lects/metodologia_oissg.pdf) (Information Systems Security Assessment Framework)

**5.**[**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 we have mentioned the list, testers should not blindly create their test methodology based on 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. 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 on 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 any.

**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.

# **Web Pen Testing Approach:**

It can be conducted in 3 phases:



## **1) Planning Phase (Before Testing):**

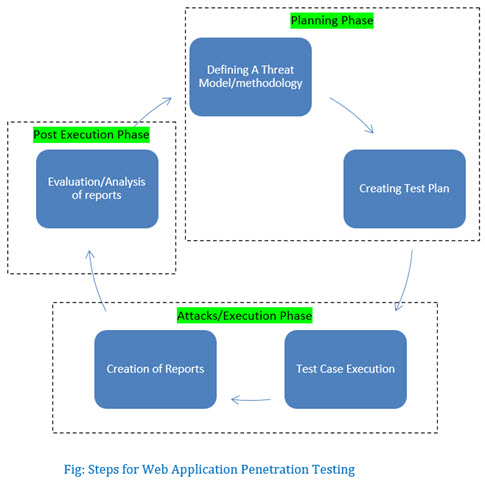
Before testing starts, it is advisable to plan what types of testing will be performed, how the testing will be performed, determine if QA needs any additional access to tools etc.

* **Scope definition –** This is same as our functional testing where we define the scope of our testing before starting our test efforts.
* **Availability of Documentation to Testers –** Ensure Testers have all the required documents like documents detailing the web architecture, integration points, web services integration etc. The tester should be aware of the HTTP/HTTPS protocol basics and know about the Web Application Architecture, traffic interception ways.
* **Determining the Success Criteria –** Unlike our functional test cases, where we can derive expected results from user requirements/functional requirements, pen testing works on a different model. The Success criteria or the test case passing criteria needs to be defined and approved.
* **Reviewing the test results from the Previous Testing –** If prior testing was ever done, it is good to review the test results to understand what vulnerabilities existed in the past and what remediation was taken to resolve. This always gives a better picture to the testers.
* **Understanding environment –**Testers should gain knowledge about the environment before starting testing. This step should ensure to give them an understanding on firewalls, or other security protocols which would be required to be disabled to perform the testing. Browser to be tested should be converted into an attack platform, usually done by changing proxies.

## **2) Attacks/Execution Phase (During Testing):**

Web Penetration testing can be done from any location, given the fact that there shouldn’t be restrictions on ports and services by the internet provider.

* **Ensure to run a test with different user roles –** Testers should ensure to run tests with users having different roles since the system may behave differently with respect to users having different privilege.
* **Awareness on how to handle Post-Exploitation –** Testers must follow the Success Criteria defined as part of Phase 1 to report any exploitation, also they should follow the defined process of reporting vulnerabilities found during testing. This step mainly involves the tester to find out what needs to be done after they have found that the system has been compromised.
* **Generation of Test Reports –** Any Testing done without proper reporting doesn’t help organization much, same is the case with penetration testing of web applications. To ensure the test results are properly shared with all stakeholders, testers should create proper reports with details on vulnerabilities found, the methodology used for testing, severity and the location of the problem found.



## **3) Post Execution Phase (After Testing):**

Once the testing is complete and test reports shared with all concerned teams, the following list should be worked upon by all-

* **Suggest remediation –** Pen Testing shouldn’t just end by identifying vulnerabilities. The concerned team including a QA member should review the findings reported by Testers and then discuss the remediation.
* **Retest Vulnerabilities –** After the remediation is taken and implemented, testers should retest to ensure that the fixed vulnerabilities did not appear as part of their retesting.
* **Cleanup –**As part of the Pentest, testers make changes to the proxy settings, so clean up should be done and all changes reverted.

**Examples of Free and Commercial Tools** –  
**Open Source:** [Nmap](http://nmap.org/), [Nessus](http://www.nessus.org/), [Metasploit](http://www.metasploit.com/" \t "_blank" \o "Metasploit), [Wireshark](http://www.wireshark.org/), [OpenSSL](http://www.openssl.org/), [Cain & Abel](http://www.oxid.it/cain.html), THC Hydra, [w3af](http://w3af.sourceforge.net/),ZAP, Acunetix, NetSparker

**Commercial services**: [Pure Hacking](http://www.purehacking.com/), [Torrid Networks](http://www.torridnetworks.com/), [SecPoint](http://www.secpoint.com/" \t "_blank" \o "SecPoint), [Veracode](http://www.veracode.com/" \t "_blank" \o "Veracode)

We have **37** Powerful Penetration Testing **Tools** for Every Penetration Tester.

[1) Metasploit](http://www.softwaretestinghelp.com/penetration-testing-tools/#1_Metasploit)

[2) Wireshark](http://www.softwaretestinghelp.com/penetration-testing-tools/#2_Wireshark)

[3) w3af](http://www.softwaretestinghelp.com/penetration-testing-tools/#3_w3af)

[4) CORE Impact](http://www.softwaretestinghelp.com/penetration-testing-tools/#4_CORE_Impact)

[5) Back Track](http://www.softwaretestinghelp.com/penetration-testing-tools/#5_Back_Track)

[6) Netsparker](http://www.softwaretestinghelp.com/penetration-testing-tools/#6_Netsparker)

[7) Nessus](http://www.softwaretestinghelp.com/penetration-testing-tools/#7_Nessus)

[8) Burpsuite](http://www.softwaretestinghelp.com/penetration-testing-tools/#8_Burpsuite)

[9) Cain & Abel](http://www.softwaretestinghelp.com/penetration-testing-tools/#9_Cain_Abel)

[10) Zed Attack Proxy (ZAP)](http://www.softwaretestinghelp.com/penetration-testing-tools/#10_Zed_Attack_Proxy_ZAP)

[11) Acunetix](http://www.softwaretestinghelp.com/penetration-testing-tools/#11_Acunetix)

[12) John The Ripper](http://www.softwaretestinghelp.com/penetration-testing-tools/#12_John_The_Ripper)

[13) Retina](http://www.softwaretestinghelp.com/penetration-testing-tools/#13Retina)

[14) Sqlmap](http://www.softwaretestinghelp.com/penetration-testing-tools/#14_Sqlmap)

[15) Canvas](http://www.softwaretestinghelp.com/penetration-testing-tools/#15_Canvas)

[16) Social Engineer Toolkit](http://www.softwaretestinghelp.com/penetration-testing-tools/#16_Social_Engineer_Toolkit)

[17) Sqlninja](http://www.softwaretestinghelp.com/penetration-testing-tools/#17_Sqlninja)

[18) Nmap](http://www.softwaretestinghelp.com/penetration-testing-tools/#18_Nmap)

[19) BeEF](http://www.softwaretestinghelp.com/penetration-testing-tools/#19_BeEF)

[20) Dradis](http://www.softwaretestinghelp.com/penetration-testing-tools/#20_Dradis)

**21)Ettercap**

**22)Hydra**

**23)Veracode**

**24)Satan**

**25)Shodan**

**26)Aircracker-ng**

**27)Arachni**

**28)PunkSpider**

**29)IBM AppScan**

**30)Nagios**

**31)Nikto**

**32)WebScarabNG**

**33)Maltego**

**34)IronWASP**

**35)HconSTF**

**36)OpenVAS**

**37)SecuniaPSI**

**Limitations of Pentest tools:** Sometimes these tools can flag false positive output which results in spending more developer time on analyzing such vulnerabilities which are not present.