Security testing methodologies

Various open source methodologies have been created to address the security assessment's needs. Using these assessment methodologies, one can strategically accomplish the time-critical and challenging task of assessing the system's security regardless of its size and complexity. Some methodologies focus on the technical aspect of security testing, while others focus on managerial criteria, and very few address both sides. The basic idea behind formalizing these methodologies with your assessment is to execute different types of tests step-by-step in order to accurately judge the security posture of a system.

Therefore, you will be introduced to several well-known security assessment methodologies that provide you with an extended view of the assessing network and application security by highlighting their key features and benefits. These include the following:

- Open Source Security Testing Methodology Manual
- Information Systems Security Assessment Framework
- Open Web Application Security Project Testing Guide
- Web Application Security Consortium Threat Classification
- Penetration Testing Execution Standard

All of these testing frameworks and methodologies will assist security professionals choose the best strategy that adheres to their client's requirements. The first two provide you with general guidelines and methods of security testing for almost any type of information asset. The testing frameworks provided by **Open Web Application Security Project** (**OWASP**) and **Web Application Security Consortium** (**WASC**) primarily deal with the assessment of application security. **Penetration Testing Execution Standard** (**PTES**) will provide you with guidance on all types of penetration testing efforts. It is, however, important to note that security is an on-going process in itself and a penetration test is a snapshot that details the security posture at the time of the test. Any minor change in the target environment may affect the entire process of security testing and could introduce errors in the final results. Additionally, adapting any single methodology does not necessarily provide you with a complete picture of the risk assessment process. It is left to the security auditor to select the best strategy that could address the target testing criteria.

There are many security testing methodologies; choosing the best one requires a careful selection process through which one can determine the cost and effectiveness of the assessment. Thus, determining the right assessment strategy depends on several factors, including the technical details provided about the target environment and resource availability, pentester's knowledge, business objectives, and regulatory concerns. From a business standpoint, efficiency and cost control is of extreme importance. Each of the following testing methodologies have very detailed and well-written documentation at their respective sites. We provide a brief summary of each, but to truly understand how they work in detail, you need to go to their respective websites and carefully study the documentation and implementation details provided by their creators.

Open Source Security Testing Methodology Manual (OSSTMM)

Open Source Security Testing Methodology Manual (**OSSTMM**) (http://www.isecom.org/research/osstmm.html) is a recognized international standard created by Pete Herzog and developed by ISECOM for security testing and analysis. It's being used by many organizations in their day-to-day assessment cycle. From a technical perspective, its methodology is divided into four key groups—**scope** , **channel** , **index** , and **vector** . The scope defines a process of collecting information on all assets that operate in the target environment. A channel determines the type of communication and interaction with these assets, which can be physical, spectrum, and communication. All of these channels depict a unique set of security components that must be tested and verified during the assessment period. These components are comprised of physical security, human psychology, data networks, wireless communication medium, and telecommunication. The index is a method that is used to classify target assets that correspond to their particular identifications, such as MAC Address and IP Address. At the end, a vector concludes the direction through which an auditor can assess and analyze each functional asset. The whole process initiates a technical roadmap that evaluates the target environment thoroughly and is known as **audit scope**.

There are different forms of security testing that have been classified under the OSSTMM methodology, and their organization is presented within six standard security test types:

- **Blind**: Blind testing does not require any prior knowledge about the target system. However, the target is informed before the execution of an audit scope. Ethical hacking and war gaming are examples of blind type testing. This kind of testing is also widely accepted because of its ethical vision of informing a target in advance.
- **Double blind**: In double blind testing, an auditor neither requires any knowledge about the target system, nor is the target informed before the test execution. Black box auditing and penetration testing are examples of double blind testing. Most of the security assessments today are carried out using this strategy, thus putting a real challenge for the auditors to select the best of breed tools and techniques in order to achieve their required goal.
- **Gray box**: In gray box testing, an auditor holds limited knowledge about the target system and the target is also informed before the test is executed. Vulnerability assessment is one of the basic examples of gray box testing.
- **Double gray box**: Double gray box testing works in a way that is similar to gray box testing, except that the time frame for an audit is defined and there are no channels and vectors being tested. White box audit is an example of double gray box testing.
- **Tandem**: In tandem testing, the auditor holds minimum knowledge to assess the target system and the target is also notified in advance, before the test is executed. Note that tandem testing is conducted thoroughly. Crystal box and in-house audit are examples of tandem testing.
- **Reversal**: In reversal testing, an auditor holds full knowledge of the target system and the target will never be informed of how and when the test will be conducted.

The technical assessment framework provided by OSSTMM is flexible and capable of deriving certain test cases that are logically divided into five security components of three consecutive channels, as mentioned previously. These test cases generally examine the target by assessing its access control security, process security, data controls, physical location, perimeter protection, security awareness level, trust level, fraud control protection, and many other procedures. The overall testing procedures focus on what is to be tested, how it should be tested, what tactics should be applied before, during, and after the test, and how to interpret and correlate the final results. Capturing the current state of the protection of a target system is considerably useful and invaluable. Thus, the OSSTMM methodology has introduced this terminology in the form of **RAV** (**Risk Assessment Values**). The basic function of RAV is to analyze the test results and compute the actual security value based on three factors, which are operational security, loss controls, and limitations. This final security value is known as **RAV score** . By using RAV score, an auditor can easily extract and define the milestones based on the current security posture to accomplish better protection. From a business perspective, RAV can optimize the amount of investment required on security and might help you with the justification of investing in more effective security solutions.

Key features and benefits

The following are the key features and benefits of OSSTMM:

- Practicing the OSSTMM methodology substantially reduces the occurrence of false negatives and false positives and provides reproducible security measurements.
- The framework is adaptable to many types of security tests, such as penetration testing, white box audit, vulnerability assessment, and so forth.
- It ensures that the assessment should be carried out thoroughly and the results are collected in a consistent, quantifiable, and reliable manner.
- The methodology itself follows a process of four individually connected phases, namely, definition phase, information phase, regulatory phase, and controls test phase. Each of these obtains, assesses, and verifies the information regarding the target environment.
- RAV calculates the actual security value based on operational security, loss controls, and limitations. The given output, known as the RAV score, represents the current state of target security.
- Formalizing an assessment report using the **Security Test Audit Report** (**STAR**) template can be advantageous to management as well as the technical team when reviewing the testing objectives, risk assessment values, and the output of each test phase.
- The methodology is regularly updated with new trends of security testing, regulations, and ethical concerns.
- The OSSTMM process can be coordinated with industry regulations, business policy, and government legislations. Additionally, a certified audit can also be eligible for accreditation from **ISECOM** (**Institute for Security and Open Methodologies**) directly.

Information Systems Security Assessment Framework (ISSAF)

Information Systems Security Assessment Framework (**ISSAF**) (www.oissg.org/issaf) is another open source security testing and analysis framework. Its framework has been categorized into several domains to address the security assessment in a logical order. Each of these domains assesses different parts of a target system and provides field inputs for the successful security engagement. By integrating its framework into a regular business life cycle, it may provide the accuracy, completeness, and efficiency required to fulfill an organization's security testing requirements. ISSAF was developed to focus on two areas of security testing—technical and managerial. The technical side establishes the core set of rules and procedures to follow and create an adequate security assessment process, while the managerial side accomplishes engagement with the management and the best practices that should be followed throughout the testing process. It should be remembered that ISSAF defines the assessment as a process instead of an audit. As auditing requires a more established body to proclaim the necessary standards, its assessment framework does include the planning, assessment, treatment, accreditation, and maintenance phases. Each of these phases holds generic guidelines that are effective and flexible for any organizational structure.

The output is a combination of operational activities, security initiatives, and a complete listing of vulnerabilities that might exist in the target environment. The assessment process chooses the shortest path to reach the test deadline by analyzing its target against critical vulnerabilities that can be exploited with minimum effort.

ISSAF contains a rich set of technical assessment baselines to test the number of different technologies and processes. However, this has introduced another problem of maintenance to keep updating the framework in order to reflect new or updated technology assessment criteria. When compared to the OSSTMM methodology, these obsolescence issues affect the OSSTMM less, because the auditor is able to use the same methodology over the number of security engagements using a different set of tools and techniques. On the other hand, ISSAF also claims to be a broad framework with up-to-date information on security tools, best practices, and administrative concerns to complement the security assessment program. It can also be aligned with OSSTMM or any other similar testing methodology, thus combining the strengths of each other.

Key features and benefits

The following are the key features and benefits of ISSAF:

- ISSAF provides you with a high value proposition to secure the infrastructure by assessing the existing security controls against critical vulnerabilities.
- It addresses different key areas of information security. These include risk assessment, business structure and management, controls assessment, engagement management, security policies development, and general best practices.
- ISSAF penetration testing methodology examines the security of a network, system, or application. The framework can transparently focus on target-specific technology that may involve routers, switches, firewalls, intrusion detection and prevention systems, storage area networks, virtual private networks, various operation systems, web application servers, databases, and so forth.
- It bridges the gap between the technical and managerial view of security testing by implementing the necessary controls to handle both areas.
- It enables the management to understand the existing risks that float over an organization's perimeter defenses and reduces them proactively by identifying the vulnerabilities that may affect the business integrity.

> **Note**
>
> OSSTMM and ISSAF can be used in combination with each other to assess the security of an enterprise environment.

Open Web Application Security Project (OWASP)

The **Open Web Application Security Project** (**OWASP**) open community brings its **top 10 project** forward to increase the awareness of application security. The project provides you with a necessary foundation to integrate security through secure coding principles and practices. OWASP also provides you with a wonderful testing guide as part of the OWASP Testing Project (https://www.owasp.org/index.php/OWASP_Testing_Project) that should be carefully reviewed to determine if this framework can assist you in your efforts.

The OWASP top 10 project categorizes the application security risks by evaluating the top attack vectors and security weaknesses in relation to their technical and business impact. While assessing the application, each of these risks demonstrates a generic attack method that is independent of the technology or platform being used. It also provides you with specific instructions on how to test, verify, and remediate each vulnerable part of an application. The OWASP top 10 mainly focuses on the high risk problem areas rather than addressing all the issues that surround the web application's security. However, some essential guidelines are available in the OWASP community for developers and security auditors to effectively manage the security of web applications:

- **The Testing Guide**: https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Content
- **The Developer's Guide**: www.owasp.org/index.php/Guide
- **The Code Review Guide**: www.owasp.org/index.php/Category:OWASP_Code_Review_Project

The OWASP top 10 changes on a year-to-year basis. For detailed information, visit the project's website at https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

Key features and benefits

The following are the key features and benefits of OWASP:

- Testing web applications against OWASP top ten security risks ensures that the most common attacks and weaknesses are avoided and the confidentiality, integrity, and availability of an application is maintained.
- The OWASP community has developed a number of security tools that focus on the automated and manual web application tests. A few of these tools are WebScarab, Wapiti, JBroFuzz, and SQLiX, which are also available under the Kali Linux operating system.
- When considering the security assessment of web infrastructure, the OWASP Testing Guide provides you with technology-specific assessment details; for instance, the testing of Oracle is approached differently than MySQL. Such a guide provides you with a wider and more collaborative look at multiple technologies, which helps an auditor choose the best-suited procedure for testing.
- It encourages the secure coding practices for developers by integrating security tests at each stage of development. This will ensure that the production application is robust, error-free, and secure.
- It provides industry-wide acceptance and visibility. The top ten security risks can also be aligned with other web application security assessment standards, thus helping you achieve more than one standard at a time with a little more effort.

Web Application Security Consortium Threat Classification (WASC-TC)

Identifying the application's security risks requires a thorough and rigorous testing procedure, which can be followed throughout the development's life cycle. WASC threat classification is another such open standard to assess the security of web applications. Similar to the OWASP standard, it is also classified into a number of attacks and weaknesses but addresses them in a much deeper fashion. Practicing this black art for identification and verification of threats that are hanging over the web application requires standard terminology to be followed, which can quickly adapt to the technology environment. This is where the WASC-TC comes in very handy. The overall standard is presented in three different views to help developers and security auditors understand the vision of web application security threats:

- **Enumeration view**: This view is dedicated to providing the basis for web application attacks and weaknesses. Each of these attacks and weaknesses have been discussed individually with its concise definition, type, and examples of multiple programming platforms. Additionally, it is in line with its unique identifier, which can be useful for referencing. A total of 49 attacks and weaknesses are collated with a static WASC-ID number (1 to 49). Note that this numeric representation does not focus on the risk severity but serves the purpose of referencing instead.
- **Development view**: The development view takes the developer's panorama forward by combining the set of attacks and weaknesses into vulnerabilities, which are likely to occur at any of three consecutive development phases. This could be a design, implementation, or deployment phase. The design vulnerabilities are introduced when the application's requirements do not fulfill the security at the initial stage of requirement gathering. The implementation vulnerabilities occur due to insecure coding principles and practices. The deployment vulnerabilities are the result of the misconfiguration of the application, web server, and other external systems. Thus, the view broadens the scope for its integration into a regular development life cycle as a part of best practices.
- **Taxonomy cross-reference view**: Referring to a cross-reference view of multiple web application security standards can help auditors and developers map the terminology presented in one standard with another. With a little more effort, the same facility can also assist you in achieving multiple standard compliances at the same time. However, each application's security standard defines it own criteria to assess the applications from different angles and measures their associated risks in general. Thus, each standard requires different efforts to be made to scale up the calculation for risks and their severity levels. The WASC-TC attacks and weaknesses presented in this category are mapped with OWASP top 10, Mitre's **Common Weakness Enumeration** (**CWE**), Mitre's **Common Attack Pattern Enumeration and Classification** (**CAPEC**), and SANS-CWE top 25 list.

Note

More details regarding Mitre's CWE can be found at https://cwe.mitre.org/.

More information regarding Mitre's CAPEC can be found at http://capec.mitre.org/.

SANS-CWE top 25 list can be found at http://www.sans.org/top25-software-errors/.

More details regarding WASC-TC and its views can be found at http://projects.webappsec.org/Threat-Classification.

Key features and benefits

The following are the key features and benefits of the WASC-TC:

- WASC-TC provides you with in-depth knowledge to assess the web application environment against the most common attacks and weaknesses.
- The attacks and weaknesses presented by WASC-TC can be used to test and verify any web application platform using a combination of tools from the Kali Linux operating system.
- The standard provides you with three different views, namely, enumeration, development, and cross-reference. Enumeration serves as a base for all the attacks and weaknesses found in the web applications. The development view merges these attacks and weaknesses into vulnerabilities and categorizes them according to their occurrence in the relative development phase. This could be a design, implementation, or deployment phase. The cross-reference view serves the purpose of referencing other application security standards with WASC-TC.
- WASC-TC has already acquired industry-level acceptance and its integration can be found in many open source and commercial solutions, mostly in vulnerability assessment and managerial products.
- It can also be aligned with other well-known application security standards, such as OWASP and SANS-CWE.