Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Chapter 2. Penetration Testing Methodology

Penetration testing, often abbreviated as **pentest** , is a process that is followed to conduct an in-depth security assessment or audit. A **methodology** defines a set of rules, practices, and procedures that are pursued and implemented during the course of any information security audit program. A **penetration testing methodology** defines a roadmap with practical ideas and proven practices that can be followed to assess the true security posture of a network, application, system, or any combination thereof. This chapter offers summaries of several key penetration testing methodologies. Key topics covered in this chapter include:

- A discussion on two well-known types of penetration testing—black box and white box
- Describing the differences between the vulnerability assessment and penetration testing
- Explaining several industry-acceptable security testing methodologies and their core functions, features, and benefits
- A general penetration testing methodology that incorporates the 10 consecutive steps of a typical penetration testing process
- The ethical dimension of how the security testing projects should be handled

Penetration testing can be carried out independently or as a part of an IT security **risk management** process that may be incorporated into a regular development life cycle (for example, Microsoft SDLC). It is vital to notice that the security of a product not only depends on the factors that are related to the IT environment but also relies on product-specific security best practices. This involves the implementation of appropriate security requirements, performing risk analysis, threat modeling, code reviews, and operational security measurement.

Penetration testing is considered to be the last and most aggressive form of security assessment. It must be handled by qualified professionals and can be conducted with or without prior knowledge of the targeted network or application. A pentest may be used to assess all IT infrastructure components including applications, network devices, operating systems, communication medium, physical security, and human psychology. The output of penetration testing usually consists of a report divided into several sections that address the weaknesses found in the current state of the target environment, followed by potential countermeasures and other remediation recommendations. The use of a methodological process provides extensive benefits to the **pentester** to understand and critically analyze the integrity of current defenses during each stage of the testing process.

Types of penetration testing

Although there are different types of penetration testing, the two most general approaches that are widely accepted by the industry are the black box and white box. These approaches will be discussed in the following sections.

Black box testing

While applying this approach, the security auditor will be assessing the network infrastructure and will not be aware of any internal technologies deployed by the targeted organization. By employing a number of real-world hacker techniques and going through organized test phases, vulnerabilities may be revealed and potentially exploited. It is important for a pentester to understand, classify, and prioritize these vulnerabilities according to their level of risk (low, medium, or high). The risk can be measured according to the threat imposed by the vulnerability in general. An ideal penetration tester would determine all attack vectors that could cause the target to be compromised. Once the testing process has been completed, a report that contains all the necessary information regarding the targets' real-world security posture, categorizing, and translating the identified risks into a business context, is generated. Black box testing can be a more expensive service than white box testing.

White box testing

An auditor involved in this kind of penetration testing process should be aware of all the internal and underlying technologies used by the target environment. Hence, it opens a wide gate for a penetration tester to view and critically evaluate the security vulnerabilities with minimum possible efforts and utmost accuracy. It does bring more value to the organization in comparison to the black box approach in the sense that it will eliminate any internal security issues lying at the target infrastructure's environment, thus making it more difficult for a malicious adversary to infiltrate from the outside. The number of steps involved in white box testing is similar to that of black box testing. Moreover, the white box approach can easily be integrated into a regular development life cycle to eradicate any possible security issues at an early stage before they get disclosed and exploited by intruders. The time, cost, and knowledge level required to find and resolve the security vulnerabilities is comparably less than with the black box approach.

Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Vulnerability assessment versus penetration testing

There is always a need to understand and practice the correct terminology for security assessment. Throughout your career, you may run into commercial grade companies and non-commercial organizations that are likely to misinterpret the term penetration testing when trying to select an assessment type. It is important that you understand the differences between these types of tests.

Vulnerability assessment is a process to assess the internal and external security controls by identifying the threats that pose serious exposure to the organizations' assets. This technical infrastructure evaluation not only points to the risks in the existing defenses, but also recommends and prioritizes the remediation strategies. The internal vulnerability assessment provides you with an assurance to secure the internal systems, while the external vulnerability assessment demonstrates the security of the perimeter defenses. In both testing criteria, each asset on the network is rigorously tested against multiple attack vectors to identify unattended threats and quantify the reactive measures. Depending on the type of assessment being carried out, a unique set of testing processes, tools, and techniques are followed to detect and identify vulnerabilities in the information assets in an automated fashion. This can be achieved using an integrated **vulnerability management** platform that manages an up-to-date vulnerabilities database and is capable of testing different types of network devices while maintaining the integrity of configuration and change management.

A key difference between the vulnerability assessment and penetration testing is that the penetration testing goes beyond the level of identifying vulnerabilities and hooks into the process of **exploitation**, **privilege escalation**, and **maintaining access** to the target system(s). On the other hand, vulnerability assessment provides you with a broad view of any existing flaws in the system without measuring the impact of these flaws to the system under consideration. Another major difference between both of these terms is that the penetration testing is considerably more intrusive than the vulnerability assessment and aggressively applies all of the technical methods to exploit the live production environment. However, the vulnerability assessment process carefully identifies and quantifies all the known vulnerabilities in a non-invasive manner.

Tip

Why penetration testing?

When there is doubt that mitigating controls such as firewalls, intrusion detection systems, file integrity monitoring, and so on are effective, a full penetration test is ideal. Vulnerability scanning will locate individual vulnerabilities; however, penetration testing will actually attempt to verify that these vulnerabilities are exploitable within the target environment.

This perception, while dealing with both of these assessment types, might confuse and overlap the terms interchangeably, which is absolutely wrong. A qualified consultant always attempts to work out the best type of assessment based on the client's business requirement rather than misleading them from one over the other. It is also the duty of the contracting party to look into the core details of the selected security assessment program before taking any final decision.

Note

Penetration testing is an expensive service in comparison to vulnerability assessment.

Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

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.

Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Penetration Testing Execution Standard (PTES)

The **Penetration Testing Execution Standard** (**PTES**) was created by some of the brightest minds and definitive experts in the penetration testing industry. It consists of seven phases of penetration testing and can be used to perform an effective penetration test on any environment. The details of the methodology can be found at http://www.pentest-standard.org/index.php/Main_Page.

The seven stages of penetration testing that are detailed by this standard are as follows (source: www.pentest-standard.org):

- Pre-engagement interactions
- Intelligence gathering
- Threat modeling
- Vulnerability analysis
- Exploitation
- Post-exploitation
- Reporting

Each of these stages is provided in detail on the PTES site along with specific mind maps that detail the steps required for each phase. This allows for the customization of the PTES standard to match the testing requirements of the environments that are being tested. More details about each step can be accessed by simply clicking on the item in the mind map.

Key features and benefits

The following are the key features and benefits of the PTES:

- It is a very thorough penetration testing framework that covers the technical as well as other important aspects of a penetration test, such as scope creep, reporting, and protecting you as a penetration tester
- It has detailed instructions on how to perform many of the tasks that are required to accurately test the security posture of an environment
- It is put together for penetration testers by experienced penetration testing experts who perform these tasks on a daily basis
- It is inclusive of the most commonly found technologies as well as ones that are not so common
- It is easy to understand and you can adapt it to your own testing needs

Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

General penetration testing framework

Kali Linux is a versatile operating system that comes with a number of security assessment and penetration testing tools. Deriving and practicing these tools without a proper framework can lead to unsuccessful testing and might produce unsatisfied results. Thus, formalizing the security testing with a structured framework is extremely important from a technical and managerial perspective.

The general testing framework presented in this section will constitute both the black box and white box approaches. It offers you a basic overview of the typical phases through which an auditor or penetration tester should progress. Either of these approaches can be adjusted according to the given target of assessment. The framework is composed of a number of steps that should be followed in a process at the initial, medial, and final stages of testing in order to accomplish a successful assessment. These include the following:

- Target scoping
- Information gathering
- Target discovery
- Enumerating target
- Vulnerability mapping
- Social engineering
- Target exploitation
- Privilege escalation
- Maintaining access
- Documentation and reporting

Whether applying any combination of these steps with the black box or white box approaches, it is left to the penetration tester to decide and choose the most strategic path according to the given target environment and its prior knowledge before the test begins. We will explain each stage of testing with a brief description, definition, and its possible applications. This general approach may be combined with any of the existing methodologies and should be used as a guideline rather than a penetration testing catch-all solution.

Target scoping

Before starting the technical security assessment, it is important to observe and understand the given scope of the target network environment. It is also necessary to know that the scope can be defined for a single entity or set of entities that are given to the auditor. The following list provides you with typical decisions that need to be made during the target scoping phase:

- What should be tested?
- How should it be tested?
- What conditions should be applied during the test process?
- What will limit the execution of the test process?
- How long will it take to complete the test?
- What business objectives will be achieved?

To lead a successful penetration testing, an auditor must be aware of the technology under assessment, its basic functionality, and its interaction with the network environment. Thus, the knowledge of an auditor does make a significant contribution towards any kind of security assessment.

Information gathering

Once the scope is finalized, it is time to move into the reconnaissance phase. During this phase, a pentester uses a number of publicly available resources to learn more about his or her target. This information can be retrieved from Internet sources such as:

- Forums
- Bulletin boards
- Newsgroups
- Articles
- Blogs
- Social networks

- Commercial or non-commercial websites

Additionally, the data can also be gathered through various search engines, such as Google, Yahoo!, MSN Bing, Baidu, and others. Moreover, an auditor can use the tools provided in Kali Linux to extract the network information about a target. These tools perform valuable data mining techniques to collect information through DNS servers, trace routes, Whois database, e-mail addresses, phone numbers, personal information, and user accounts. As more information is gathered, the probability of conducting a successful penetration test is increased.

Target discovery

This phase mainly deals with identifying the target's network status, operating system, and its relative network architecture. This provides you with a complete image of the interconnected current technologies or devices and may further help you in enumerating various services that are running over the network. By using the advanced network tools from Kali Linux, one can determine the live network hosts, operating systems running on these host machines, and characterize each device according to its role in the network system. These tools generally implement **active** and **passive** detection techniques on the top of network protocols, which can be manipulated in different forms to acquire useful information such as operating system fingerprinting.

Enumerating target

This phase takes all the previous efforts forward and finds the open ports on the target systems. Once the open ports have been identified, they can be enumerated for the running services. Using a number of port scanning techniques such as full-open, half-open, and stealth scan can help determine the port's visibility even if the host is behind a firewall or **Intrusion Detection System** (**IDS**). The services mapped to the open ports help in further investigating the vulnerabilities that might exist in the target network's infrastructure. Hence, this phase serves as a base for finding vulnerabilities in various network devices, which can lead to a serious penetration. An auditor can use some automated tools given in Kali Linux to achieve the goal of this phase.

Vulnerability mapping

Up until the previous phase, we have gathered sufficient information about the target network. It is now time to identify and analyze the vulnerabilities based on the disclosed ports and services. This process can be achieved via a number of automated network and application vulnerability assessment tools that are present under the Kali Linux OS. It can also be done manually but takes an enormous amount of time and requires expert knowledge. However, combining both approaches should provide an auditor with a clear vision to carefully examine any known or unknown vulnerability that may otherwise exist on the network systems.

Social engineering

Practicing the art of deception is considerably important when there is no open gate available for an auditor to enter the target network. Thus, using a human attack vector, it is still possible to penetrate the target system by tricking a user into executing malicious code that should give backdoor access to the auditor. Social engineering comes in different forms. This can be anybody pretending to be a network administrator over the phone forcing you to reveal your account information or an e-mail phishing scam that can hijack your bank account details. Someone imitating personnel to get into a physical location is also considered social engineering. There is an immense set of possibilities that could be applied to achieve the required goal. Note that for a successful penetration, additional time to understand human psychology may be required before applying any suitable deception against the target. It is also important to fully understand the associated laws of your country with regards to social engineering prior to attempting this phase.

Target exploitation

After carefully examining the discovered vulnerabilities, it is possible to penetrate the target system based on the types of exploits that are available. Sometimes, it may require additional research or modifications to the existing exploit in order to make it work properly. This sounds a bit difficult but might get easier when considering a work under advanced exploitation tools, which are already provided with Kali Linux. Moreover, an auditor can also apply client-side exploitation methods mixed with a little social engineering to take control of a target system. Thus, this phase mainly focuses on the target acquisition process. The process coordinates three core areas, which involve pre-exploitation, exploitation, and post-exploitation activities.

Privilege escalation

Once the target is acquired, the penetration is successful. An auditor can now move freely into the system, depending on his or her access privileges. These privileges can also be escalated using any local exploits that match the system's environment, which, once executed, should help you attain super-user or system-level privileges. From this point of entry, an auditor might also be able to launch further attacks against the local network systems. This process can be restricted or non-restricted depending on the given target's scope. There is also a possibility of learning more about the compromised target by sniffing the network traffic, cracking passwords of various services, and applying local network spoofing tactics. Hence, the purpose of privilege escalation is to gain the highest-level access to the system that is possible.

Maintaining access

Sometimes, an auditor might be asked to retain access to the system for a specified time period. Such activity can be used to demonstrate illegitimate access to the system without performing the penetration testing process again. This saves time, cost, and resources that are being served to gain access to the system for security purposes. Employing some secret tunneling methods, which make a use of protocol, proxy, or end-to-end connection strategy that can lead to establishing a backdoor access, can help an auditor maintain his or her footsteps into the target system as long as required. This kind of system access provides you with a clear view on how an attacker can maintain his or her presence in the system without noisy behavior.

Documentation and reporting

Documenting, reporting, and presenting the vulnerabilities found, verified, and exploited will conclude your penetration testing activities. From an ethical perspective, this is extremely important because the concerned managerial and technical team can inspect the method of penetration and try to close any security loopholes that may exist. The types of reports that are created for each relevant authority in the contracting organization may have different outlooks to assist the business and technical staff understand and analyze the weak points that exist in their IT infrastructure. Additionally, these reports can serve the purpose of capturing and comparing the target system's integrity before and after the penetration process.

Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

The ethics

The ethical vision of security testing constitutes rules of engagement that have to be followed by an auditor to present professional, ethical, and authorized practices. These rules define how the testing services should be offered, how the testing should be performed, determine the legal contracts and negotiations, define the scope of testing, prepare the test plan, follow the test process, and manage a consistent reporting structure. Addressing each of these areas requires careful examination and the design of formal practices and procedures must be followed throughout the test engagement. Some examples of these rules are discussed as follows:

- Offering testing services after breaking into the target system before making any formal agreement between the client and auditor is completely forbidden. This act of unethical marketing can result in the failure of a business and might have severe legal implications depending on the jurisdictions of a country.
- Performing a test beyond the scope of testing and crossing the identified boundaries without explicit permissions from a client is prohibited.
- Binding a legal contract that should limit the liability of a job unless any illegal activity is detected. The contract should clearly state the terms and conditions of testing, the emergency contact information, the statement of work, and any obvious conflicts of interest.
- The test plan concerns the amount of time that is required to assess the security of a target system. It is highly advisable to draw up a schedule that does not interrupt the production of business hours.
- The test process defines the set of steps that are required to be followed during the test engagement. These rules combine technical and managerial views to restrict the testing process with its environment and people.
- Scope definition should clearly define all the contractual entities and the limits imposed on them during the security assessment.
- Test results and reporting must be presented in a clear and consistent order. The report must mark all the known and unknown vulnerabilities and should be delivered confidentially to the authorized individual only.

Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Summary

In this chapter, we have discussed several penetration testing methodologies. We have also described the basic terminology of penetration testing, its associated types, and the industry contradiction with other similar terms. The summary of these key points is highlighted as follows:

- Penetration testing can be broken into different types such as black box and white box. The black box approach is also known as **external testing**, where the auditor has no prior knowledge of the target system. The white box approach refers to an **internal testing**, where the auditor is fully aware of target environment. The combination of both types is known as a gray box.
- The basic difference between vulnerability assessment and penetration testing is that the vulnerability assessments identify the flaws that exist in the system without measuring their impact, while the penetration testing takes a step forward and exploits these vulnerabilities in order to evaluate their consequences.
- There are a number of security testing methodologies but very few provide stepwise, consistent instructions on measuring the security of a system or application. We have discussed five such well-known open source security assessment methodologies, highlighting their technical capabilities, key features, and benefits. These include OSSTMM, ISSAF, OWASP, PTES, and WASC-TC.
- We also presented a simplified and structured testing framework for penetration testing. This process involves a number of steps, which have been organized according to the industry approach towards security testing. These include target scoping, information gathering, target discovery, enumerating target, vulnerability mapping, social engineering, target exploitation, privilege escalation, maintaining access, and documentation and reporting.
- Finally, we discussed the ethical view of penetration testing that should be justified and followed throughout the assessment process. Considering ethics during every single step of assessment engagements leads to a successful arrangement between auditor and business entity.

The next chapter will guide you through the strategic engagement of acquiring and managing information taken from the client for the penetration testing assignment.