Ethical Hacker

A Cisco/NetAcad Course

Introduction to Ethical Hacking and Penetration Testing — Explain the importance of methodological ethical hacking and penetration testing.
Planning and Scoping a Penetration Testing Assessment — Create penetration testing preliminary documents.
Information Gathering and Vulnerability Scanning — Perform information gathering and vulnerability scanning activities.
Social Engineering Attacks — Explain how social engineering attacks succeed.
Exploiting Wired and Wireless Networks — Explain how to exploit wired and wireless network vulnerabilities.
Exploiting Application-Based Vulnerabilities — Explain how to exploit application-based vulnerabilities.
Cloud, Mobile, and IoT Security — Explain how to exploit cloud, mobile, and IoT security vulnerabilities.
Performing Post-Exploitation Techniques — Explain how to perform post-exploitation activities.
Reporting and Communication — Create a penetration testing report.
Tools and Code Analysis — Classify pentesting tools by use case.

Module 1: Introduction to Ethical Hacking and Penetration Testing

Module Objective: Explain the importance of methodological ethical hacking and penetration testing.

4 types of threat actors:

- Organized Crime
- Hacktivists
- State-Sponsored Attackers
- Inside Threats

3 most-common environment considerations for pen tests:

- Network Infrastructure Tests
  - Testing of the network infrastructure can mean a few things. For the purposes of this course, we say it is focused on evaluating the security posture of the actual network infrastructure and how it is able to help defend against attacks. This often includes the switches, routers, firewalls, and supporting resources, such as authentication, authorization, and accounting (AAA) servers and IPSs. A penetration test on wireless infrastructure may sometimes be included in the scope of a network infrastructure test. However, additional types of tests beyond a wired network assessment would be performed. For instance, a wireless security tester would attempt to break into a network via the wireless network either by bypassing security mechanisms or breaking the cryptographic methods used to secure the traffic. Testing the wireless infrastructure helps an organization to determine weaknesses in the wireless deployment as well as the exposure. It often includes a detailed heat map of the signal disbursement.

- Application-Based Tests
  - This type of pen testing focuses on testing for security weaknesses in enterprise applications. These weaknesses can include but are not limited to
    - misconfigurations,
    - input validation issues,
    - injection issues, and
    - logic flaws.

  - Because a web application is typically built on a web server with a back-end database, the testing scope normally includes the database as well. However, it focuses on gaining access to that supporting database through the web application compromise. A great resource that we mention a number of times in this book is the **Open Web Application Security Project**

(OWASP).

- Penetration Testing in the Cloud
    - Cloud service providers (CSPs) such as Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP) have no choice but to take their security and compliance responsibilities very seriously. For instance, Amazon created the Shared Responsibility Model to describe the AWS customers' responsibilities and Amazon's responsibilities in detail (see https://aws.amazon.com/compliance/shared-responsibility-model).
    - The responsibility for cloud security depends on the type of cloud model (software as a service [SaaS], platform as a service [PaaS], or infrastructure as a service [IaaS]). For example, with IaaS, the customer (cloud consumer) is responsible for data, applications, runtime, middleware, virtual machines (VMs), containers, and operating systems in VMs. Regardless of the model used, cloud security is the responsibility of both the client and the cloud provider. These details need to be worked out before a cloud computing contract is signed. These contracts vary depending on the security requirements of the client. Considerations include disaster recovery, service-level agreements (SLAs), data integrity, and encryption. For example, is encryption provided end to end or just at the cloud provider? Also, who manages the encryption keys–the CSP or the client?
    - Overall, you want to ensure that the CSP has the same layers of security (logical, physical, and administrative) in place that you would have for services you control. When performing penetration testing in the cloud, you must understand what you can do and what you cannot do. Most CSPs have detailed guidelines on how to perform security assessments and penetration testing in the cloud. Regardless, there are many potential threats when organizations move to a cloud model. For example, although your data is in the cloud, it must reside in a physical location somewhere. Your cloud provider should agree in writing to provide the level of security required for your customers. As an example, the following link includes the AWS Customer Support Policy for Penetration Testing: https://aws.amazon.com/security/penetration-testing.

One of the tools that we talk about more in a later module is the Social-Engineer Toolkit (SET), created by Dave Kennedy. This is a great tool for performing social engineering testing campaigns.

Regarding Bug Bounties:

Before diving in, make sure you've got:

1. Basic web and network security knowledge
    - HTTP, DNS, SSL/TLS, common web vulnerabilities (OWASP Top 10).
2. Familiarity with pentesting tools
    - Burp Suite (or OWASP ZAP), nmap, curl, your browser's dev tools.
3. Programming/scripting comfort
    - At least one language (Python, JavaScript, Bash) to automate tests.

3 “Testing Environments”:

- Unknown-Environment Test
    - In an unknown-environment penetration test, the tester is typically provided only a very limited amount of information.
        - For instance, the tester may be provided only the domain names and IP addresses that are in scope for a particular target.
    - The idea of this type of limitation is to have the tester start out with the perspective that an external attacker might have.
    - Typically, an attacker would first determine a target and then begin to gather information about the target, using public information, and gain more and more information to use in attacks.
        - The tester would not have prior knowledge of the target’s organization and infrastructure.
    - Another aspect of unknown-environment testing is that sometimes the network support personnel of the target may not be given information about exactly when the test is taking place.
        - This allows for a defense exercise to take place as well, and it eliminates the issue of a target preparing for the test and not giving a real-world view of how the security posture really looks.

- Known-Environment Test
    - In a known-environment penetration test, the tester starts out with a significant amount of information about the organization and its infrastructure.
        - The tester would normally be provided things like network diagrams, IP addresses, configurations, and a set of user credentials.
    - If the scope includes an application assessment, the tester might also be provided the source code of the target application. The idea of this type of test is to identify as many security holes as possible.
    - In an unknown-environment test, the scope may be only to identify a path into the organization and stop there.
        - With known-environment testing, the scope is typically much broader and includes internal network configuration auditing and scanning of desktop computers for defects.
    - Time and money are typically deciding factors in the determination of which type of penetration test to complete.
    - If a company has specific concerns about an application, a server, or a segment of the infrastructure, it can provide information about that specific target to decrease the scope and the amount of time spent on the test but still uncover the desired results.
    - With the sophistication and capabilities of adversaries today, it is likely that most networks will be compromised at some point, and a white-box approach is not a bad option.

- Partially Known Environment Test
    - A partially known environment penetration test is somewhat of a hybrid approach between unknown- and known-environment tests.
    - With partially known environment testing, the testers may be provided credentials but not full documentation of the network infrastructure.
        - This would allow the testers to still provide results of their testing from the perspective of an external attacker’s point of view.
    - Considering the fact that most compromises start at the client and work their way throughout the network, a good approach would be a scope where the testers start on the inside of the network and have access to a client machine.
        - Then they could pivot throughout the network to determine what the impact of a compromise would be.

Standards/Methodologies

MITRE ATT&CK

- The MITRE ATT&CK framework (https://attack.mitre.org) is an amazing resource for learning about an adversary's tactics, techniques, and procedures (TTPs).
- Both offensive security professionals (penetration testers, red teamers, bug hunters, and so on) and incident responders and threat hunting teams use the MITRE ATT&CK framework today.
- The MITRE ATT&CK framework is a collection of different matrices of tactics, techniques, and subtechniques.
  - These matrices–including the Enterprise ATT&CK Matrix, Network, Cloud, ICS, and Mobile–list the tactics and techniques that adversaries use while preparing for an attack, including gathering of information (open-source intelligence [OSINT], technical and people weakness identification, and more) as well as different exploitation and post-exploitation techniques.

- You will learn more about MITRE ATT&CK throughout this course.

OWASP WSTG

- The OWASP Web Security Testing Guide (**WSTG**) is a comprehensive guide focused on web application testing. It is a compilation of many years of work by OWASP members.
- OWASP WSTG covers the high-level phases of web application security testing and digs deeper into the testing methods used.
  - For instance, it goes as far as providing attack vectors for testing cross-site scripting (**XSS**), XML external entity (**XXE**) attacks, cross-site request forgery (**CSRF**), and SQL injection attacks; as well as how to prevent and mitigate these attacks.
- You will learn more about these attacks in Module 6, "Exploiting Application-Based Vulnerabilities."
- From a web application security testing perspective, OWASP WSTG is the most detailed and comprehensive guide available.
  - You can find the OWASP WSTG and related project information at https://owasp.org/www-project-web-security-testing-guide/.

NIST SP 800-115

- Special Publication (SP) 800-115 is a document created by the National Institute of Standards and Technology (**NIST**), which is part of the U.S. Department of Commerce.
- NIST SP 800-115 provides organizations with guidelines on planning and conducting information security testing. It superseded the previous standard document, SP 800-42. SP 800-115 is considered an industry standard for penetration testing guidance and is called out in many other industry standards and documents.
- You can access NIST SP 800-115 at https://csrc.nist.gov/publications/detail/sp/800-115/final.

OSSTMM

The Open Source Security Testing Methodology Manual (**OSSTMM**), developed by Pete Herzog, has been around a long time. Distributed by the Institute for Security and Open Methodologies (**ISECOM**), the OSSTMM is a document that lays out repeatable and consistent security testing (https://www.isecom.org).

- It is currently in version 3, and version 4 is in draft status.

The OSSTMM has the following key sections:

- Operational Security Metrics
- Trust Analysis
- Work Flow
- Human Security Testing
- Physical Security Testing
- Wireless Security Testing
- Telecommunications Security Testing
- Data Networks Security Testing
- Compliance Regulations
- Reporting with the Security Test Audit Report (STAR)

PTES

The Penetration Testing Execution Standard (**PTES**) (http://www.pentest-standard.org) provides information about types of attacks and methods, and it provides information on the latest tools available to accomplish the testing methods outlined.

PTES involves seven distinct phases:

1. Pre-engagement interactions
2. Intelligence gathering
3. Threat modeling
4. Vulnerability analysis
5. Exploitation
6. Post-exploitation
7. Reporting

ISSAF

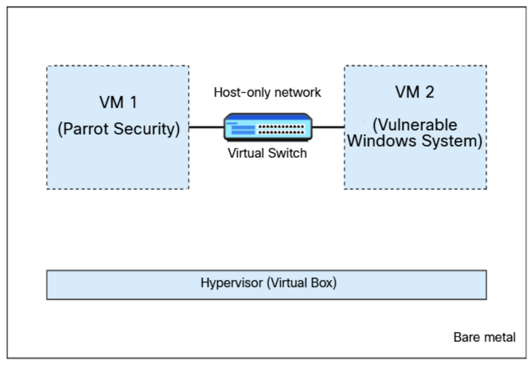
The Information Systems Security Assessment Framework (**ISSAF**) is another penetration testing methodology similar to the others on this list with some additional phases. ISSAF covers the following phases:

- Information gathering
- Network mapping
- Vulnerability identification
- Penetration
- Gaining access and privilege escalation
- Enumerating further
- Compromising remote users/sites
- Maintaining access
- Covering the tracks

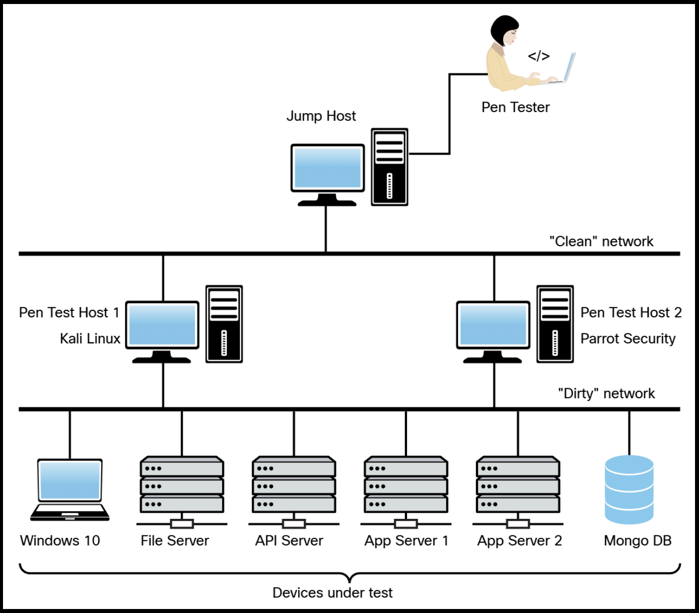
– – Continued – –

Building a Pen-Testing Lab

Basic Penetration Testing Lab Environment with Two VMs →



More Elaborate Penetration Testing Lab Environment:



Typical pen testing environment rules:

Closed network — Ensure controlled access to and from the lab environment, with restricted access to the Internet.

Virtualized computing environment — Allows easy deployment and recovery of devices being tested.

Realistic environment — The testing environment should match the real environment as closely as possible.

Health monitoring — When something crashes, you must be able to determine why it happened.

Sufficient hardware resources — Ensure lack of resources is not the cause of false results.

Multiple operating systems — Test or validate findings from different operating systems to compare results.

Duplicate tools — Validate findings by running the same test with different tools and comparing results.

Practice targets — Practice using tools against targets that are known to be vulnerable.

Module 2: Planning and Scoping a Penetration Testing Assessment

Module 3: Information Gathering and Vulnerability Scanning
Module 4: Social Engineering Attacks
Module 5: Exploiting Wired and Wireless Networks
Module 6: Exploiting Application-Based Vulnerabilities
Module 7: Cloud, Mobile, and IoT Security
Module 8: Performing Post-Exploitation Techniques
Module 9: Reporting and Communication
Module 10: Tools and Code Analysis

Final Activity