1.0 Penetration Testing with Kali Linux: General Course Information

Welcome to the ***Penetration Testing with Kali Linux*** (PWK) course!

PWK was created for System and Network Administrators and security professionals who would like to take a serious and meaningful step into the world of professional penetration testing. This course will help you better understand the attacks and techniques that are used by malicious entities against computers and networks.

The ultimate purpose of the course is to provide an understanding of, and intuition for, these attacks at a deep enough level to be able to replicate them. By leveraging the ability to perform them, we can develop a powerful insight into what kind of security defenses are important and how to improve them. Congratulations on taking that first step. We're excited you're here.

PWK consists of two types of overarching learning modalities: ***Learning Modules*** and ***Challenges Labs***. Learning Modules all cover specific penetration testing concepts or techniques, while Challenge Labs require the learner to apply the skills acquired via the Modules.

Learning Modules are divided into ***Learning Units*** : atomic pieces of content that help the learner achieve specific ***Learning Objectives***.

In this Learning Module we will cover the following Learning Units:

- *Getting Started with PWK*
- *How to Approach the Course*
- *Summary of PWK Learning Modules*

1.1 Getting Started with PWK

This Learning Unit covers the following Learning Objectives:

- *Take inventory over what's included in the course*
- *Set up an Attacking Kali VM*
- *Connect to the PWK VPN*

Much like learning to play a musical instrument, security training requires equal parts of conceptual knowledge and hands-on practice. In this Learning Unit we'll learn what kind of material is included with PWK, how to set up our attacking Kali VM, and how to reach the PWK labs over a VPN connection.

1.1.1 PWK Course Materials

The course includes online access to the Learning Modules and their accompanying course videos. The information covered in the Modules and the videos overlap, meaning you can read the Modules and then watch the videos to fill in any gaps or vice versa. In some cases, the book modules are more detailed than the videos. In other cases, the videos may convey some information better than the Modules. It is important that you pay close attention to both.

The Learning Modules also contain various exercises. Completing the Module exercises will help you become more efficient with discovering and exploiting the vulnerabilities in the lab machines.

Some Module exercises have a simple question-and-answer where the learner is tasked with retrieving the solution from the text. Other Module exercises have three components: a question, a machine (or a group of machines), and a flag. In these cases, the question asks you to perform a specific action or set of actions on the provided machine. Once you have successfully completed the objective, you will receive a flag in the form **OS{random-hash}**. You can then submit the flag into the ***OffSec Learning Portal*** (OLP), which will tell you if you have inserted the correct flag or not. The OLP will then save your progress, and track the number of your correct submissions provided to date.

It is worth noting that flags are dynamically generated at machine boot and expire at machine shutdown. If the solution is obtained to a question and the machine is reverted, and only after the revert the original answer is submitted, the OLP will not accept the flag.

The flag must be submitted before reverting or powering off the machine.

As an additional note, the way Module exercises are implemented allows us to use the same remote IP and port multiple times. On the Module Exercise VMs that require an SSH connection, we suggest issuing the SSH command with a couple of extra options as follows:

```
ssh -o "UserKnownHostsFile=/dev/null" -o "StrictHostKeyChecking=no"
learner@192.168.50.52
```

Listing 1 - The recommended way to SSH into Module Exercise VMs

The **UserKnownHostsFile=/dev/null** and **StrictHostKeyChecking=no** options have been added to prevent the known-hosts file on our local Kali machine from being corrupted.

Module Exercises are currently supported on the x86-64 Kali Linux version exclusively.

We will go over the design of different kinds of Module exercises in a section below.

1.1.2 Student Mentors and Support

Discord[1], our community chat platform, can be accessed via the Profile drop-down at the upper right hand corner of the OffSec Learning Portal. Live Support will allow you to directly communicate with our learner Mentors and learner Technical Services Teams.

The Technical Services Team is available to assist with technical issues, while the Student Mentors will be able to clarify items in the course material and exercises. In addition, if you have tried your best and are completely stuck on an exercise or lab machine, Student Mentors may be able to provide a small hint to help you on your way.

Remember that the information provided by the Student Mentors will be based on the amount of detail you are able to provide. The more detail you can give about what you've already tried and the outcomes you've been able to observe, the more they will be able to help you.

1.1.3 Setting Up Kali

The Module Exercises and Challenge Labs are to be completed using virtual machines (VMs) operating in our lab environment. When we refer to a lab environment, we mean the combination of the following components:

- *Your Kali Linux VM*
- *The OffSec Learning Portal*
- *A lab containing deployable target machines*
- *A VPN connection between your Kali VM and the lab*

Let's look at these components individually.

Kali Linux[2] is an operating system (like Windows or macOS) that comes with a curated set of tools that are specifically useful for penetration testing and other information security activities. Kali Linux is open source and free to use.

If you're already familiar with cybersecurity, you may have Kali Linux installed and can skip ahead to the next section.

If not, we *strongly* recommend installing Kali on a VM, which provides the functionality of a physical computer system running another operating system (OS) within a program called a hypervisor. The benefit of using a VM is that it allows us to run a guest OS within a host OS. Although we could physically install Kali on a dedicated machine, it is more convenient, safe, and efficient to install Kali within our host system. Among other reasons, this ensures that we have easy access to all the tools available to both.

For example, we may be using a desktop computer running Windows or a laptop running macOS. We could install VMware Workstation Player on our Windows machine or VMware Fusion on our Mac to install the Kali Linux VMware image. When this virtual image is installed, Kali will run alongside our primary operating system in a window, or full-screen if we like. If configured properly, Kali Linux will have access to

[1] (OffSec, 2023), https://discord.gg/offsec
[2] (OffSec, 2023), https://help.offensive-security.com/hc/en-us/articles/360049796792-Kali-Linux-Virtual-Machine

the network with its own IP address and will behave as if it's installed on a dedicated machine for the most part.

> ***From a terminology standpoint, we call the physical system running Windows or macOS our host machine and we call the Kali VM a guest machine.***

The VMware image that we recommend is a default 64-bit build of Kali Linux. We recommend using the latest VMware image available on the OffSec VM image download page.[3] Note that although the VirtualBox image, the Hyper-V image, or a dedicated installation of Kali should work, we can only provide support for the indicated VMware images.

In the next section, we'll set up the VPN connection that will connect us to the lab.

1.1.4 Connecting to the PWK Lab

Many of the Module exercises and all of the lab machines will require you to connect to a ***Virtual Private Network*** (VPN).

A VPN essentially creates an encrypted tunnel that allows your data to traverse an open network such as the public Internet, and connect to another otherwise isolated network securely.

We'll connect to the VPN from our Kali machine, granting us access to the lab. When a learner connects to the lab, the specific segment of the network they connect to is private to them. In other words, each connection is to a unique environment in which the learner can work at their own pace without worrying about interrupting, or being interrupted by, other learners.

Even though each lab is private, it is prudent to consider the labs as a hostile environment and you should not store sensitive information on the Kali Linux virtual machine used to connect to the VPN. Client-to-client VPN traffic is strictly forbidden and could result in termination of access from the course and its materials.

Fortunately, connecting to a VPN is a quick and easy process. If you're using Kali as a VM, go ahead and start the machine. Then on the Kali machine, open up a browser and navigate to the OffSec Learning Portal and sign in.

Next, let's navigate to the Course drop-down menu and select the PEN200 course. This will take us to the main course page. At the top right corner of the page but to the left of your account name, you'll see the download drop-down menu for VPN. Clicking this option will generate a VPN pack for this course and download it in the form of a .ovpn text file. Be sure to note the location of the download.

[3] (OffSec, 2023), https://help.offensive-security.com/hc/en-us/articles/360049796792-Kali-Linux-Virtual-Machine

Next, let's use the Kali Linux terminal to connect to the VPN. Clicking the black terminal icon at the top-left of the Kali VM will present a window like this:

```
┌──(kali㉿kali)-[~]
└─$
```

Listing 2 - The Kali Terminal

If we chose a different username during setup, our prompt will include that name:

```
┌──(ArtVandelay㉿kali)-[~]
└─$
```

Listing 3 - The Kali terminal with a different username

In some cases, your screen may differ from what's shown in the course material. This is rarely problematic, but we will often point out these potential inconsistencies.

This is the command prompt, which accepts our user commands. For simplicity we will switch to a less-complex version of the terminal with **Ctrl + P** as shown in Listing 4.

```
kali@kali:~$
```

Listing 4 - Switching to the one-line command prompt

Next, we'll focus on the VPN pack (i.e., the .ovpn file we downloaded). We should have downloaded it to the Kali VM, but if it was downloaded to the host machine, we should either copy it over or re-download it from Kali. Let's use updatedb and locate to find the file:

```
kali@kali:~$ sudo updatedb
[sudo] password for kali:

kali@kali:~$ locate pen200.ovpn
/home/kali/Downloads/pen200.ovpn
```

Listing 5 - Finding the .ovpn file

Note that we used the sudo command to invoke updatedb, because this particular command requires elevated permissions. The updatedb command creates or updates a database that is used by the locate command to find files across the entire filesystem. The sudo command will require us to enter our

password. Note that the cursor will not move and no asterisk (*) characters will appear as we type the password. We'll type in our password and press **Return**.

Based on this output, we are using the filename pen200.ovpn. We can check the browser's download history to determine the exact name of the file.

Once we have located the .ovpn file, we'll cd to its directory, which is /home/kali/Downloads in this case:

```
kali@kali:~$ cd /home/kali/Downloads

kali@kali:~/Downloads$
```

Listing 6 - Changing Directories with cd

Although this command doesn't produce any output (unless we entered the command incorrectly), we can check for the .ovpn file with ls, which lists files in this directory. Note that the output of the below command on your machine may appear different depending on what files are in the Downloads directory:

```
kali@kali:~/Downloads$ ls
pen200.ovpn
```

Listing 7 - Listing file contents with ls

Executing files from Downloads can be a little bit messy, since that particular directory can change so often. Instead, let's create a new directory and move the .ovpn file there:

```
kali@kali:~/Downloads$ mkdir /home/kali/offsec

kali@kali:~/Downloads$ mv pen200.ovpn /home/kali/offsec/pen200.ovpn

kali@kali:~/Downloads$ cd ../offsec

kali@kali:~/offsec$
```

Listing 8 - Creating a new directory and moving the .ovpn file

Here we create a new directory using mkdir, move the .ovpn file with mv and then change our working directory with cd.

We're now ready to connect to the VPN. We'll connect with the openvpn command followed by the full name of the .ovpn file. Once again we must use sudo, since openvpn requires elevated permissions. Note that sudo caches our password for a short time. If we enter this second sudo command shortly after the first, we will not need to re-enter the password.

```
kali@kali:~/offsec$ sudo openvpn pen200.ovpn

2021-06-28 10:20:12 Note: Treating option '--ncp-ciphers' as '--data-ciphers'
(renamed in OpenVPN 2.5).
2021-06-28 10:20:12 DEPRECATED OPTION: --cipher set to 'AES-128-CBC' but missing in
-- data-ciphers (AES-128-GCM). Future OpenVPN version will ignore --cipher for
cipher  negotiations. Add 'AES-128-CBC' to --data-ciphers or change --cipher
'AES-128-CBC' to  --data-ciphers-fallback 'AES-128-CBC' to silence this warning.
2021-06-28 10:20:12 OpenVPN 2.5.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO]
[LZ4]  [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on May 14 2021
2021-06-28 10:20:12 library versions: OpenSSL 1.1.1k 25 Mar 2021, LZO
2.10 2021-06-28 10:20:12 TCP/UDP: Preserving recently used remote
address:  [AF_INET]192.95.19.165:1194
2021-06-28 10:20:12 UDP link local: (not bound)
2021-06-28 10:20:12 UDP link remote: [AF_INET]192.95.19.165:1194
2021-06-28 10:20:12 [offensive-security.com] Peer Connection Initiated
With [AF_INET]192.95.19.165:1194
2021-06-28 10:20:13 TUN/TAP device tun0 opened
2021-06-28 10:20:13 net_iface_mtu_set: mtu 1500 for tun0
2021-06-28 10:20:13 net_iface_up: set tun0 up
2021-06-28 10:20:13 net_addr_v4_add: 192.168.49.115/24 dev tun0
2021-06-28 10:20:13 WARNING: this configuration may cache passwords in memory --
Use the auth-nocache option to prevent this
2021-06-28 10:20:13 Initialization Sequence Completed
```

Listing 9 - Connecting to the labs VPN

The output of Listing 8 may seem intimidating at first. For now, simply note that the last line of the output reads “Initialization Sequence Completed”, indicating that we have connected successfully to the VPN. Make sure that you can find it on your own connection!

We must leave this command prompt open. Closing it will disconnect the VPN connection.

We can open another terminal tab by clicking File > New Tab.

Once we are connected to the PWK VPN, we will be provided with a TUN0 network interface, which we can view with the ip a command. The address assigned to the TUN0 interface will be `192.168.119.X`, where `X` is some value between 1 and 255. Every time we reconnect to the VPN, we might get assigned a different value for `X`.

In addition, all lab machines within the PWK environment will have addresses that follow the format `192.168.X.Y`, where `X` is the same value as the third octet of our TUN0 address, and `Y` is the specific octet associated with the machine.

In the course material, we will be using different IP addresses for our TUN0 network interface as well as for the lab machines. Please make sure you are using the IP addresses assigned to you via TUN0 and via the OLP so that you can access the machines properly.

Lab time starts when your course begins and is metered as continuous access.

If your lab time expires, or is about to expire, you can purchase a lab extension at any time. To purchase additional lab time, use the Extend link available at top right corner of the OffSec Training Library. If you purchase a lab extension while your lab access is still active, you can continue to use the same VPN connectivity pack. If you purchase a lab extension after your existing lab access has ended, you will need to download a new VPN connectivity pack via the course lab page in the OffSec Learning Portal

Learners who have purchased a subscription will have access to the lab as long as the subscription is active. Your subscription will be automatically renewed, unless canceled via the billing page.

1.2 How to Approach the Course

This Learning Unit covers the following Learning Objectives:

- *Conceptualize a learning model based on increasing uncertainty*
- *Understand the different learning components included in PWK*

1.2.1 A Model of Increasing Uncertainty

Penetration testing - and information security in general - is fundamentally about reasoning under uncertainty. Consider how a game like chess is different from a game like poker. In chess, you know everything that your opponent does about the game state (and vice versa). You may not know what they are thinking, but you can make predictions about their next move based on the exact same information that they are using to determine it. When playing poker, however, you do not have all of the information that your opponent possesses, so you must make predictions based on incomplete data.

In this regard, penetration testing is a lot closer to poker than chess. When we simulate an attack, we will never know everything there is to know about the machine/system/network/organization we are targeting. We therefore must make assumptions and estimate probabilities - sometimes implicitly and sometimes explicitly. Conversely, as the defender, we will not be aware of every potential attack vector or vulnerability we might be exposed to. We therefore need to hedge our bets and make sure that our attack surfaces that are most likely to be vulnerable are adequately protected.

> ***As a general rule, the only reason why hacking a machine takes any time at all is because there are things about it that we don't know. In a majority of cases, if we knew everything there was to know about a specific target ahead of time, then we would already know the precise few commands or lines of code necessary to compromise it.***

With this in mind, we can think about PWK as teaching two sets of different skills at the same time: one relating to penetration testing technique, and one relating to methodology, approach, and attitude.

The object level set of skills is taught explicitly via the Modules' Learning Objectives. You will read about how to gather information, find and exploit perimeter defenses, escalate your privileges, move laterally between machines, and pivot to other networks. All of this information is covered extensively and inside the PWK Modules themselves.

However, the structure of the course enables a second order of learning. This second layer is arguably the more important one, though it is much more difficult to quantify. It provides learners with a framework for how to think, feel, and act in novel scenarios. And since penetration testing is about novel scenarios (i.e. uncertainty), it is critical that we become comfortable orienting them.

PWK contains seven learning modalities:

1. **Learning Modules**
2. **Demonstration Module Exercises**
3. **Application Module Exercises**
4. **Capstone Module Exercises**
5. **The Assembling the Pieces Module**
6. **Challenge Labs (type one)**
7. **Challenge Labs (type two)**

We can think about these learning modalities as points along a spectrum, where our uncertainty about the space we're operating in increases as we progress through the course. Let's consider each mode one by one.

1.2.2 Learning Modules

As mentioned above, the text-based Learning Modules all cover specific penetration testing concepts, techniques, and skills. They are each approximately between 30 and 50 pages in length, and they are accompanied by videos that go over the same concepts in a visual and interactive manner. They are logically ordered in a way that allows for progressively building on top of previously learned skills.

In our model of uncertainty, they are considered to be *no/low uncertainty*, because the learner only needs to passively read or watch the content. However, we encourage you to start the relevant lab machines and follow along by typing the commands and clicking around in the same manner as demonstrated. This helps you internalize the material.

1.2.3 Demonstration Module Exercises

There are several types of Module exercise. The objective of the first kind is for the learner to actually absorb the content by following the demonstration.

This type of exercise asks the learner to either input some factual, knowledge based answer to the question, or to obtain a randomized flag by copying the exact same commands and input shown in the course material.

The amount of uncertainty here is still very low, because the learner can obtain the solution directly by reading or watching the Module.

For example, the *Client Side Attacks* Module has a Learning Unit about exploiting Microsoft Office. In that Learning Unit, the learner will be asked to perform the demonstrated techniques on a copy of the original machine used to create the demonstration.

1.2.4 Applied Module Exercises

Here we start to slowly increase the amount of uncertainty. Instead of the learner needing to copy exactly the same steps, the learner now must apply their skills in novel but limited scenarios.

For example, the previously mentioned Learning Unit on Microsoft Office contains a second machine that is slightly modified from the first. The learner needs to use the same type of techniques, but the modifications on the second machine will require that the learner adapt to the new situation.

This kind of exercise helps the learner reinforce what they learned in the demonstration, and also gives them the opportunity to think outside of the box.

1.2.5 Capstone Module Exercises

While demonstration and application exercises are constrained to specific Learning Units, Capstone Exercises have a wider scope. In particular they encompass the entire Module. This increases the amount of uncertainty present, because the learner may not know which techniques or concepts from the module are specifically required to complete the exercise.

In addition to a Learning Unit on exploiting Microsoft Office, the Client Side Attacks Module also contains Learning Units on reconnaissance, and another on Windows Library files. So a capstone exercise for this Module might include a directive to attack a specific machine with one of the client-side attacks, but it won't necessarily be clear which one to use without exploration of the machine.

The purpose of Capstone exercises is to provide ample opportunities to actually hack machines from beginning to end, but still under relatively constrained parameters. In particular, the learner knows the kind of attacks to use, and they know which machines to use them on.

1.2.6 Assembling The Pieces

There are 20 Modules in PWK (aside from this introduction and the final module) and for each of them the learner will go through the process of:

1. **Reading and watching the Module and preferably following along**
2. **Completing the Demonstration exercises by copying the input**
3. **Working through the Application exercises by using specific techniques**
4. **Attacking machines from start to finish via the Capstone Exercises**

At this point, learners will be just about ready for the Challenge Labs. The Assembling the Pieces Module represents a bridge between the Modules and the Labs. It provides a full walkthrough of a small penetration test and allows the learner to follow along with all demonstrated steps. In a sense, this Module is the equivalent of a demonstration exercise for the entire set of Challenge Labs.

1.2.7 Challenge Labs 1-3

There are two types of Challenge Labs. The first three are called *scenarios*. Each scenario consists of a set of networked machines and a short background story that puts those machines in context. Your goal is to obtain access to a Domain Administrator account on an Active Directory domain, and compromise as many machines on the network as possible.

In the same way that Capstone Exercises test the learner on the material of multiple Learning Units, so too do these scenarios test the learner on the material of multiple Learning Modules. The uncertainty here is high, because you will not know which machines are vulnerable to what types of attacks. In addition, each of the three Challenge Labs progressively increases in complexity due to additional machines, subnetworks, and attack vectors.

Further, you will not know that any *specific* machine is directly vulnerable in the first place. Some machines will be dependent on information, credentials, or capabilities that will be found on other machines. And some machines may not even be (intentionally) exploitable until after the Domain Controller is compromised.

All machines contain either a local.txt file, a proof.txt file, or both. The contents of these files are randomized hashes that can be submitted to the OLP to log each compromise. Just like the Module exercise flags, the contents of these files will change on every revert of the machine. We'll discuss more details related to these scenarios in the final Module of PWK.

1.2.8 Challenge Labs 4-6

The second type of Challenge Lab consists of an OSCP-like experience. They are each composed of six OSCP machines. The intention of these Challenges is to provide a mock-exam experience that closely reflects a similar level of difficulty to that of the actual OSCP exam.

Each challenge contains three machines that are connected via Active Directory, and three standalone machines that do not have any dependencies or intranet connections. All the standalone machines have a local.txt and a proof.txt.

While the Challenge Labs have no point values, on the exam the standalone machines would be worth 20 points each for a total of 60 points. The Active Directory set is worth 40 points all together, and the entire domain must be compromised to achieve any points for it at all.

All the intended attack vectors for these machines are taught in the PEN-200 Modules, or are leveraged in the first three Challenge Labs. However, the specific requirements to trigger the vulnerabilities may differ from the exact scenarios and techniques demonstrated in the course material. You are expected to be able to take the demonstrated exploitation techniques and modify them for the specific environment.

Also included with your initial purchase of the PWK course is an attempt at the *OSCP certification exam*[4] itself. The exam is optional, so it is up to you to decide whether or not you would like to tackle it.

To schedule your OSCP exam, go to your exam scheduling calendar. The calendar can be located in the OffSec Learning Portal under the course exam page. Here you will find your exam expiry date, as well as schedule the exam for your preferred date and time.

Keep in mind that you won't be able to select a start time if the exam labs are full for that time period so we encourage you to schedule your exam as soon as possible.

We will cover the exam in more detail in the final Learning Module of this course. For additional information, please visit our support page[5].

1.3 Summary of PWK Learning Modules

This Learning Unit covers the following Learning Objectives:

- *Obtain a high level overview of what's covered in each PEN-200 Learning Module*

In the previous Learning Units, we went over the general structure and specific components of PWK. In this Learning Unit, we will summarize each of the Learning Modules included within the course.

1.3.1 Getting Started: Optional Ramp-up Modules

Introduction to Cybersecurity provides a broad survey on the current state of the world of Cybersecurity. It covers how Cybersecurity is practiced as a discipline and what kinds of threats and threat actors exist. It also covers security principles, controls and strategies, Cybersecurity laws, regulations and frameworks, and career opportunities within the industry.

Effective Learning Strategies is a practical introduction to learning theory that explains OffSec's unique approach to teaching. This module begins with an overview of how learning happens and then explores the construction of OffSec materials. The second half of the module is immediately applicable for learners and includes tactics, strategies, and specific, practical steps.

Finally, we continue with a Module on *Report Writing for Penetration Testers*. This Module provides a framework, some advice, and some tips on writing notes as you progress through a penetration test. It also covers how you might think about writing a penetration testing report. The OSCP exam requires each learner to submit a report of their exam penetration test, so it is recommended to practice your note taking and report writing skills as you proceed with the Module exercises and Challenge Lab machines.

[4] (OffSec, 2023), https://help.offensive-security.com/hc/en-us/categories/360002666252-General-Frequently-Asked Questions-FAQs
[5] (OffSec, 2023), https://help.offensive-security.com/

1.3.2 Web Application and Client Side Attacks

We then dive into one of the most important aspects of penetration testing: *Information Gathering*. Often called by its synonym enumeration, the vast majority of one's time during a penetration test is spent on information gathering of one form or another. However, this Module is specifically about how to approach a network at the very outset of an engagement.

We extend our information gathering toolkit by exploring the concept of *Vulnerability Scanning*[6]. Vulnerability scanning offers us several techniques to narrow our scope within a particular network. It helps us identify machines that are especially likely to be vulnerable. Attack vectors on such machines are often colloquially called *low-hanging fruit*, as the imagery of reaching up to take the easy pieces of fruit off a tree is particularly powerful.

1.3.3 Web Application and Client Side Attacks

It is now time to start learning some perimeter attacks. By perimeter attacks, we mean methods of infiltration that can be reliably done from the internet. In other words, attacks that can be initiated without any sort of access to an organization's internal network.

We begin with an extensive exploration of Web Application attacks. There are two primary reasons for starting here. The first is that Web vulnerabilities are among the most common attack vectors available to us, since modern web apps usually allow users to submit data to them. The second is that web applications are inherently visual and therefore provide us with a nice interface for understanding why our attacks work in the way that they do.

Introduction to Web Applications begins by covering a methodology, a toolset, and an enumeration framework related to web applications that will help us throughout the course. It then covers our first vulnerability class: *Cross-Site Scripting* (XSS)[7]. XSS is an excellent vulnerability to start with because it targets the user of a web application as opposed to the server running it. Since the vast majority of our regular day-to-day usage of web applications is as normal users, XSS can be unusually intuitive, compared to other types of attacks.

> ***Due to the fact that XSS targets users, it can be considered both a Web Application attack and a Client-Side Attack as we'll soon learn.***

We continue our exploration of web application attacks in *Common Web Application Attacks*, where we survey four different kinds of vulnerabilities. *Directory Traversal*[8] provides us with an example of how we can obtain access to information that we're not supposed to. *File Inclusion* shows us what can happen when certain configurations are not set up judiciously by a web administrator. *File Upload Vulnerabilities*[9] demonstrate how we can take advantage of the ability to upload our own files to a web server. Finally, *Command Injection*[10] allows us to run code of our choice on the web server itself.

[6] (Wikipedia, 2023), https://en.wikipedia.org/wiki/Vulnerability_scanner
[7] (OffSec, 2022), https://www.offsec.com/offsec/clarifying-hacking-with-xss/
[8] (OWASP, 2023), https://owasp.org/www-community/attacks/Path_Traversal
[9] (OWASP, 2023), https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
[10] (OWASP, 2023), https://owasp.org/www-community/attacks/Command_Injection

Our examination of web-based attacks concludes with a dedicated Module on *SQL Injection*, otherwise known as *SQLi*[11]. This vulnerability class is particularly important not only because of how common it is, but because it teaches us how weaknesses can arise in a system due to multiple components interacting with each other in complex ways. In the case of SQLi, a web server and a database need to both be set up in precise ways so that we as attackers cannot abuse them.

Client-Side Attacks are another very common external class of attacks. They generally deal with methods of taking advantage of human users of computer systems. In this Module, we'll learn how to perform reconnaissance on a system, attack users of common programs like Microsoft Office, and even how to abuse Microsoft Library Files.

1.3.3 Other Perimeter Attacks

It is relatively common to encounter various types of external-facing services on a penetration test that are vulnerable to different kinds of attacks. However, as penetration testers we will rarely have time to write our own exploits from scratch in the middle of an engagement.

Luckily, there are several ways in which we can benefit from the experience of the information security community. Locating Public Exploits will portray several different means of working with exploits that are available on Kali Linux and *on the internet*[12]. Then, *Fixing Exploits* will help us adapt these exploits to suit our specific needs.

We then explore the very surface of a very exciting subject: *Anti Virus Evasion*. While *Anti Virus* (AV) evasion isn't itself a perimeter attack, having some knowledge of how to avoid AV will be helpful since most modern day enterprises do deploy AV solutions.

Finally, we complete our review of perimeter attacks with an analysis of cryptography and *Password Attacks*. Weak or predictable passwords are extremely common in most organizations. This Module covers how to attack network services and how to obtain and crack various kinds of credentials.

1.3.4 Privilege Escalation and Lateral Movement

Once we obtain access to a machine, we suddenly have a whole set of new actions and activities open to us. We may want to increase our *privileges*[13] on the machines so that we can fully control it, or we might want to use it to gain access to other machines on the network.

Windows Privilege Escalation demonstrates how after compromising a Windows target, we can use our new legitimate permissions to become an Administrator. We will learn how to gather information, exploit various types of services, and attack different Windows components.

Then, *Linux Privilege Escalation* goes through the same process with Linux targets and obtaining root level permissions. It reinforces the methodology learned in the previous Module and covers Linux-specific techniques.

[11] (OffSec, 2023), https://www.offsec.com/offsec/start-studying-security-with-sqli
[12] (OffSec, 2023), https://www.exploit-db.com/
[13] (Wikipedia, 2023), https://en.wikipedia.org/wiki/Privilege_escalation

Escalating permissions is instrumentally important on an engagement because doing so gives us more access. But as penetration testers, we always want to ask ourselves what the biggest impact our attacks can have on the network to provide the most value for our clients. Sometimes, it can be even more effective to gain access to another machine owned by the organization. When we move from one machine to another on the same network, we call this *pivoting*[14], and when we move into another subnetwork we call this *tunneling*[15]. *Port Redirection and SSH Tunneling* covers the basics of these persistence skills, while *Tunneling through Deep Packet Inspection* showcases a particular technique that can be used to evade a common network-layer defense.

We wrap up this portion of the course with an exploration of *The Metasploit Framework (MSF)*[16]. MSF is a powerful set of tools that help us automate many of the enumeration and exploitation steps we've learned so far.

1.3.5 Active Directory

Active Directory[17] is one of the most complex and important technologies for us to learn as penetration testers because it is ubiquitous in today's enterprise environment. PWK dedicates three Modules to this area: *Active Directory Introduction and Enumeration* paints a picture of how to think specifically about Windows machines in the context of an Active Directory domain. We will learn how to gather information and set ourselves up to more thoroughly compromise a network.

Then, *Attacking Active Directory Authentication* provides us with several techniques to increase our presence within the network by attacking or bypassing authentication protocols. Finally, *Lateral Movement in Active Directory* helps us understand how to apply many of the pivoting concepts we've previously learned in complex AD environments.

1.3.6 Challenge Lab Preparation

The final two PWK Modules represent a bridge between the text, video, and exercise based learning modalities and the Challenge Labs themselves. By this point the learner will have completed over 300 exercises, including the compromise of approximately 25 machines. Now it's time to put it all together. In *Assembling the Pieces*, we walk the learner through a simulated penetration test of five machines. Techniques from *Information Gathering* all the way through *Lateral Movement in Active Directory* are required to successfully compromise the domain. Learners will be able to follow along and see exactly how we think about targeting a new environment from start to finish.

Finally, *Trying Harder: The Challenge Labs* provides a set of instructions and some further detail on the Challenge Labs. We highly recommend completing all the Modules including *Assembling the Pieces* before beginning with the Challenge Labs!

---

[14] (NIST, 2022), https://csrc.nist.gov/glossary/term/pivot#:~:text=Definition(s)%3A,persistent%20threat%20(APT)%20attacks.
[15] (Wikipedia, 2023), https://en.wikipedia.org/wiki/Tunneling_protocol
[16] (Rapid7, 2022), https://www.metasploit.com/
[17] (Microsoft, 2022), https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain services-overview

This introduction Module helped orient us to begin with PEN200. We've set up our attacking environment and connected to the PWK labs. We learned a little bit about the pedagogical design of the course, and received a summary of each Module. Now it's time to roll up our sleeves and get started!