This report has been prepared to report my learnings from the Udemy course titled 'Ethical Hacking from Scratch: Complete Bootcamp 2023'. This course can be accessed at Ethical Hacking from Scratch: Complete Bootcamp 2023 | Udemy. This course has been created by Andrei Dumitrescu and the Crystal Mind Academy.

The following sections of this course have been covered in this report:

1. Getting Started (8min)
2. Setting the Hacking Environment (20min)
3. Cryptography and Stenography (2hr 37min)

## Section 1: Getting Started

Before beginning the ethical hacking learning journey, we first need to understand what ethical hacking is and why we need it. All around us, we hear news about critical businesses and sometimes governments facing huge losses as a result of their software systems and networks being compromised by malicious attackers. As more and more individuals and businesses move onto the internet on a daily basis, it is now more important than ever to be able to protect and safeguard our digital systems. As the world becomes more connected, concepts of network security become paramount.

The unfortunate fact is that most networks and devices that we use in our day-to-day lives can be easily exploited if not properly secured and defended. A well-secured network system helps protect us and our valuable software systems from the risk of data loss and theft. The number of cybersecurity jobs are on a steep rise owing to the ever-growing number of cybercriminals and cybercrimes.

Having established the need to better secure our digital systems, now we need to understand what ethical hacking is. Ethical hacking is the process of attempting to break into a protected system with the explicit permission of the system owner. Such attempts will help security professionals identify and patch critical security vulnerabilities that might otherwise have been attacked by malicious actors. An ethical hacker's objective is to attempt to think and act from the point of view of a malicious actor. This approach helps to better defend against real-world attack scenarios. A key point to note here is that ethical hacking can only be performed once the owner of the system has clearly stated their permission to such processes.

The most critical objective of the security industry is to protect three critical properties, namely, confidentiality, integrity and availability. These three properties together form the Security Triad. Let's look at each of these properties one by one.

# Section 2: Setting the Hacking Environment

In this course, we will be using the Linux operating system to set up the ethical hacking lab environment. Since I am working on a Windows system, we need to install a Linux virtual machine to be used as the hacking and penetration testing operating system. Many hundreds of Linux distributions exist and are available. And hacking labs do not depend on any specific Linux distribution. However, the course instructor recommends the use of Kali Linux.

Kali Linux is the world's most popular offensive security optimized Linux distribution. Kali Linux includes more than 600 penetration testing tools including well-known Metaspoilt, Nmap (Network Mapper) and Aircrack-ng. Apart from this, Kali Linux also offers a wide variety of obscure and specialized tools including tools ued for information gathering, vulnerability analysis, wireless attacks, web application assessment, exploitation, stress testing and forensic tools. It is important to note that Kali Linux is specifically optimized for penetration testers and security specialists. Also, Kali is free and opensource. The major advantage of Kali is that we do not have to spend time and effort to separately set up required tools as most of them will already be installed, configured and ready to be launched.

Another Linux distribution that is popularly used by cybersecurity professionals is the Parrot OS. One reason that an individual might prefer Parrot OS instead of Kali Linux is because the former is extremely lightweight and easy on resources, hence making it suitable for anyone who might be using an old system. Like the Kali Linux distribution, Parrot OS is also based on Debian and designed with the specific use of security and privacy in mind. Kali Linux and Parrot OS are the two most popular Linux distributions in the cybersecurity community.

Next, we are going to look at how to install a Kali Linux virtual machine. For this, first we have to download, install and set up the Oracle VirtualBox, a free software that helps us manage the different virtual machines that we are going to use. As an alternative, VMware can also be used to achieve the same purpose. The Oracle VirtualBox needs a minimum of 5 GB RAM and 20 GB space on the disk to run smoothly. It can be downloaded from Downloads – Oracle VM VirtualBox. Once the download is successful, we can run the executable file to install the software. In addition to installing the Oracle VirtualBox, it is also recommended to install the VirtualBox Extension Pack which adds some highly beneficial features to VirtualBox like USB and audio device support. The extension pack can also be downloaded and installed from the same link.

Now that we have successfully installed the Oracle VirtualBox, we can move ahead with the installation of Kali Linux. There are different options available to download and install the Kali Linux OS.

i. It can be installed as the main OS on the system. For this, we download the ISO file and create a bootable disk. Then, we can boot the system using this disk to install Kali Linux as the main OS.

ii. The second option is to run it directly from the disk and this way, it does not have to be installed on the system.

iii. The third method, which is what we are going to use for the purpose of this course, is to install it in a virtual machine.

To install Kali Linux in a virtual machine, we can download it from Get Kali | Kali Linux. Alternatively, it can also be downloaded from Kali Linux | OffSec. The downloaded file will be an OVA (Open Virtualization Appliance) file. OVA format is used to compress installable versions of a virtual machine. When we run the OVA file, the virtual machine will be imported to the virtualization software that is available on the system (which is Oracle VirtualBox in our case). While installing the virtual machine, we will be able to change a few settings including the amount of RAM and storage space, but we will go with the default values. These parameters can be changed later as well, if required. Importing the virtual machine might take a few minutes. Once installed, we can open the VM and login. Both the username and password is set to `'kali'` by default. Please note that this user is non-privileged. By default, the network is set to NAT mode, which means it is connected to the public network and can be used to access the internet. These settings can be changed using the 'Network' tab which can be opened from the bottom-right corner of the window. To verify if we are able to connect to the internet, open the command promt and run following command:

```
ping www.google.com
```

If the ping is successful, we can confirm that the network being used is public and we can connect to the internet through this network.

After the successful installation of the virtual machine, we need to set up a few more things. Firstly, the `'kali'` user that we are logged in with is not a root user. Instead, it is a non-privileged user. Many administrative functions that we need to perform as part of ethical hacking will require root privileges and hence, we need to obtain root privileges. For this, we can use the command:

```
sudo su
```

When prompted, we can use 'kali' as the password. To verify that we have obtained root privileges, execute command - '`id`'. The desired output will be of the following form:

```
uid=0(root) gid=0(root) groups=0(root)
```

Now, we have successfully obtained root privileges and all commands executed in this terminal will have root privileges. However, it is important to note that this privilege is only temporarily available for this terminal. If we logout (with command '`exit`') and log back into another terminal with the '`kali`' user, we will again return back to being non-privileged. To attain root privilege again in the new terminal, we need to execute '`sudo su`' again. If we do not wish to temporarily become root, we can execute following command:

```
sudo apt install nmap
```

With this, we have completed the installation and setup of the Kali Linux virtual machine.

# Section 3: Cryptography and Stenography

## 3.1 Hash Algorithms

Hash algorithms are a critical part of cryptography. Hashes have a wide variety of applications like digital certificates, security certificates, blockchain technologies and many more. Understanding hash algorithms and how they work is critical to the ethical hacking learning journey. In simple words, a hash algorithm is nothing but a mathematical process which generates a unique output from a given input. The output might be referred to as a message digest or checksum. The input of the hash algorithm can take any form, from binary information to a word or phrase. There are three important properties that a mathematical function should satisfy in order to qualify as a good hash algorithm.

i. **Determinism**: This means that the output of a hash function should not change between executions given the same input.

ii. **One-wayness property**: A good hash function should be strictly one-way. Given the output, we should not be able to retrieve the input even if the hash function algorithm is known. If an input, `x` is used to generate output, `H(x)` using a hash algorithm, `H(.)`, then we should not be able to recover the value of `x` given the value of `H(x)` and knowledge of `H(.)`.

iii. **Strong Collision Resistance**: This property states that there should not be two unequal input values that will result in the same hashed output value. That is, there should be no `x` and `y`, given `x` not equal to `y`, where `H(x)= H(y)`.

iv. **Weak Collision Resistance**: This property mandates that given `x`, one should not be able to `y` (which is not equal to `x`), such that `H(x)= H(y)`.

v. **Fixed Length Output:** The output of a particular hash function should always have output of the same length, regardless of the input length.

vi. **Avalanche Effect:** Altering any one bit of the input must result in a complete different hash output from the original output.

The above three properties are a pre-requisite for a mathematical function to be useful as an effective hash algorithm in a practical security scenario. Let us look at a practical example of a hash function using Linux. Let us generate a hashed output of a password stored at '`/etc/passwd`'. Let's assume the content of this file is '`password`'. To generate the hash, execute the following command:

```
sha256sum /etc/passwd
```

```
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d
```

In the above command, the hash algorithm being used is `sha256`. sha stands for Secure Hash Algorithm and belongs to a family of popularly used hash protocols called SHA-2. `sha256sum` is the Linux application or tool which is used to implement `sha256` hash algorithm. The input to this hash function is the content of the file '`/etc/passwd`'.

Another way to implement a hash function is using the `openssl` command. To generate a hash of the '`/etc/passwd`' using the sha3-256 algorithm, we can use the following command:

```
openssl dgst -sha3-256 /etc/passswd
```

```
c0067d4af4e87f00dbac63b6156828237059172d1bbeac67427345d6a9fda48
```

Alternatively, to generate the hash of any random string ('Linux' in the used example below), using any hash algorithm (RIPEMD-160 used in the given example) we can use the following command:

```
echo -n "Linux" | openssl dgst -rmd160
```

```
45708bae009e82c05573455e659219615cdd8901
```

Two other hash functions that are now considered insecure and obsolete are md5 and sha1. To calculate the hash of any string ('Linux' in the following example) using these algorithms, we can use following commands:

SHA-1:

```
echo -n 'Linux' | sha1sum
```

```
83ad8510bbd3f22363d068e1c96f82fd0fcccd31
```

MD5:

```
echo –n 'Linux' | md5sum

edc9f0a5a5d57797bf68e37364743831
```

To get a list of hash algorithms that can be implemented using openssl, simply type `'help'` on the terminal and scroll down to section which reads `'Message Digest commands'`. All algorithms listed in this section can be used as part of openssl.

An important point to note regarding hash functions is that no matter what method is used, the same hash algorithm will always produce the same hash output. To verify this, lets generate the hash output of the word 'Linux' with the sha256 algorithm in different methods.

```
echo –n "Linux" | sha512sum

4828e60247c1636f57b7446a314e7f599c12b53d40061cc851a1442004354fe


echo –n "Linux" | openssl dgst -sha512

4828e60247c1636f57b7446a314e7f599c12b53d40061cc851a1442004354fed
```

As we can see above, both types of command generate the exact same output. We can also verify this using an external online hash generator like [Generate All Hashes - MD5, SHA1, SHA3, CRC32 - Online - Browserling Web Developer Tools.]

Hash functions are popularly used in a wide range security applications. Some of the most common applications of hash functions include:

- **Password Storage**: If user passwords are stored plaintext, an attacker who hacks the password database will be able to access the account of all users. An alternative is storing the passwords in an encrypted form. However, this is also not secure since if the decryption key is compromised, then again the passwords of all users can be retrieved. Instead, a secure way to store passwords are as hashes. The first time the user enters a password, it is stored as a hash on the password database. Every subsequent time the user enters the password in order to authenticate, the password is then hashed and compared with the stored hash. If the values match, the user is authenticated.

- **File Integrity:** Using hash functions, we can ensure the integrity of the file due to its one-wayness and collision resistance properties.

- **Digital Signatures and Certificates**

- **Data Structures (Programming Languages)**

Although hash functions are widely used in a variety of security applications, it is important to remember that hash functions can also be vulnerable to attacks. Some of the attacks that can be implemented on cryptographic hash functions include:

i. **Collision Attack:** This attack works by finding two messages that have the same output. This is possible on hash algorithms which are not collision resistant. Collision attacks have been successfully performed against both SHA-1 and MD5 hash algorithms, which is why both these algorithms are now obsolete and useless. The SHA-1 hash algorithm was broken by Google Research Security in 2017 and required "computing power equivalent to 6500 years of single-CPU computations". Details of this can be found at SHAttered. It is important to understand here that many collisions exist in all hash algorithms, including the ones popularly used today, But their security is based on the fact that a collision cannot be found in a reasonable amount of time given the current computational capabilities.

ii. **First pre-image attack**: This involves finding the input value given its hash output. This attack can be performed if the hash algorithm does not satisfy the one-wayness property.

iii. **Second pre-image attack**: In this type of attack, given a message, the attacker attempts to find another message that has the same hash output. This attack may be possible on hash algorithms that are not weak collision resistant.

## 3.2 Full Disk Encryption

The easier it is to carry electronic devices with us these days, the easier it is for them to be lost or stolen. Mobile phones, tablets, laptops and USB sticks might have valuable and confidential information that if stolen might have dangerous consequences. This can be especially risky for a business organization which trusts its employees with keeping their devices safe. An alarmingly large number of breaches occur due to lost or stolen devices. Password protection does not mean that the files on the device cannot be hacked. Simple hacking techniques exist which allow attackers to not only hack a system and retrieve confidential data but also create a new user with root permission on the device.

One technique to prevent such breaches due to stolen/lost devices to perform full disk encryption. This is a cryptographic method that implements encryption on the entire disk. All the files, software programs and even the operating system are encrypted. Through this course, we will learn how to perform full disk encryption through dm-crypt, which is the standard when it comes to encrypting disks on Linux. Dm-crypt performs encryption at the kernel-level and offers transparent disk encryption. It also provides plausible deniability. LUKS (Linux Unified Key Setup) is designed to provide a standardized key setup and platform-independent standard on-disk format for use in different tools. With a disk being

fully encrypted by dm-crypt and LUKS, the confidential data present on a disk cannot be recovered even in the event that the device is stolen/lost.

Lets now take a look at how to fully encrypt a USB stick using dm-crypt and LUKS. Once encryption is done, we might need to install another software. For example, we need to install LibreCrypt in Windows to be able to access the encrypted drive. LibreCrypt is the only way to read UKS volumes on Windows. First, we need to install cryptsetup which is used to set up encrypted filesystems on dm-crypt. This can be done using the following command:

```
apt install cryptsetup
```

Now, we can insert the USB drive into the computer. While running Linux in a VM, we will able to see the USB drive in the properties of the VM in the USB section. This feature is only available if the VirtualBox Extension Pack was installed. To identify the name of the USB, you can use the command `'fdisk -l'`. In this case, the name of the USB is `'sdc'`. The next command will delete all data on the disk that is being encrypted. So, it is important to make sure that all required data is properly backed up.

```
cd if=/dev/urandom of=/dev/sdb status=progress
```

The above command is a way of generating random data and inserting into the USB named `sdb`. This might take a few minutes to complete depending on the size of the disk. The next steo is to initialize the LUKS partition and set the initial paraphrase either through prompts or through a key file. However, this will fail if the partition is already mounted.

```
cryptsetup –y –v luksformat /dev/sdb
```

In the above command, the `–y` option is used to interactively ask for a passphrase and the `–v` flag comes from verbose. The above command will overwrite the data on `sdb` and it cannot be recovered. Now, execute the following command:

```
cryptsetup luksOpen /dev/sdb secretdata
```

The above command will create a mapping between the disk and the spatial device file. Once the mapping is complete, we need to format the filesystem.

```
mkfs.ext4 /dev/mapper/secretdata
```

As the last step, we need to mount the encrypted filesystem to the main file tree so that we can use the disk normally. Once the filesystem is mounted, we can copy, move or erase files on the disk. It can be mounted using the below command:

```
mount /dev/mapper/secretdata /mnt
```

# Summary

In the first section, we understood what ethical hacking is and why it is needed in the first place. We also looked at how ethical hacking can help us understand the various techniques that a real-world attacker might use to attack our systems. We also took time to understand the three properties, which together form the Security Triad. These properties, which include Confidentiality, Integrity and Availability are three critical properties re the most critical objectives for any security professional trying to defend their systems. We also mentioned the importance of obtaining explicit approvals from the system owner or administrator before performing ethical hacking of a system.

In section 2, we created a virtual environment from scratch to enable us to exercise and practice ethical hacking techniques. We downloaded and set up the Oracle VirtualBox and at the same time, we also understood the additional features offered by the VirtualBox Extension Pack that we used later in Section 3 to perform a full disk encryption. We downloaded the Kali Linux virtual machine and also went through why Kali Linux is one of the most popular Linux distributions among cybersecurity professionals. We also set up a public NAT network on the Kali Linux virtual machine that allowed us to communicate with the internet.

In the third section, we gathered an in-depth understanding of hashing algorithms. We looked at why hash functions are so popular and their many critical applications in the modern security industry. We also looked at the properties of hash functions that made them useful and secure in real-world scenarios. We went through the different types of hashing algorithms including SHA-256, SHA-512, MD5 and RIPEMD-160. We looked at why the SHA-1 and MD5 hash algorithms are now considered broken and obsolete. We also looked at the possible weaknesses of hash functions and how they can be attacked.

Further, in Section 3, we also covered how to perform a full disk encryption and why it is a useful technique. It helps us protect our critical and confidential information even if our personal devices are lost or stolen. Finally, we looked at how to perform a full encryption of a USB drive.