AZERBAIJAN STATE OIL AND INDUSTRY UNIVERSITY (ASOIU)

Program: "Collaboration program to develop the Bachelor of Computer Engineering" between Azerbaijan State Oil and Industry University and Georgia State University (United States of America) / University of Siegen (Germany) – BBA Program / ZU Program

Specialty: Computer Engineering

Group: ZU 058

GRADUATE WORK TASKS

1. Subject: Types of data encryption and vulnerability detection at software infrastructures
2. Initial data:
3. Outline:
4. Simulations: Portswigger vulnerable application analyzing
5. Deadline: 10.05.2022
6. Date of receiving of the initial data:
7. Program Director: Prof. Aliyev Rafig A
8. Supervisor of Graduate work: Rzayev Ilgar E
9. Student: Movlamov Nasir S.

ABSTRACT

The numbers of security vulnerabilities that are being found today are much higher in applications than in operating systems. This means that the attacks aimed at web applications are exploiting vulnerabilities at the application level and not at the transport or network level like common attacks from the past. At the same time, the quantity and impact of security vulnerabilities in such applications has grown as well.

Many transactions are performed online with various kinds of web applications. Almost in all of them the user is authenticated before providing access to the backend database for storing all the information.

A well-designed injection can provide access to malicious or unauthorized users and mostly achieved through SQL injection and Cross-site scripting (XSS). In this thesis we are providing a vulnerability scanning and analyzing tool for various kinds of SQL injection and Cross Site Scripting (XSS) attacks. Our approach can be used with any web application, not only the known ones. As well as it supports the most famous Database management servers, namely MS SQL Server, Oracle, and MySQL. We validate the proposed vulnerability scanner by developing experiments to measure its performance. We used some performance metrics to measure the performance of the scanner which include accuracy, false positive rate, and false negative rate. We also compare the performance results of it with performance of similar tools in the literature.

TABLE OF CONTENTS

LIST OF FIGURES*

LIST OF ABBREVIATIONS

CVE	Common Vulnerabilities and Exposures
DDOS	Denial of service
CD-ROM	Compact Disc Read-Only Memory
FBI	Federal Bureau of Investigation
NSA	National Security Agency
IoT	Internet of Things
DER	Distributed Energy Resources
KGB	Komitet Gosudarstvennoy Bezopasnosti
OS	Operating system
IT	Information technology
OPSEC	Operations Security
IP	Internet Protocol
PIN	Personal identification number
2FA	Two factor authentication
CIO	Chief information officer
CSO	Chief security officer
CEO	Managing Director
HR	Human Resources
BYOD	Bring your own device
VPN	Virtual private network
AES	Advanced Encryption Standard
ICANN	Internet Corporation for Assigned Names and Numbers
DoS	Denial-of-service
TCP	Transmission Control Protocol
ICMP	Internet Control Message Protocol
XSS	Cross-site scripting

INTRODUCTION

The number of cyber security offenses is always rising. Major corporations are providing an increasing amount of information regarding new cyber risks and data breaches perpetrated by various malevolent organizations or individuals. The dangers to networks, security systems, operating systems, and data's confidentiality, integrity, and availability are continually rising. While awareness of security concerns is growing, and experts and ordinary people alike desire to learn more about security and security-related problems, so is awareness of those who aim to inflict harm and disruption in cyberspace. The war between so-called White Hat Hackers and Black Hat Hackers is an endless cycle in which both parties play a cat and mouse game day after day.

Unfortunately, the criminals who carry out these heinous actions are extremely cunning and smart, and they have the expertise and resources to breach even the most sophisticated security measures. This is a major threat to every person, community, business, and government on the planet. The increased demand for security professionals and penetration testers has sparked interest among politicians, the media, educational institutions, and those in charge of information technology, allowing for greater cyber security education and knowledge sharing.

Sharing and training others on how to run private data and systems safely can reduce the attack surface for attackers and, ideally, reduce the success rate of potential exploitations. The end-user who is subjected to socialengineering assaults is perhaps one of the most significant security dangers. Weak passwords on Internet-facing services are the second largest concern, exposing personal and protected data to outsiders. The third most serious danger is the remaining unpatched vulnerabilities in software and operating systems.

Chapter 1 Classification of attack vectors and methods

Any offensive maneuver that targets computer information systems, computer networks, infrastructures, or personal computer devices is referred to as a cyberattack. An attacker is a person or entity that tries to gain unauthorized access to data, functions, or other restricted regions of the system, possibly with malicious intent. Cyber attacks can be classified as cyber warfare or cyber terrorism, depending on the situation. Cyber attacks can be launched by sovereign states, individuals, groups, societies, or organizations, and they can come from anywhere. A cyber weapon is a product that aids in the execution of a cyber attack.

By hacking into a vulnerable system, a cyber-assault can steal, change, or destroy a specific target. Cyber assaults can vary from the installation of malware on a single computer to the attempted destruction of whole nations' infrastructure. Legal experts are working to confine the term's use to occurrences that result in bodily harm, as opposed to more common data breaches and broader hacking activity. [1]

1.1 Vulnerabilities and attacks

A flaw in design, implementation, operation, or internal control is referred to as a vulnerability. The Common Vulnerabilities and Exposures (CVE) database contains the majority of the vulnerabilities discovered. At least one functional attack or "exploit" exists for an exploitable vulnerability. Using automated tools or customized scripts, vulnerabilities can be studied, reverse-engineered, hunted, or exploited. To safeguard a computer system, it's crucial to understand the threats that may be launched against it. These threats are often categorized into one of the following categories:

Backdoor

Any secret way of evading regular authentication or security procedures in a computer system is known as a backdoor, cryptosystem, or algorithm. They could exist for a variety of reasons, such as inadequate configuration or original design. They may have been inserted by a trusted entity to give legitimate access, or by a hostile attacker for bad purposes, but they always create a vulnerability. Backdoors are notoriously difficult to detect, and they are usually discovered by someone with access to the source code of the application or extensive knowledge of the computer's operating system.

Denial-of-service attack

Denial of service (DoS) attacks use backdoors to make a system or network resource unavailable to its intended users. Attackers can deny service to individual victims, such as by repeatedly inputting incorrect passwords until the victim's

account is locked, or they can overburden a machine's or network's capabilities and block all users at once. While a network assault from a single IP address can be banned by adding a new firewall rule, many different types of Distributed denial of service (DDoS) attacks are feasible, and protecting against them is far more complicated. Such attacks can come from a botnet's zombie computers or from a variety of different sources.

Direct-access attacks

Unauthorized users who get physical access to a computer are likely to be able to copy data directly from it. They could also jeopardize security by tampering with operating systems, installing software worms, keyloggers, and covert listening devices, or employing wireless microphones. Even if normal security measures are in place, they can be circumvented by booting another operating system or tool from a CD-ROM or other bootable media. These attacks are guarded against by disk encryption and the Trusted Platform Module. [2]

Eavesdropping

Eavesdropping is the act of listening in on a private computer "conversation" (communication), usually between hosts on a network, without being detected. The FBI and NSA, for example, have employed tools like Carnivore and NarusInSight to spy on internet service provider systems. Even machines that function in a closed system (that is, without any contact with the outside world) can be eavesdropped on by monitoring the hardware's feeble electromagnetic signals; the NSA has a specification for these attacks called TEMPEST.

Multi-vector, polymorphic attacks

A new class of multi-vector, polymorphic cyber assaults emerged in 2017, combining various forms of attacks and changing shape to bypass cybersecurity protections as they spread.

Phishing

Phishing is the deception of users in order to get sensitive information such as usernames, passwords, and credit card information. Phishing is most commonly carried out through email spoofing or instant messaging, and it frequently urges people to enter information on a false website that looks and feels nearly identical to the authentic one. Personal information, such as log-in details and passwords, is frequently requested on the bogus website. The individual's real account on the real website can then be accessed using this information. Phishing is a type of social engineering that takes advantage of a victim's confidence. Attackers are employing ingenious methods to get access to legitimate accounts.

Attackers frequently email false electronic bills to people, claiming that they have

purchased music, applications, or other items and asking them to click a link if the purchases were not allowed. [3]

Privilege escalation

Privilege escalation defines a situation in which an attacker with restricted access can elevate their privileges or access level without authorization. A regular computer user, for example, might be able to exploit a system vulnerability to get access to protected data, or even become "root" and have full unlimited access to the system.

Reverse engineering

Reverse engineering is the process of dismantling a man-made thing in order to disclose its designs, code, architecture, or extract knowledge from it; it is analogous to scientific study, with the exception that scientific research is concerned with natural phenomena. [4]

Side-channel attack

Any computing system has an impact on its surroundings in some way. This impact on its surroundings can be caused by a variety of factors, ranging from electromagnetic radiation to residual effects on RAM cells, which can lead to a Cold boot attack, to hardware implementation flaws that allow access to and/or guessing of additional variables that are typically unavailable. In a side-channel attack, the attacker would gather such knowledge about a system or network in order to predict its internal state and, as a result, gain access to information that the victim believes is secure. [5]

Social engineering

In the context of computer security, social engineering entails impersonating a senior executive, a bank, a contractor, or a customer to persuade a user to reveal secrets such as passwords, card numbers, and so on, or to grant physical access by, for example, impersonating a senior executive, a bank, a contractor, or a customer. This usually entails taking advantage of people's trust and relying on their cognitive biases. A popular fraud involves impersonating their CEO and sending emails to accounting and finance department officials asking immediate action. The FBI reported in early 2016 that "business email compromise" (BEC) schemes had cost US businesses more than $2 billion in the previous two years.

Spoofing

Spoofing is the act of impersonating a legitimate entity by falsifying data (such as an IP address or login) in order to obtain information or resources that are otherwise unavailable. Spoofing can take several forms, including:

- When an attacker forges an email's sending (or source) address, this is known as email spoofing.
- An attacker changes the source IP address of a network packet to hide their identity or impersonate another computing machine.
- MAC spoofing is when an attacker changes their network interface controller's Media Access Control (MAC) address to hide their identity or impersonate someone else.
- Biometric spoofing occurs when an attacker creates a phony biometric sample in order to impersonate another user.

Tampering

A malicious modification or alteration of data is referred to as tampering. Examples include the so-called Evil Maid assaults and the installation of surveillance capability into routers by security services.

Malware

Malicious software (malware) placed on a computer can leak personal information, give the attacker control of the machine, and permanently erase data.

1.2. Systems at risk

The growing number of computer systems, as well as the rising dependence on them by individuals, corporations, industries, and governments, means that an increasing number of systems are vulnerable. [6]

Financial systems

Financial regulators and financial institutions' computer systems, including as the US Securities and Exchange Commission, SWIFT, investment banks, and commercial banks, are frequent targets for hackers attempting to influence markets and profit illegally. Websites and programs that take or store credit card numbers, stockbroker accounts, and bank account information are also common hacking targets due to the potential for immediate financial gain from shifting money, making purchases, or selling the information on the black market. Customer account information and PINs were also obtained by interfering with in-store payment systems and ATMs.

Utilities and industrial equipment

Many utilities, including telecommunications, the electrical grid, nuclear power plants, and the opening and closing of valves in water and gas networks, employ computers to control functions. Such devices are vulnerable to assault if they are

linked to the Internet, but the Stuxnet virus proved that even equipment managed by computers that are not connected to the Internet can be vulnerable. The Department of Homeland Security's Computer Emergency Readiness Team examined 79 hacking incidents involving energy companies in 2014.

Aviation

The aviation sector relies heavily on a number of complicated systems that could be targeted. Controlling aircraft over the oceans is very perilous since radar observation only extends 175 to 225 miles offshore, and a single power loss at one airport can have worldwide consequences. Much of the structure is predicated on radio signals that might be disrupted, and controlling planes overseas is especially risky because radar monitoring only reaches 175 to 225 miles offshore. An inside-the-plane strike is also a potential.

Air navigation service providers are moving to develop their own dedicated networks in Europe, with the (Pan-European Network Service) and NewPENS, and in the United States, with the NextGen initiative.

A successful attack can result in everything from loss of secrecy to system integrity, air traffic control disruptions, aircraft loss, and even death.

Consumer devices

Desktop computers and laptops are frequently targeted with the purpose of obtaining passwords or financial account information, or to build a botnet to attack another target. Smartphones, tablet computers, smart watches, and other mobile devices with sensors such as cameras, microphones, GPS receivers, compasses, and accelerometers may be exploited and collect personal information, including sensitive health information. Any of these devices' WiFi, Bluetooth, or cell phone networks might be exploited as attack vectors, and sensors could be remotely enabled after a successful breach.

The growing number of home automation gadgets, such as the Nest thermostat, might be targets as well. [7]

Large corporations

Targeting large corporations is a typical occurrence. Many attacks involve data breaches and are targeted at financial gain through identity theft. Home Depot, Staples, Target Corporation, and the most recent Equifax breach are just a few examples of companies that have lost millions of customers' credit card information.

Medical records have been used in identity theft, health insurance fraud, and impersonating patients in order to obtain prescription medicines for recreational or

reselling purposes. Despite the fact that cyber risks are on the rise, 62% of all businesses did not expand their security training in 2015.

Automobiles

Engine timing, cruise control, anti-lock brakes, seat belt tensioners, door locks, airbags, and advanced driver-assistance systems are all standard features on numerous models. Additionally, connected cars may talk with onboard consumer gadgets and the cell phone network via WiFi and Bluetooth. Self-driving cars will presumably be considerably more complicated. All of these systems pose some level of security risk, and such concerns have gotten a lot of attention.

A malicious compact disc might be used as an attack vector, and the car's inbuilt microphones could be utilized to listen in on conversations. The threat is significantly greater if access is acquired to a car's internal controller area network – and in a highly publicized 2015 test, hackers remotely carjacked a vehicle from a distance of 10 miles and drove it into a ditch. Manufacturers are responding in a variety of ways, with Tesla in 2016 putting out "over the air" security updates to its cars' computer systems. In September 2016, the US Department of Transportation issued some preliminary safety requirements for autonomous vehicles and urged states to develop standardized laws.

Government

Activists and foreign countries frequently assault government and military computer networks. Because local and regional government infrastructure is now entirely computerized, such as traffic light controls, police and intelligence agency communications, employee records, student data, and financial systems, they are all possible targets. Passports and government identification cards that control entry to RFID-enabled facilities are subject to cloning.

Internet of things and physical vulnerabilities

The Internet of Things (IoT) is a network of physical items including gadgets, cars, and buildings that are equipped with electronics, software, sensors, and network connectivity to collect and share data. There have been concerns that this is being created without adequate consideration of the security risks.

While the Internet of Things allows for a more direct integration of the physical world with computer-based systems, it also allows for abuse.

As the Internet of Things grows in popularity, cyberattacks are more likely to become a physical (rather than just a virtual) hazard. If a front door's lock is connected to the Internet and can be locked/unlocked from a phone, a thief might break into the house with the touch of a button on a stolen or hacked phone. In a world governed by IoT-enabled gadgets, people could lose a lot more than their

credit card details. Thieves have also employed electronic techniques to get around hotel door locks that aren't connected to the Internet. A cyber-kinetic attack is a type of cyber-attack that targets physical infrastructure and/or human lives. As IoT devices and appliances become more popular, cyber-kinetic attacks may become more common and devastating.

Medical systems

Medical devices, including in-hospital diagnostic equipment and implanted devices such as pacemakers and insulin pumps, have either been successfully hacked or have potentially lethal vulnerabilities demonstrated. Many allegations of hospitals and healthcare institutions being hacked have surfaced, including ransomware attacks, Windows XP exploits, viruses, and data breaches involving sensitive data kept on hospital servers. The US Food and Drug Administration issued suggestions on how medical device manufacturers should maintain the security of Internet-connected devices on December 28, 2016, but no enforcement mechanism.

Energy sector

According to Daily Energy Insider, the possibility of a cyber attack on distributed generation systems is serious. An attack could result in a long-term loss of power in a vast area, and such an attack could have just as serious implications as a natural disaster. The District of Columbia is considering establishing a Distributed Energy Resources (DER) Authority, with the purpose of providing customers with more insight into their personal energy use and allowing Pepco, the local electric company, to better predict energy demand. The proposal in Washington, D.C., on the other hand, would "enable third-party vendors to build additional points of energy distribution, potentially increasing the opportunity for cyber attackers to endanger the electric grid."

1.2.1 Impact of security breaches

Security breaches have resulted in massive financial losses, but because there is no standard technique for calculating the cost of an incident, the only information available is what the businesses involved make public. Various computer security consultancies estimate the total global damages caused by virus and worm assaults, as well as other dangerous digital activities. In 2003, these businesses expected losses ranging from $13 billion (for computer viruses alone) to $226 billion. (This covers all types of covert assaults). The validity of these figures is heavily contested, and the approach employed is largely anecdotal.

Rough estimates of the financial cost of security breaches, on the other hand, can help businesses make sensible investment decisions. The amount a company spends to protect information should be a small fraction of the expected loss (i.e.,

the expected value of the loss resulting from a cyber/information security breach), according to the classic Gordon-Loeb Model for determining the optimal level of investment in information security. [8]

1.2.2 Attacker motivation

The motivations for computer security breaches differ from attacker to attacker, just as they do with physical security. Some are thrill seekers or vandals, while others are activists or criminals looking for a quick buck. State-sponsored attackers are increasingly frequent and well-funded, although they began with amateurs like Markus Hess, who hacked for the KGB, as Clifford Stoll recounts in The Cuckoo's Egg.

Recent attacker motivations can also be linked to extremist organizations seeking political advantage or disrupting social agendas. The expansion of the internet, mobile technologies, and low-cost computer devices has increased capabilities while also posing a threat to surroundings deemed critical to operations. All key targeted environments are vulnerable to compromise, prompting a series of proactive research on how to mitigate risk by considering the goals of these types of attackers. There are several significant contrasts between the motivations of hackers and those of nation-state actors trying to attack based on ideological preferences.

Identifying what would trigger an assault on a system and who might be motivated to breach it is a typical aspect of threat modeling for every system. Depending on the system to be secured, the level and detail of measures will vary. Even when the fundamental technology is identical, the risks that a home personal computer, a bank, and a classified military network confront are vastly different. [9]

1.3. How to avoid and prevent the most common types of cyber attacks

An attack vector is a mechanism or method by which an attacker or hacker gains access to a computer or network server in order to deliver a payload or damaging impact. Hackers can utilize attack vectors to exploit system weaknesses, such as the human factor. A security hole can be found in a piece of software or a computer's operating system (OS). A security vulnerability can occur as a consequence of a programming error in an application or a faulty security setup. Low-tech hacks are also feasible, such as obtaining an employee's security credentials or breaking into a facility. Hackers are always looking for possible access points into organizations and people' systems, applications, and networks. They may even target physical facilities or identify vulnerable users and internal personnel who may intentionally or unintentionally divulge their information

technology (IT) access credentials. Although these terms are commonly used interchangeably, they are not synonymous. An attack vector differs from an attack surface in that the vector is how an intruder gains access, whereas the attack surface is the target of the attack. [10]

How can attackers use attack vectors to their advantage?

There are several ways to compromise computer systems, infrastructure, networks, operating systems, and IoT devices by exposing, altering, disabling, destroying, stealing, or gaining unauthorized access to them.

Attack vectors can be divided into passive and aggressive attacks in general.

Vector Exploits for Passive Attacks

Passive attack vector exploits, such as typosquatting, phishing, and other social engineering-based assaults, are attempts to acquire access or make use of information from the system without affecting system resources.

Exploits using Active Attack Vectors

Active attack vector exploits include malware, exploiting unpatched vulnerabilities, email spoofing, man-in-the-middle attacks, domain hijacking, and ransomware, to name a few examples.

However, the majority of attack vectors are similar:

- A possible victim is identified by the attacker.
- Using social engineering, malware, phishing, OPSEC, and automated vulnerability scanning, the attacker acquires knowledge about the victim.
- The information is used by attackers to determine potential attack vectors and construct or utilize tools to exploit them.
- Attackers gain unauthorized access to a computer system and steal sensitive information or install dangerous software.
- Attackers keep a close eye on the computer or network, steal information, and take use of computational resources.

Your third- and fourth-party suppliers and service providers are an often-overlooked attack vector. No matter how advanced your internal network and information security is, if suppliers have access to critical data, they are just as much of a risk to your company. This is why measuring and mitigating third-party and fourth-party risk is critical. As a result, it must be included in your information security policy and risk management program. Consider investing in threat intelligence technologies that automate vendor risk management and automatically monitor and alert you if your vendor's security posture deteriorates. A third-party risk management framework, vendor management strategy, and vendor risk

management program are now required for every firm. Perform a cybersecurity risk assessment before choosing a new vendor to understand what attack vectors you could be bringing to your business by utilizing them, and inquire about their SOC 2 compliance. [11]

The SolarWinds supply chain assault was one of the most well-known hacks. The attack paths were investigated, however the intrusion might have been the consequence of compromised credentials or possibly access through SolarWinds' Orion IT management software's development environment.Intruders are continuously seeking out new attack vectors. Attack vectors include the following:

1. **Software vulnerabilities.** An attacker can employ a threat vector, such as malware, to obtain unauthorized access if a network, operating system, computer system, or application has an unpatched security vulnerability.
2. **Compromised user credentials.** Users can intentionally or unintentionally leak their user IDs and passwords. This can be accomplished verbally, but cyber attackers can also use a brute-force attack to get access to credentials by attempting countless combinations of user IDs and passwords until an approved pair of credentials is located. The hacker then uses these credentials to gain access to a network, system, or application.
3. **Weak passwords and credentials.** Brute-force assaults concentrate cyber attackers' efforts on weak or readily guessed user IDs and passwords. Hackers can also steal passwords by monitoring public Wi-Fi networks for when users enter their login credentials. A hacker may, for example, install keylogging software on a user's computer via an infected website or email. The keylogging application records all keystrokes made by the user, including the user's ID and password. Hackers can also get access by persuading consumers to click unsolicited email attachments with malicious links to phony websites that persuade them to hand over personal information (PII). [12]
4. **Malicious employees.** Employees who are malicious or unhappy can use their security clearances to hack into networks and systems and obtain sensitive information like as customer lists and intellectual property (IP), which they can either demand a ransom for or sell to others for nefarious purposes.
5. **Poor or missing encryption.** In rare cases, employees — or IT — may neglect to encrypt vital data stored on PCs and smartphones while out in the field. In other cases, encryption methods contain known design flaws or only encrypt and protect data with a limited number of keys.
6. **Ransomware.** Ransomware is a type of malware that encrypts data on a victim's computer and threatens to publish or prohibit access to it unless a

ransom is paid to the attacker. Ransomware may encrypt a user's files and then demand money to unlock them. The majority of ransomware is accidentally downloaded onto a computer or network by a user. It can take the form of a worm, which is malware that spreads over a network, or a Trojan, which embeds harmful software code in a downloaded file and then demands money.

7. **Phishing.** Phishing is a deceitful activity in which an attacker sends emails seeming to be from a respected firm in order to trick people into divulging personal information such as passwords or credit card details. Spear phishing is a highly focused assault that seeks unauthorized access to valuable corporate information from a single victim.
8. **Misconfigured devices.** Companies' software and hardware security might be misconfigured, leaving them open to hackers. Vendor security settings on equipment are lax, and security breaches can occur if IT does not change this equipment before placing it on networks. In other circumstances, businesses buy equipment but fail to completely set security.
9. **Trust relationships.** Security is frequently entrusted to third-party system and network suppliers, cloud providers, and business partners. When these third-party systems are hacked, the information obtained by the hackers may include sensitive information from the firms these providers serve. When a big credit card company's network gets hacked, or when a hospital system's network is hacked and sensitive patient data is taken, these are examples.
10. **Distributed denial-of-service (DDoS) attacks.** DDoS assaults flood victims with fake emails, making their system or network unworkable and their services inaccessible to their intended recipients. These assaults are frequently directed towards the web servers of financial, commercial, and government entities, and are frequently used to divert attention away from other network attacks. [13]

How can devices be protected against common vector attacks?

In order to get access to company IT assets, attackers employ a range of methods. IT's responsibility is to identify and apply the policies, tools, and strategies that are most successful in defending against these assaults as they change. A list of effective protective strategies is provided below:

- Set up strict password policies. Ascertain that usernames and passwords are of sufficient length and strength, and that the same credentials are not used to access multiple applications and systems. Use two-factor authentication (2FA) or verification procedures such as a password and a personal identification number to provide an extra layer of protection to system access (PIN).

- Install software for security monitoring and reporting. Once a possible attack by an unknown or unauthorized user or source is detected, this program monitors, detects, alarms, and even shuts down entry points to networks, systems, workstations, and edge technologies.
- On a regular basis, audit and test IT resources for vulnerabilities. IT vulnerability testing should be performed at least once a quarter, and IT resources should be examined for vulnerability by an outside IT security audit company once a year. In light of these findings, all security regulations, methodologies, and preventative tactics should be changed as soon as possible.
- Maintain a high level of IT security. Security investments are costly, and a chief information officer (CIO) and a chief security officer (CSO) must get approval from the CEO and the board of directors before proceeding. This necessitates frequent briefings and education for C-level executives so that they are aware of the importance of IT security and the consequences for the firm and its brand if IT is left unprotected.
- Users must be educated. All new workers should receive thorough training in IT security rules and procedures, with current staff receiving yearly refresher training. IT workers, particularly those working in the security field, should be up to date on the most recent security rules and procedures.
- Collaborate with the human resources department (HR). At least once every two to three years, an outside security audit company should conduct a social engineering vulnerability audit. If an employee engages in questionable behavior, IT should notify HR promptly so that appropriate action may be taken, such as meeting with the individual, restricting access, coaching the employee, or dismissing the person.
- Install all updates right away. When a hardware, firmware, or software update is released, IT should install it as soon as possible. If devices are utilized in the field, security updates should be delivered through push notifications, which automatically update software or firmware.
- Companies with a bring-your-own-device (BYOD) policy should use thin clients. All business data should be stored in a secure cloud or other enterprise system so that users may log in from home or on their own devices using a virtual private network (VPN), which is confined to a small group of users and is not exposed to the general public. This removes the storage of sensitive data on distant devices.
- On portable devices, use robust data encryption. Data encryption should be utilized anywhere sensitive data is held, whether it is on a laptop, a smartphone, a sensor, or any other form of edge device. This can be accomplished by using a robust data encryption method, such as Advanced

Encryption Standard (AES) . The US government encrypts data via AES, which employs 192- and 256-bit keys.

- Review and configure all security settings for operating systems, web browsers, security software, network hubs, and edge devices including sensors, cellphones, and routers. Companies frequently fail to alter the security settings on systems, browsers, hubs, and internet of things (IoT) devices, which come with limited default security settings. Companies should check and, if required, reset security on all new IT as a normal process.
- Physical security is essential. Even though most data breaches and security breaches target IT, physical access intrusions can occur. Data centers, computers in multiple corporate divisions and remote field offices, medical equipment, field-based sensors, and even physical file cabinets in offices are all potential targets for hackers. They should be regularly protected, safeguarded, and examined.

1.4. Stages and methods of the cyber attack

In the never-ending battle between cyber security professionals and hackers, change is unavoidable. Critical infrastructure cyber-attacks are growing increasingly widespread, complicated, and innovative. This provides a continuous challenge for cyber security teams, who must understand where their operations are vulnerable to attacks before hackers can locate them. [14]

Hackers' motivations have shifted in several recent high-profile events. Attacks are increasingly aimed at disrupting services rather than stealing data for financial benefit. Hackers have also been employing a previously unseen attack vector. Instead of directly assaulting their major targets, they have targeted less secure suppliers that those targets rely on.

Phase one: Reconnoitering a target for hacking

During the reconnaissance phase, hackers locate a susceptible target and investigate potential exploits. Anyone in the corporation might be the initial target. Attackers simply have only a single point of entry to get started. In this stage, targeted phishing emails are a frequent means of delivering malware. The goal is to get to know the target. At this point, hackers are wondering who the main individuals in the firm are, who they do business with, and what publicly available

data there is about the target organization. Company websites and online contact services like Linkedin are two obvious places to start when looking for key people in firms. Identifying suppliers and customers may require social engineering,' in which a hacker makes phony sales calls to the firm.

Hackers obtain Internet Protocol (IP) address information from publicly accessible sources and execute scans to ascertain what hardware and software the target firm is running. They look through the online register database of the Internet Corporation for Assigned Names and Numbers (ICANN).

The more time hackers spend gathering knowledge about the company's employees and systems, the more effective the hacking effort will be.

Phase two: Weaponizing information on a company

The hacker utilizes the previously collected knowledge to devise ways to get access to the target's network during the weaponization phase. This might entail crafting convincing spear phishing emails that appear like emails the target could receive from a reputable vendor or other business contact. Another hacking method is to build 'watering holes,' or false websites that seem identical to those of a merchant or a bank. This is intended to collect usernames and passwords, or to provide a free download of a malware-infected document or something else of interest. The attacker's final task in this phase is to gather the tools needed to successfully exploit any vulnerabilities discovered after gaining access to the target's network. [15]

Phase three: ‘Delivering’ the attack

The assault begins during the delivery phase. Phishing emails are sent, 'watering hole' web pages are put to the internet, and the attacker waits for all the data they require to arrive. If the phishing e-mail contains a weaponized attachment, the attacker waits for someone to open it and the malware within to 'call home' to the hacker.

Phase four: Exploiting the security breach

During the exploitation phase, the hacker begins to reap the benefits of planning and carrying out the assault. As usernames and passwords are sent to the attacker,

they are tested against web-based e-mail systems or virtual private network (VPN) connections to the enterprise network. If malware-infected attachments were emailed, the attacker gains remote access to the systems. The hacker investigates the targeted network to obtain a better understanding of its traffic flow, which systems are connected to it, and how they might be abused.

Phase five: Installing a persistent backdoor

During the installation process, the attacker assures uninterrupted network access. The hacker will do this by installing a permanent backdoor, creating administrator accounts on the network, and disabling firewall restrictions. They may even provide remote desktop access on network servers and other computers. The hacker's purpose at this stage is to ensure that they will remain in the system for as long as necessary to fulfill their goals.

Phase six: Exercising command and control

They now have unrestricted access to the whole network and administrator accounts, and all of the tools necessary for the command and control phase are in place. The attacker has the ability to look at everything, impersonate any network user, and even send emails from the CEO to all staff. Now that they have power, the hacker can lock a company's IT users out of the whole network, possibly demanding a fee to recover access.

Phase seven: Achieving the hacker’s objectives

The period of action on objectives has now begun. This might include stealing information on staff, clients, and product designs, among other things. Alternatively, an attacker might begin to disrupt the target company's activities. Not all hackers are looking for monetizable data or embarrassing emails to publish. Some people just wish to wreck havoc or inflict suffering on a corporation. If a corporation gets online orders, a hacker may, for example, shut down the ordering system or remove orders. They might even generate orders and have them dispatched to clients of the firm. If a hacker obtains access to an Industrial Control System, they can disable alarms, shut down equipment, and input new set points.

Know your enemy for greater cyber security

Following recent high-profile cyber-attacks on critical infrastructure, DNV is seeing an increase in requests from clients looking for help with the company's full cyber security expertise for operational technologies and information systems. "Demand for our unique domain knowledge for cyber security spans a wide range of industries, from energy to maritime and healthcare," Solberg noted. The often observed seven stages of a cyber-attack, on the other hand, are crucial to understanding how hackers obtain access to systems and exploit vulnerabilities." [16]

After understanding phases of cyber attack we can talk about methods of cyber attacks. Most common cyber attacks:

1. Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks

   A denial-of-service attack exhausts a system's resources, preventing it from responding to service requests. A distributed denial of service (DDoS) assault is likewise an attack on system resources, but it is launched from a large number of additional host machines infected with malicious software controlled by the attacker. In contrast to assaults that are meant to allow the attacker to obtain or enhance access, denial-of-service attacks do not deliver direct benefits to the attacker. For some of them, the gratification of service denial is sufficient. If, on the other hand, the targeted resource belongs to a commercial competitor, the profit to the attacker may be substantial. Another goal of a DoS assault is to take a system down so that another type of attack may be initiated. Session hijacking is a popular example, which I'll go into later. TCP SYN flood assault, teardrop attack, smurf attack, ping-of-death attack, and botnets are the most prevalent forms of DoS and DDoS attacks.

2. TCP SYN flood attack

   An attacker launches this attack by exploiting the buffer space during a Transmission Control Protocol (TCP) session's first handshake. The attacker's device sends connection requests to the target system's in-process queue, but it does not respond when the target system responds. When the connection queue on the target system is full, the system times out while waiting for a response from the attacker's device, causing the system to crash or become useless.

3. Teardrop attack

   This attack causes the length and fragmentation offset fields of successive Internet Protocol (IP) packets on the attacked host to overlap; the attacked system attempts but fails to reconstruct packets during the process. The target system becomes confused and crashes as a result. If users do not have the necessary fixes to protect themselves against this DoS attack, deactivate SMBv2 and block ports 139 and 445.

4. Smurf attack

   To flood a target network with traffic, this attack employs IP spoofing and the ICMP protocol. ICMP echo requests are sent to broadcast IP addresses in this sort of attack. These ICMP queries are coming from a forged "victim" address. For example, if the attacker's intended target address is 10.0.0.10, he may send a faked ICMP echo request from 10.0.0.10 to 10.255.255.255. This request would be sent to all IP addresses in the range, with all answers returning to 10.0.0.10, overloading the network. This process is repeatable and, if automated, has the potential to cause massive network congestion. To prevent your devices from this attack, disable IP-directed broadcasts on routers. This stops network devices from responding to ICMP echo broadcasts.

5. Ping of death attack

   This sort of attack employs IP packets with IP sizes exceeding the maximum of 65,535 bytes to 'ping' a target machine. Because IP packets of this size are not permitted, the attacker fragments the IP packet. The target system may encounter buffer overflows and other problems after reassembling the packet. Ping of death attacks can be prevented by employing a firewall that checks fragmented IP packets for maximum size.

6. Botnets

   Botnets are networks comprising millions of systems infected with malware and controlled by hackers in order to launch DDoS assaults. These bots or zombie systems are used to launch assaults on target systems, frequently overloading the target system's bandwidth and processing capacity. These

DDoS attacks are tough to track down since botnets are scattered around the globe.

Botnets can be mitigated by:

- RFC3704 filtering, which will refuse traffic from faked addresses and aid in tracing traffic back to its proper originating network. RFC3704 filtering, for example, will discard packets from bogon list addresses.
- Black hole filtering, which filters out unwanted traffic before it reaches a secure network. When a DDoS assault is identified, the BGP (Border Gateway Protocol) host should communicate routing modifications to ISP routers such that all traffic to victim servers is routed to a null0 interface at the next hop.

7. Session hijacking

   An attacker hijacks a session between a trusted client and a network server in this form of MitM attack. While the server continues the session, believing it is conversing with the trusted client, the attacker machine replaces its IP address for the trusted client.

   For instance, the attack may go as follows:

   - A connection is established between a client and a server.
   - The client is taken over by the attacker's machine.
   - The attacker's workstation disconnects the client from the server.
   - The attacker's computer spoofs the client's sequence numbers and inserts its own IP address for the client's.
   - The attacker's machine maintains contact with the server, and the server believes it is still in contact with the client. [17]

8. IP Spoofing

   An attacker uses IP spoofing to convince a system that it is interacting with a known, trusted entity, granting the attacker access to the system. The attacker transmits a packet to a target host using the IP source address of a known, trusted host rather than its own IP source address. The packet may be accepted and acted upon by the destination host.

9. Replay

A replay attack happens when an attacker intercepts and stores past communications before attempting to transmit them again as one of the participants. This kind is readily defeated by using session timestamps or nonces (a random number or a string that changes with time). There is currently no one technique or configuration that can prevent all MitM attacks. Encryption and digital certificates, in general, constitute a strong barrier against MitM attacks, ensuring both the secrecy and the integrity of communications. A man-in-the-middle attack, on the other hand, can be placed into the middle of conversations and render encryption ineffective — for example, attacker "A" intercepts person "P's" public key and replaces it with his own public key. Anyone wishing to send an encrypted message to P using P's public key is inadvertently using A's public key. As a result, A can read the message meant for P and then transmit it to P, encrypted with P's true public key, and P will never discover that the message was compromised. A might potentially change the message before resending it to P. As you can see, P is encrypting his data and believes it is safe, but it is not because of the MitM attack. As a result, A can read the message meant for P and then transmit it to P, encrypted with P's true public key, and P will never discover that the message was compromised. A might potentially change the message before submitting it to P. As you can see, P is encrypting his data and believes it is safe, but it is not because of the MitM attack. So, how can you be certain that P's public key belongs to P and not A? To address this issue, certificate authorities and hash algorithms were developed. When person 2 (P2) wants to send a message to P and P wants to ensure that A does not read or change the message and that the message originated from P2, the following approach must be used:

- P2 generates a symmetric key that is encrypted using P's public key.
- P2 provides to P the encrypted symmetric key.
- P2 computes the message's hash function and digitally signs it.
- P2 uses the symmetric key to encrypt his message and its signed hash before sending it to P.
- P may get the symmetric key from P2 since he is the only one who holds the private key to decrypt the encryption.

- Because he owns the symmetric key, P and only P can decode the symmetrically encrypted message and signed hash.
- He may verify that the message has not been tampered with by computing the hash of the received message and comparing it to the digitally signed one.
- P can additionally confirm to himself that P2 was the sender because only P2 can sign the hash and verify it using P2's public key.

10. Phishing and spear phishing attacks.

The practice of sending emails that look to be from reputable sources in order to get personal information or persuade users to do something is known as phishing. It mixes social engineering and technical sleight of hand. It might be an attachment to an email that contains malware that infects your machine. It might potentially be a link to a malicious website designed to fool you into installing malware or disclosing sensitive information.

Spear phishing is a type of phishing that is extremely targeted. Attackers spend time researching their targets and crafting messages that are personal and relevant to them. As a result, spear phishing might be difficult to identify and much more difficult to combat. Email spoofing is one of the most straightforward techniques for a hacker to begin a spear phishing attack. This is when the information in the email's "From" field is forged, making it appear to be from someone you know, such as your management or a partner organization. Website copying is another tactic employed by con artists to lend credence to their story. They impersonate respectable websites in order to dupe you into providing personally identifiable information (PII) or login credentials. [18]

You can use the following tactics to lower your chances of getting phished:

- **Hovering over the links.** Hover your cursor over the link without clicking! Simply move your cursor over the link to see where it takes you. Use critical thinking to comprehend the URL.
- **Analyzing email headers.** Email headers describe how an email arrived at your address. The "Reply-to" and "Return-Path" options should point to the same domain as the one specified in the email.

- **Sandboxing.** You may test email content in a sandbox environment, tracking behavior such as opening attachments and accessing links within emails.
- **Critical thinking.** Do not accept an email as genuine only because you are busy, stressed, or have 150 unread messages in your inbox. Take a moment to review the email.

11. Drive-by attack

Drive-by download attacks are a common technique for malware to spread. Hackers seek out susceptible websites and introduce harmful scripts into the HTTP or PHP code on one of the pages. This script may directly install malware on the computer of a site visitor, or it may redirect the victim to a site controlled by the hackers. When you access a site, read an email, or see a pop-up window, you may experience a drive-by download. Unlike many other types of cyber security assaults, a drive-by attack does not require a user to actively authorize the attack — you do not need to click a download button or open a malicious email attachment to become infected. A drive-by download might exploit a security flaw in an app, operating system, or web browser as a result of failed or missed updates. Keep your browsers and operating systems up to date to avoid drive-by attacks, and avoid websites that may contain malicious malware. Stick to the sites you're familiar with, but keep in mind that even these might be hacked. Keep as few unneeded applications and apps on your smartphone as possible. The more plug-ins you have, the more opportunities for drive-by assaults to exploit. [19]

12. Password attack

Obtaining passwords is a frequent and successful attack tactic since passwords are the most often used way to authenticate users to an information system. Looking around a person's desk, "sniffing" the network connection to collect unencrypted passwords, employing social engineering, acquiring access to a password database, or simply guessing can all be used

to obtain access to a person's password. The last method can be used in either a random or systematic manner:

- **Brute-force.** Brute-force password guessing entails employing a random strategy by attempting various passwords and hope that one of them works. Using logic, attempt passwords relating to the person's name, work title, hobbies, or comparable stuff.
- **Dictionary attack.** A dictionary attack uses a dictionary of common passwords to gain access to a user's computer and network. One way is to duplicate an encrypted file holding the passwords, then encrypt a dictionary of frequently used passwords using the same encryption and compare the results.

You should set an account lockout policy that locks the account after a few invalid passwords tries to protect yourself from dictionary or brute-force assaults. You can set it up appropriately by following these account lockout recommended practices.

13. SQL injection attack

With database-driven websites, SQL injection has become all too frequent. It happens when a bad guy uses the input data from the client to conduct a SQL query on the database. In order to perform specified SQL instructions, SQL commands are placed into data-plane input (instead of the login or password, for example). A successful SQL injection exploit can read sensitive data from a database, edit database data (insert, update, or delete), perform database management activities (such as shutdown), retrieve the content of a specified file, and, in certain situations, issue commands to the operating system.

For example, a web form on a website may ask for a user's account name and then send it to the database using dynamic SQL to retrieve the related account information:

“SELECT * FROM users WHERE account = ‘“ + userProvidedAccountNumber +”’;”

While this works for consumers who enter their account number correctly, it exposes a security hole for attackers. For example, if someone entered "' or '1' Equals '1' in the account number field, the query string would be:

"SELECT * FROM users WHERE account = '' or '1' = '1';"

Because '1' = '1' always evaluates to TRUE, the database will return the data for all users instead of just a single user.

The fact that SQL makes no distinction between the control and data planes makes it vulnerable to this form of cyber security assault. SQL injections are most effective when a website employs dynamic SQL. Due to the ubiquity of older functional interfaces, SQL injection is particularly frequent in PHP and ASP applications. Because of the nature of the programmatic interfaces provided, J2EE and ASP.NET applications are less prone to have easily exploited SQL injections. Apply the least privilege model of permissions in your databases to defend yourself from SQL injection attacks. Stick to stored procedures and prepared statements (as long as they don't include any dynamic SQL) (parameterized queries). In order to prevent injection attacks, the code that is performed on the database must be robust. Additionally, at the application level, check input data against a white list. [20]

14. Cross-site scripting (XSS) attack

In SS attacks, third-party online resources are leveraged to run scripts in the victim's web browser or scriptable application. The attacker injects a payload containing malicious JavaScript into a website's database. When a client requests a page from the website, the webpage transmits the requested page to the victim's browser, including the attacker's payload contained in the HTML body, which executes the malicious script. It may, for example, send the victim's cookie to the attacker's server, from which the attacker could extract it and use it to hijack the victim's session. When XSS is utilized to exploit further vulnerabilities, the most disastrous results occur. An attacker can exploit these issues to steal cookies, track keystrokes, capture screenshots, discover and collect network information, and remotely access and administer the victim's PC. While XSS may be used in VBScript,

ActiveX, and Flash, JavaScript is the most commonly exploited, owing to its widespread use on the internet.

Developers can sanitize data submitted by users in an HTTP request before reflecting it back to prevent XSS attacks. Before echoing anything back to the user, such as the values of query parameters during searches, make sure all data is checked, filtered, or escaped. Special characters like?, &, /, >, and spaces are converted to their HTML or URL encoded equivalents. Allow users to opt out of client-side scripting. [21]

15. Eavesdropping attack

Interception of network communication is used in eavesdropping attacks. Passwords, credit card numbers, and other personal information that a user may be transferring over the network can be obtained by eavesdropping. Passive or aggressive eavesdropping is possible:

- **Passive eavesdropping.** By listening to the network's message transmission, a hacker can discover the information.
- **Active eavesdropping.** By impersonating a friendly unit and sending inquiries to transmitters, a hacker actively obtains information. Probing, scanning, or meddling are all terms for the same thing.

Passive eavesdropping assaults are frequently more difficult to detect than active eavesdropping attacks, because active attacks require the attacker to first gather information about friendly units through passive eavesdropping.

Data encryption is the most effective eavesdropping deterrent.

16. Birthday attack

Birthday attacks target hash algorithms, which are used to check the integrity of messages, software, and digital signatures. A message digest (MD) of constant length is produced by a hash function, regardless of the length of the input message; this MD uniquely describes the message. When a hash function is used to process two random messages, the birthday attack refers to the likelihood of discovering two random messages that yield the same

MD. If an attacker calculates the same MD for his message as the user, he may securely replace the user's message with his, and even if the receiver checks MDs, he will not be able to detect the replacement. [22]

17. Malware attack

Malicious software is unwelcome software that is installed on your computer without your permission. It can spread by attaching itself to legal code, hiding in beneficial apps, or replicating itself throughout the Internet. The following are some of the most frequent malware types:

- **Macro viruses.** These viruses affect Microsoft Word and Excel, among other programs. Macro viruses attach themselves to the initialization sequence of a program. The virus executes instructions before handing control to the program when it is opened. The virus multiplies and attaches itself to other programs on the computer system.
- **File infectors.** Viruses that infect executable code, such as.exe files, are known as file infectors. When the code is loaded, the virus is installed. Another variant of a file infector links to a file by producing a virus file with the same name but a.exe extension. As a result, the viral code will run when the file is opened.
- **System or boot-record infectors.** On hard drives, a boot-record virus attaches itself to the master boot record. When the system boots up, it checks the boot sector for viruses and loads them into memory, where they can spread to other drives and computers.
- **Polymorphic viruses.** These viruses hide their presence through a series of encryption and decryption cycles. A decryption software first decrypts the encrypted virus and its accompanying mutation engine. The virus then infects a section of code. The virus encrypts the mutation engine and a copy of the virus with an algorithm corresponding to the new decryption procedure, and the mutation engine produces a new decryption routine. The mutation engine and virus's encrypted package is connected to fresh code, and the process is repeated. Because of the numerous alterations to their source code, such viruses are difficult to detect yet have a high amount of entropy.

This characteristic may be used to detect them by anti-virus software or free programs like Process Hacker.

- **Stealth viruses.** These viruses hide their presence through a series of encryption and decryption cycles. A decryption software first decrypts the encrypted virus and its accompanying mutation engine. The virus then infects a section of code. The virus encrypts the mutation engine and a copy of the virus with an algorithm corresponding to the new decryption procedure, and the mutation engine produces a new decryption routine. The mutation engine and virus's encrypted package is connected to fresh code, and the process is repeated. Because of the numerous alterations to their source code, such viruses are difficult to detect yet have a high amount of entropy. This characteristic may be used to detect them by anti-virus software or free programs like Process Hacker.
- **Trojans.** A Trojan, often known as a Trojan horse, is a harmful software that hides in a helpful application. Trojans do not self-replicate, which is one of the primary differences between viruses and Trojans. A Trojan can provide a back door that attackers can use in addition to performing assaults on a system. A Trojan, for example, can be configured to open a high-numbered port so that a hacker can listen and then launch an assault.
- **Logic bombs.** A logic bomb is malicious software that is added to a program and is activated when a specified event occurs, such as a logical condition or a specific date and time.
- **Worms.** Worms vary from viruses in that they do not connect to a host file and instead spread through networks and computers as self-contained programs. Worms are typically propagated via email attachments, with the worm software being activated when the attachment is opened. The worm sends a copy of itself to every contact in an infected computer's email address in a conventional worm attack. A worm propagating throughout the internet and overloading email servers can cause denial-of-service attacks on network nodes in addition to undertaking malicious activities.
- **Droppers.** A dropper is an application that is used to infect computers with viruses. Virus-scanning software may not identify the dropper in

many cases since it is not infected with dangerous code. A dropper can also connect to the internet and obtain virus updates from a compromised system's virus software.

- **Ransomware.** Ransomware is a sort of software that prevents the victim from accessing his or her data and threatens to publish or erase it unless a ransom is paid. While some simple computer ransomware can be unlocked by a skilled person, more powerful software employs a technique known as cryptoviral extortion, which encrypts the victim's data in such a way that they are virtually impossible to recover without the decryption key.
- **Adware.** Advertising banners are displayed while any program is running, and adware is a software application utilized by businesses for marketing goals. Adware may be downloaded to your system automatically when surfing any website and viewed through pop-up windows or a bar that displays on your computer screen.
- **Spyware.** Spyware is a sort of application that is placed on a user's computer or browser to collect information about them. It secretly records everything you do and delivers the information to a remote user. It can also use the internet to obtain and install additional malicious apps. Spyware is similar to adware in that it is a distinct software that is installed unintentionally when you install another freeware program.

Understanding the offensive is essential to mounting a strong defense. The ten most prevalent cyber-security attacks used by hackers to disrupt and infiltrate information systems were discussed in this article. As you can see, attackers have a variety of techniques for gaining unauthorized access to critical infrastructure and sensitive data, including DDoS attacks, malware infection, man-in-the-middle interception, and brute-force password guessing. The methods used to counteract these dangers differ, but the fundamentals of security remain the same: Keep your systems and anti-virus databases up to date, train your employees, configure your firewall to whitelist only the ports and hosts you need, use strong passwords, use a least-privilege model in your IT environment, make regular backups, and audit your IT systems for suspicious activity on a regular basis. [23]

Chapter 2 Analysis and detection of vulnerabilities in web application

The scanning stage is the most important part of the process. At this stage, the tester should take a more proactive position in the information gathering process, using previous targets' provided information to begin looking for new vulnerabilities. During this procedure, a tester should analyze the application for entry points and weaknesses, then test and validate the findings (see figure 30). At this point, the tester begins to play a more active role in testing, interacting directly with the application and its services. [24]

Figure 1.1: Main Procedures of the Scanning Stage

Active scan. During the scanning stage, a tester should take a more proactive role in scanning, interacting directly with target components. It can be performed using automated or human technologies to discover and evaluate security in view of the potential vulnerabilities (see figure 1.2 and figure 2).

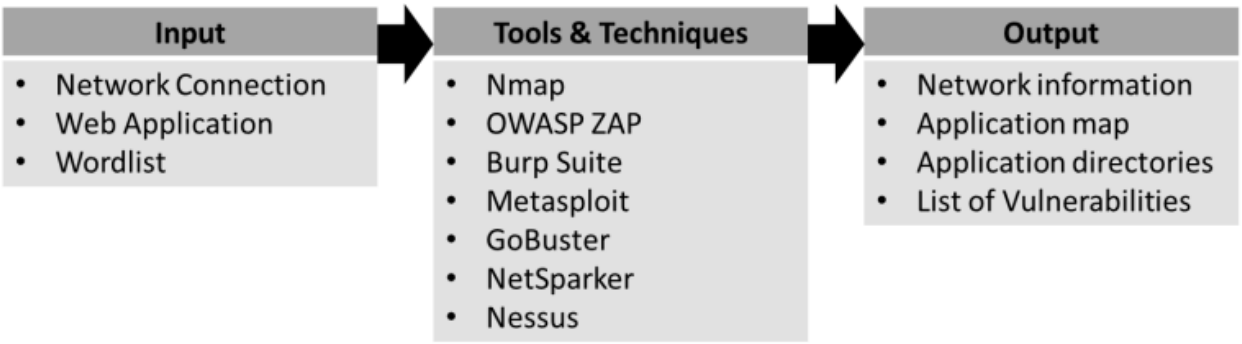


Figure 1.2: Active Scan (ITTO)

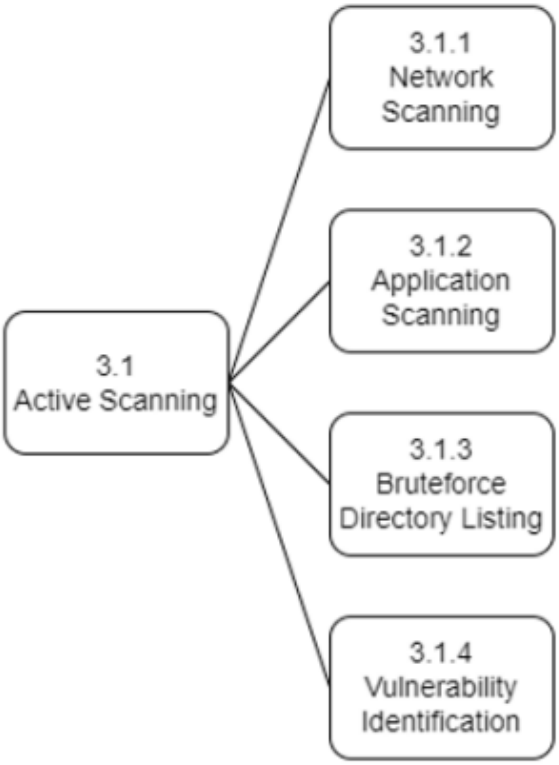


Figure 2: Sub-processes of the Active Scan

Network scanning is mostly accomplished via the use of automated tools to provide the tester with a rudimentary overview of what may be accessible on the target network. This step towards the application scan usually begins during the reconnaissance stage, while performing the port scan; however, in this stage, it is intended to do a more in-depth analysis of the network, searching for network configuration errors, vulnerabilities in Voice over IP technologies, and testing for vulnerabilities on the services and protocols used.

Generic application scanning is a technique for detecting general problems in an application. It is possible to detect problems in the program using either manual or automated crawling (see figure 4). During the scan, a tester may come across form fields that may be used to try SQL injections or XSS. While crawling the application, critical information might be found in error details or on the Viewstate. Burp Suite (PortSwigger, 2021b), Nessus (Tenable, 2021b), and OWASP ZAP (OWASP, 2021a) are the most often used tools for this since they all feature built-in crawlers that search the application for vulnerabilities. [25]

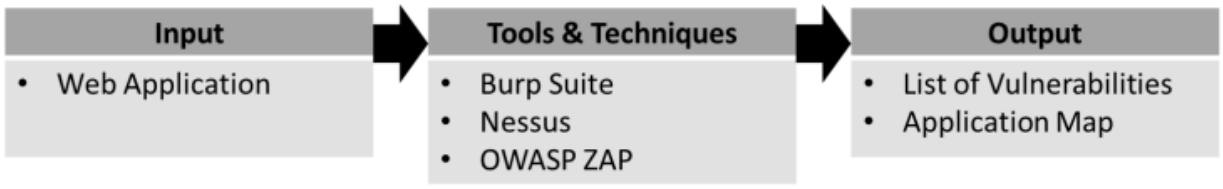


Figure 3: General Application Scanning (ITTO)

Bruteforce directory listing is a method for locating directories that were not discovered during the previous phases of the penetration test. A scanner will look for common and accessible directories. However, while using brute force directory listing, a tester may utilize standard wordlists or even customize wordlists based on keywords identified in the application (found through prior scans and reconnaissance) to locate administrative or sensitive directories, so broadening the engagement assault area (see figure 4). This process may occasionally cause the program to fail or inundate the website with requests, resulting in a DDOS. Nmap (Nmap, 2021), NetSparker (Invicti, 2021b), and GoBuster are some of the most commonly used tools for this (Christian Mehlmauer, 2021).

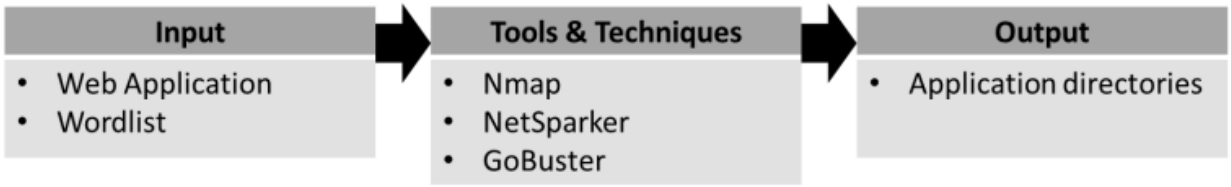


Figure 4: – Bruteforce Directory Listing (ITTO)

A tester may discover additional vulnerabilities inside the scope while completing the preceding steps for the active scan of the application. An example of this is when a tester runs a general application scan and discovers form fields that are vulnerable to SQL injections. All discovered vulnerabilities must be investigated and recognized in order to be exploited later (see figure 5). To do so, a tester must understand how the vulnerability was discovered and discovered, how the vulnerability works, and what impact its exploit may have. [26]

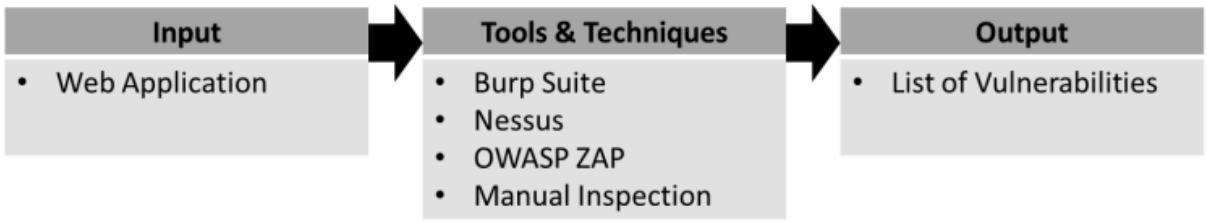


Figure 5:– Vulnerability Identification (ITTO)

A passive scan occurs when a tester scans the program in a passive manner (see figure 6).

Figure 6. - Passive Scan

It is primarily accomplished through the use of a combination of automated and manual tools, with the automatic tools seeking to identify and gather information and the manual tools assisting the tester in analyzing and assessing the outputs (see figure 7).

Figure 7: Sub-processes of the Active Scan

A tester can monitor traffic to assist determine the details of an operating system or device for passive scanning of the application. During this procedure, a tester may discover misconfigurations, unprotected data transfers, or even sensitive information (see figure 8). Wireshark (Wireshark, 2021), Tcpdump (The Tcpdump Group, 2021), and WinDump are the most often used programs for this (Riverbed Technology, 2021) [27]

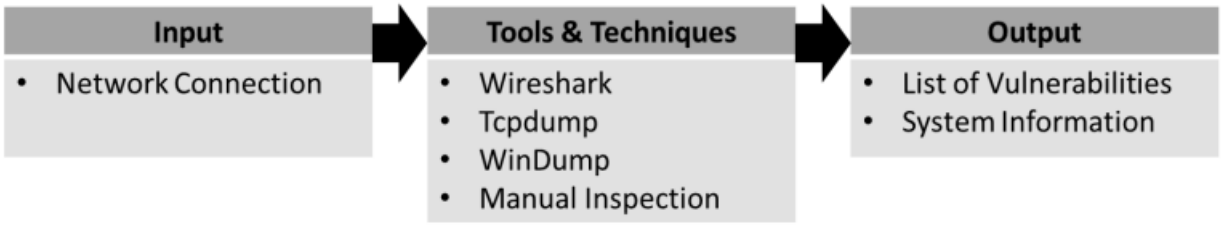


Figure 8: – Traffic Monitoring (ITTO)

Metadata analysis is examining and assessing data that describe data rather than the data itself. When reviewing a file, for example, the metadata information may include details such as the document author, the date the document was written, and other information that may include custom metadata. Internal addresses and pathways to the server, IP addresses, and other information can be found inside this metadata, allowing testers to acquire further access or information to uncover new entry points (see figure 9). This is mostly done manually, however testers can utilize tools like FOCA (Josep, 2021) or MetaCrawler (Metacrawler, 2021).

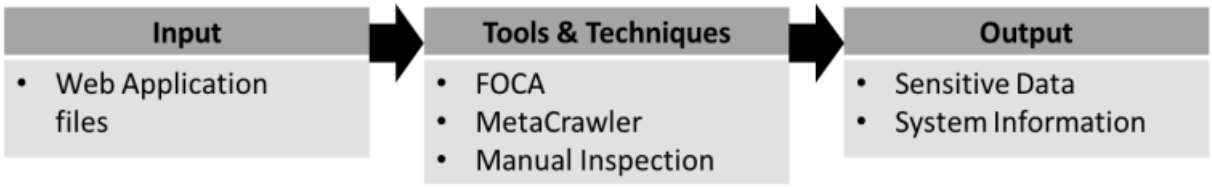


Figure 9: - Traffic Monitoring (ITTO)

Vulnerability testing is the practice of identifying vulnerabilities and security problems in an application. This may be done while the application is being actively scanned, and it must always be customized to the test objectives, rules of engagement, and end purpose. A tester will investigate vulnerabilities within the

defined depth and breadth of the criteria for this method, ensuring that the assessment results fulfill the expectations (see figure 10). This should be done utilizing active methodologies and tools, which will assist the tester in verifying the accuracy of the results. For this technique, it is customary to employ tools such as Nessus, OWASP ZAP, or Burp Suite.

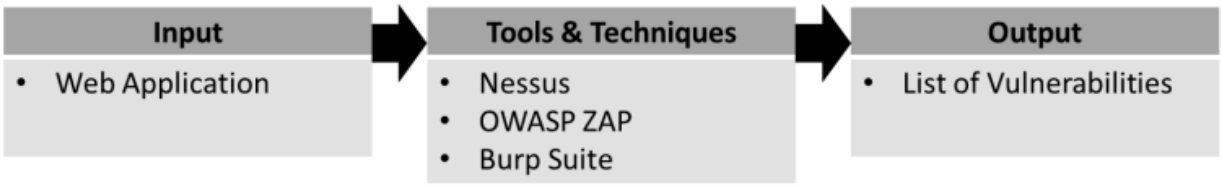


Figure 10. - Vulnerability testing

It is also necessary to manually verify and evaluate the code connected with the application, the source code, or the analysis of requests made to the application during the scanning step (see figure 11).

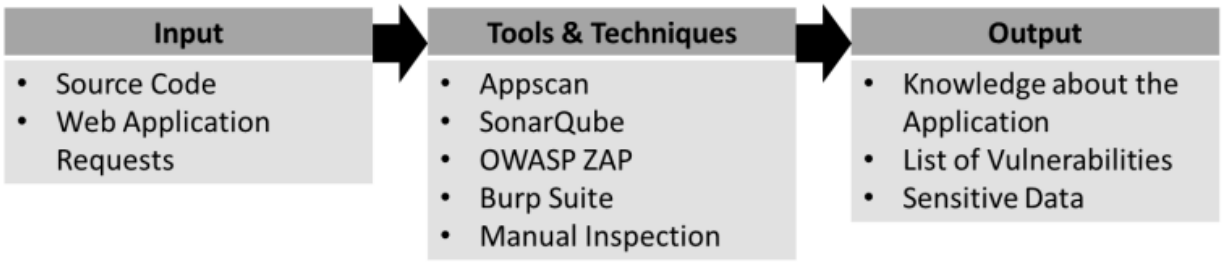


Figure 11: Traffic Monitoring (ITTO)

Depending on the method to the pentest, a tester may be permitted to peek at the application's source code (in the case of white-box testing), which, albeit a lengthy task, can help the tester better understand the application's workflow. It is also feasible to evaluate with automated tools, however most testers would agree that hand inspection is superior.

Source code analysis is the method by which a tester examines the source code of an application in search of security problems, as many security defects go undiscovered by vulnerability scanners. Any information related to security issues is always there in the source code. With access to the source code, a tester may

precisely determine all of the methods within the program and eliminate the guesswork associated with security testing. Findings vary and may be connected to faulty business logic, concurrency issues, easter eggs, or even cryptography, with the latter being the most dangerous vulnerabilities in online applications (see figure 12).

The study of the source code may be a lengthy task because some of it must be done manually. This process, however, can be facilitated with the help of tools such as SonarQube (SonarSource, 2021) or Appscan (HCL Software, 2021), as these tools analyze and review the code, searching for coding errors, security vulnerabilities, and design flaws, and offering remediation measures to ensure stable and secure code.

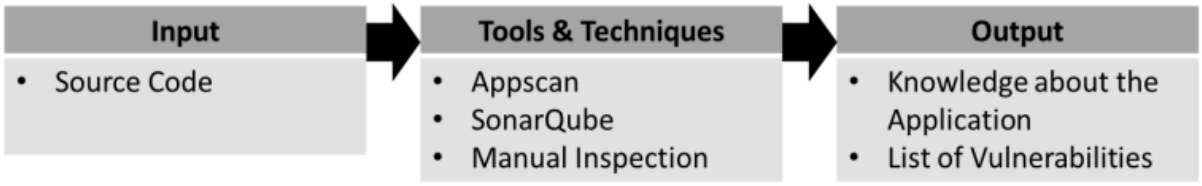


Figure 12: – Source Code Analysis (ITTO)

Observing, analyzing, and modifying requests made between the program and the browser is another method for discovering sensitive information or vulnerabilities. Requests such as GET HTTP requests frequently reveal sensitive data that may or may not be cryptographed, allowing the tester to reveal new application gateways (see figure 13). This analysis may be performed using Burp Suite (through the Intruder) or OWASP ZAP, since both tools have a customizable proxy to intercept requests as well as plugins to assist the tester in transforming encoded or raw data into usable information.

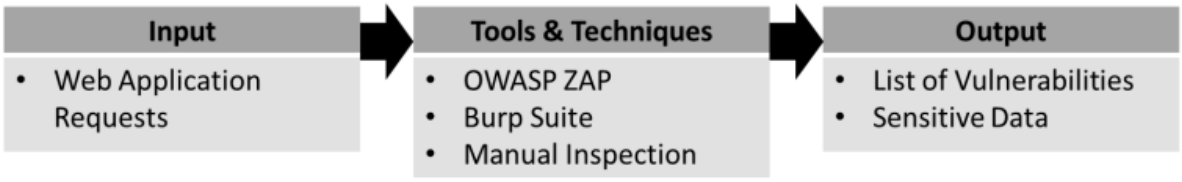


Figure 13: – Request Analysis (ITTO)

The validation of the data will enable the tester to better correlate the chosen tools and their limits. This stage should be taken into account throughout most penetration tests since it is vital for the results to validate every input and output of the conducted activities and their outcomes. This technique is critical in assisting the tester in narrowing down the number of detected vulnerabilities to only those that are legitimate (see Figure 14). This may be accomplished by either testing the vulnerabilities or inspecting the relevant defect.

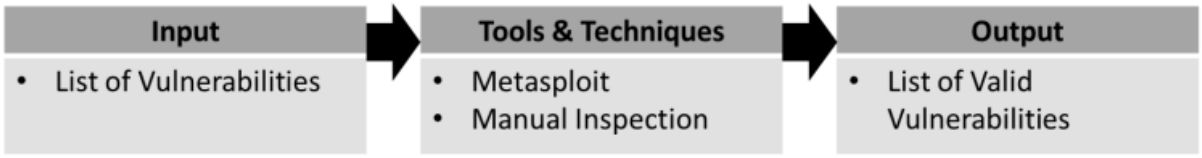


Figure 14: – Validation (ITTO)

The vulnerability analysis step is the methodology's fourth stage. At this step, the tester will concentrate his efforts on discovering and assessing the results, as well as developing a plan of action for the exploitation and testing of the discovered vulnerabilities (see figure 15). There are no software requirements for this methodological step because it should be completed mostly by process reviews, manual inspection, and public research.

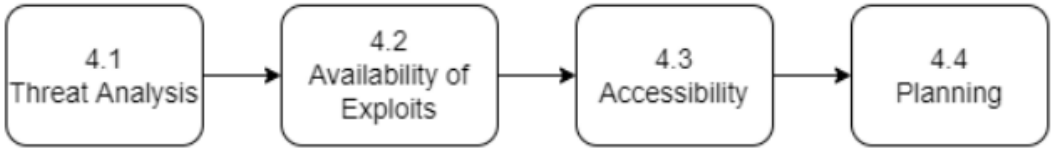


Figure 15: - Main Procedures of the Vulnerability Analysis Stage

A tester should study all results after completing all prior scanning methods. The tester will have a better understanding of how to carry out the exploitations, the impact they may have, the vulnerabilities connected with the results, and the restrictions within the rules of engagement and the established scope. Most vulnerabilities may be discovered by conducting public research on the topic. However, in order to conduct a more in-depth investigation of the scenario, a tester may need to put up a duplicated environment.

The purpose of threat analysis is to enable the tester to discover potential attack vectors related to the end goal of the tests and analyze their impact. Online systems such as NVD (National Vulnerability Database), OSVDB (Open Source Vulnerability Database), Security Advisories, and Issue Trackers are excellent vulnerability databases that enable testers to search for and learn more about the researched flaws. CVE (Common Vulnerabilities and Exposures) is also a valuable source of information on vulnerabilities discovered based on system components, categories of flaws, and CVE numbers, which make research more precise because they provide the identifier for specific vulnerabilities (see figure 16).

Figure 16: – Threat Analysis (ITTO)

A tester should evaluate his capacity to access or produce exploits or payloads needed to test the environment while completing a vulnerability analysis. Not only by examining the availability and accessibility of such exploits or payloads, but also by taking into account the use of third parties and the customisation of particular methodologies to evaluate the authenticity of the findings. This phase is critical for preparing the exploitation stage since it allows the tester to validate his results and better understand how to exploit particular design faults (see figure 17). In terms of custom exploits, Metasploit is a commonly used program that provides a collection of exploits that is always growing and is available for usage by any tester.

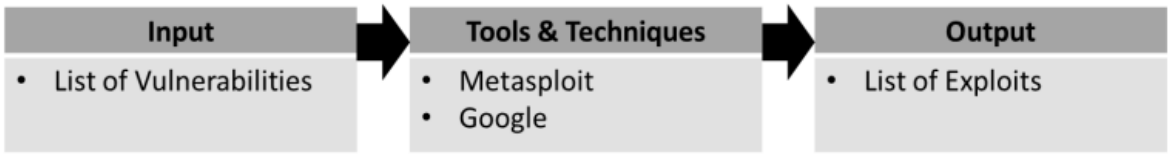


Figure 17: – Availability of Exploit (ITTO)

Accessibility analysis entails gaining access to certain vulnerabilities or entry points in order to construct a specific scenario and execution route for the exploitation (see figure 18). A tester should outline all of the criteria and needs to reach the entry point while keeping the target's protective measures and workflow in mind.

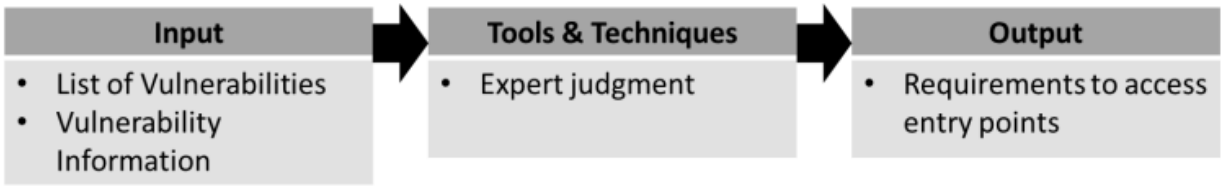


Figure 18:– Accessibility (ITTO)

Taking all of the preceding duties into account, the preparation of the subsequent phases should result in a clear and simple product. Taking into account the scope and rules of engagement, as well as all findings, the tester can determine a path of execution for his exploits with as much information as possible, detailing the goals of exploitation, all of the entry points, how to access them, what payloads or exploits to use, the tools and methods to use, and how to configure the tools given the established goals (see figure 19).

Figure 19: – Planning (ITTO)

The exploitation stage is the methodology's another stage. The major purpose of this stage is to get access to the targets by circumventing security measures. Assume that all of the preceding phases were completed appropriately. In that scenario, this step should be rather simple because the majority of the entry points have been determined, the approach and engagement have been planned, and all extra information has been considered for the application of payloads and exploits.

If the assault vector is well established, there is a fair chance of success based on the declared aims. This stage may result in a loop between the exploitation and vulnerability analysis stages since new vulnerabilities might be discovered when performing exploitation on prior discoveries or while doing additional penetration and permission escalation (see figure 20). [28]

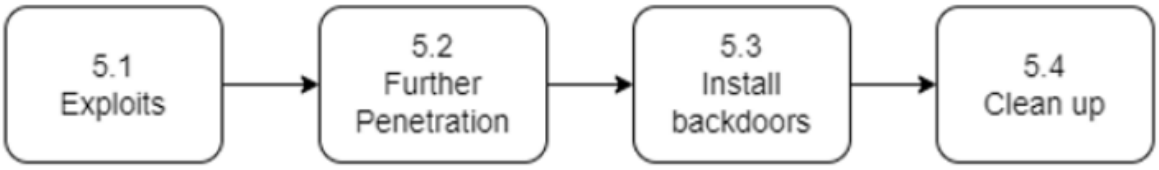


Figure 20: - Main Procedures of the Exploitation Stage

Attack vectors should be precise and elusive when performing exploitation. This means that, because the entire process is intended to simulate an actual attack, the exploitation stage is the application of all of the Design and Development 40 accumulated research done on the target, combined with evasive measures to allow a tester to execute precise attacks on the target without being detected during the penetration test (see figure 21).

It is not always feasible to avoid detection in brute force or flooding exploits, hence extra precautions must be taken to conceal the testers' identities.

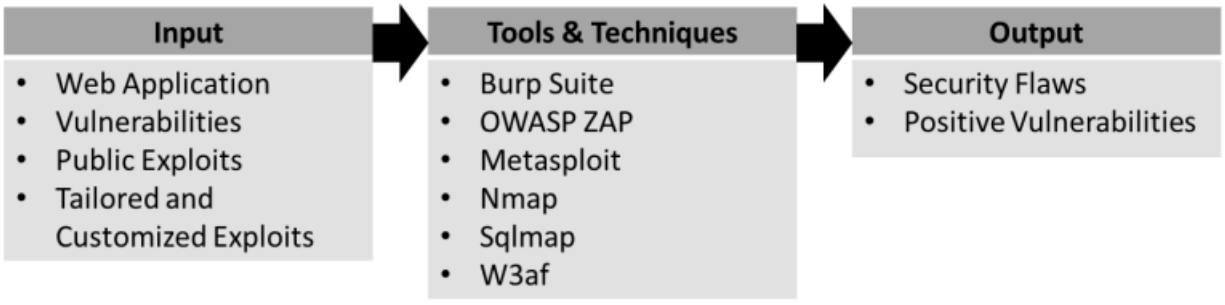


Figure 21: – Exploits (ITTO)

There are primarily two sorts of exploits: public and tailored and customized exploits; both have their applications and should be employed in accordance with the tester's demands (see figure 22). Burp Suite, OWASP ZAP, Metasploit, Nmap,

Sqlmap (Bernardo Damele A. G. & Miroslav Stampar, 2021), and w3af (W. Org., 2021) are among the most popular tools for exploit execution.

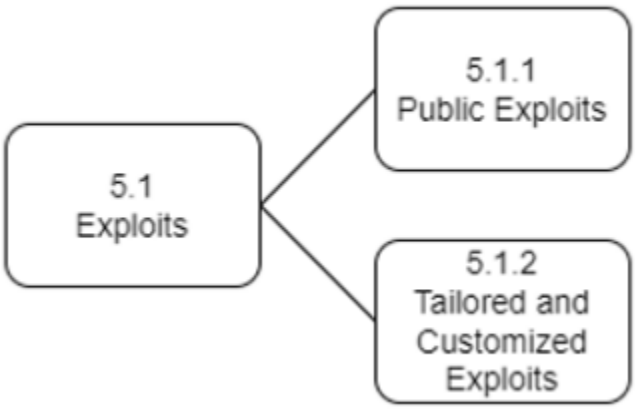


Figure 22: - Sub-processes of the Exploits

A tester will seek to get access to as many resources as possible in order to increase the odds of successfully exploiting a target's vulnerability. With all of the information gathered in the preceding phases, a tester can search for public exploits on the internet or in other sorts of documentation. These exploits are created in response to specific vulnerabilities in certain hardware or software setups, and they can be used by the tester if all prerequisites are satisfied. During this process, a tester should exercise caution about culpability and the source of the vulnerability, as there are various false exploits meant to harm or destroy their user's PC.

Exploit-DB (Security, 2021), Searchsploit (K. Org., 2021), Metasploit, and even Google (Google, 2021) are examples of trustworthy exploit databases (see figure 24).

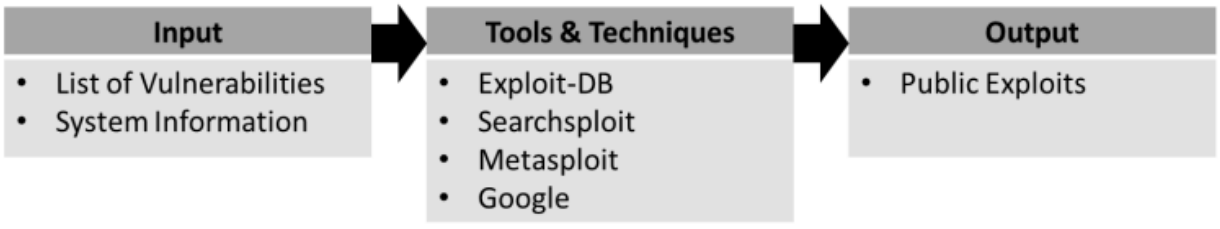


Figure 23: – Public Exploits (ITTO)

Because each assault is unique in terms of how the exploitation happens, the tester may need to tweak and personalize the exploitation to meet the testing environment in order to produce effective results. A tester's chances of a successful attack rise if he understands the testing environment and the applicability of an exploit. Because public exploits may be overly particular to certain versions of operating systems or apps, exploits must be changed and customized for execution.

This approach may need an environment simulation to test the modifications and ensure their success. Even if a tester gets all of the system knowledge, having a functional infrastructure and system to test the exploits will make the exploitation process easier. This is warranted because memory address changes occur as a result of service packs or new version releases. A tester will need to know how to read code written in several programming languages and understand how payloads operate for each exploitation in order to tailor them to his needs for this activity (see figure 24).

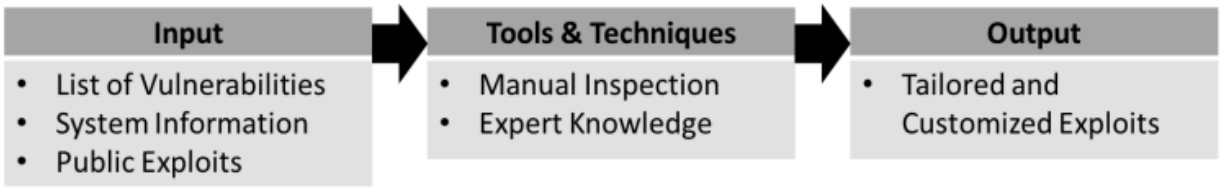


Figure 24: – Tailored and Customized Exploits (ITTO)

After successfully running exploits, a tester may enumerate and acquire access to other systems on the client's infrastructure, depending on the rules of engagement and the scope of the test. A tester may be able to execute actions within the compromised system using the access granted during vulnerability exploitation, allowing him to upload tools into the system, enumerate DNS of the internal network, execute brute force attacks, execute remote exploits, and abuse of compromised credentials.

A tester might also use the hacked system to proxy to an internal network, set port forwarding, restrict information access, and even alter authentication. This is mostly accomplished through the use of shell commands within the hacked system (see figure 25). [29]

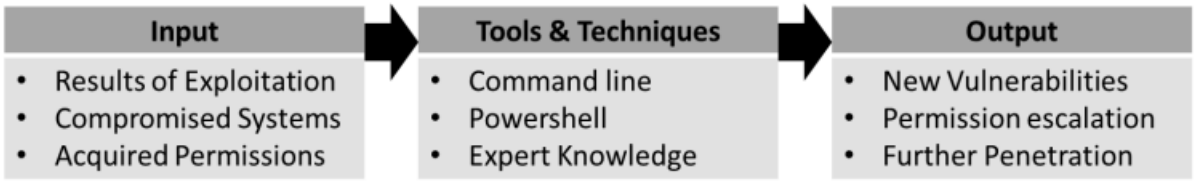


Figure 25: – Further Penetration (ITTO)

A tester may discover additional vulnerabilities to test when exploiting vulnerabilities or further penetrating the system as permissions climb and access to other areas of the system is given (see figure 26). These findings will need to be analyzed before they can be evaluated. Because new vulnerabilities were discovered, a tester must assess the risk and effect of the discoveries, which will result in a loop between the fourth and fifth phases of this technique (as represented on figure 6).

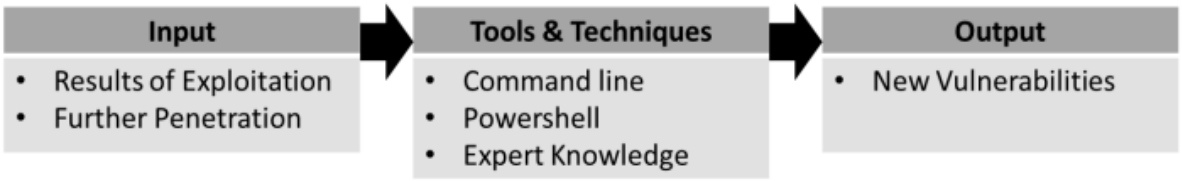


Figure 26: – Further Penetration (ITTO)

With the capacity to edit and control configurations within infiltrated systems, testers can install backdoors that will remain in the system (see figure 27). Backdoors installed should need authentication (to prevent unattended attackers) and, if feasible, remain on the system after reboots. After breaching a system, backdoors allow testers to circumvent the standard authentication procedure. This can be done to preserve or make future access easier, or to further abuse the system. A tester can use tools developed for this purpose or simple utilities like NetCat ("Hobbit," 2021), whose primary mission is to read and write data over network connections, to install a backdoor. A tester can use this to keep access to the system even if he is offline for whatever reason.

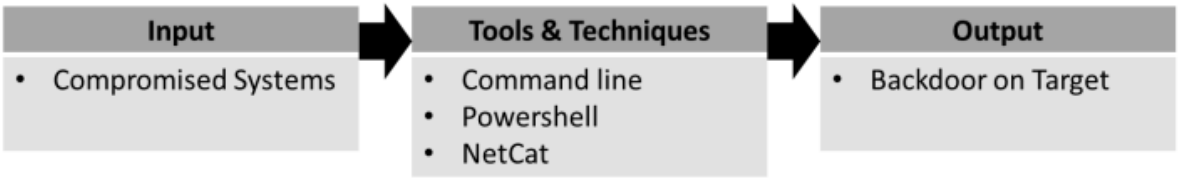


Figure 27: – Install Backdoors (ITTO)

Before completing a system clean-up, a tester should check that all exploitation processes have been documented and that there are no more tests to run on the system. The clean-up technique involves a tester cleaning up the system of any traces of the penetration test. This includes removing all executables, restoring all system settings and configurations to their original values, removing all backdoors installed, removing any user accounts created to connect to compromised systems, and restoring the database from backup to ensure no data was damaged during the process (see figure 28).

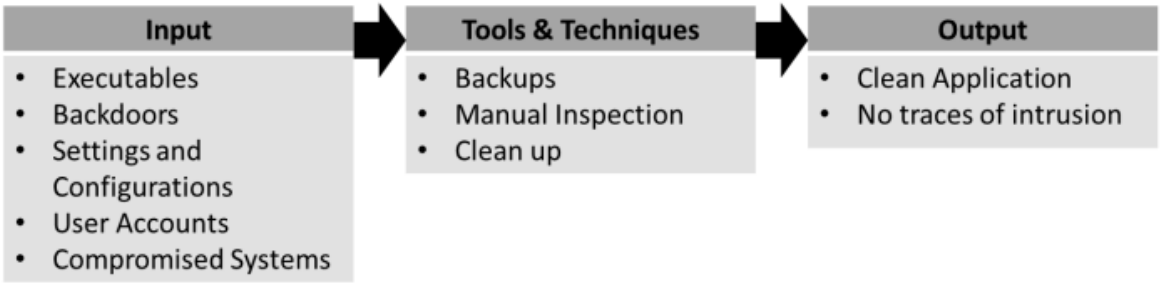


Figure 28: – Clean-Up (ITTO)

The analysis of findings step is the methodology's last stage. A tester is intended to assess the whole process at this step to check whether all of the rules of engagement were followed, if the set goals were accomplished, if all metrics are appropriate, and to analyze the overall impact of the penetration test on the application. This step is split into three major tasks: a review of defined engagement rules, goals, and metrics, a study of the penetration tester's impact, and an examination of the penetration test technique as a whole (see figure 29).

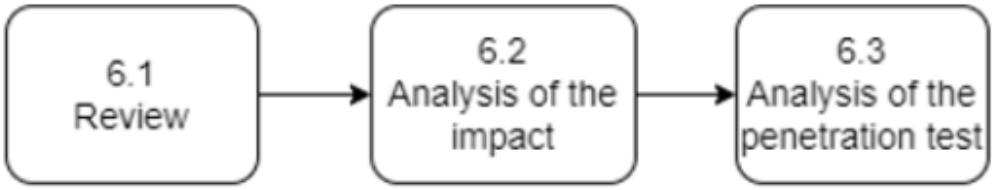


Figure 29:- Main Procedures of the Analysis of Results Stage

To assess conformity with previously defined standards and norms, these must be evaluated to ensure that the entire method was carried out correctly (see figure 30 and figure 31). For this evaluation, the tester must determine whether the overall strategy for penetration testing matches what was intended.

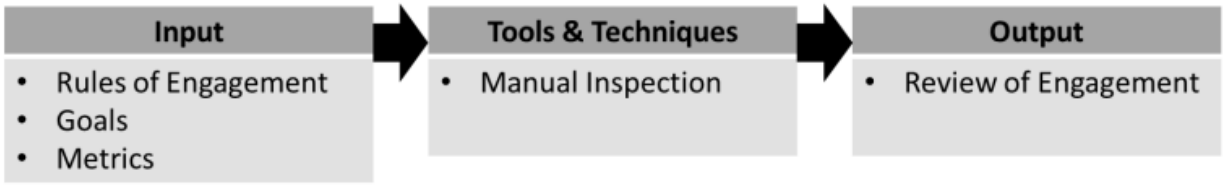


Figure 30: – Review (ITTO)

Figure 31: - Sub-processes of the Review

Because the rules of engagement are in place to safeguard both parties (testing and client), the penetration test should be assessed in accordance with the guidelines (see figure 32). A tester should evaluate all of the operations carried out throughout the penetration test to ensure that each activity was carried out correctly. Following a review of these, the tester should express the findings of the analysis in the report on the reporting stage.

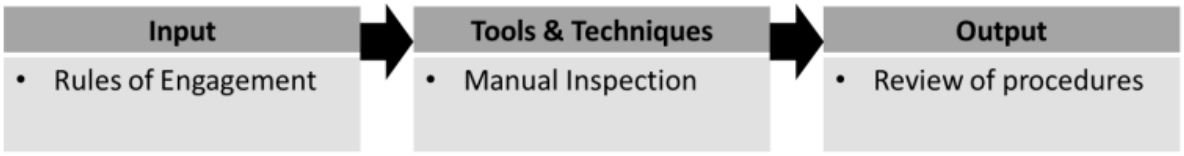


Figure 32: – Review Rules of Engagement (ITTO)

The entire penetration testing procedure was carried out for a cause, which is defined by the previously set goals (see figure 33). A tester should review the penetration test findings and confirm that they satisfy the objectives. This information must be included in the report created at the reporting step.

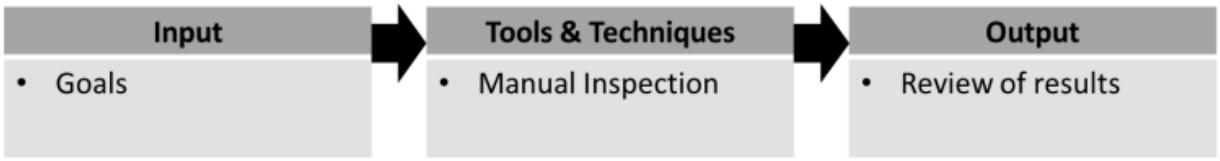


Figure 33: Process 6.1.2 – Review the goals (ITTO)

To assess the performance and efficacy of the penetration test, a study of the metrics and their correlation with the findings is required (see figure 34). Assume the measurements are well-defined and well-established. In such situation, all parties will have a more clear and succinct view of the outcomes, not only for the client to evaluate the tester's job, but also for the tester to analyze what happened as expected. This information must be included in the report created at the reporting step.

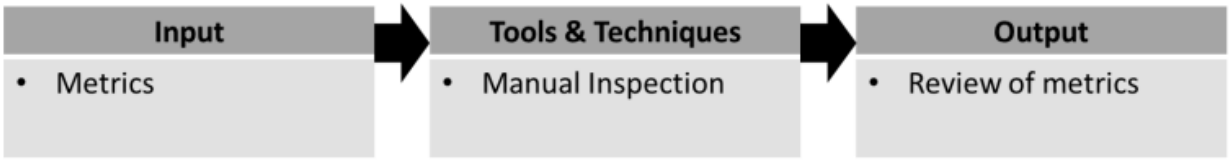


Figure 34: Review Metrics (ITTO)

After completing all security tests and assessments, as well as clearing up any traces of the penetration test, a tester should analyze and evaluate the overall impact of the procedure on the targets (see figure 35). If done correctly, there will be no influence on the systems; nonetheless, the tester should do this assessment to ensure the system's integrity and the procedures utilized.

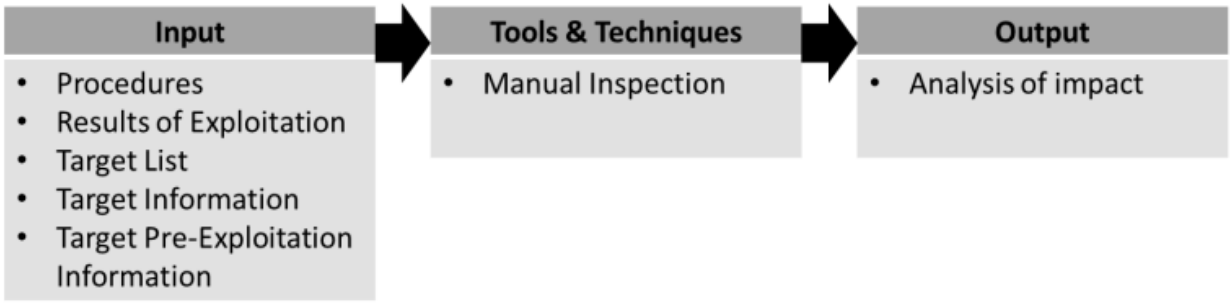


Figure 35: – Analysis of the impact (ITTO)

Taking everything into account, an ethical tester should examine and appraise all processes performed throughout the penetration test (see figure 36). What met expectations, what did not, which tools Design and Development 46 were best suited for certain tasks, and any other information should be reviewed. This is mostly a technique for evaluating the tester's approach and skills. Self-evaluation is an essential component of self-improvement and growth. This material may be included in the final report but is not required.

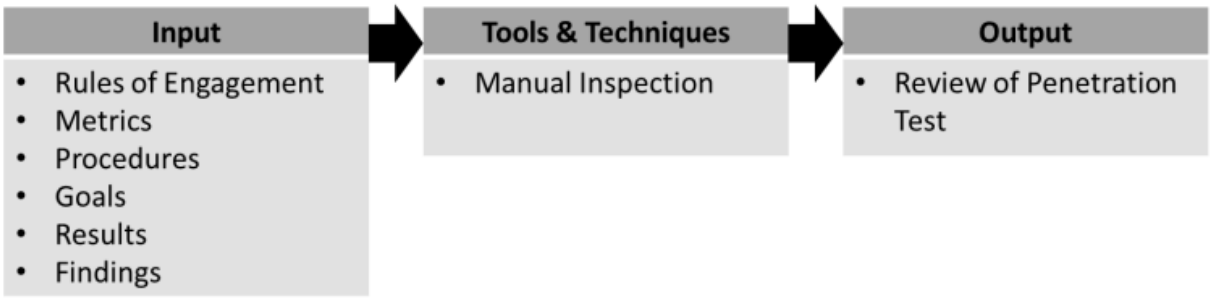


Figure 36: – Analysis of the penetration test (ITTO)

2.1. Environment

For the Design and Development stage, the environment consisted of a Virtual Machine with Kali Linux OS loaded. During the Pre-Engagement process, it was determined that the scope of the Pentest should only include a certain port of software installation with a specified address for connection, which needed access to a Virtual Private Network (VPN). Permissions to use the program were granted, but with limited access, preventing direct connection to the database using the provided credentials. Within the WebApp, all-access was given, but not to all folders specified in terms of application permissions.

While working at VTXRM, I learned about Accipiens and some of the frameworks and infrastructures that were utilized to host the application. However, there were certain limits on access to the core source code as well as the network and application architecture. Keeping this in mind, the pentest technique was designed using a gray box approach. [30]

2.2. How Vulnerability Assessment Tools Work

The initial component of the Pentest was to determine which tools were optimal for doing a vulnerability and security assessment among the various Open Source scanners, as well as some commercial scanners. The tools were chosen based on previous research conducted during the development of section 2.7, taking into account Shay Chen's review of SecTool (Chen, 2016),

which evaluates (commercial and open-source) vulnerability scanners based on their features, keeping in mind that Accipiens authentication is performed via NTMLv2 (an authentication protocol used by Windows to authenticate network users) via the VPN connection and by consulting

Nmap, Metasploit, Burp Suite, OWASP ZAP, Nessus, w3af, and Acunetix were the first tools chosen. This list was later cut down because the w3af utility was removed from Kali Linux and its installation was not feasible due to Module issues and a lack of ConfigParser. Acunetix was removed from the list because its use was limited and no access for academic purposes was given. A free license for Burp Suite Pro was requested by PortSwigger, but it was not granted, resulting in the use of the free version, Burp Suite Community Edition. [31]

2.2.1 Nmap

As previously stated, Nmap is an open-source program for port scanning and OS fingerprinting; its primary applications are network discovery and security audits. Nmap analyzes raw IP packets to determine available hosts, services, operating systems, potential firewalls or packet filters, and a variety of other properties. It was picked owing to its popularity among security testers and its diversity of tools, which allow it to scan many distinct components and features of a specific network, providing fundamental output throughout a network scan. [26]

According to the Nmap Organization, it is flexible because it includes many port scanning mechanisms, powerful because it has been used to scan massive networks with thousands of machines, portable because it is compatible with all major operating systems, easy to use, free because it is available to any user for free download, well documented and translated in multiple languages, supported by a large community of developers and users, acclaimed with many awards, and popular.

2.2.2 Metasploit

Metasploit is a highly adaptable framework that is simple to configure and works with the majority of operating systems. This framework, which was included into Kali, makes it easier for pentesters to remotely evaluate the security of systems. With several commercial-grade vulnerabilities and a rich exploit creation environment, this application allows users to design and run attacks via command line or GUI. This framework is separated into modules that contain a large number of scripts, tools, and plugins, as well as a library that allows exploiting multiple vulnerabilities without writing additional code by simply declaring an attack and its payload.

2.2.3 Burp Suite Community Edition

Burp Suite is a platform for web application security testing. It is a tool that can map and analyze an application environment. A Repeater, a Decoder, a Sequencer, a Comparer, a Burp Intruder, HTTP(s), and WebSocket's proxy are all included in the Community Edition. With a plethora of plugins and functions, this program is one of the most popular among professional testers.

Because the use of Burp Suite is limited to the Community Edition, the following functions are available:

Repeater — A straightforward tool for manually modifying and reissuing individual HTTP and WebSocket messages in order to examine the application's answers.

Decoder — This tool recognizes multiple encoding formats and converts encoded or raw data into its canonical, encoded, or hashed form.

Sequencer — This tool enables testers to validate untrustworthy data like as session tokens, antiCSRF (Cross-Site Request Forgery) tokens, password reset tokens, and more.

Comparer - The Compare is a straightforward tool for comparing any two pieces of data.

Proxy - Burp Suite allows you to run a web proxy server between the browser and the targeted application, intercepting, inspecting, and modifying raw traffic in both directions.

The Intruder - The Intruder is a tool for automating customized assaults. It may be highly customized to accomplish a wide range of activities, from brute-force guessing of web directories to active exploitation of complicated vulnerabilities.

2.2.4 OWASP ZAP

The OWASP ZAP is an open-source web application security scanner that is maintained by a team of worldwide volunteers. It is one of the most widely used web app scanners in the world. It may be used as a proxy server to alter the total traffic that travels through it, allowing users to map web application directories and resources through the ZAP spider. This tool is simple to install and use, and it finds and detects security flaws such as SQL injection, broken authentication, sensitive data exposure, broken access control, security misconfiguration, Cross-Site Scripting (XSS), insecure deserialization, components with known vulnerabilities, and missing security headers. It is feasible to set up in all main operating systems due to its interoperability. [32]

2.2.5. Nessus

Nessus is a free and open-source remote security scanning tool that is primarily used for vulnerability assessments and penetration testing. Testing each port on a computer may detect what service it is running and then test it for any known vulnerabilities to prevent attackers from launching harmful attacks. It can look for gaps in authentication and access to sensitive data, as well as misconfigurations, denial of service problems, software flaws, malware, and missing updates. It has a server-client architecture and supports the installation of many plugins to handle a wide range of vulnerability tests.

2.3 Methodology used

This dissertation's approach is separated into seven steps, comparable to the technique created by PTES. The testing process begins with a scoping discussion, during which an NDA (non-disclosure agreement) is signed between the tester and the client, ensuring the customer that no proprietary or sensitive material would be revealed or shared in any way. Following the agreement, the scope of the test should be determined in terms of infrastructure constraints, needs, metrics to evaluate security, and test objectives (database, application, network, social engineering, and others).

Additional information about the target should be disclosed in accordance with the scope and expected approach (if a tester is expected to use a black-box approach, no information should be disclosed; if it is a white-box approach, the tester should have access to all information on the system as well as its source code; and if it is a grey-box approach, only relevant information should be disclosed), as well as technical details of the infrastructure and ot With this information, a tester is expected to specify a time estimate for the test, as well as the tools to be used and the methodology to be used. [33]

The Reconnaissance stage of this process follows, in which the tester should research additional information about the application, the network, and the systems where it is housed. It is possible to map the network, open host ports, and running services, map some of the application directories, and discover information about the database used, the system where it is running on, disclosed addresses, versions, instance, and server name, and the TCP port where it is running using tools such as Nmap, Burp Suite, or Metasploit. After acquiring basic target information, the tester should begin the Scanning step by adopting a more active approach to the application.

Using automatic scanners saves time, however certain human examinations should still be performed. A tester can use Nessus to test the application security and scan for active vulnerabilities for the active scan; OWASP ZAP

can also be used with the same purpose and utilize the built-in crawler to map the application deeply, identify new directories, or vulnerabilities. Burp Suite is a fantastic tool for manually reviewing intercepted messages between the browser and the application in order to study and analyze various interactions. The Vulnerability Analysis step begins with evaluating the implications of previously discovered vulnerabilities, assessing their entry points, and categorizing them based on severity.]

After analyzing all of the information on the vulnerabilities discovered, the Exploitation stage begins, in which a tester attempts to exploit the entry points to confirm or deny the presence of a vulnerability on the target, as well as test possible entry points or exploits that were not discovered during the scanning stage but for which the application may be vulnerable. [34]

A tester may discover new vulnerabilities throughout this approach. In such a scenario, those should be re-analyzed, establishing a loop between the Vulnerability Analysis and the Exploit phases until no new vulnerabilities are discovered. Assume there are no further vulnerabilities to investigate.

In that case, the tester should proceed to the Analysis of Results stage to evaluate and assess the results of the exploitation, concluding on the potential impact of the exploitation on the system, the risks associated with it, how to prevent it, and the overall result of the pentest in terms of the metrics defined. The final stage of the methodology is the Reporting stage, in which a tester is expected to fill out and deliver to the customer a report containing detailed information about the scope, the vulnerabilities discovered, the steps taken to discover and exploit them, the impact they may have, remediation for the vulnerability, and any other technical detail deemed relevant. [35]

Chapter 3 Data Encryption

Data encryption is another method of data protection. Stream encryption is commonly used to secure data. Stream encryption encrypts each plaintext bit one at a time with the appropriate bit of the key stream to produce a bit of the cipher text stream. Using shift registers, the pseudo random key stream is created sequentially from a random seed value. Stream ciphers, on the other hand, are extremely vulnerable to known-plaintext assaults. Many digital services necessitate dependable security in digital data storage and transfer. Because of the fast rise of the internet in today's digital world, information security has become increasingly crucial and has garnered a lot of attention. Steganography (SG) is one of several techniques used to safeguard digital data. It is a technique in which two persons communicate in a clandestine manner utilizing a cover item. SG is a very old method of covert communication that may be traced back to spies' use of methods such as invisible ink and microdots. Image steganography is a major field of information concealment techniques that may be used to conceal private communication over public channels. Recently, information concealment strategies have piqued the interest of researchers in the field of information security. In general, the embedding procedure in SG necessitates the use of a digital media to transport the data. Images and multimedia components, such as video and music files, are commonly utilized and transferred through the internet. Because they do not draw notice, such mediums are ideal for concealing messages.

In the digital age, information confidentiality is a must, and encryption is one of the technologies used to safeguard it. Encryption is a means of safeguarding information against unwanted assaults by changing it into an unrecognizable format to the attackers. The purpose of encryption is to provide a simple and low-cost method of encryption and decryption to all authorized users who have the relevant key, and vice versa for all other users who do not have the key. Data decryption is the inverse of data encryption, and it retrieves the original data.

Data encryption systems are classified into text encryption, audio encryption, image encryption, and video encryption based on the type of plaintext. Some encryption standards have been established in order to provide a general cryptosystem that can encrypt digital data such as text/image/audio/video. DES, RSA, AES, and IDEA are among the most extensively used. [36]

3.1. Cryptosystems

Based on the kind of key, cryptosystems are classed as symmetric or asymmetric, with the former using secret keys and the latter using public keys. Symmetric cryptosystems are further classified as block and stream ciphers. Block ciphers operate on big blocks of plaintext message with a fixed transformation, whereas stream ciphers operate on individual plaintext bits with a variable transformation over time. Stream ciphers are most commonly used in military, telecommunications, and corporate applications [16]. The security of a stream cipher is dependent on the development of an unexpected sequence known as a key stream, which must be of appropriate length and randomness . As a result, the key stream generator is an extremely important building component for stream cipher algorithms. [37]

3.2. Encryption of streams

Stream ciphers are a type of symmetric encryption method. The Vernam (One-Time-Pad) cipher, which encrypts by XORing the plaintext with a random key, influenced their core design philosophy. The Vernam cipher's disadvantage is that the key must be a real random sequence shared by both the sender and the recipient, and it can only be used once. In terms of key creation and distribution, this presents a practical issue. Stream ciphers, on the other hand, expand a given short random key into a pseudo-random key stream, which is then XOR'ed with the plaintext to produce the cipher text. The fundamental form of a stream cipher comprises the production of a pseudorandom sequence of bits, which is XOR'ed bit by bit with the plaintext at the transmitter to create the cipher text. The plain text is retrieved at the receiver by producing an identical pseudorandom sequence of bits that is perfectly synchronized with the incoming encrypted text stream. Stream encryption systems are divided into two types: synchronous and self-synchronous. The key stream in the former is created independently of the message, therefore a missing character during transmission demands resynchronization of the transmission and reception key 13 generators. Figure 2- 3 depicts the synchronous stream encryption block diagram. That is, the encryption of a character is not spread throughout a message block length. As a result, error propagation does not occur in synchronous stream ciphers.

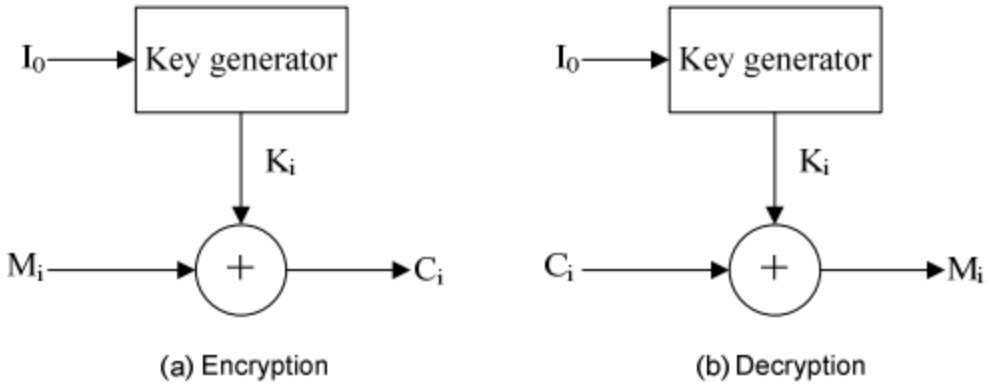


Figure 2-3: Block Diagram of Stream Encryption.

Each key character in a self-synchronous stream cipher is produced from a set number, n, of the previous cipher text characters, giving birth to the term cipher feedback. If a cipher text character is lost during transmission in such a system, the error propagates for n characters, but the system resynchronizes itself once n correct cipher text characters are received. Figure 2-4 depicts the synchronous stream encryption block diagram. [38]

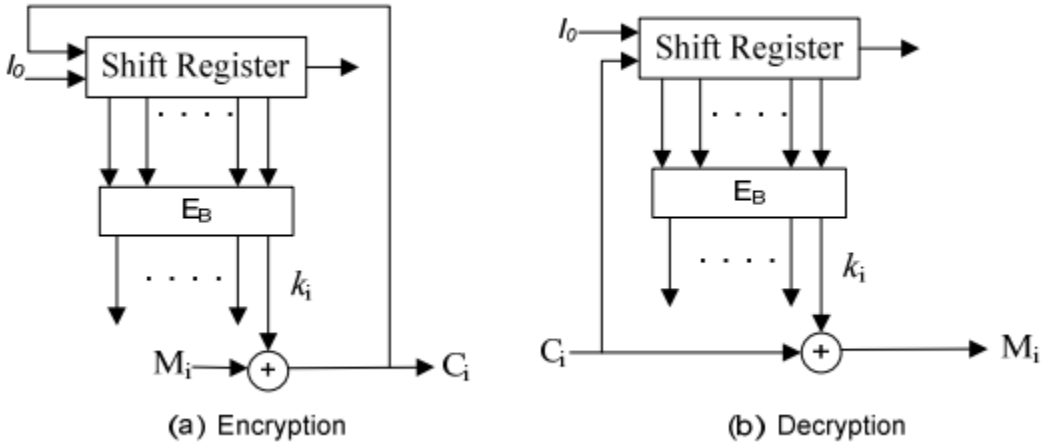


Figure 2-4: Block Diagram of Self-Synchronous Stream Cipher

Pseudo-random numbers can be created using either hardware or software.

The most basic method of producing the pseudo-random number sequence using hardware is to employ linear feedback shift registers, which may be used in stream

ciphers and are ideally suited to low power or high speed needs. RC4 is one method of producing a pseudo random number sequence using software. These, like any other stream cipher, may be used for encryption by mixing them with plain text via a bit-wise exclusive-or operation. These bits are identical to the Vernam encryption except that produced pseudo random bits are used instead of a prepared stream.

3.3 Short-comings of Stream Encryption

However, pseudo random bit streams generated by any of the aforementioned approaches are inefficient for encryption since they are prone to several types of attacks. Because of the linear combination of PRBG bits and ciphered bits, a pseudo-random sequence generated with LFSRs, for example, is extremely vulnerable to known plaintext attacks. Even RC4, which is notable for its simplicity and speed in software, has flaws that make it unsuitable for use in cryptosystems. When the beginning of the output stream is not deleted, or when nonrandom or linked keys are utilized, RC4 is extremely susceptible. To combat the disadvantages of this linear combination of the PRBG and the ciphered bits, one way is to make the combination non-linear by permutation, confusion, or by providing cipher/plain text feedback. We may also apply non-linear functions or chaotic functions between the PRBG and ciphered bits to make the result non-linear. Many features of chaotic systems have matching equivalents in classical cryptosystems, indicating a tight 1 link between chaos and cryptography.

Many articles on chaotic encryption schemes have recently been published. Ergodicity, high sensitivity to beginning circumstances, lengthy periodicity, high unpredictability, and mixing are all features of chaotic systems. With all of these benefits, scientists were meant to create new and powerful chaotic cryptography methods. Because of their common features, chaotic systems have sparked a lot of interest in secure communication and cryptography. In the context of cryptography, most characteristics satisfy some conditions such as diffusion and mixing. In order to increase communication security, several attempts have been made to research chaotic image encryption techniques. The chaotic pseudorandom bit generators (PRBGs) play a fundamental role in a number of proposed chaotic cryptosystems, including generating cryptographic keys and randomly initializing variables in cryptographic protocols. As chaotic cryptology research has progressed, certain

catastrophic flaws have been identified, discouraging practical use of these cryptosystems. The equivalency between the beginning condition and the chaotic symbolic trajectory, for example, renders cryptosystems exceedingly weak such chaotic PRBGs are difficult to discretize in the limited digitized state space or to construct with low complexity digital hardware requirements. [39]

3.4. Image Encryption Techniques

Image encryption is one of the strategies for safeguarding digital photos. It is the process of realigning the original image into an unintelligible image that is unrecognizable in 16 appearances. Traditional data encryption techniques, such as DES, triple DES, RSA, IDEA, or AES, are unsuitable for picture encryption due to image features such as high redundancy and significant correlation among pixels . Shannon proposed that confusion and diffusion are the two primary approaches for dealing with high redundancy and strong correlations. Many picture encryption algorithms based on chaotic functions have been proposed. The chaotic pseudo random key streams play a vital role in a variety of proposed chaotic cryptosystems , including generating cryptographic keys and randomly initializing variables in cryptographic protocols. As chaotic cryptology research has progressed, certain catastrophic flaws have been identified, discouraging practical use of these cryptosystems. The equivalence between the starting condition and the chaotic symbolic trajectory, for example, renders cryptosystems very vulnerable.

3.5. Summary

Although there are numerous data concealing and data encryption techniques in the literature, there is still much room for improvement. Some solutions to the aforementioned problems are proposed in this thesis. We strive to confound the relationship between the pseudo-random sequence and the message bits in stream encryption. In this thesis, we provide two new data encryption techniques that make use of graphics as well as classic stream cipher principles. The primary goal of any data concealing method is data security. Our data hiding algorithm satisfies all requirements, including secrecy, integrity, availability, and authenticity. [40]

The suggested methodology proposes a novel method of encrypting data that is based on misinterpreting the relationship between the output of PRBG and the message bits. The output of the PRBGs will be used as a pointer to a single bit in

an image in this suggested data encryption scheme. To acquire the ciphered bits, a logical operator such as XOR'ing is used between the message bits and the image bits. The intrinsic qualities of the pictures, such as bulk data capacity, high redundancy, and strong pixel correlation, are employed to increase the algorithm's security. [41]

Chapter 4 Simulation of vulnerable applications and exploiting vulnerabilities.

Exploiting vulnerabilities will be described in this chapter to assist you comprehend the ultimate process. The scan checklist includes the most lethal assaults against Web applications. Possible attack routes and mitigation mechanisms are also presented. That can be a variety of things, depending on the application. We will create a virtual lab for demonstration of simulation. List of applications for configuration of virtual lab.

1. Virtualbox
2. Kali Linux
3. BurpSuite
4. PortSwigger demo vulnerable applications

As a main operating system we will select Linux. We will continue Ubuntu distribution of Linux. Installation of Virtual Box with images [42]

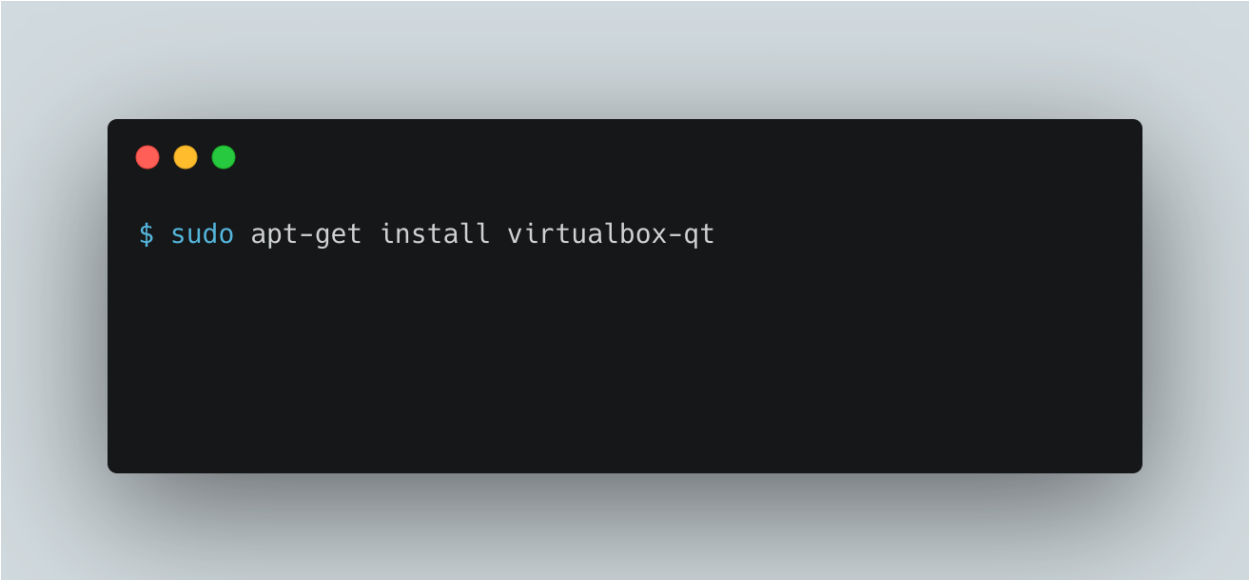


Figure 37: Terminal of Linux

This command will install Virtualbox for Linux. We will go to the Kali Linux https://www.kali.org/get-kali/ downloading image for Virtualbox. We will import Kali Linux image inside VirtualBox [43]

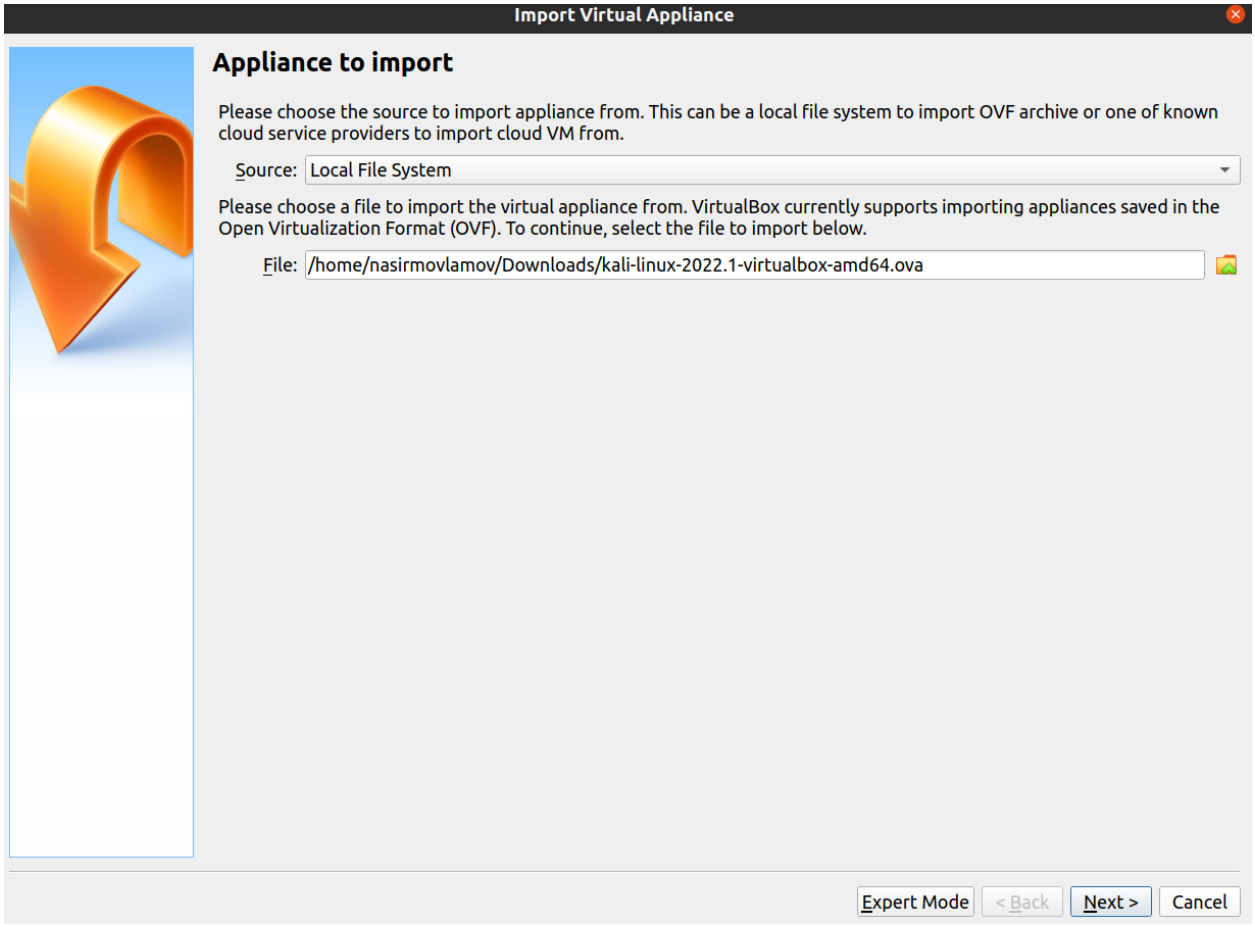


Figure 38: Virtualbox importing Kali Linux

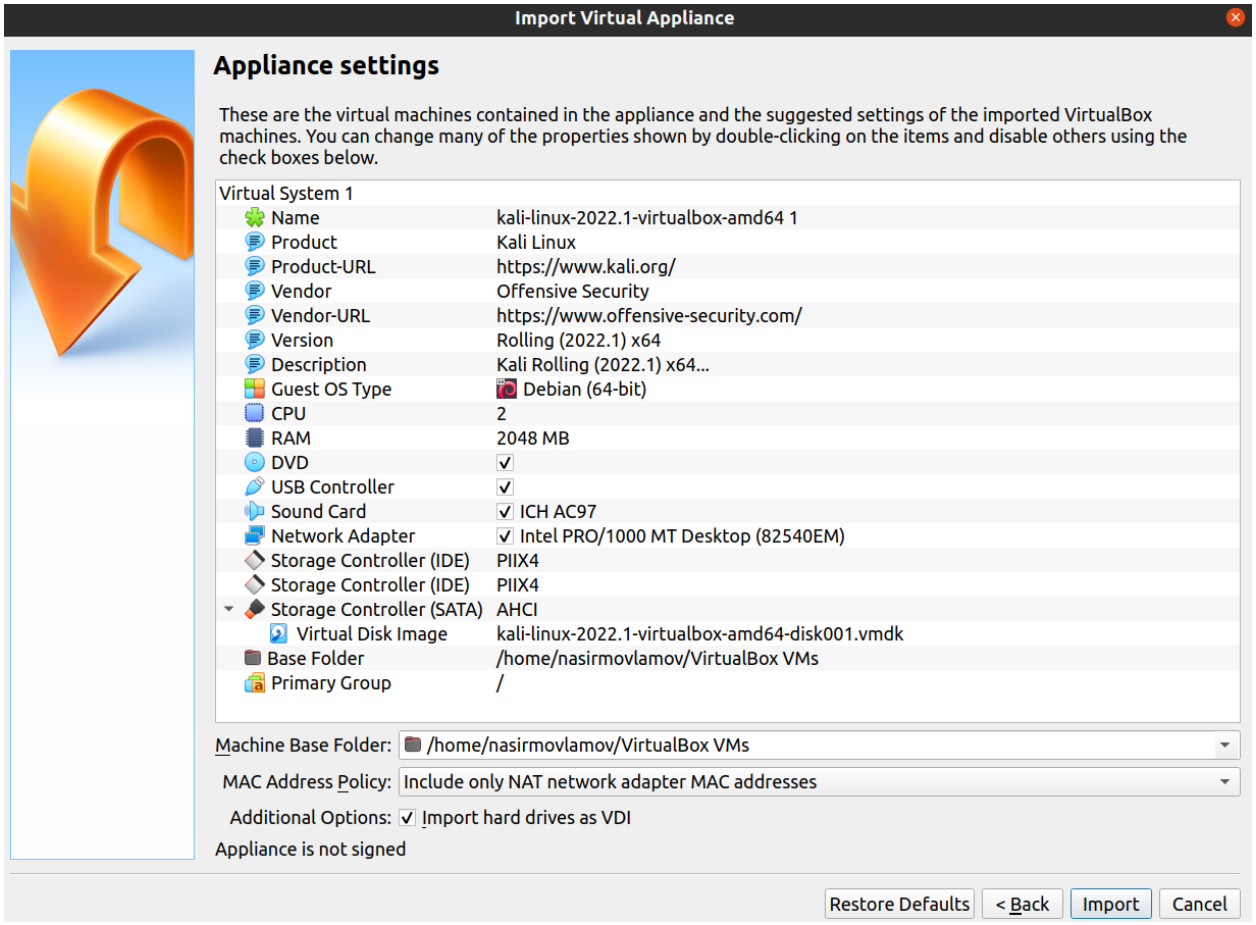


Figure 39: VirtualBox import settings

After adding Kali Linux inside Virtualbox it will appear as image below [44]

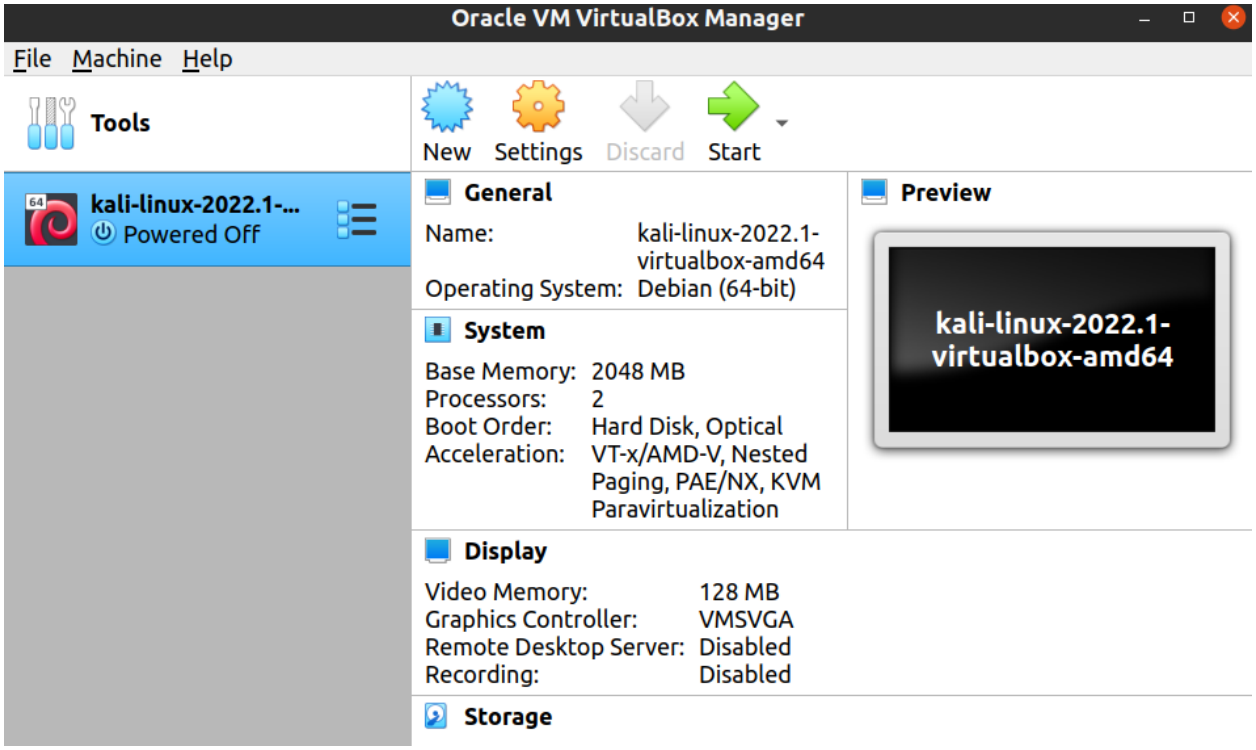


Figure 40: Starting Kali Linux

We will start Kali Linux from VirtualBox

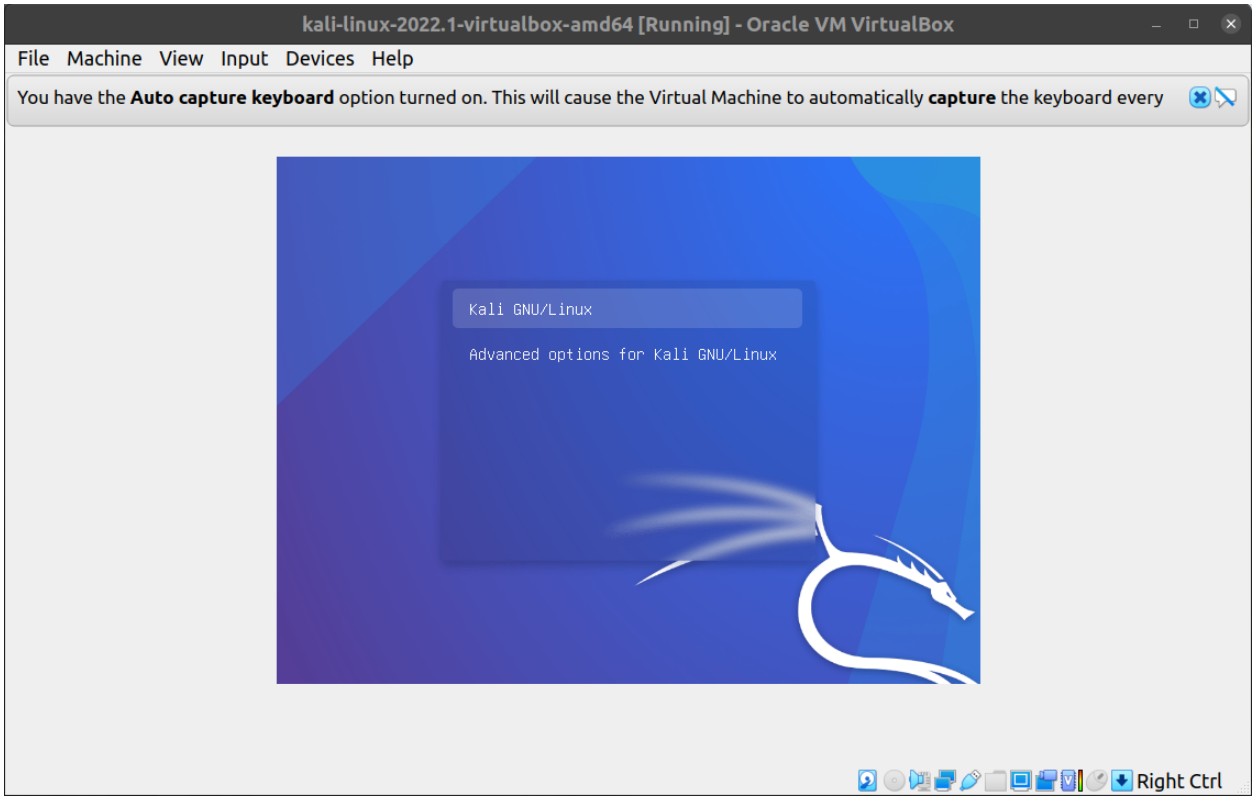


Figure 41: Opening Kali Linux

Then we will enter Kali Linux

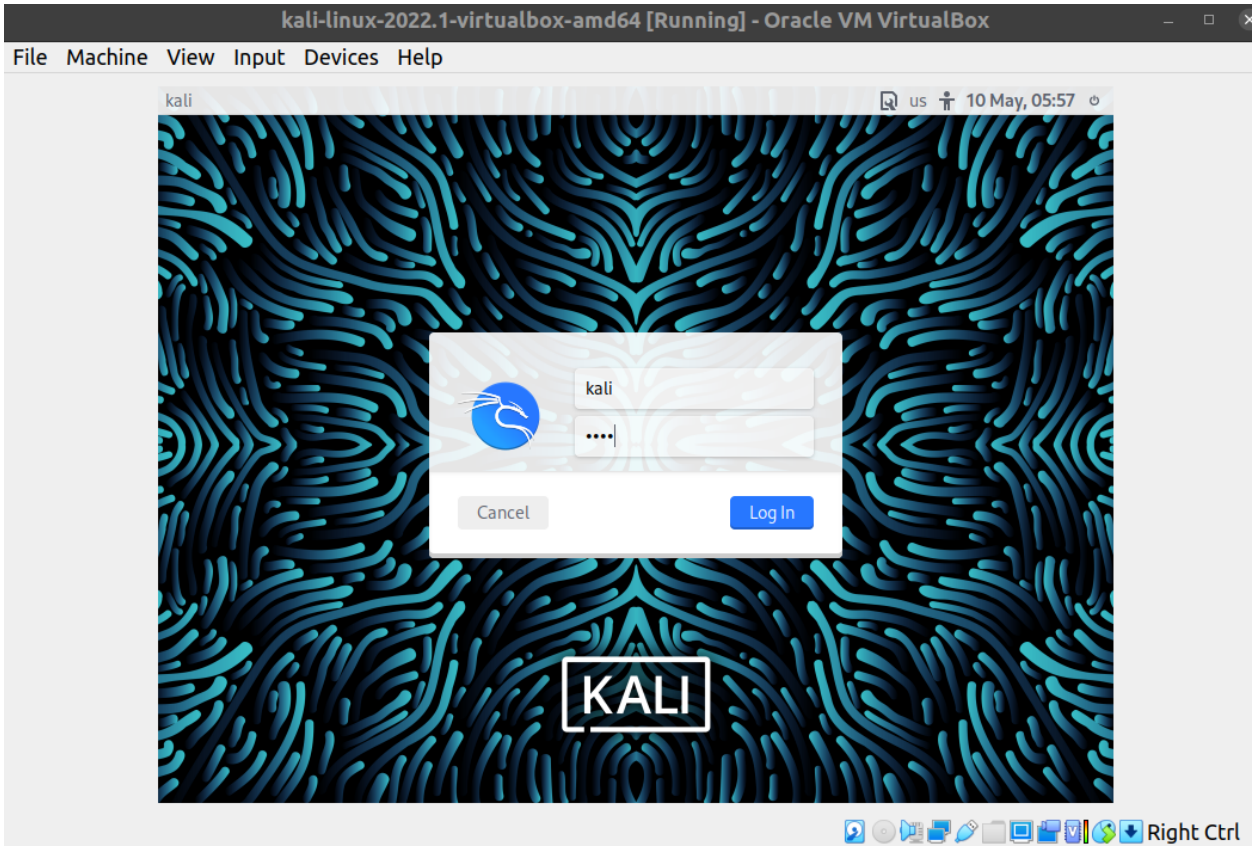


Figure 42: Kali Linux

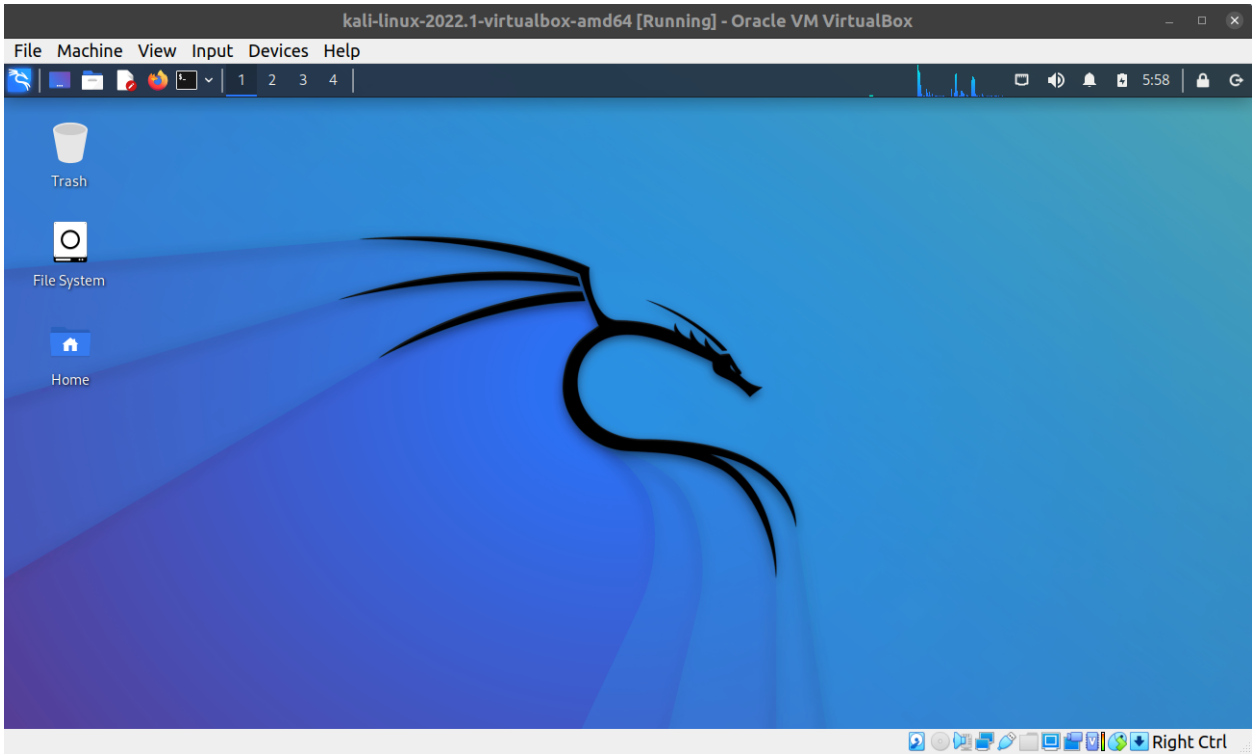


Figure 43: Kali Linux start menu

Our virtual lab is ready for penetration testing of different applications. For testing we will use online vulnerable apps from portswigger.

4.1. Sql injection .

SQL injection is a cybersecurity flaw that allows an attacker to meddle with database queries made by an application. It typically enables an attacker to examine data that they would not otherwise be able to get. This might include data belonging to other users or any other data that the program has access to. An attacker can often edit or destroy this data, resulting in lasting changes to the application's content or behavior. An attacker can escalate a SQL injection attack to compromise the underlying server or other back-end infrastructure, or launch a denial-of-service attack in specific circumstances.

What are the consequences of a successful SQL injection attack?

Unauthorized access to sensitive data, such as passwords, credit card information, or personal user information, can arise from a successful SQL injection attack. SQL injection attacks have been the cause of several high-profile data breaches in

recent years, resulting in reputational harm and regulatory fines. In some situations, an attacker can get a persistent backdoor into an organization's systems, resulting in a long-term compromise that can go undetected for a long time. There are several SQL injection vulnerabilities, attacks, and strategies that can occur in a number of settings. Examples of typical SQL injection include:

- Retrieving hidden data, in which you may change a SQL query to retrieve more results.
- Subverting application logic is the process of changing a query in order to interfere with the application's logic.
- UNION attacks allow you to obtain data from several database tables.
- Examining the database, where you may extract information about the database's version and structure.
- The results of a query you control are not returned in the application's answers due to blind SQL injection.

Vulnerable application for sql injection from portswigger [45]

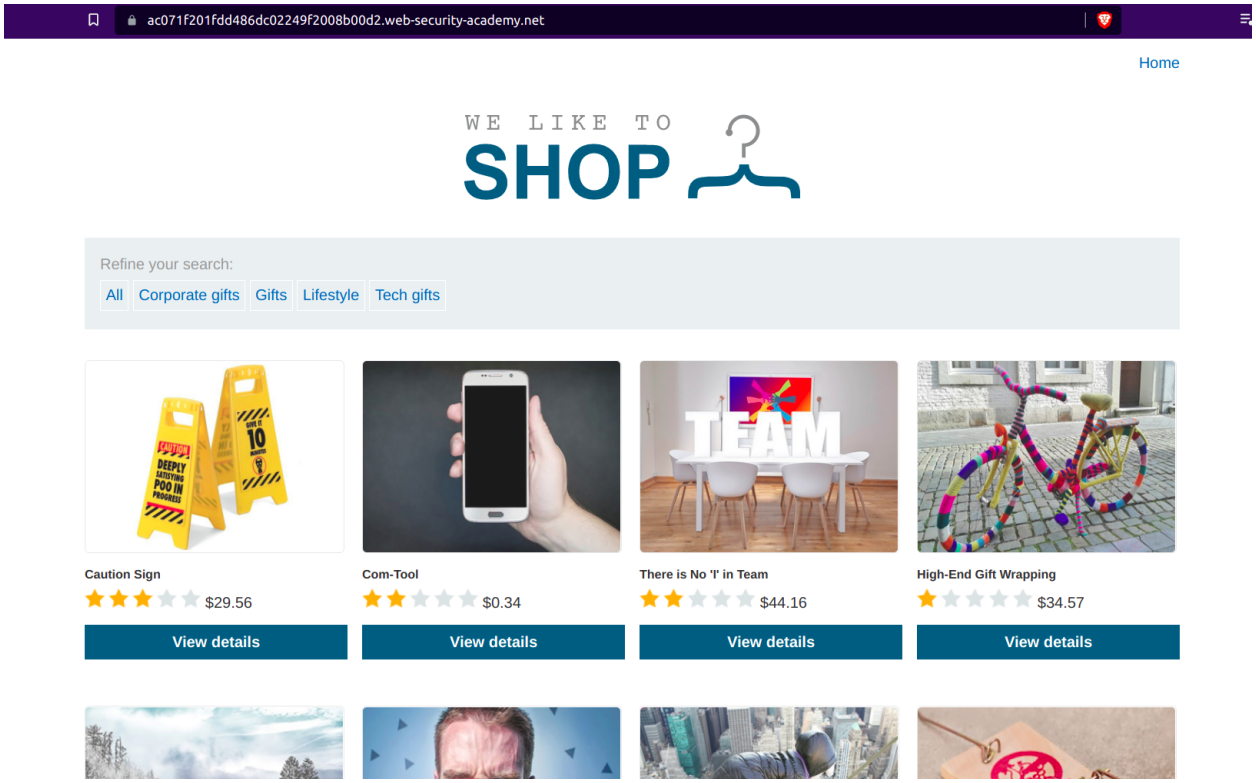


Figure 44: Demo vulnerable app

It is a simple e-commerce application based on selling different gifts. We will try fuzzy parameters on the link to access hidden data in this application.

https://ac071f201fdd486dc02249f2008b00d2.web-security-academy.net/filter?category=Gifts+test

In the link we tried a simple test word to see if it will appear in application or not.

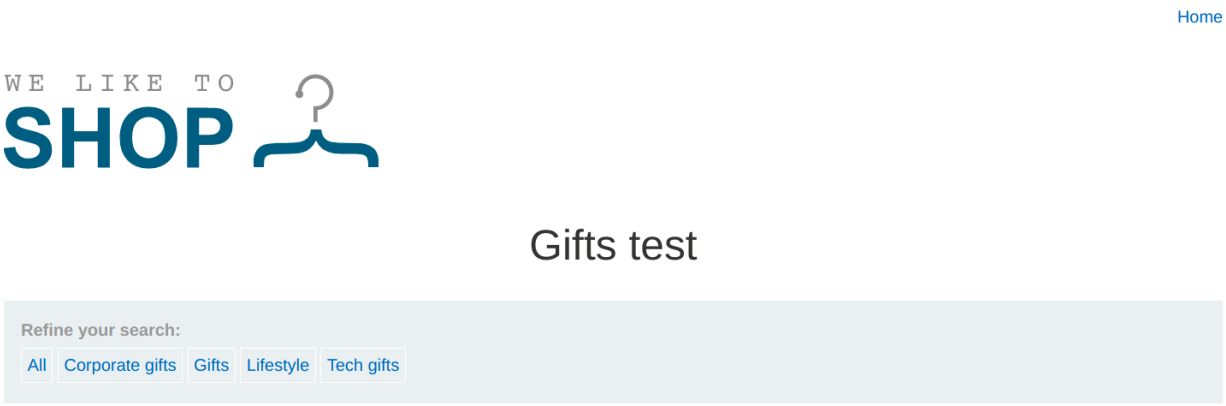


Figure 45: Demo vulnerable app

That’s all it appeared inside the application. It has the potential risk of sql injection. We will modify the category parameter, giving it the value '+OR+1=1--.

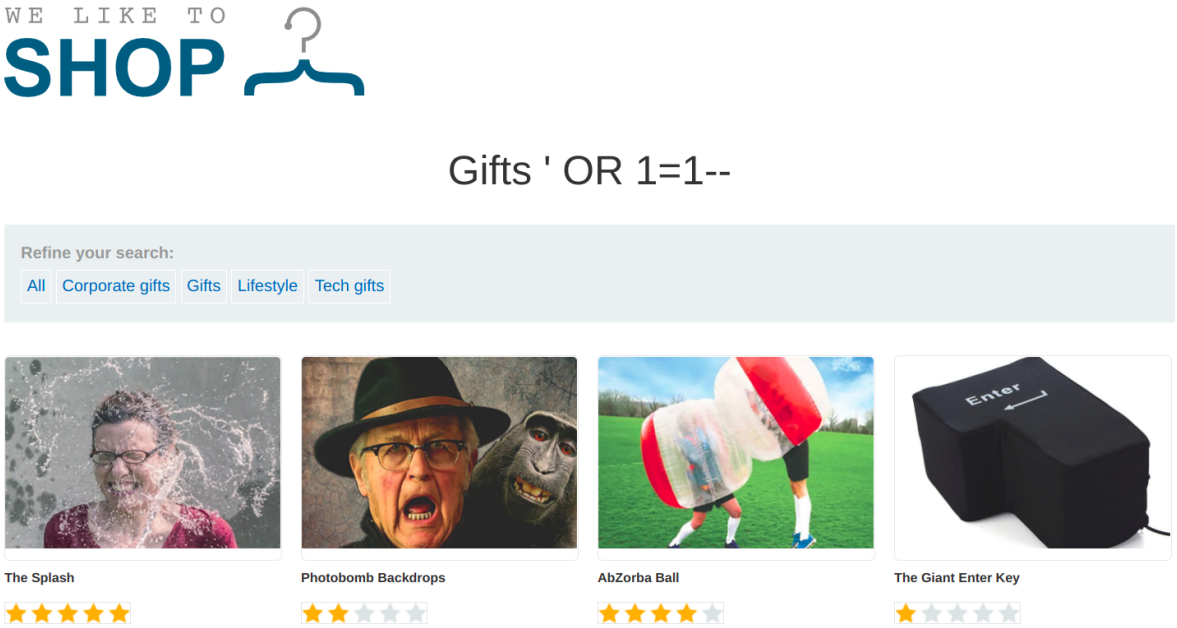


Figure 46: Demo vulnerable app

That’s all we have access to hidden data from the database.

In this part we will look at login bypass with sql injection exploitation.

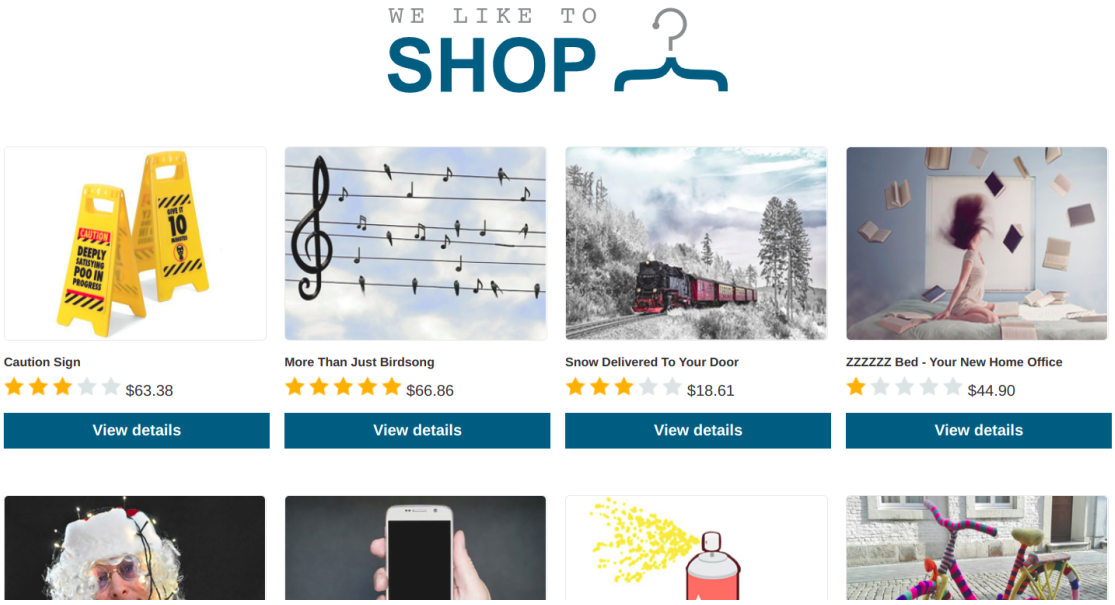


Figure 47: Demo vulnerable app

For this demo we will use BurpSuite tool from Kali Linux

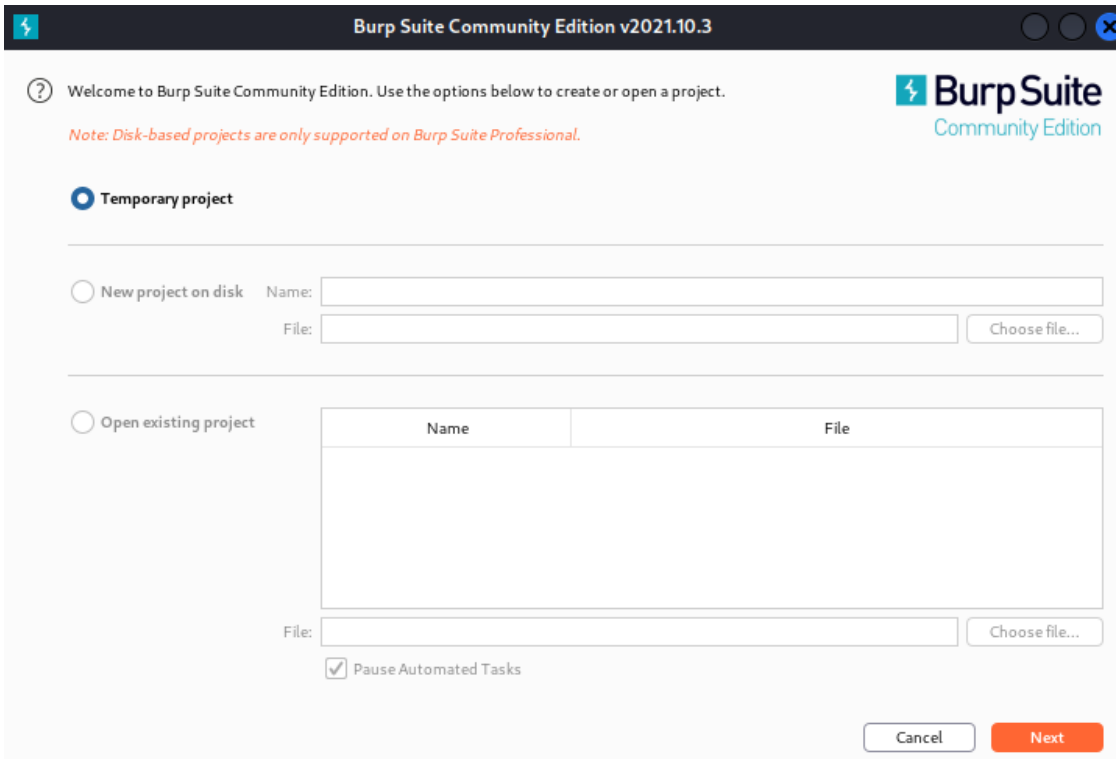


Figure 46: BurpSuite

Inside BurpSuite will come to the proxy tab and open with the browser. We will open a link to our demo vulnerable application.

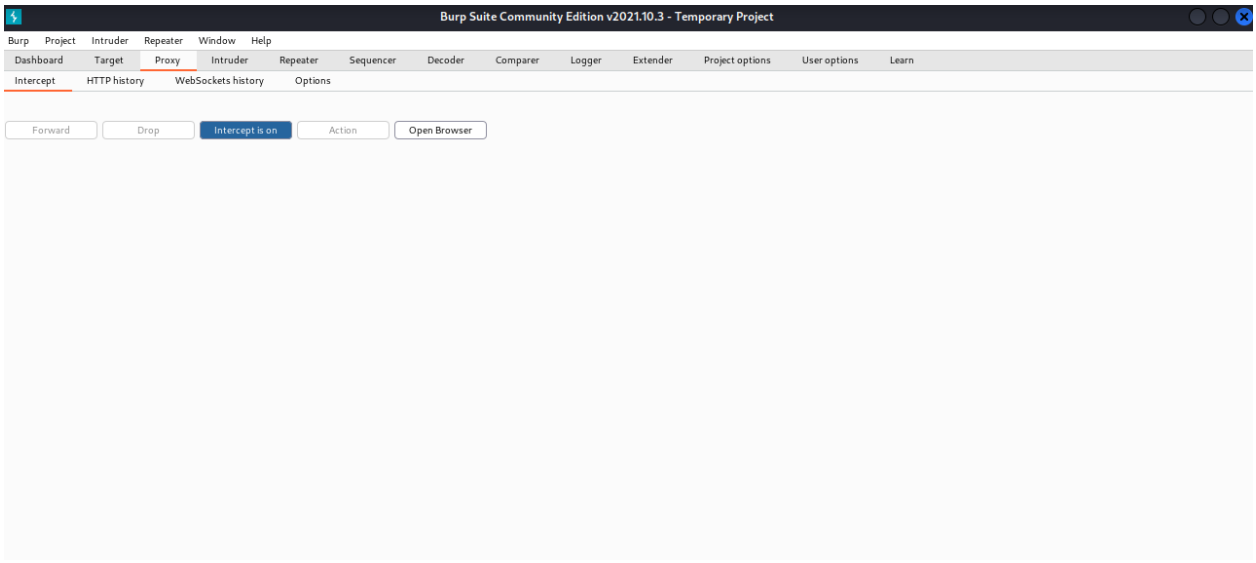


Figure 47: BurpSuite

In login page we will try to bypass application and gain access to administrator

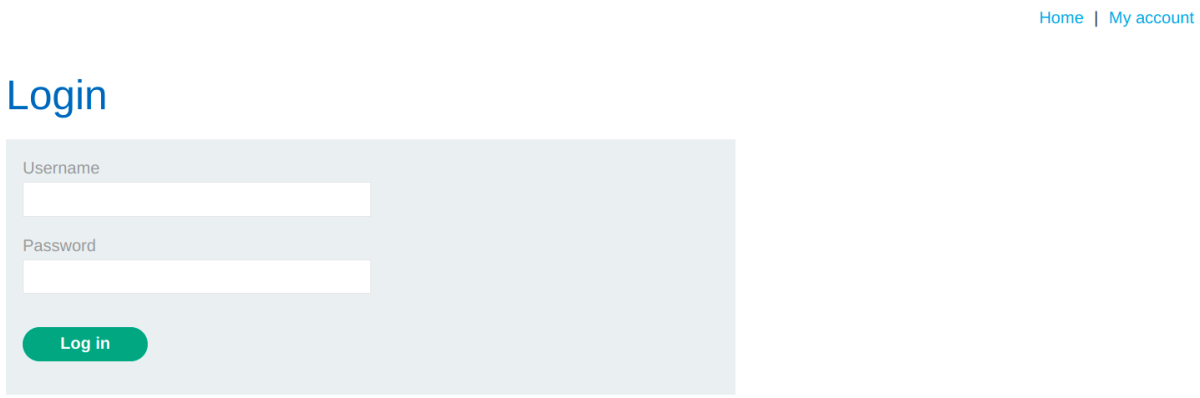


Figure 48: Demo vulnerable app

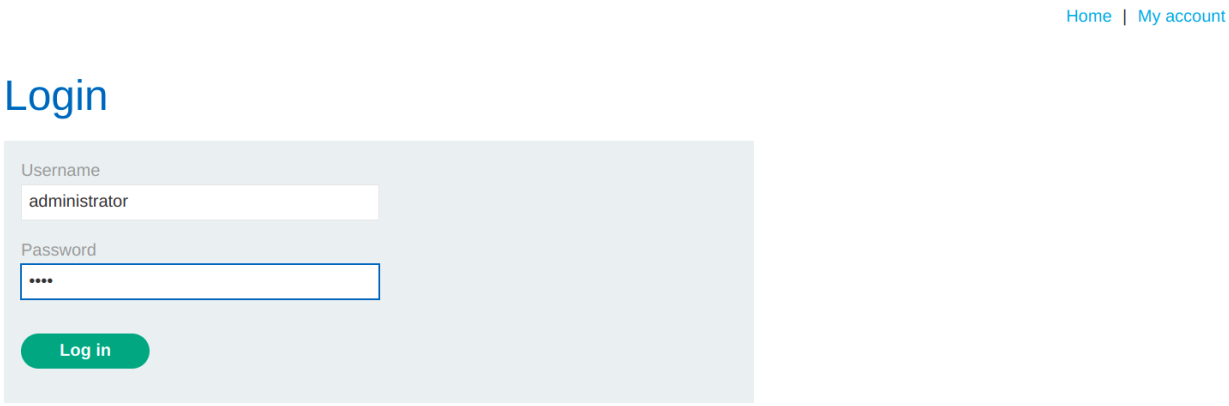


Figure 49: Demo vulnerable app

We have entered name and password.

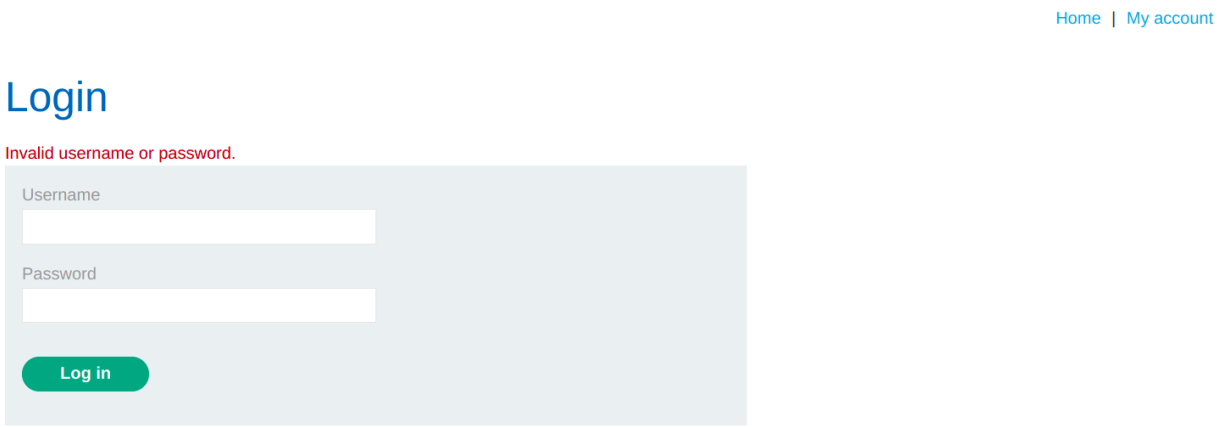
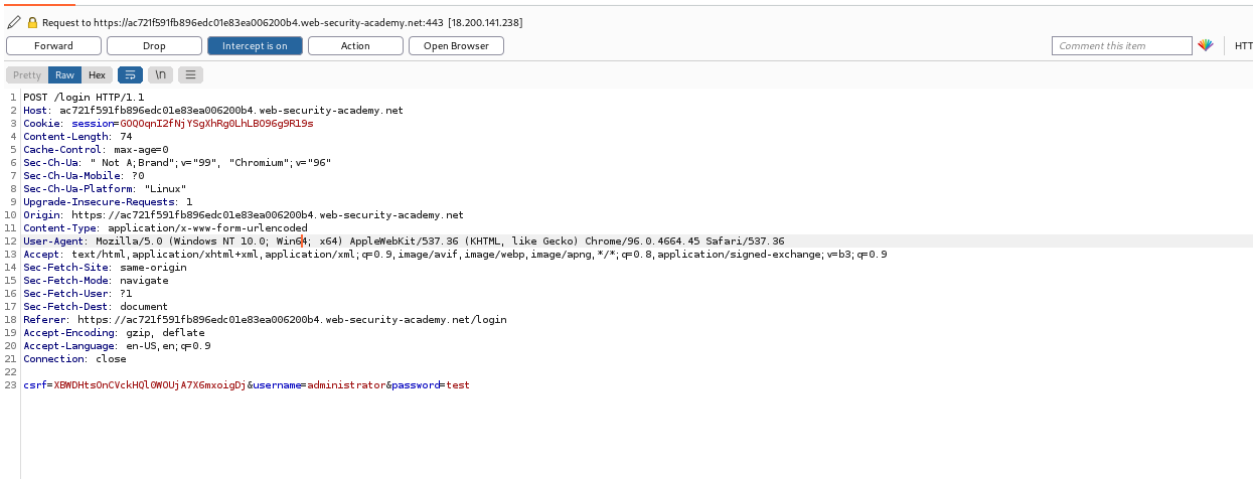


Figure 49: Demo vulnerable app

Response from the application will be like this. Now, we will try to manipulate application request headers.




```
csrf=XBWDHtsOnCVckHQl0WOUjA7X6mxoigDj&username=administrator'--&password=test
```

Figure 50: BurpSuite

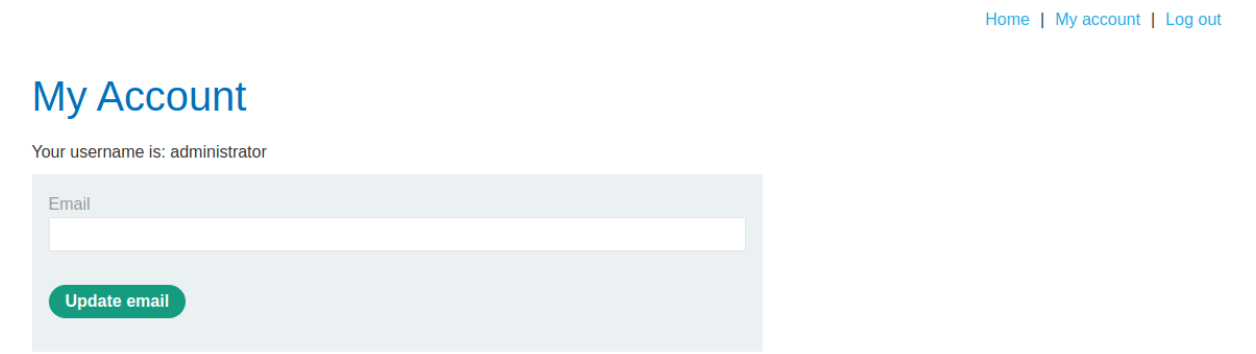


Figure 51: Demo vulnerable app

Now we gained access to the administrator account.

Conclusion

The most critical vulnerabilities that affect today's Web have been discussed in previous chapters. To ensure the security of Web systems, every Web application developer should be aware of at least these vulnerabilities and be able to identify and avoid them. Companies must recognize that their Web application is frequently an invitation to the attacker.

This thesis implies that one should not rely solely on automatic Web scanning tools. The best thing to do is to have a professional penetration tester evaluate the Web application on a frequent basis, after each modification. However, this is not always achievable. In such instances, the approach can still be provided to actual software developers. This thesis proposes such an approach, and the experiment's findings indicate that it is a viable option for detecting the most critical Web vulnerabilities. Even while a professional penetration test is typically far more extensive, it might be costly. As a result, teaching the process to software developers might be considered as a low-cost option. If the site is too huge for a manual penetration test, using an automated Web vulnerability scanner may be a smart choice. However, it is important to understand which dangers a Web scanner will detect and which will not, and it should be used with caution. It may be effective in avoiding compromise by the least competent attackers, referred to as "script kiddies" and automated malware. Penetration testers should not depend on the results of Web scanners and should confirm every discovered vulnerability if they want to use them.

As an optimal way to perform penetration testing seems a combination of automation and manual elaboration to find vulnerabilities faster. First, the penetration tester should examine the application, gathering all possible attack points. Then it is possible to use tools that automatically try to detect vulnerabilities for these attack points. If there is a suspicion of a presence of a vulnerability, the tool should alert the penetration tester. The penetration tester then should manually evaluate the alert to avoid false positives.

The experiment also indicated one promising study avenue. The attackers have their own method of assaulting the websites. These fingerprints might be gathered, for example, in the form of server logs, and used to identify the attacker. In this manner, it may also be feasible to distinguish between automated and manual attacks, assuming that the automated assault is carried out by malware of some type and follows a clearly recognizable pattern.

References

1. Adam Ali.Zare Hudaib, ”The Principles of Modern Attacks Analysis for Penetration Tester”, International Journal of Computer Science and Security (IJCSS), Volume (9):Issue(2):201

2. Nuno Antunes and Marco Vieira, ”Portugal Penetration Testing for Web Services”, University of Coimbra, 0018-9162/14/2014 IEEE5

3. Kennedy A. Torkura, Muhammad I.H. Sukmana, Christoph Meinel, HassoPlattner-Institute University of Potsdam, Potsdam ”Integrating Continuous Security Assessments in Microservices and Cloud Native Applications.”, https://doi.org/10.1145/3147213.3147229

4. Jain, T., & Jain, N. (2019). Framework for Web Application Vulnerability Discovery and Mitigation by Customizing Rules Through ModSecurity. 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 643–648. https://doi.org/10.1109/SPIN.2019.8711673

5. Mukhopadhyay, D., Karmakar, S., Meshram, A., & Jadhav, A. (2019). A Prototype of IoT based Remote Controlled Car for Pentesting Wireless Networks. 2019 Global Conference for Advancement in Technology (GCAT), 1–7. https://doi.org/10.1109/GCAT47503.2019.8978354

6. Farah, T., Alam, D., Kabir, Md. A., & Bhuiyan, T. (2015). SQLi penetration testing of financial Web applications: Investigation of Bangladesh region. 2015

World Congress on Internet Security (WorldCIS), 146–151. https://doi.org/10.1109/WorldCIS.2015.7359432

7. 11. G. Pujolle, A. Serhrouchni, and I. Ayadi, “Secure session management with cookies”, in 2009 7th International Conference on Information, Communications and Signal Processing (ICICS), 2009, pp. 1–6. DOI: 10.1109/ICICS.2009.5397550.

8. Pereira, R., & Serrano, J. (2020). A review of methods used on IT maturity models development: A systematic literature review and a critical analysis. Journal of Information Technology, 35(2), 161–178. https://doi.org/10.1177/0268396219886874

9. Y.-W. Huang, F. Yu, C. Hang, C.-H. Tsai, D.-T. Lee, and S.-Y. Kuo, “Securing web application code by static analysis and runtime protection”, in Proceedings of the 13th International Conference on World Wide Web, ser. WWW ’04, New York, NY, USA: ACM, 2004, pp. 40–52, ISBN: 1- 58113-844-X. DOI: 10.1145/988672.988679. [Online]. Available: http: //doi.acm.org/10.1145/988672.988679.

10. 38. Wright, C.S., 2007. SANS Institute, A Taxonomy of Information Systems Audits, Assessments and Reviews.

11. 35. Thomas, D., 2002. Hacker culture, Minneapolis: University of Minnesota Press

12. Skibell, R., 2003. The Phenomenon of Insecure Software in a Security-focused World. Available at: http://grove.ufl.edu/~techlaw/vol8/issue2/skibell.html [Accessed August 25, 2010].

13. M. Johns and J. Winter, “Requestrodeo: Client side protection against session riding?”, Feb. 2019

14. A. W. Marashdih and Z. F. Zaaba, “Cross site scripting: Removing approaches in web application”, Procedia Computer Science, vol. 124, pp. 647 –655, 2017, 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November

2017, Bali, Indonesia, ISSN: 1877-0509. DOI: https: //doi.org/10.1016/j.procs.2017.12.201. [Online]. Available: http: //www.sciencedirect.com/science/article/pii/S1877050917329691

15. Y. Demchenko, L. Gommans, C. de Laat, and B. Oudenaarde, "Web services and grid security vulnerabilities and threats analysis and model", in The 6th IEEE/ACM International Workshop on Grid Computing, 2005., 2005, 6 pp.–. DOI: 10.1109/GRID.2005.1542751

16. WhiteHat Security, 2007. WhiteHat Website Security Statistics Report.

17. E. Fong, R. Gaucher, V. Okun, P. E. Black, and E. Dalci, "Building a test suite for web application scanners", in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008, pp. 478–478. DOI: 10.1109/HICSS.2008.79.

18. T. Groß, B. Pfitzmann, and A.-R. Sadeghi, "Browser model for security analysis of browser-based protocols", in Computer Security – ESORICS 2005, S. d. C. di Vimercati, P. Syverson, and D. Gollmann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 489–508, ISBN: 978- 3-540-31981-8.

19. Sauer, J. & Lee, A., 2010a. What is Black Box Testing? - A Word Definition From the Webopedia Computer Dictionary. Available at: http://www.webopedia.com/TERM/B/Black_Box_Testing.html [Accessed August 26, 2010].

20. BlackDuck, Open source security and risk analysis report, 2018. [Online]. Available: https : / / www . blackducksoftware . com / open - source - security-risk-analysis-2018.

21. B. Eshete, A. Villafiorita, and K. Weldemariam, "Early detection of security misconfiguration vulnerabilities in web applications", in 2011 Sixth International Conference on Availability, Reliability and Security, 2011, pp. 169–174. DOI: 10.1109/ARES.2011.31.

22. First.org, Common vulnerability scoring system v3.0, 2015. [Online]. Available: https : / / www . first . org / cvss / cvss - v30 - specification - v1.8.pdf.

23. L. K. Shar and H. B. K. Tan, “Defending against cross-site scripting attacks”, Computer, vol. 45, no. 3, pp. 55–62, 2012, ISSN: 0018-9162. DOI: 10.1109/MC.2011.261.

24. D. Huluka and O. Popov, “Root cause analysis of session management and broken authentication vulnerabilities”, in World Congress on Internet Security (WorldCIS-2012), 2012, pp. 82–86.

25. X.-W. Huang, C.-Y. Hsieh, C. H. Wu, and Y. C. Cheng, “A token-based user authentication mechanism for data exchange in restful api”, in 2015 18th International Conference on Network-Based Information Systems, 2015, pp. 601–606. DOI: 10.1109/NBiS.2015.89.

26. Nmap. (2021). Nmap. Nmap. https://nmap.org/

27. Scambray, J., 2006. Hacking exposed : Web applications 2nd ed., New York: McGrawHill.

28. K. Singh, A. Moshchuk, H. J. Wang, and W. Lee, “On the incoherencies in web browser access control policies”, in 2010 IEEE Symposium on Security and Privacy(SP), vol. 00, 2010, pp. 463–478. DOI: 10.1109/ SP.2010.35. [Online]. Available: doi.ieeecomputersociety.org/10. 1109/SP.2010.35.

29. Stallings, W., 2007. Network security essentials : applications and standards 3rd ed., Upper Saddle River NJ: Pearson Education.

30. M. M. Hassan, S. Nipa, M. Akter, R. Haque, F. Deepa, M. Mostafijur Rahman, M. A. Siddiqui, and M. H. Sharif, “Broken authentication and session management vulnerability: A case study of web application”, International Journal of Simulation: Systems, Science & Technology, vol. 19, Apr. 2018. DOI: 10.5013/IJSSST.a.19.02.06.

31. M. Johns, “Code injection vulnerabilities in web applications - exemplified at cross-site scripting”, it - Information Technology, vol. 53, pp. 256–, Sep. 2011. DOI: 10.1524/itit.2011.0651.

32. Sima, C., 2005. DevCity.NET :: Security Risk Assessment and Management in Web Application Security. Available at: http://devcity.net/Articles/187/1/article.aspx [Accessed August 27, 2010].

33. Ferraiolo, D.F. & Kuhn, D.R. (October 1992). "Role-Based Access Control" (PDF). *15th National Computer Security Conference*: 554–563.

34. Schatz, Daniel; Bashroush, Rabih; Wall, Julie (2017). "Towards a More Representative Definition of Cyber Security". *Journal of Digital Forensics, Security and Law*. **12** (2). ISSN 1558-7215.

35. Penetration Test Guidance Special Interest Group 2015. Information Supplement: Penetration Testing Guidance. PCI Security Standards Council. Cited 1.3.2016, https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pd f/.

36. Kessler, Gary (November 17, 2006). "An Overview of Cryptography". *Princeton University*.

37. Data Encryption in Transit Guideline | Information Security Office". *security.berkeley.edu*.

38. Unisys, Dr Glen E. Newton (2013-05-07). "The Evolution of Encryption". *Wired*. ISSN 1059-1028. Retrieved 2020-04-02.

39. Bek, E. (19 May 2016). "Protect Your Company from Theft: Self Encrypting Drives". *Western Digital Blog*. Western Digital Corporation. Retrieved 8 May 2018.

40. Sotirov, A., Stevens, M. & Appelbaum, J., 2008. MD5 considered harmful today. Available at: http://www.win.tue.nl/hashclash/rogue-ca/ [Accessed August 30, 2010].

41. Yan Li; Nakul Sanjay Dhotre; Yasuhiro Ohara; Thomas M. Kroeger; Ethan L. Miller; Darrell D. E. Long. "Horus: Fine-Grained Encryption-Based Security for Large-Scale Storage" (PDF). *www.ssrc.ucsc.edu*. Discussion of encryption weaknesses for petabyte scale datasets.

42. Kali Linux Official Documentation, 2016. What is Kali Linux. Cited 14.3.2016, http://docs.kali.org/introduction/what-is-kali-linux/.

43. Virtualbox Official Documentation, 2022. https://www.virtualbox.org/wiki/Documentation

44. BurpSuite Official Documentation, 2022. https://portswigger.net/burp/documentation

45. PortSwigger Labs. https://portswigger.net/web-security/all-labs