University of South-Eastern Norway

www.usn.no

FMH606 master's Thesis MT-31-19

Process Technology

Software Testing, Data Security and GDPR

May 14, 2019

Michael Ogechi Agbamoro

Faculty of Technology, Natural sciences and Maritime Sciences

Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 master's Thesis MT-31-19

Title: < Software Testing, Data Security and GDPR

Number of pages: <61>

Keywords: Data security, security testing, GPDR, threats, trends, Edemso, test tools, test software.

Student: Michael Ogechi Agbamoro

Supervisor: Hans-Petter Halvorsen

External partner: Cevia Solutions

Availability: Open

Summary:

Software and data security are an important part of a modern-day business strategy of any organization as it forms the basis for guaranteed security of information, data, and assets of any organization. This project is principally focused on performing security test on the Edemso software to find vulnerabilities, based on the security test results determine the security level of the Edemso software and propose improvements regarding the general security of the Edemso software.

In this project, extensive and detailed analyses of common security threats, various software security testing tools, security testing method, and GDPR were performed. Based on the analysis a preferred security testing method was determined and the Edemso software was properly examined and tested and the findings meticulously documented and analyzed.

After proper and extensive examination of the Edemso software, about 36 vulnerabilities or loopholes were discovered, 75% of which are low risk, 25% are critical risk level vulnerabilities and there was zero severe high-risk level vulnerability. Attempts to exploits the discovered vulnerabilities were unsuccessful and proposed solutions to eliminating these critical vulnerabilities found in the project are; an update of the OpenSHH on port 22tcp and reconfigurations in the internet information service manager of the Edemso software. Based on the project results the security level of the Edemso software was determined to be in good shape and if documented recommendations for improvements are implemented there will be a significant improvement of the security level of the Edemso software.

Preface

Protection of network systems and platforms from cyber-attack and unauthorized access is an important issue facing all organizations and businesses in this era of internet of things. This project work focuses on the overall and complete protection strategy of a cloud-based startup company.

Special thanks to my Supervisor Hans-Petter Halvorsen for the great supervision and encouragement during this project

Special thanks to Erling Ekrene the Founder of Cevia Solutions for his quick response to all my inquiries

To my wife Gift and son Nathan, thanks for the many midnights of staying awake with me.

Porsgrunn, 14th of May 2019

Michael Agbamoro

Contents

Nomenclature

- OWASP- Open Web Application Security Project
- ISO/IEC 270021- International Standard Organization/ International
- CVE - Common Vulnerabilities and Exposures (CVE)
- GDPR- General Data Protection Regulation
- AWS -Amazon Web Services
- JavaFX- software platform for creating and delivering desktop applications
- GPS- Global Positioning System
- EU- European Union
- EEA- European Economic Area
- IP- Internet Protocol address
- HTTP- Hypertext Transfer Protocol
- DoS- Denial of Service
- XSS- Cross-site scripting
- XXE- XML External Entities
- XML- Extensible Markup Language
- SQL- Structured Query Language
- OS- Operating System
- ORM- Object Relational Mapping
- LDAP- Lightweight Directory Access Protocol
- EL-Expression Language
- OGNL-Object Graph Navigation Library
- URL- Uniform Resource Locator
- HTML- Hypertext Markup Language
- API- Application Programming Interface
- DOM- Document Object Model
- ZAP-Zed Attack Proxy
- OWTF- Offensive Web Testing Framework
- CPE-Common Platform Enumerators
- VM-Virtual Machine
- SDLC-Systems Development Life Cycle
- CD- Compact Disc
- USB- Universal Serial Bus
- RAM- Random Access Memory
- CMS - Content Management System
- TCP- Transmission Control Protocol
- SSH- Secure Shell
- SSL- Secure Sockets Layer
- WAF Web Application Firewall
- IPS -Intrusion Prevention System

1 Introduction

This chapter covers the background studies of the project, the objectives, methods, and scope of the project and the structure of reporting. It also covers the basic description of the software, how the software works and the software architecture

1.1 Background

Organizations, corporations and businesses both big and small have become increasingly dependent on the internet, information technology, cloud computing, social media, automation, machine learning, and big data as they try to move and secure their information, data and asset online. This has resulted in more extensive research in recent times in software and data security. Software and data security are not only a concern of businesses but countries across the world have also beefed up its capacity and resources to tackle the new security threat of the internet age which have become issues of national security. The increased focus on software and data security by all stakeholders on a local, national and global level is mostly because of the critical role software and data security plays as the world transits into the 4th industrial revolution called the internet of things. The implications are that any bridge in the security of a software or data infrastructure can result in catastrophic consequences like losses of millions of lives, financial losses, disruption of essential services and production process. A recent example was the attack on one of the world's largest producer of aluminum, Norway's Norsk Hydro by ransom attackers in March 2019 which cost the company around $50million in losses.

Across the world, organizations, corporations, businesses, and countries have reported the various attack on its software and data security infrastructure, some of which were successful and resulted in huge losses. The attackers continue to invent more advanced and innovative ways to exploit flaws, loopholes, and weakness in various online platforms, software, and network systems. Some of these attacks are because of flaws and loopholes in the design and implementation of the system, poor system configuration, using insecure networks, human errors and sometimes complexity of the system. Based on this backdrop, data and software security will continue to be an important research area for capacity building, innovation, and global cooperation as the world fully embraces the 4th industrial revolution.

The project focuses on the extensive data & software security testing of a cloud-based software called Edemso. It is expected to handle highly proprietary information and therefore will be required to be certified as secured and compliant with all standard data & software security testing. In this project, Edemso will undergo extensive scanning process to identify, measure and document all possible vulnerabilities. Security test will be performed on the Edemso software to test against all documented vulnerabilities, loopholes, and potential weakness. The security level will be determined, and the suggestive improvement recommendation will be documented. Figure 1-1 illustrates the security testing set up to be used in this project. Security tools will be used by the security tester to examine and test the Edemso software infrastructure and all findings will be documented and analyzed.

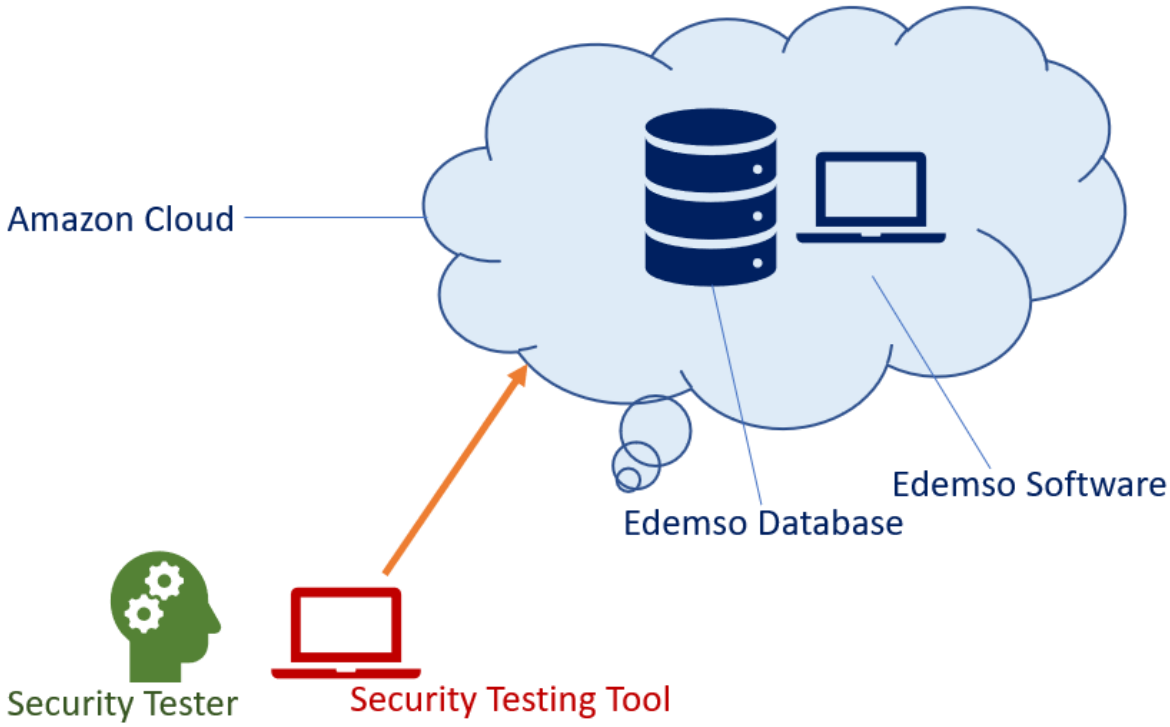

Figure 1-1 Overview of Security Testing System

Figure 1-2 shows the project overview and the method to be used in the project. There will be a general study covering every area of data security and software security testing. Security testing tools, operating systems, and platforms will be analyzed and the most suitable will be selected. Detailed test plans, test cases, and test environment will be developed for the project. Based on the test plan and method various security testing will be carried on the cloud-based software. The security test results are analyzed and interpreted, final recommendations for improvements are documented.

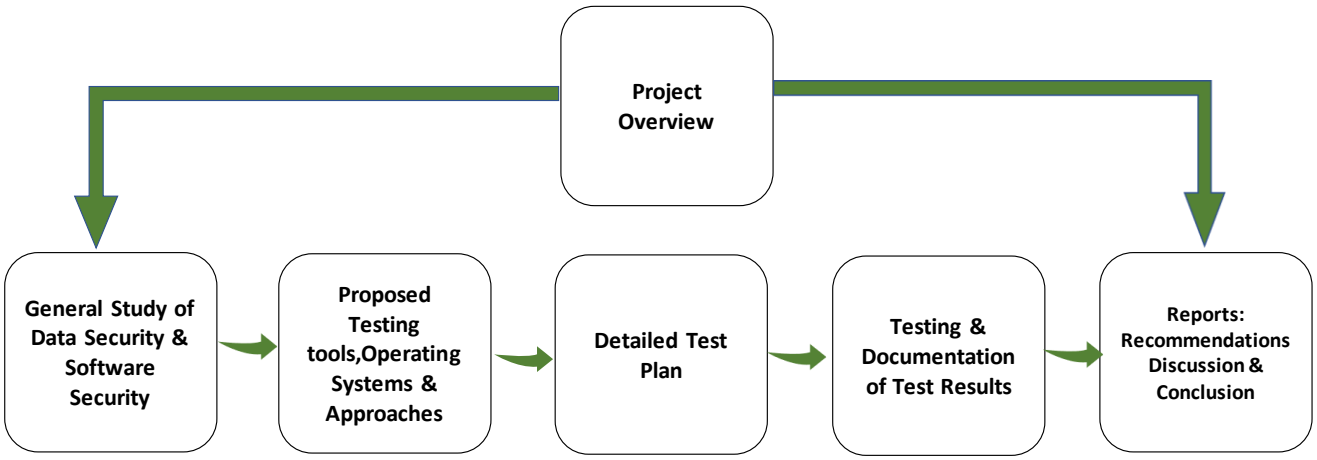

Figure 1-2 Project Overview

Chapter 2 covers detailed explanation on data security, the various common data security threats that can affect the Edemso software, data security platforms, the GDP, and ISO/IEC 27001 security standards

Chapter 3 gives a detailed overview of software security testing, explaining and analyzing various software security threats, common vulnerabilities according to Open Web Application Security Project (OWASP) which will be investigated in the Edemso software, the different Open Web Application Security Project (OWASP) projects different software testing technique and software testing operating platforms and systems.

Chapter 4 covers the details of the actual security testing to be performed on the Edemso software. it gives extensive literature studies on the security testing approach and various stages of the security testing of Edemso

Chapter 5 gives details on the different test plans, environments, and methods that can be used for testing the Edemso Software. the chapter also discusses the testing methods used to perform the security test on the Edemso software.

Chapter 6 gives details of the different security tools used to perform the security test on the Edemso software, operating platform, operating systems, and the security testing phases carried out in the project.

Chapter 7 documents the results of the software security test carried out on the Edemso software with the results of the security test for each phase is documented and analyzed using charts, figures, and tables.

1.2 Cloud-Based Software: Edemso

Edemso is a cloud-based management software designed to provide enhanced business efficiency to small and medium scale organizations in sectors like engineering, maritime and oil and gas. It's a working platform that tracks the movement of documents in an organization which could be internally (within departments in the organization, employee to employee) or externally (with customers, suppliers, contractors, etc.). Currently, in the tech market, there are many competing solutions trying to solve the problem of project management, workflow tracking, document storage, and documentation management, but none is known to provide the solution Edemso is providing which is a working platform that controls and manages documentation flow and movement. [1]

With Edemso, organizations can have an effective and efficient documentation system and track the movement of the documents. Edemso is unique in that with-it organizations can effectively run its product management, project management with no technical training required as the platform is user-friendly. It can store data up to 100GB and it accepts any file type. [1]

1.2.1 Structure & Architecture of Edemso

Edemso was built using Java 8, spring (boot, data, and security) for the backend and ReactJS for the frontend. The software was deployed on AWS (ec2, s3, and db) through ngnix and spring boot. Gitlab used as a repository and team city as ci. [2]

The software is web-based and has a desktop application that was written using JavaFX with intermediate RabbitMQ server. [2]

1.2.2 How Edemso Works

The login into Edemso can be done using a laptop, desktop computer, tablet, and smartphones. To work on a file on Edemso, the file program should be installed in the user device. The Edemso platform has an administrative setting that consist subsection which includes: account info, users, user groups, security levels, setup, and template. Each subsection serves different functions. [1]

Figure 1-3 shows the administrative platform of Edemso with different subsections. The Users subsection is where the users are added and given user groups and security level. The User group subsection is where different modules are set up and managed for each user group. The security level subsection is where users are given restricted access to specific documents, products, projects or quotations. Setup subsection is where documents are tagged to different groups. [1]

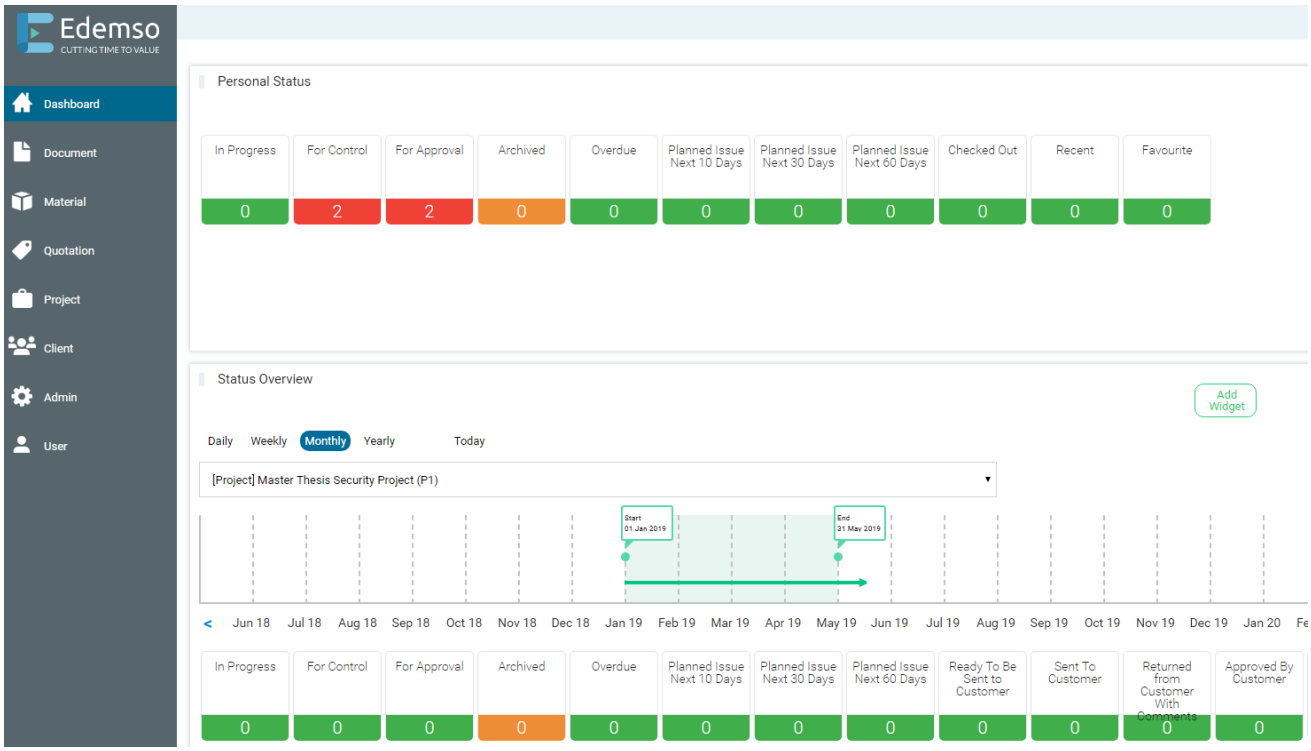

Figure 1-3 Edemso Administrative Platform [1]

Edemso is made up of four modules which are the document module, the product module, quotation module, and the client module. Figure 1-4 shows the Data model showing the relationship of all four modules in Edemso. [3]

From Figure 1-4 documents are created in the document module and linked with a product which is created in the product module. A product is linked to a quotation or a project. The document attached to the product will also be linked and accessible from a quotation (which can be created in the quotation module) or a project (which can be created in the project module). The client can be added in the client module and linked with all products, quotations, and projects associated with the client. The client will have a listing showing all associated product, quotation, and projects.

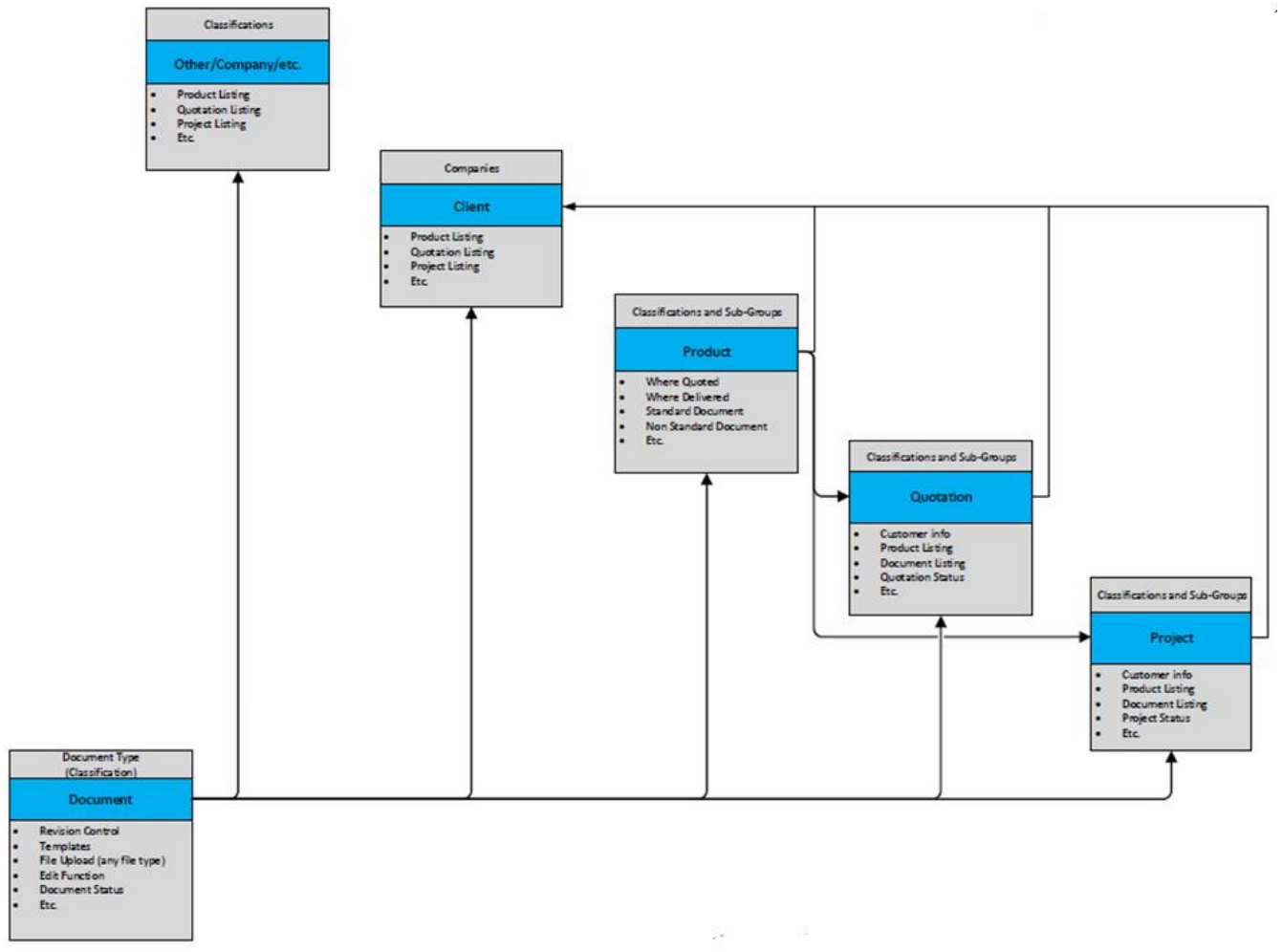

Figure 1-4 Data Model showing Edemso Modules [3]

Document Module

The document module houses all documents. A document can consist of several files which carry different information. Each file in the document can be worked on separately. All document attached to a product or a quotation will also be listed on the document list in the document module. [1]

Product Module

The product module manages the products and all attached document to any given product. Within a product module there are subsections which includes my product which shows the products and the responsible, all products which shows all product within the company, new product which is for creating new product and Edemso software which shows information about the product like document attached to the products, projects and quotations the product is under. [1]

Project Module

The project module is a system that manages and controls all projects and its individual attached documents and products. It is made of subsections which include my project which shows the project the logged in user is responsible for, all projects which shows all the project within the company and new project which is used to create new projects. [1]

Quotation Module

The quotation module is a system within Edemso that manages and controls all quotations and its individual documents and products attached to them. It is made up of subsections which are my quotations, which shows the quotation a logged in user is responsible for, all quotations which show all quotations within the company and new quotation which is used to create a new quotation. [1]

Client Module

The client module is a system that manages the client listing showing all products, quotations, projects, and documents attached to each client. [1]

2 Data Security

This chapter goes into a detailed explanation of data security and GDPR. It gives an overview of the evolving trends of data security. It discusses the various data security platforms, operating systems.

2.1 Introduction

In this age of big data and massive transition of information and data storage from the traditional physical systems to the modernized, digitized system of storage like cloud storage life has become more convenient for people but also the risk of data bridge which could result in harm has become even higher. Data security is, therefore, a crucial aspect of security for any organization in this technology age. Data security is a deliberate and systematic attempt to protect digital data or digital information using standards and technology, from any form of attack, unauthorized access or exploitation which could cause harm to an individual or organization. [4]

2.2 Trends and Threats

Organizations and individuals are constantly generating data in various forms which are mostly available on the internet. This data comes in forms like multimedia, GPS, smart homes, databases, documents and data collected during the process of using online services, like streaming, using social media and using digital devices like phones and tablets. [4] This has resulted in an unprecedented amount of data available on the internet which can be used by cybercriminals for harmful purposes. To address some the issues of security of data international laws and standards like the GDPR and the ISO/IEC 27001.2013 have been enacted. The guiding principles of these laws and standards are that all stored data should be owned to be clear on who is responsible to protect and control access to the stored data. [4]

Some known threats of data security are discussed below;

- Malware
- Spyware
- Ransomware
- Trojans
- Viruses
- Worms
- Rootkits
- Adware

2.2.1 Malware

Malware is a general name for any form of malicious software, code or script is written or designed with the principal objective of causing harm, damage or loss to the target computer system, software, application or network system. [5] The malware will carry out this objective after it has been introduced into the target system or application. Examples of known malware are viruses, worms, trojan horses, ransomware, spyware, adware, and bootkits. Figure 2-1

shows the various categories of malware and their degree of occurrence. To prevent these threats from attacking a target system or network protective measures like installing antivirus software and firewalls are important strategies to be adopted. [5]

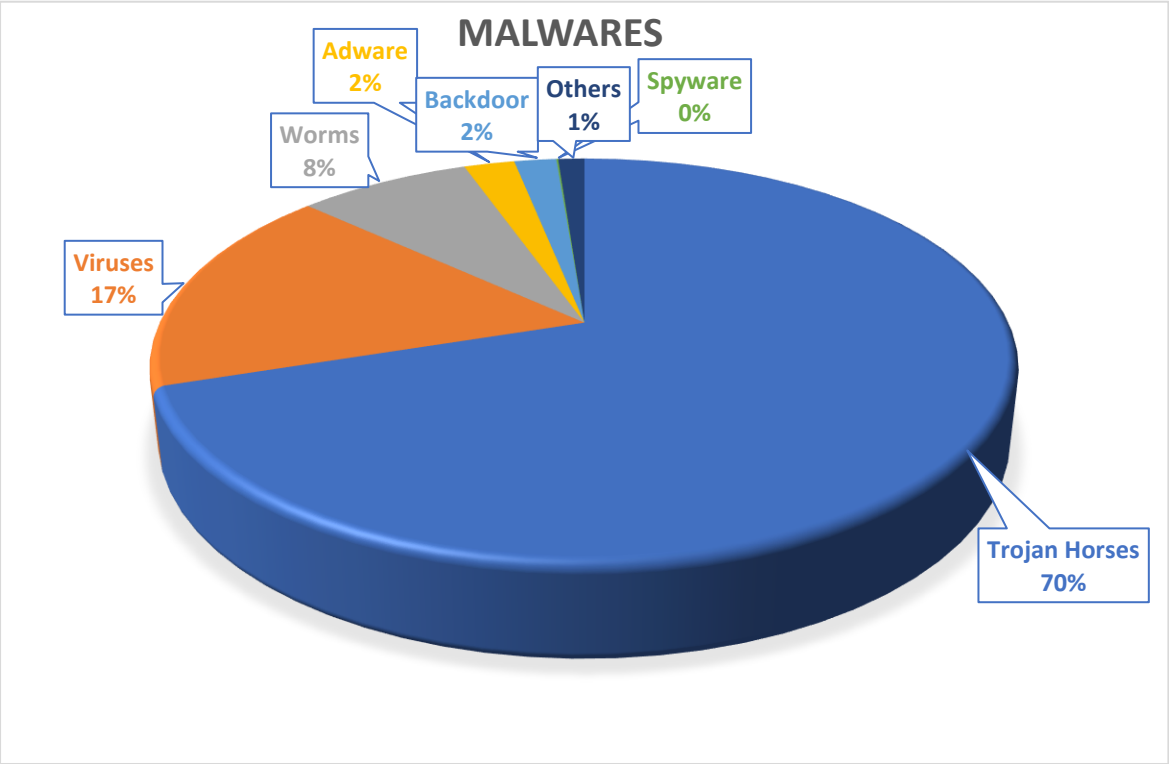

Figure 2-1 Various Categories of Malware

2.2.2 Spyware

Spyware is an information gathering software used to spy on individuals, systems, and organizations without their knowledge to get information about them for gaining control, access into the system or network of the organization illegally. Spyware could come in four different forms which are adware, system monitor, tracking cookies and trojans. To protect a system or application from spyware threats various anti-spyware software have been developed to provide real-time protection and to also detect any spyware in a system or application. [6]

2.2.3 Ransomware

Ransomware is malicious software that is used as an attack tool which is most often disguised as a verified software, file or link for a user to click, open or install when they come across them through emails or other means. Once this malicious software is installed into the system or network, the ransomware could lock the system out making it difficult with open for even security experts, the attacker thereafter holds the target ransom until a ransom is paid. [7] Like other security threats ransomware attacks can be prevented by using security software to protect the network system.

2.2.4 Trojans Horse

Trojan horse is a malicious program designed to misled users from its original purpose thereafter gaining unauthorized access to user information with the intent of causing harm to the target user. Trojan horse is commonly used by attackers for ransomware attacks and they generally do not infect other files or try to propagate once they gain access into a system. [8]

2.2.5 Viruses

Viruses are malicious software that is designed to reproduce itself once they are implemented infecting an entire system, network or application without the permission or knowledge of the target user of the system or network. Viruses are designed with different objectives in mind ranging from holding the target ransom to political purposes to illustrating that a system is vulnerable to such attacks. There are three main parts of a virus which are; the infection mechanism that reproduces the virus to infect the system, the trigger which is the action that triggers the virus to be activated it could be an action like clicking the file and finally the payload which is the actual part of the virus that executes the malicious activity. To prevent this form of threat in a system or network antiviruses are used to protect the system and detect all forms of the virus in a system before it infects the system. [9] [10]

2.2.6 Worms

Worms are malicious software that is developed simply to reproduce itself and spread into other computers to disturb the network and consume bandwidths. Worms are not designed to cause serious harm in a system but to simply spread across computers. [9] [10]

2.2.7 Rootkits

Rootkits are an exploitation tool used by attackers to exploit known vulnerabilities like privileged escalation or password attack by gaining unauthorized access into a system for causing harm to the target system. Rootkits do not grant unauthorized access into a system or network, but it conceals other payload tools from been detected which goes ahead to gain unauthorized access and cause damage to the target. [9] [10]

2.2.8 Adware

This is a software that generates unwanted advertisements to users which in turn produces revenue for the developer. These advertisements sometimes come as pop-up or in an unlosable window which makes it very irritating, discomforting to users of the software or system. To prevent this adware there are programs developed to detect and remove them from a system or application. [9] [10]

2.3 Data Security Platforms

Organizations are continuously working steps ahead to protect their data from attackers and cybercriminals as data production continues to increase and the need to secure it. Moving away from the traditional system of data security organizations are adopting the data security platform. Data security platforms are new ways of protecting data of an organization that incorporates every aspect of data security in one platform. The data security platform as shown in Figure 2-2 covers the data discovery process, to data classification, data analysis, and data protection all in one platform. This single data management system helps organizations apply uniform policies across the entire spectrum of data management.

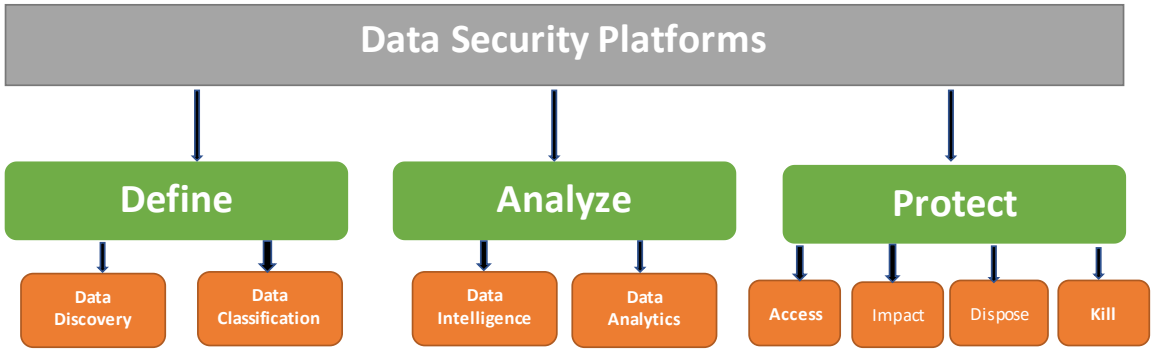

Figure 2-2 Data Security Platform

2.4 GDPR

GDPR is a regulatory guideline that requires all businesses that carry out dealings and transactions within the European Union (EU) and the European Economic Area (EEA) member countries to protect the personal information and privacy of EU and EEA citizens also regulate the exportation of the personal information outside the EU and EEA member states. [11] It is the European Union's new data protection law adopted by both the European Parliament and the European Council in April 2016 to replace the Data Protection Act which went into force in 1995 but has since become outdated as it did not account for many of the data concerns created by the internet in this digital age. The GDPR implementation as a law in Europe was given a two years preparation period to allow businesses and organizations that fall under the law to prepare, plan and implement the regulation. On the 25th of May 2018, the GDPR came into force and has since been an important requirement for all organizations doing business in Europe. [11]

The primary objective of the GDPR is to give individual citizens control over their personal information, have a uniform data protection guide for the EU and EEA member countries and to compel organizations operating within Europe or using the data of the citizens of Europe to have data protection principles which must be implemented to protect personal data of individuals in the EU and EEA. The GDPR requires companies operating within Europe to protect the personal information of the EU citizens.

2.5 ISO/IEC 27001

The international organization for standards and the international electrotechnical commission published the ISO/IEC 27001, an information security standard in 2013. ISO/IEC 27001 is a management system standard that requires management control of information security. [12] The ISO/IEC 27001 standard covers a wide range of requirement which organizations must comply with to become certified. They include; the scope of the standard, how the document is referenced, reuse of terms and definitions in ISO/IEC 27000, organizational context and stakeholders, information security leadership and high-level support policy, risk assessment, backup the information security system operations, evaluating the systems performance and corrective actions. [12] For an organization to become certified they must undergo an audit by an accredited certification body which must confirm that the organization meets all requirement according to the ISO/IEC 27001 standard. [12]

3 Overview of Software Security Testing

This chapter gives a detailed overview of software security testing. It explains and analyses various software security threats, common vulnerabilities according to Open Web Application Security Project (OWASP), the different Open Web Application Security Project (OWASP) projects different software testing technique and operating platforms and systems.

3.1 Introduction

Software security testing is the test carried out on a software, web application to find any loopholes and identify any weaknesses, vulnerabilities, threats, and the risk that could result in damages or losses to the software or web application. The process of software security testing involves detecting possible security threat to the system, measuring the degree of weakness and finding solutions that address the problems sometimes through coding [13].

3.2 Trends and Threats

With increasing internet connectivity in the global network also comes increasing the number, frequency, and sophistication of attacks of systems in the cyberspace. The trends of attack have continued to increase over the last few years with more devices getting connected to the internet with a crime relating to cyber-attack reaching £52 billion in 2007 [14]. Some common threats frequently experienced in the cyber world are discussed below;

- Computer crime
- Vulnerabilities
- Eavesdropping
- Keyloggers
- Exploits
- Backdoors
- Logic bombs
- Payloads
- Denial of service

3.2.1 Computer Crime

When a crime is committed involving a computer, a network system, internet or software system as a tool or a target, it is referred to as a computer crime or a cybercrime [15] [16]. Cybercrimes are therefore activities targeted against individuals or group of individuals with the objective to purposely do damage to the reputation of the target which could result in physical or mental harm, losses in financial terms which could directly or indirectly use available modern information technology tools. Cybercrimes like intercepting and revealing confidential information, copyright infringement, hacking and unwarranted mass surveillance have become a major concern in the cyber ecosystem. [15] [16]

Cybercrimes cover a whole range of activities some of which includes financial fraud crimes and cyberterrorism. Financial fraud crimes include all forms of activities like false representation, unauthorized processing and altering information to mislead the target.

Cyberterrorism includes all illegal cyber activities directed at advancing a political or social objective. Cyberextortion includes activities on websites, servers or network systems like denial of service by attackers with the aim of extorting money from the targets. [15] [16]

It was estimated that in the United States almost $1.5 billion was lost to online credit and debit card theft in 2012 [17].In 2014 a report showed that the yearly harm caused by cybercrimes was estimated to be close to $450billion [18]. A more recent study done Centre for Strategic and International Studies (CSIS), in partnership with McAfee in 2018 reported that close to $600 billion is lost in cybercrime annually in the United States alone. [19]

3.2.2 Vulnerability

Vulnerability is a loophole in a system which could be exploited by an attacker and result in unauthorized access, illegal activities and actual harm to a system. Vulnerability can be a security risk but not all vulnerability is a security risk for instance when an exploitable vulnerability is exploited and the affected part of the system is of no value to the entire system then it is not a security risk even though it is a vulnerability [20]. A vulnerability is any loophole or weakness of a system that can be exploited by an attacker to gain unauthorized access into a system or privileged control over a system for causing damage or loss to the owner and users of the target system. [21]

There are various forms of classifying vulnerability depending on the system they are related to. They include:

- Hardware vulnerability which covers vulnerability to humidity, dust, soiling and protected storage.
- Software vulnerability covers vulnerabilities from design errors and inadequate testing
- Network vulnerabilities cover insecure communications lines and network architecture flaws
- Personnel and organizational vulnerabilities cover poor security education and the absence of proper security audit and continuity plans. [22]

The vulnerability of a system or a network could be caused by various factors like the complexity of the system, using common and well-known codes or software, number of accessible ports and protocols, software bugs and unchecked inputs. Examples of common software vulnerabilities include code injection, email injection, HTTP header injection, HTTP response splitting, SQL injections, and format string attack. [22]

3.2.3 Eavesdropping

Eavesdropping is the illegal act of covertly listening to private communications of others without their permission and knowledge with the aim of using any obtained information against the target. Eavesdropping is a form of a network attack where the attacker tries to collect information from the network transmitted by computers and use the information for harmful purposes. It is most effective where encryption services are not available as part of security measures. [9] [10]

3.2.4 Keyloggers

This involves using software or hardware to secretly record users when they are using a computer system. There are two types of keyloggers which are the hardware-based keyloggers

and software-based keyloggers. The hardware-based keyloggers do not require software installation to become active as they come in hardware from within a computer system or network system. The software-based keyloggers are software built to function in target computer systems or networks. [23]

3.2.5 Exploits

Exploits are a collection of data, a software or a series of command that can infiltrate a loophole or weakness in a network system or software application for causing harm in the target system. These harms could come in the form of gaining unauthorized access to the system or performing a denial of service attack. Exploits can be classified into two based on how they relate to the target software or application, they are remote exploits and local exploits. A remote exploit attacks the security of a vulnerable network without any previous access to the network or application, while a local exploit must first gain access to the system through its loopholes before attacking the system by removing security barriers to all for actual attacks like denial of service. [22]

3.2.6 Backdoors

A backdoor is a secret route of bypassing typical verification or authentication of a system to access the system or application. A backdoor to be in a variety of forms which may include being in the form of a hidden program, code or part of an operating system. Trojan Horse is sometimes used by an attacker to create backdoors into a system. The trojan horse will appear as a verified software which when installed will create a backdoor into the system to gain access to confidential information. [24]

3.2.7 Logic Bombs

This type of threat is trigger code deliberately encoded into an application, a network system or application to trigger specific actions as soon as certain conditions are activated or experienced. These actions are mostly harmful, and the codes come in the form of viruses or malicious worms to trigger certain payload. The payload is usually unknown and unwanted by the target system or application.

3.2.8 Payloads

When a malicious malware is used to attack a system, network or application, the actual part of the malware that carries out the attack or malicious activity on the system is the payload. The payload, therefore, is the most important part of the virus or worm used for attacking a target system or application.

3.2.9 Denial of Service

Denial of service attack (DoS attack) is a common exploitation used by attackers to deny access to services or make the service unavailable to users by interrupting services of a host connected to the internet. The attacker uses these exploitations by jamming the target system or application with the redundant request in a quest to overload the system and thereby prevent actual requests from being fulfilled. When the attacker is sending the redundant request from

multiple sources so that it can't be stopped by blocking a single source of the attack it is called a distributed denial-of-service attack. [25]

3.3 Common Vulnerabilities

The Open Web Application Security Project (OWASP) is a global not for profit organization dedicated to improving, innovating and making available the best practices, standards, technology, and software for cybersecurity. [26] All their software and materials are open sourced and are available to the public. Vulnerabilities are loopholes or weakness an attacker uses to gain access into a system and exploit the system.

Figure 3-1 shows how an attacker can use various attack vectors to find a loophole, enter into the system, exploit the system and cause huge damage that could result in a negative impact in an organization. [26] .

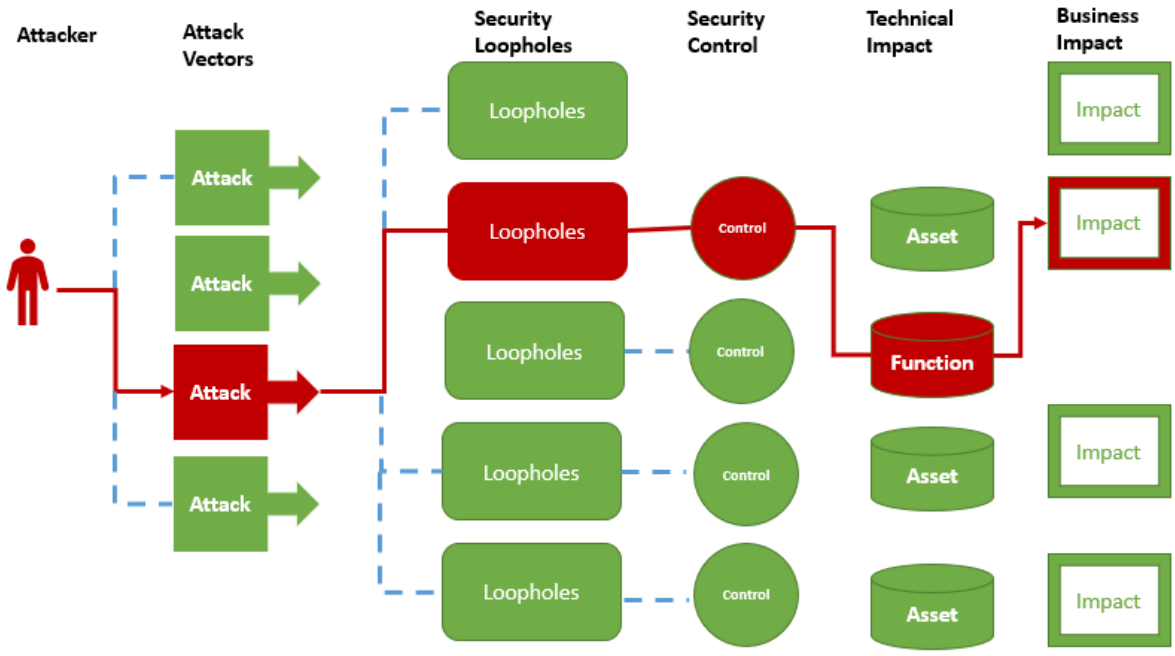

Figure 3-1 Diagram of how a system can be attacked

Figure 3-1 shows the path a system can be attacked which can sometimes be easily found and sometimes not easily found. Also, the impact of the attack could be very minimal or consequential as to result in damages. To determine the security risk each path has on a system there must be an evaluation of the all components involved which includes the attacker, attack vector, loopholes or weaknesses, security control, the technical and business impact. In 2017 OWASP released a list of top 10 vulnerabilities facing the cyber community to educate and enlighten security experts and developers on these common weaknesses, the likely impact they

can have on an organization that uses various network systems and platforms and how to address them. [26] . These top 10 vulnerabilities are discussed below, and a summary of these vulnerabilities is shown in Table 3-1

- Injections
- Broken Authentication and Session Management
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting (XSS)
- Insecure Deserialization
- Using Components with Known Vulnerabilities
- Insufficient Logging & Monitoring

Table 3-1 Summary of Top OWASP Vulnerabilities

Vulnerabilities	Exploitability	Weakness Prevalence	Weakness Detectability	Impact
Injections	Easy	Common	Easy	Severe
Broken Authentication and Session Management	Easy	common	Average	Severe
Sensitive Data Exposure	Average	Widespread	Average	Severe
XML External Entities (XXE)	Average	Common	Easy	Severe
Broken Access Control	Average	Common	Average	Severe
Security Misconfiguration	Easy	Widespread	Easy	Moderate
Cross-Site Scripting (XSS)	Easy	Widespread	Easy	Moderate
Insecure Deserialization	Difficult	Common	Average	Severe
Using Components with Known Vulnerabilities	Average	Widespread	Average	Moderate
Insufficient Logging & Monitoring	Average	Widespread	Difficult	Moderate

Table 3-1 gives a summary of the top 10 common vulnerabilities covering their exploitability, ease of detection, prevalence and the potential impact it could have on the system owners. Each vulnerability is discussed in more details below.

3.3.1 Injections

Injections occur when attackers use data as a tool of an attack like sending invalid data to a web application for doing something different from what the application is designed to do. Injection flaws in a web application are very common and some examples include; SQL, NoSQL, OS command, Object Relational Mapping (ORM), LDAP, and Expression Language (EL) or Object Graph Navigation Library (OGNL) injection. Injections typically result in loss of data, data corruption, denial of service and unauthorized disclosure. However, injection flaws are easy to discover using scanners. [26]

3.3.2 Broken Authentication

Broken authentication vulnerabilities are vulnerabilities that an attacker can use to try to gain partial or full access into a system illegally for causing harm to the target. Cyber attackers use the available millions of valid user names and password combinations to carry out the credential stuffing, they use common administrative account lists, automated brute force and dictionary attack tools to exploit broken authentication vulnerabilities. The broken authentication vulnerabilities are widespread but can be detected using scanners and prevented using a two-factor authentication method. [26]

3.3.3 Sensitive Data Exposure

Sensitive data exposure is a common vulnerability that results in exposing data that needs protection like passwords, credit card numbers, health information, and other personal information. Sensitive data exposure vulnerability can occur if personal data that fall under privacy laws like the GDPR are not protected, data is transmitted in clear text, sensitive data is stored in clear text, old and weak cryptographic algorithms are used by default and encryption is not enforced. [26]

3.3.4 XML External Entities (XXE)

XML External entities are vulnerabilities that occur in systems that process XML inputs. When a poorly configured XML performs a syntax analysis on processes that contains an XML input which could have references to external entities a loophole is created which can be exploited by attackers to gain access to classified information. [26]

3.3.5 Broken Access Control

Broken access control vulnerability occurs when a user can gain access beyond the given permission. The user gains unauthorized access to the system to cause harm to the target. This access can be gained through reconfiguring the URL, internal application state or the HTML page to go around the access control framework. Also, unauthorized access can be gained through the attacker acting as a user or admin without been logged in, metadata manipulation, unauthorized API access, and force browsing. [26] Broken access control vulnerability is a common vulnerability because this form of vulnerability can’t be detected automatically and therefore manual testing is the must be used to detect this vulnerability. [26]

Broken access vulnerability can be prevented in a system by ensuring that access to privileged information is denied by default, apply access control mechanism throughout the system and ensure that within the web roots there are no metadata files and backup files. [26]

3.3.6 Security Misconfiguration

Security misconfiguration vulnerability can occur when features that are not important like unnecessary ports, services, pages are enabled in a system, default admin accounts and passwords are not changed and not enabling latest security features during a system upgrade. Security misconfiguration vulnerability occurs in the web server, application server, database, and frameworks. [26] It can be detected using automated scanners. To prevent security misconfiguration vulnerability reduces features that are not important, do not install unused features in the system and create an automated process to confirm the effectiveness of configurations in every part of the system. [26]

3.3.7 Cross-Site Scripting (XSS)

Cross-site scripting vulnerability could occur in three different forms which are; reflected XSS, stored XSS, DOM XSS. Reflected XSS involves using a user input that is not validated to attack a target browser thereby inputting malicious links to redirect the target to the attacker-controlled page. Stored XSS is regarded as very high risk because it stores unverified user input that could be viewed subsequently by another user. [26] DOM XSS vulnerability could be in the form of frameworks and APIs that comes with attacker-controlled data. XSS is a prevalent vulnerability that can be detected using automated tools. XSS can be prevented by ensuring that the active browser content does not have unverified data. [26]

3.3.8 Insecure Deserialization

Insecure deserialization vulnerability occurs when an attacker sends unfriendly or altered objects to a target system and the system tries to deserialize the object. The attacker can exploit these vulnerabilities in two forms which are object and data structure attacks and data tampering attacks. To detect these vulnerabilities, human inputs and tools will be used together to validate the loopholes. To prevent this form of vulnerability all forms of serialized objects from unverified sources should not be accepted by the system. [26]

3.3.9 Using Components with Known Vulnerabilities

When using components with known vulnerabilities it’s important that information about these components are known such as the versions and they are up to date to avoid any form of related vulnerabilities. Furthermore, all related components used in the system must be sourced from verified sources. These forms of vulnerabilities are widespread and can be detected using scanners. [26]

3.3.10 Insufficient Logging & Monitoring

These forms of vulnerabilities most times form the foundation of attack for major incidents of attack of a system. When a system or application does not login important transactions like failed logins, does not monitor its APIs for suspicious activities and stores logs only locally it

can become vulnerable to insufficient logging & monitoring vulnerabilities. To detect these forms of vulnerabilities the logs should be examined after performing a penetration test to see if there was enough recording of all activities of the penetration test. [26]

3.4 OWASP Projects

OWASP Testing systems cover in details various projects that have produced valuable results in security testing.

3.4.1 Zed Attack Proxy (ZAP)

This is an open sourced penetrative tool that is used to find weaknesses automatically in a web application during the development and testing stage. It can serve as a proxy server thereby allowing the user to intercept and manipulate the traffic passing through it. Some built-in features include an automated scanner, forced browsing, fuzzer, and web socket support. [27]

3.4.2 Web Testing Environment (WTE)

Web Testing Environment (WTE) is an open-source platform that consists of selected security testing tools which include virtual machines which are designed to create a testing environment for security testing of various systems. [27]

3.4.3 OWTF

OWTF is designed to align the penetration testing process to OWASP testing guideline, to make penetration testing more efficient, innovative and more detailed. This helps provide a big picture in the penetration testing process and that the same time makes provision to investigate in details complex vulnerabilities within a system. [27]

3.4.4 Dependency Check

Dependency check is a tool of OWASP used to find known vulnerabilities of every aspect of a project. It is currently supported by Java, Net, Ruby, Node.js, and python. Dependency-check has an inbuilt command line interface, an ant task, and a Jenkins plugin. It also has analysers that check the project dependencies and gathers information about the dependencies. The gathered information is then used to identify the Common Platform Enumerators (CPE)of each dependency. For each CPE identified a listing of related Common Vulnerabilities & Exposure (CVE) will be reported. [27]

3.4.5 Security Shepherd

The security shepherd project is a security training platform for web and mobile application built to enhance and promote security awareness in the cybersecurity ecosystem. The project focuses on building the penetration testing skills of the users through exposure to security risk concepts, security challenges, common vulnerabilities and impact of those vulnerabilities can have on a system. [27]

3.4.6 DefectDojo

DefectDojo is an open source security application tool written in python for vulnerability management through streamlining the testing process by providing templates, generating a report and other self-service tools. The main objective of this tool is to reduce the time spent logging vulnerabilities. [27]

3.4.7 Juice Shop

This project is a web application that is open to be used for testing, demonstration, and training. It contains all the common vulnerabilities and many other security loopholes which can be found in real systems. The Juice Shop is a training platform where various trials, training, and practices can be carried out by penetration testers. [27]

3.4.8 Security Knowledge Framework

The Security Knowledge Framework is a high-level web-based application that uses the OWASP Application Security Verification Standard to serve as a guide for creating and securing software. The security knowledge framework also serves as a reference for security knowledge and as a tool for sourcing all security requirements during development of an application. [27]

3.4.9 Dependency Track

The Dependency Track Project is a platform that monitors third-party elements used in making applications for organizations. The platform combines various vulnerability databases and uses it to identify the possible weakness of a third-party component used to make an application. [27]

3.5 Two-factor Authentication Mechanism

The rise of cybercrime and increase in illegal and authorized access to user information and credential has resulted in the need for a multi-step or multi-level authentication to access a user's information in some software and application. The idea behind this mechanism is the discovery that just using a password to gain access to a system or application is not sufficiently secured as attackers can gain access to this password through various attack tools. But the two-factor authentication mechanism is a security measure used by applications or systems to further reduce the chances of illegal or unauthorized access to a user's information. [28] The two-factor authentication mechanism is a form of multi-factor authentication where a user will have to confirm its identity using two different factors like something they know and something they have. Example of the two-factor authentication mechanism is requiring in addition to a password a user will also need a one-time password produced by an authenticator like a token or a phone to gain access to a system or application. This method of securing user information is known to reduce the likelihood of attacks like identity theft and other online fraud because of the two-step verification required. The one step of password information will not be enough to gain access to a user's information when a two-step mechanism is in place. However, some two-factor authentication mechanism can still be vulnerable to man-in-the-middle attacks, man-in-the-browser attacks, and phishing. [29]

3.6 Software Security Platforms and Operating System

Most security software works in the generally known operating systems such as Linux, macOS, Windows, Solaris, and Open Solaris. To run this security software on any operating system two components must be present. They include a virtual machine and software security operating system.

3.6.1 Virtual Machine

The virtual machine is a virtual platform that can create a virtual environment for developers and testers. Table 3-2 shows some generally used virtual machines by Software security testers. [30]

Table 3-2 List of commonly used Virtual machines [31]

Virtual Machine	Licencing	Operating Systems
Oracle VM VirtualBox	Free	Windows, Linux, Mac OS, and Chrome OS
ConEmu-Maximus5	Free	Windows and Chrome OS
Parallels Desktop	Paid	Mac OS
VMware Fusion	Paid	Mac OS only
VMware Workstation Pro	Paid	Windows and Linux
Quick Emulator, QEMU	Free	Web Browser
Portable-VirtualBox	Free	Windows and Chrome OS
Docker	Free	Windows, Linux, Mac OS, and Chrome OS
KVM (Kernel-based Virtual Machine)	Free	Linux
Xen	Free	Web Browser

3.6.2 Software Security Operating System

Software security operating systems are security-based operating systems that are equipped with security testing tools which are using in carrying out security testing operations, ethical hacking activities like network scanning, mapping, and exploitation. This operating system creates a perfect environment where proper and detailed security testing can be carried out. Table 3-3 shows a list of the ten most commonly used software security operating systems.

This security software is mostly Linux based, all open sourced and available for free download on the internet. [30]

Table 3-3 lists of commonly used software security operating system. [30]

Software	Operating System
Kali Linux	Debian-based OS
Parrot Security OS	Debian-based OS
BackBox Linux	Ubuntu-based OS
Samurai Web Testing Framework	
Pentoo Linux	
DEFT Linux	Ubuntu-based OS
Caine	Ubuntu-based OS
Network Security Toolkit	Fedora-based
Bugtraq	Debian, Ubuntu, and OpenSuSe OS
BlackArch	

4 Software Security Testing

This chapter covers the details of the actual security testing to be performed on the Edemso software. it gives extensive literature studies on the security testing approach and various stages of the security testing of Edemso.

4.1 Introduction

The internet of things industrial revolution is resulting in an increasing number of complex and more complicated software and systems. The automation of many human activities and an increasing number of computers connecting to the internet are increasing the vulnerabilities and chances of an attack in the system. [32] [33] [34]Vulnerability of a system is a loophole in the system which could be a flaw from the design or implementation process which allows attackers to get unauthorized access into a system and cause harm to the system. Hackers and attackers could exploit vulnerabilities of a system to get confidential information, cause harm and steal proprietary information. The security of a system can be strengthened through the identification of all potential vulnerabilities and removing them. This is made possible by an important security process used in the cyber community called Vulnerability Assessment and Penetration Testing. [34]

4.2 Vulnerability Assessment

Vulnerability Assessment is a process of scanning a system or software, identifying, enumerating, quantifying and reporting potential vulnerabilities of the system or software. These vulnerabilities are possible loopholes which an attacker can exploit to gain unauthorized access into a system or software. Some possible vulnerabilities include control vulnerability, boundary condition vulnerability, input validation vulnerability, authentication vulnerabilities, configuration weakness Vulnerabilities, and exception Handling Vulnerabilities, etc. [32] [33] [34]Figure 4-1 shows the vulnerability Assessment Cycle of a system. Vulnerability Assessment is the first phase of penetration testing.

Previous research works have been done on Vulnerability Assessment some of which includes works by Ivan Krsul, [35]Steven E Noel et al, [36] Stefan Kals et al, [37] Sushil Jajodia & Steven Noel, [38] and Christopher Kruegel [39]. Ivan Krsul worked on the visualization and detection of computer vulnerabilities [35]. Stefan Kals et al did extensive work on vulnerability scanner tools and developed a scanner tool called SecuBat, [36]. Steven E Noel et al did extensive work on the impact the interrelationship between multiple vulnerabilities and exploits in a single network. [34] [37] Sushil Jajodia et al worked on Topological Vulnerability Analysis approach which focused on the and potential attack path into a computer network and vulnerability interdependencies [38] Christopher Kruegel et al did a detailed study on execution after redirect vulnerabilities. [34] [39]

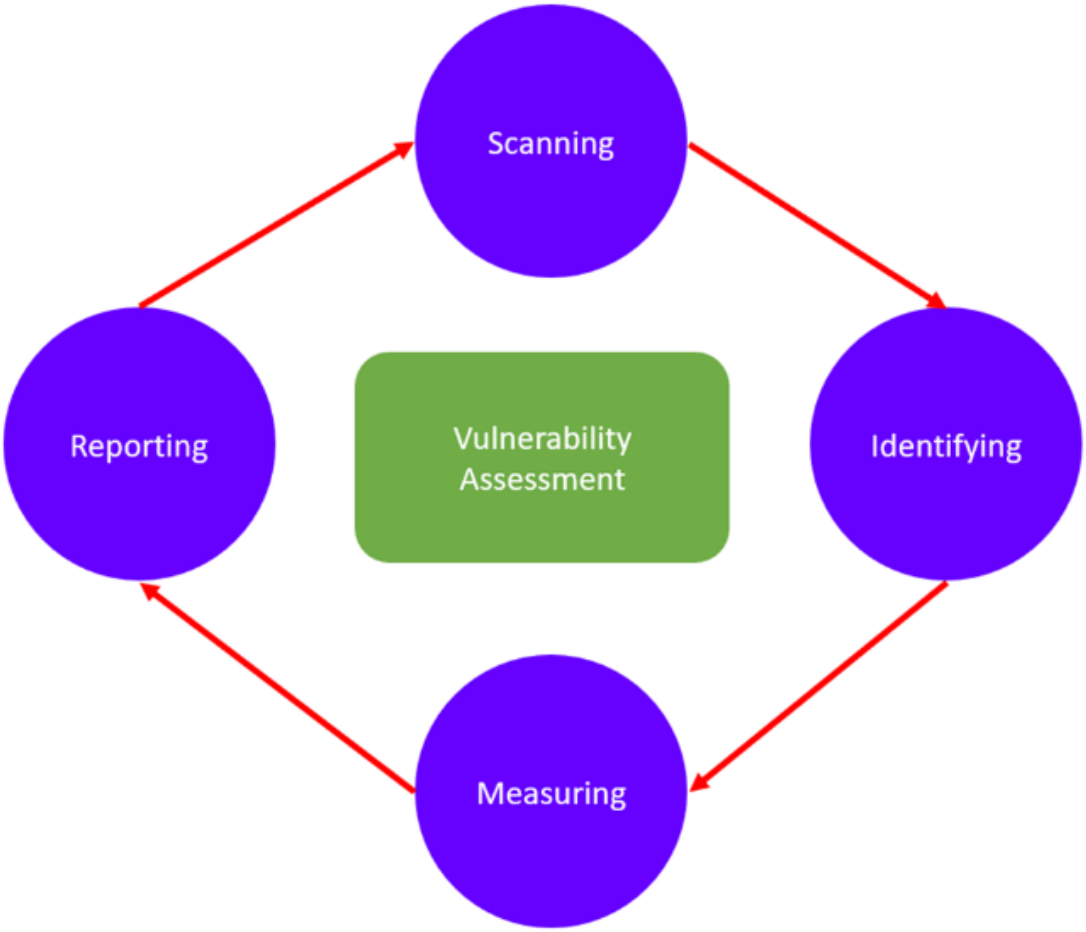

Figure 4-1 Vulnerability Assessment Cycle [32]

Vulnerability assessment is divided into three stages which are

- Vulnerability analysis
- Web application analysis
- Database analysis

4.2.1 Vulnerability Analysis

Vulnerability analysis stage covers the scanning and identifying weakness and loopholes in the target network, target host server and other wireless networks involving the target.in this phase the following scan will be performed; [32] [33] [34]

- The network-based scan covers wired and wireless networks. It focuses on identifying any loopholes or weakness that can be identified in the wired or wireless network of the target.
- The host-based scan which covers the scanning and discovering of vulnerabilities in the host servers, ports, and another access point.

4.2.2 Web Application Analysis

Web application analysis phase is the phase where the website of the target is scanned to identify any known loopholes, weakness, and errors in the software configurations.

4.2.3 Database Analysis

Database analysis phase is focused on scanning and identifying the loopholes, weakness in a database to prevent a malicious attack like SQL injection attacks.

4.3 Penetration Testing

Penetration testing is an extension of the vulnerability assessment, it involves exploiting the vulnerabilities that have been scanned, identified, measured and reported during the vulnerability assessment. . [32] [33] [34]

5 Software Test Plan

This chapter gives details on the different test plans, environments, and methods that can be used for testing the Edemso Software.

5.1 Overview of Software

The project will be carrying out detailed security testing on a cloud-based software called Edemso used as a management tool. It covers two connected platforms which will undergo various security testing. These two platforms are;

1. https://www.edemso.com/login
2. www.ceviasolutions.com

Edemso is a web-based software which also has an application that can be installed on a desktop. The desktop application makes it possible to be able to download a document, edit the document and upload the edited document back to the Edemso platform.

5.2 Software Test Methods

In software development, the software goes through three major systems development life cycle (SDLC) environment which are the development environment, the test environment, and the production environment. In each environment, a security test can be performed depending on the focus of the developer. Figure 5-1 shows the software development life cycle environment and how a software moves from one environment to another in a cycle.

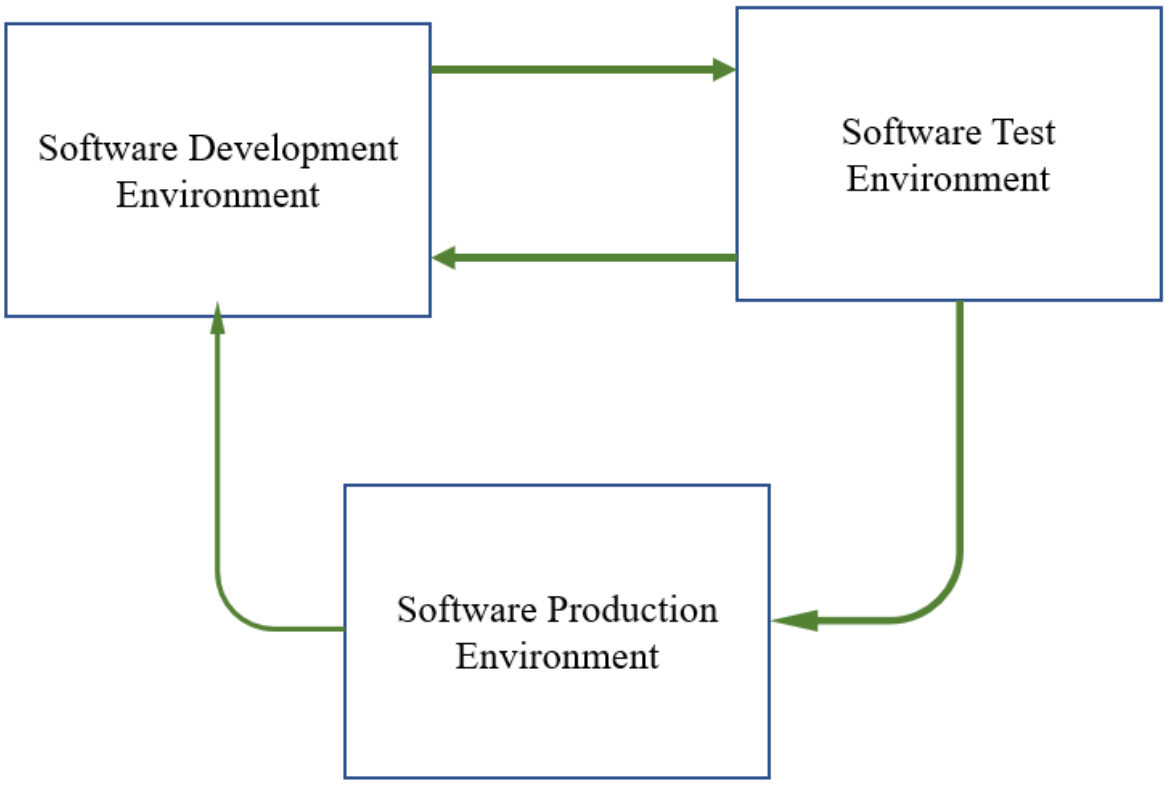

Figure 5-1 Software Development Life Cycle Environment

The software test plan can be structured in different test cases and environments which in the end is expected to give insight into the security state of the platform being tested. These different test cases and the environment is summed up as software test methods. In this project three different software test methods will be examined and documented;

1. Direct test method
2. Test environment method
3. Comparative test method

5.2.1 Direct Test Method

In this method of software testing the actual platform to be tested undergoes series of security tests which involves actual attack and hacking of the system in real time. In this method, the security tester will try to break into the security of the cloud network which is hosting the platform and thereafter breaks into the platform database and tries to access unauthorized information from the platform. From Figure 5-2 the first security barrier to be tested is the cloud network host which is the provided by Amazon web services and the second security barrier to be tested is the database which in this case is the PostgreSQL. As seen in Figure 5-2 the security barrier of the host cloud network is high because the Amazon web services protect its cloud with a strong web application firewall. Therefore, to successfully break into the cloud network the intending attacker will require very sophisticated skills.

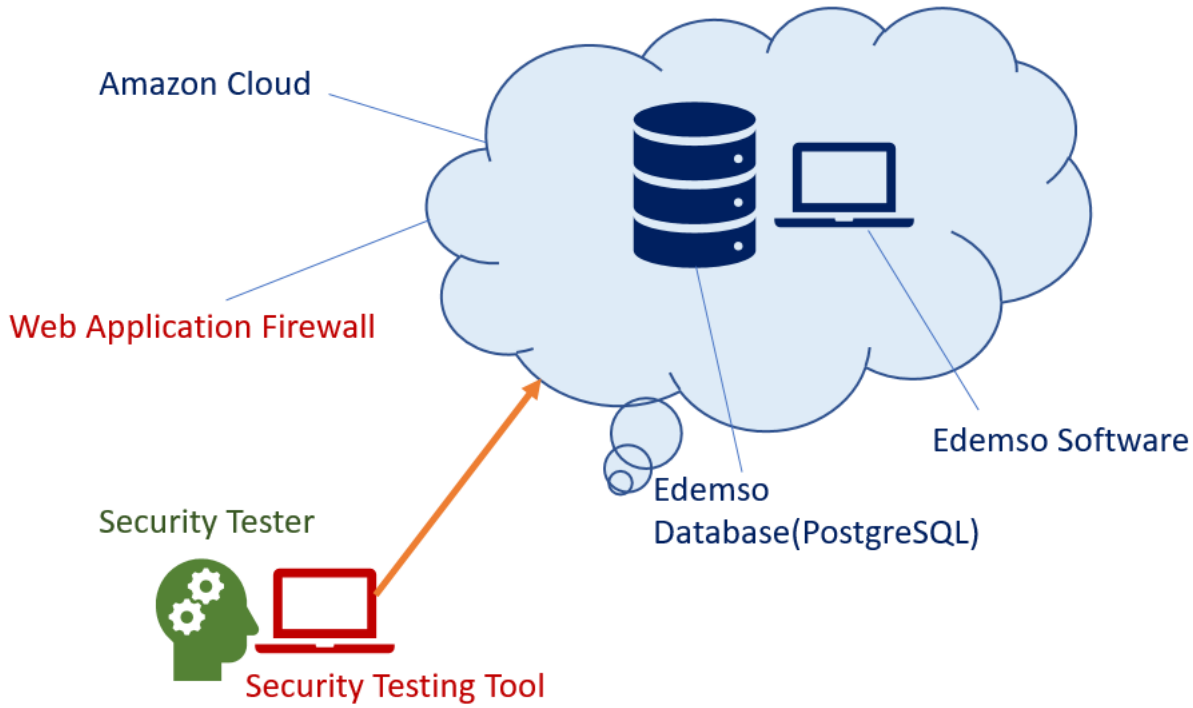

Figure 5-2 Illustration of Direct Test in Real Time

In the direct test method, the test environment is the actual environment the platform is operating on and the testing will include finding vulnerabilities of the cloud network host,

finding vulnerabilities of the database, finding vulnerabilities of the web application and thereafter exploiting those vulnerabilities to gain access the system. In this project, the direct test method is used for testing the Edemso software using the software testing tools detailed in chapter 6.

5.2.2 Testing Environment Method

In this method, a test environment is created remotely by setting up a local server in a local computer and installing the Edemso software and database into the local server. The security test is carried out using the software security testing tools detailed in chapter 6. In this method the sophisticated security barrier of the cloud host network is not available, therefore giving the opportunity to really test the security state of the Edemso platform. From Figure 5-3 it can be seen that the local host server has a no firewall protection thereby allowing for easy testing and attack of the software database and the Edemso web application to gain unauthorized access to information. The security test will be mainly to find database vulnerabilities, the web application vulnerabilities and thereafter find ways to exploit those vulnerabilities.

Figure 5-3 Illustration of Testing Using Local Server

This testing method will not be used in this project but is recommended for future studies and work in the security testing research works.

5.2.3 Comparative Test Method

In this method, the Edemso platform which is the target platform and another web application with known vulnerabilities are tested with security testing tools detailed in chapter 6. The main objective of this testing method is to verify that the security testing tools used in the project work. Figure 5-4 shows the illustration of how comparative test is carried out for both the Edemso platform and the demo web application using the same security testing tools. For both platforms, the security tests will include finding the vulnerabilities of the network, the database, and the web application. Thereafter those vulnerabilities are exploited to gain access into both systems and obtain unauthorized information.

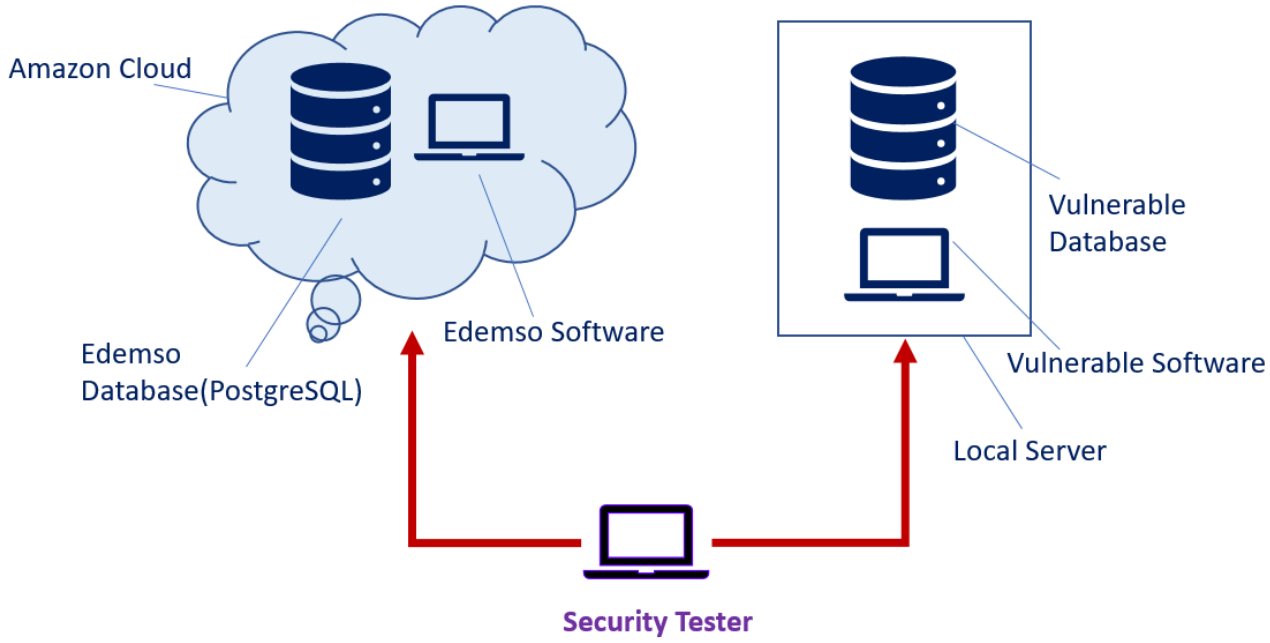

Figure 5-4 Comparative security testing

5.2.4 Summary of the Software Test Methods

Table 5-1 shows the summary of the software security test methods and those that will be used for different phases of the testing. In this project, the direct testing method will be mostly used to test the Edemso platform though there is a limitation to this testing method which is that the testing could result in real damage to the system. The comparative test method will be used for the database analysis phase of the testing as it helps to measure and analyses the security state of the database of the system.

Table 5-1 Summary of Software Security Test Method

Test Method Summary			
Security Testing Phase	**Test Environment Method**	**Comparative Test method**	**Direct Test method**
Test Strategy	Deploy Demo software in a local server & run a security test	Run similar security test on platforms of target vs one known vulnerabilities	Run security test directly on the target platform in a live environment
Limitations	Project Duration could limit implementation	Requires using platforms with similar structures & loopholes	Can cause real damage on the actual platform
Environment	Test environment	Production environment	Production environment

Information Gathering	Method not used	To be performed	To be Performed
Scanning & Discovery	Method not used	To be performed	To be Performed
Vulnerability Analysis	Method not used	Method not used	To be performed
Web Application Analysis	Method not used	Method not used	To be performed
Database Analysis	Method not used	To be performed	Method not used
Execution Phase	Method not used	Method not used	To be performed

6 Penetration Testing

This chapter covers details of the different security tools used to perform the security test on the Edemso software, operating platform, and the security testing phases carried out in this project.

6.1 Introduction

To carry out a comprehensive security testing on the Edemso platform some important tools must be available, they include a virtual platform, an operating system, and the various security testing tools.

6.1.1 Oracle VM Virtual Box

Oracle VM Virtual Box is a free and open source application used for general purpose and cross-platform virtualization, developed by Oracle Corporation for computers. It works with various operating systems like Linux, MacOS, windows, and others. It increases the capabilities of an existing computer allowing it to run multiple operating systems at the same time. For this project, the Oracle VM Virtual Box is used to run the Kali Linux which is the security platform that will be used in the project. [40]

6.1.2 Kali Linux

Kali Linux is an open sourced and free Debian-derived Linux distribution used for penetration testing. It has over 600 preinstalled penetration testing programs. Kali Linux can be used natively by installing it into a computer using a live CD or USB or by using a virtual machine. To install Kali Linux into a computer, it must have a minimum of the 1G hard disk, 512MB RAM for i386 and AMD64 architectures. In this project, Kali Linus will be installed using a virtual machine. Kali Linux is made up over 600 preinstalled testing tools which are divided based on the function of the tools. Classes of penetration tools are information gathering, vulnerability analysis, exploitation tools, wireless attack, forensics tools, web applications, and stress testing. Table 6-1 shows a classification of the penetration tools available in Kali Linux. [41]

Table 6-1 Classification of the penetration tools available in Kali Linux

Classes of Penetration Tools	Examples
Information Gathering	*Nmap, Wireshark, Fierce,*
Exploitation Tools	*Armitage, Backdoor Factory, Metasploit Framework*
Wireless Attack	*Airbase-ng, Aircrack-ng, Airdecap-ng and Airdecloak-ng, Aireplay-ng,*

Forensics Tools	*Binwalk, bulk extractor, Capstone, chntpw, Cuckoo,*
Stress Testing	*DHCPig, FunkLoad, iaxflood,*
Web Applications	*BlindElephant, Burp Suite, CutyCapt, DAVTest,*
Sniffing & Spoofing	*Wireshark, Wifi Honey, Zaproxy, Bettercap, Burp Suite,*
Password Attacks	*Maskprocessor, multiforcer, Ncrack, oclgausscrack,*
Maintaining Access	*CryptCat, Cymothoa, dbd, dns2tcp, HTTPTunnel,*
Reverse Engineering	*Apktool, dex2jar, diStorm3, edb-debugger*
Hardware Hacking	*android-SDK, apktool, Arduino*
Reporting Tools	*CaseFile, cherry tree, CutyCapt,*

6.2 Penetration Testing Tools

Penetration testing tools will be used as security testing tools in this project. The penetration testing tools used in this project are all preinstalled in the security operating system used called Kali Linux. Penetration testing tools are very effective in that they ensure that the test is faster, give more accurate and precise results, carry out advanced analysis, gather bulk information and automates most of the test processes. Some penetration tools that will be used in this project are:

6.2.1 Nmap

Nmap which stands for Network Mapper is an open source, free scanning tool used for network discovery on a computer by sending packets and analyzing the response, network scanning and security audit. It is used for network inventory, managing service upgrade schedules and monitoring hosts available on a network. Nmap is used for port scanning, identifying which ports are opened or closed in the host target and detects the version of an application and application name in any network service. [41]

6.2.2 Wireshark

Wireshark is a free and open sourced penetration testing tool used as a protocol analyzer. It is used for analysis, troubleshooting, software and communication protocol development. Wireshark is a penetration tool application that captures data and can analyze the encapsulation

of different network protocols. It is one of the preinstalled penetration tools in the Kali Linux operating system. [41]

6.2.3 Metasploit Framework

Metasploit Framework is an open source penetration test platform that can be used to discover, exploit and validate vulnerabilities. It has a combination of infrastructure, content and testing tools that are used to carry out comprehensive security auditing.it also has anti-forensic and evasive tools built into it. It is one of the preinstalled penetration tools in the Kali Linux operating system. [41]

6.2.4 OWASP ZAP Web Application Security Scanners.

OWASP ZAP web application security scanner is an open source web application scanner designed to discover vulnerabilities in web applications. It can function as a proxy server where it can be used to manipulate traffic passing through it including traffic using https. It is one of the preinstalled penetration tools in the Kali Linux operating system. [41]

6.2.5 Burp Suite

Burp Suite is a graphical penetration testing tool used for testing web application security. The burp suite is made up important tools like HTTP Proxy which intercept traffic, a scanner which performs vulnerability scan of a web application, intruder which performs an automated attack on web applications, spider for crawling activities, a repeater for manual testing of web application and others. [41]

6.2.6 John the Ripper Password Cracker

John the Ripper is a free penetration testing tool for cracking passwords. It is used for testing and breaking passwords. It's a combination of some password crackers like autodetects password hash types. It is one of the preinstalled penetration tools in the Kali Linux operating system. [41]

6.3 Penetration Testing Phases

In this project, the penetration testing will be performed in phases. For each phase of the penetration test, there will be different classes of penetration tools to be used. The penetration testing phases and the classes of penetration testing tools used for each phase are outlined below

6.3.1 Planning and Reconnaissance Phase

The penetrative tools used for this phase are the gathering information tools. In this phase, all information about the target platform to be tested is gathered using information-gathering tools. For this project, the information gathering tool to be used includes the red hawk which only works in Kali Linux, who is lookup, Geo-IP lookup, grab banners, subnet calculator lookup, subdomain scanner lookup reverse IP lookup and CMS detection

6.3.2 Scanning and Discovery Phase

The penetrative tools used in this phase are vulnerability analysis tools. In this phase, the entire system is scanned to identify possible loopholes and vulnerabilities. Also, the system of attack is mapped out to know the vulnerabilities to exploit. The penetrative tools used in this project for the scanning and discovering phase is Nmap.

Scanning and discovery phase is divided into the following;

- Vulnerability analysis
- Web application analysis
- Database analysis

6.3.3 Execution Phase

In this phase, the actual attack is carried out. This is also called the exploitation phase. This phase is divided into the following

- Password attacks
- Exploitation attacks

7 Test Result

This chapter documents the results of the software security test carried out in the project. In this chapter, the results of the security test for each phase is documented and analyzed using charts and tables.

7.1 Penetration Test Result

Each phase of the penetration test is performed using one of the three different test methods detailed in 5.2. The penetration test results cover the phases listed below;

- Planning and reconnaissance phase
- Scanning and discovery phase
- Execution phase

7.2 Planning and Reconnaissance Phase

In the planning and reconnaissance phase, information gathering tools like Dmitry (Deep magic Information Gathering Tool) and Sparta are used to gather all information about the target platforms. Figure 7-1 shows the target platforms of the Edemso software to be tested which are;

- Platform 1- **www.edemso.com/login**
- Platform 2- **www.ceviasolutions.com**

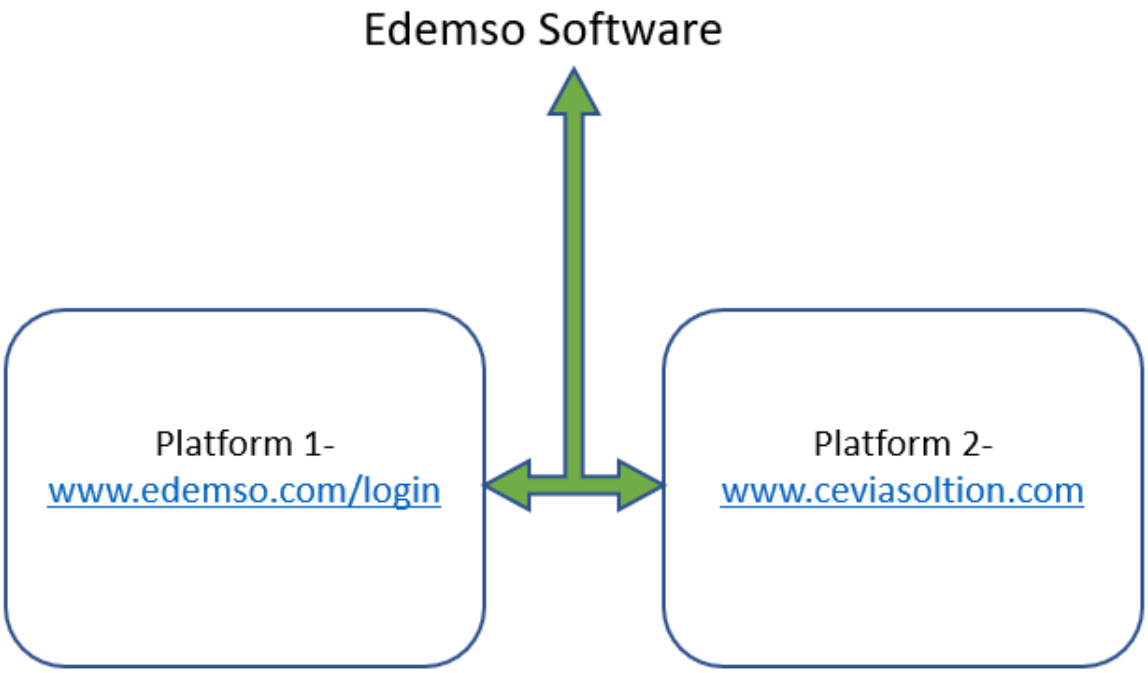

Figure 7-1 Platforms of Edemso Software

Table 7-1 shows a summary of the information gathered on the two platforms to be tested. The information collected gives more details about the platforms and determines the methods and tools to be used in the subsequent phase of the testing process.

Table 7-1 Information Gathered on Target Platforms

Items	Platform 1	Platform 2
Host IP	54.93.48.11	143.204.47.52
Domain Name	EDEMSO.COM	CEVIASOLUTIONS.COM
Range of IP	54.39.0.0 - 56.255.255.255	143.197.0.0 - 143.204.255.255
Country	EU	EU
Registry Domain ID	2256945313_DOMAIN_COM-VRSN	2014467842_DOMAIN_COM-VRSN
Name Server	NS1.HYP.NET, NS2.HYP.NET, NS3.HYP.NET	NS1.HYP.NET, NS2.HYP.NET, NS3.HYP.NET
Registrar URL	http://www.domainnameshop.com	http://www.domainnameshop.com
Creation Date	2018-04-25T20:55:13Z	2016-03-21T23:08:40Z
Registry Expiry Date	2020-04-25T20:55:13Z	2020-03-21T23:08:40Z
DNS Lookup	SOA 13 2 3600 20190531185352 20190501185352 39664	SOA 13 2 3600 20190524015306 20190424015306 47212
Sub Net Calculator Address	2a01:5b40:0:248::52	2a01:5b40:0:248::52
Sub Net Calculator Network	2a01:5b40:0:248::52 / 128	2a01:5b40:0:248::52 / 128
MX Lookup IP	194.63.252.29	104.47.13.36

MX Lookup HOSTNAME	mx09.domeneshop.no	mail-he1eur040036.inbound.protection.outlook.com

For the three different test methods discussed in 5.2, this phase of the testing is common to all and therefore must be performed in the three different methods.

7.3 Scanning and Discovery Phase

After gathering as much information about the test platforms the next phase is the scanning and discovery phase. In this phase, the target platforms are scanned extensively to discover all possible vulnerabilities, loopholes, and weakness. The scanning and discovery phase are executed in three stages which are;

- Vulnerability analysis
- Web application analysis
- Database analysis

7.3.1 Vulnerability analysis

The vulnerability analysis is performed with two penetration testing tools which are Nmap and Golismero. The analysis is performed in two steps which are;

- General scanning of both platforms using Nmap
- Vulnerability scanning of both platforms using Nmap and Golismero.

General Scanning of Both Platforms Using Nmap

Nmap is used to extensively scan the network to know the number of open and closed ports. Table 7-2 show the result summary of the Nmap scan for the target platforms. It can be seen from Table 7-2 that of the 1000 ports scanned for both platforms, platform 1 has 4 open ports and 996 closed ports while platform 2 has 2 open ports and 998 closed ports. Also, Table 7-2 shows the traceroute for both platforms goes through one path before it gets to the target platform host IP.

Table 7-2 Nmap Scan Results of Target Platforms

Ports	Platform 1	Platform 2
Scanned Ports	1000	1000
Closed Ports	996	998
Open Ports	4	2
TRACEROUTE (using port 80/tcp) **HOP RTT ADDRESS**	a- 2.81 ms 10.0.2.2 b-2.86ms ec2-54-93-48-11.eucentral-1.compute.amazonaws.com (54.93.48.11)	a-2.80ms-10.0.2.2 b-2.85ms server-143-204-47-80.osl50.r.cloudfront.net (143.204.47.80)

Details about the open ports for both platforms are summarized in Table 7-3 which shows the service of each port and the version of each port for both platforms. From Table 7-3 platform 1 has four open ports which are; port 22tcp (transmission control protocol) running ssh (secure shell) service, port 80tcp which is running an HTTP(hypertext transfer protocol) service, port 443tcp which runs SSL(secure socket layer) and HTTP services, and port 8843tcp which runs SSL and https(hypertext transfer protocol secure) services. Platform 2 has 2 open ports which are port 80tcp which is running an HTTP service and port 443tcp which runs SSL and HTTP services.

Table 7-3 Nmap Scan Results for Open Ports of Target Platforms

Platform 1	**Open Ports**	**Service**	**Version**
	22/tcp	ssh	OpenSSH 7.2p2 Ubuntu
	80/tcp	http	nginx 1.10.3 (Ubuntu)
	443/tcp	ssl/http	nginx 1.10.3 (Ubuntu)
	8443/tcp	ssl/https-alt	Linux
Platform 2	**Open Ports**	**Service**	**Version**
	80/tcp	HTTP	Amazon CloudFront
	443/tcp	ssl/http	Amazon CloudFront

Vulnerability Scanning of Both Platform Using Nmap & Golismero

With the results obtained in the general scanning of both platforms which are summarized in Table 7-2 and Table 7-3, scanning for loopholes and vulnerabilities is performed using two penetration testing tools called Nmap and Golismero. The result summary for vulnerability scanning is shown in Figure 7-2 which shows that Nmap found 7 vulnerabilities in platform 1 and zero vulnerability in platform 2. Also, from Figure 7-2 Golismero found 1 vulnerability each in both platform 1 and platform 2.

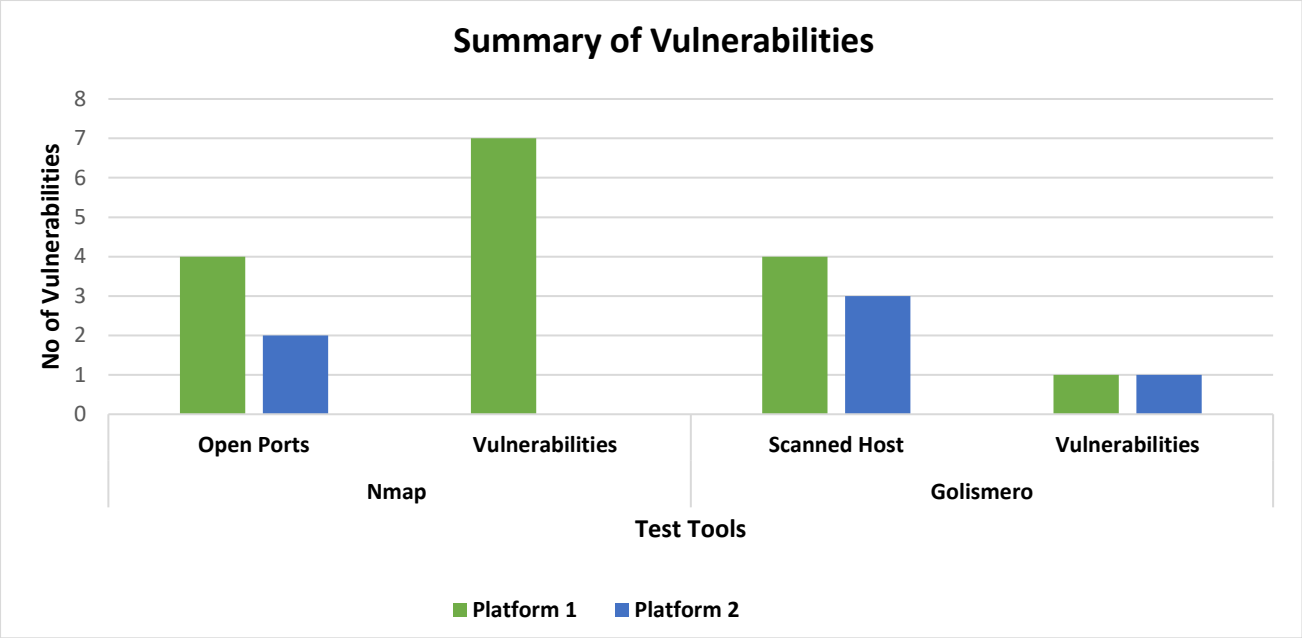

Figure 7-2 Summary of Vulnerabilities for Target Platforms

Summary of Discovered Vulnerabilities

The discovered vulnerabilities are further analyzed in

Table 7-4 for both target platforms which shows that both platform 1 and 2 each has one low-risk probability vulnerability. Also, it can be seen in Table 7-4 that platform 1 has 7 critical risk level vulnerability and platform 2 has zero critical risk vulnerability. Both platform 1 and 2 have zero severe risk vulnerability.

Table 7-4 Risk Analysis of Discovered Vulnerabilities

Risk Analysis	Risk level		
Target Platform	**Low-Risk Probability**	**Critical/Medium Risk**	**Severe/High Risk**
Platform 1	1	7	0
Platform 2	1	0	0

The 7 vulnerabilities that are under the critical risk level were all found in platform 1 which are further analyzed below;

1. CVE-2016-6515.This is a vulnerability which can be exploited because the authentication password function does not limit the length of a password which can allow an attacker to cause a denial of service using long strings of password

2. CVE-2015-8325. This is a vulnerability occurs because of the when the Use Login features are enabled, and PAM is configured to be readable. This vulnerability allows users to gain privileges by triggering a designed environment for the login program.
3. CVE-2018-15473. This is a vulnerability that allows invalid authentication to be bailed out immediately, therefore, leading to user enumeration vulnerability
4. CVE-2017-15906. This vulnerability allows attackers to create zero-length files because OpenSSH does not correctly prevent writing operation in a read-only state.
5. CVE-2018-15919. This vulnerability allows user behavior to be observable from a remote location, therefore, allowing attackers to detect the existence of users in a target network.
6. CVE-2016-10708. This vulnerability allows an attacker to cause a denial of service using an out of sequence new key messages.
7. CVE-2016-6210. This vulnerability allows an attacker to carry out password hashing on a network.

For the three different test methods discussed in 5.2, only the direct test method is applicable in this phase of the testing.

7.3.2 Web Application Analysis

The web application analysis the second stage of the scanning and discovery phase where the web application of both platforms is extensively scanned using two penetration testing tools called OWASP ZAP tool and Nikto tool. Both testing tools are used separately to scan the target platform 1 and 2 which gives a result summary shown in Figure 7-3. From the Figure 7-3 using the OWASP ZAP tool 9 vulnerabilities were found in target platform 1 and 18 vulnerabilities were found in target platform 2. Using the Nikto tool 3 vulnerabilities were found in target platform 1 and 6 vulnerabilities were found in platform 2. Both penetration testing tools found similar vulnerabilities in platform 1 and 2 with the OWASP ZAP tool finding more vulnerabilities than the Nikto tool. These extra vulnerabilities found by the OWASP ZAP tool fall under the low probability risk vulnerabilities.

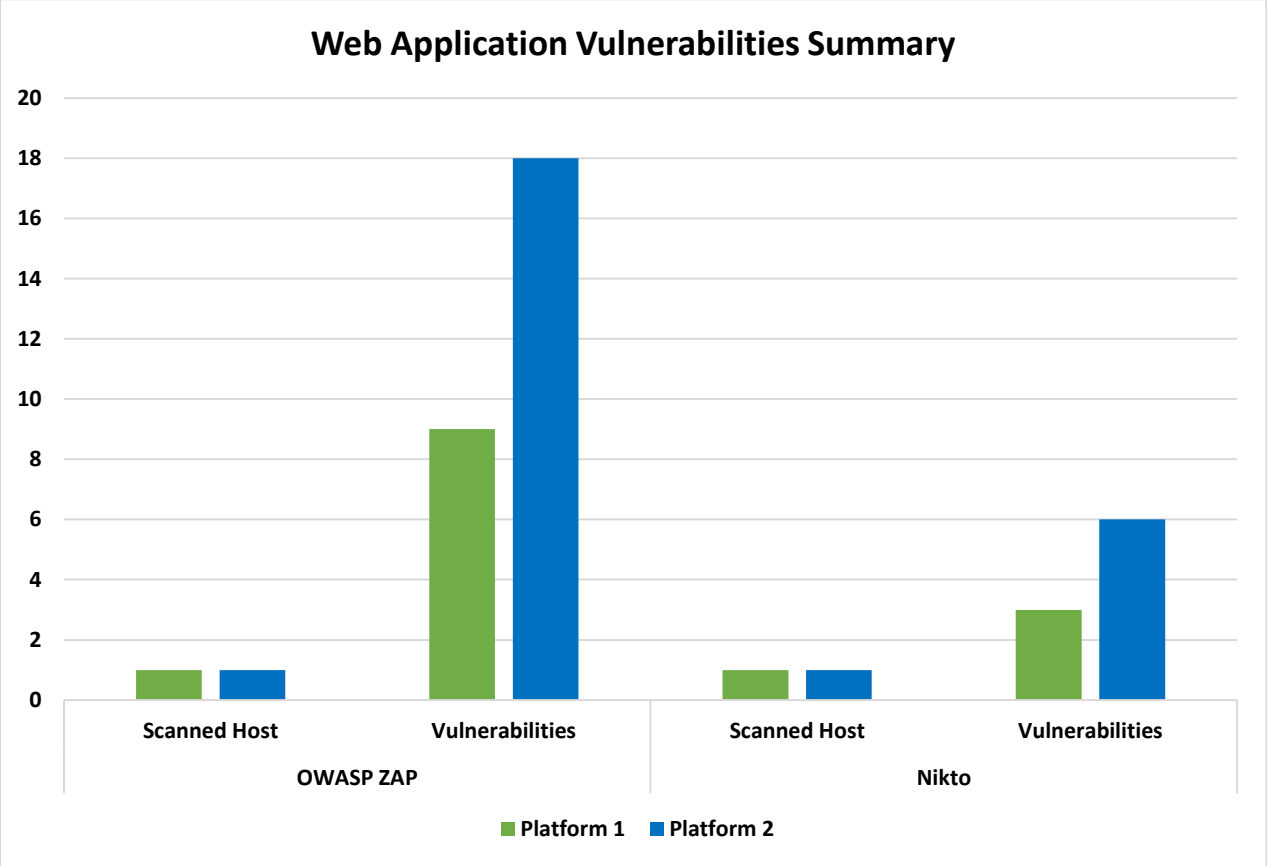

Figure 7-3 Web Application Vulnerabilities Summary

Summary of Discovered Vulnerabilities

The discovered web application vulnerabilities are further analyzed in Table 7-5 for both target platforms 1 and 2. Table 7-5 shows that platform 1 and 2 have 7 and 16 low-risk probability vulnerabilities. Also, Platform 1 and 2 each have 2 critical risk level vulnerability and both platforms 1 and 2 have zero severe risk vulnerability.

Table 7-5 Summary of Risk Analysis of Web Application Vulnerabilities

Risk Analysis	Risk level		
Target Platform	**Low-Risk Probability**	**Critical/Medium Risk**	**Severe/High Risk**
Platform 1	7	2	0
Platform 2	16	2	0

Target platform 1 and platform 2 each have two critical vulnerabilities which are similar, both of which are further analyzed below;

- The X-frame-Option header in platform 1 and 2 are absent which serves as a defense against clickjacking attack

- The cross-site scripting (XSS)-Protection header for platform 1 and 2 are not properly defined which makes it difficult to defend the platforms against some forms of cross-site scripting.

Some low-risk probability vulnerabilities are analyzed below;

- The anti-sniffing header is absent on both platform 1 and 2 which can allow for sniffing with an older version of internet explorer or google chrome.
- Improper setting of the cache-control and pragma HTTP in both platform 1 and 2.
- Third-party JavaScript present in both platform 1 and platform 2.

For the three different test methods discussed in 5.2, only the direct test method is applicable in this phase of the testing.

7.3.3 Database Analysis

The database analysis is the third stage of the scanning and discovery stage where the database of both target platform 1 and 2 undergo security testing through SQL injections to find vulnerabilities and exploit them to gain access to the database. The target platform 1 and 2 both use the PostgreSQL database.

In this stage of the testing, the comparative testing method which is one of the three security testing method discussed in 5.2 is used as the testing method. A web application with known vulnerabilities using a MySQL database is tested alone side the target platform 1 and 2. The penetration testing tools for the database analysis is SQLmap. After running an injection test on platform 1, the penetration testing tool produced the result shown by Figure 7-4 which shows that target platform 1 is protected by a WAF (web application firewall) or an IPS (intrusion prevention system). This prevents any form of injection or attack on the database of platform 1.

```
[14:38:27] [INFO] testing connection to the target URL
[14:38:33] [INFO] heuristics detected web page charset 'ascii'
sqlmap got a 301 redirect to 'https://www.edemso.com/'. Do you want to follow? [Y
[14:38:49] [INFO] checking if the target is protected by some kind of WAF/IPS
[14:38:49] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
Y
[14:40:55] [WARNING] dropping timeout to 10 seconds (i.e. '--timeout=10')
[14:40:55] [INFO] using WAF scripts to detect backend WAF/IPS protection
[14:41:31] [CRITICAL] WAF/IPS identified as 'Generic (Unknown)'
are you sure that you want to continue with further target testing? [y/N] y
[14:41:38] [WARNING] please consider usage of tamper scripts (option '--tamper')
[14:41:38] [INFO] testing if the target URL content is stable
[14:41:39] [CRITICAL] can't establish SSL connection

[*] ending @ 14:41:39 /2019-05-10/
```

Figure 7-4 Database Analysis Result for Platform 1

Also, after running an injection test on platform 2, the penetration testing tool produced the result shown by Figure 7-5 which shows that the target platform 2 is protected by a WAF (web application firewall) or an IPS (intrusion prevention system). This prevents any form of injection or attack on the database of platform 2.Figure 7-5 also shows that the IPS is CloudFront a service provided by Amazon.

```
do you want to try URI injections in the target URL itself? [Y/n/q] y
[15:02:25] [INFO] testing connection to the target URL
[15:02:26] [INFO] heuristics detected web page charset 'ascii'
sqlmap got a 301 redirect to 'https://www.ceviasolutions.com/=space2comment'. Do you want to follow? [Y/n] y
[15:02:41] [INFO] checking if the target is protected by some kind of WAF/IPS
[15:02:41] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS
do you want sqlmap to try to detect backend WAF/IPS? [y/N] y
[15:02:47] [WARNING] dropping timeout to 10 seconds (i.e. '--timeout=10')
[15:02:47] [INFO] using WAF scripts to detect backend WAF/IPS protection
[15:02:48] [CRITICAL] WAF/IPS identified as 'CloudFront (Amazon)'
```

Figure 7-5 Database Analysis result for Platform 2

In order to confirm that the penetration testing tool actually works, it is used to inject a web application with known vulnerabilities and it can be seen that from Figure 7-6 there is no WAF or IPS preventing the injection attack from proceeding thereby confirming that the database of platform 1 and 2 are secured from commonly known database vulnerabilities or loopholes.

```
[14:55:25] [INFO] testing connection to the target URL
sqlmap got a 301 redirect to 'http://www.ourdailydevotionals.com/?id=1'. Do you want to follow? [Y/n] y
[14:55:33] [INFO] testing if the target URL content is stable
[14:55:35] [WARNING] GET parameter 'id' does not appear to be dynamic
[14:55:37] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[14:55:39] [INFO] testing for SQL injection on GET parameter 'id'
[14:55:39] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:55:41] [WARNING] reflective value(s) found and filtering out
[14:56:06] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:56:28] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[14:56:38] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[14:56:45] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[14:56:55] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[14:57:34] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[14:57:36] [INFO] testing 'MySQL inline queries'
[14:57:37] [INFO] testing 'PostgreSQL inline queries'
[14:57:39] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[14:57:41] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
```

Figure 7-6 Database Analysis result for Web Application with known Vulnerability

7.3.4 Risk Analysis

The risk analysis is the summary of the classification of the risk level of various vulnerabilities found during the scanning and discovery phase. Table 7-6 shows the risk level of the vulnerabilities found in both platform 1 and platform 2. From Table 7-6 platforms 1 has 8 overall low-risk probability vulnerabilities,9 critical risk vulnerabilities, and zero severe risk

vulnerabilities. Also, platform 2 has 19 overall low-risk probability vulnerabilities, 2 critical risk vulnerability, and zero severe risk vulnerabilities

Table 7-6 Risk Analysis Summary

Target Platform	Risk Analysis	Risk Level		
Platform 1	**Scanning Phase**	**Low-Risk Probability**	**Critical/Medium Risk**	**Severe/High Risk**
	Vulnerability Analysis	1	7	0
	Web application Analysis	7	2	0
	Database Analysis	0	0	0
Platform 2	**Scanning Phase**	**Low-Risk Probability**	**Critical/Medium Risk**	**Severe/High Risk**
	Vulnerability Analysis	1	0	0
	Web application Analysis	16	2	0
	Database Analysis	0	0	0

7.4 Execution Phase

In this phase, the discovered vulnerabilities are exploited with exploitation tools to see if they are exploitable. Many attempts to exploit the critical vulnerabilities were not successful mostly because of the protect firewall provided by Amazon. The vulnerability attempted to be exploited was the password cracking as shown in Figure 7-7. The many exploitation attacks were unsuccessful because the vulnerabilities discovered during the scanning phase are not very severe or high-risk vulnerabilities which could have been easily exploited. The unsuccessful attempt to exploit the discovered vulnerabilities is not a guarantee that they can’t be exploited by more advanced and sophisticated attackers. It's therefore important that the proposed recommendations are implemented to lower the risk of attack to its minimum.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-05-14 04:23:37
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (1:1/p:1), ~1 try per task
[DATA] attacking ftp://54.93.48.11:21/
[ATTEMPT] target 54.93.48.11 - login "/usr/share/metasploit-framework/data/wordlists/user" - pass "/usr/share/metasploit-framework/data/wordlists/password" - 1
[REDO-ATTEMPT] target 54.93.48.11 - login "/usr/share/metasploit-framework/data/wordlists/user" - pass "/usr/share/metasploit-framework/data/wordlists/password
[STATUS] 2.00 tries/min, 2 tries in 00:01h, 1 to do in 00:01h, 1 active
[REDO-ATTEMPT] target 54.93.48.11 - login "/usr/share/metasploit-framework/data/wordlists/user" - pass "/usr/share/metasploit-framework/data/wordlists/password
[REDO-ATTEMPT] target 54.93.48.11 - login "/usr/share/metasploit-framework/data/wordlists/user" - pass "/usr/share/metasploit-framework/data/wordlists/password
[STATUS] 2.00 tries/min, 4 tries in 00:02h, 1 to do in 00:01h, 1 active
1 of 1 target completed, 0 valid passwords found
```

Figure 7-7 Password Cracking Exploits on Platform 1

7.5 Result Summary

Penetration test of the target platform 1 and 2 can be performed using any three methods discussed in 5.2 and a summary based on security test methods are shown in Table 7-7. For each test method, there is an environment where the test can be performed. In this project, the direct test method was mostly used, and the comparative test method was used on one occasion only. Table 7-7 shows that the target platforms 1 and 2 has 36 vulnerabilities combined and has a secured database.

Table 7-7 Overall Penetration Test Result Summary

Result Summary Based on Test Methods			
Security Testing Phase	**Test Environment Method**	**Comparative Test Method**	**Direct Test Method**
Environment	Test environment	Production environment	Production environment
Information Gathering	Information gathered	Information gathered	Information gathered
Scanning & Discovery	0	0	0
Vulnerability Analysis	0	0	9
Web Application Analysis	0	0	27
Database Analysis	0	The database is secured comparing with Platform with known vulnerabilities	Not Considered
Execution Phase	Not considered	Not considered	**Performed**

8 Discussion

In this chapter the methods used in the project are explained, the results obtained are analyzed and interpreted in detail, solutions to the inquiry of the project are laid out, recommendations to solving weaknesses and loopholes found are discussed and the overall project work is critically evaluated.

8.1 Introduction

Organizations and corporations continuously seek to innovate, reinvent and be steps ahead in the security of their online assets, platforms, and network. In this project work, the objective was to extensively explore every possible area of loopholes in the client's platform, report findings and proffer applicable recommendations that can be implemented to better secure the client's platform. To achieve this goal various security testing methods were analyzed, and a suitable method was adopted and executed which produced results that will be discussed, analyzed and document recommendations based on the results. This project documented three testing methods which could be used to perform extensive security testing on a target platform which can be seen in 5.2 In this project the direct testing method was mostly used, and the comparative test method was used on one occasion only. The third method of testing which was not used in this project is recommended for future research work in software security testing. The results obtained this project will be discussed following the order in which the security tests were performed.

8.2 Information Gathering Phase Results

Table 7-1 shows the information gathered for the two platforms examined which gave a more detailed understanding of the target platforms and served as a pool of resources used to perform the subsequent phases of the security test. The information gathered also influenced the choice penetration tools used in the succeeding phases of the security test of the target platforms

8.3 Scanning and Discovery Phase

The scanning and discovery phase results are divided into three separate parts which are the vulnerability analysis results, the web application analysis results and the database analysis results.

8.3.1 Vulnerability Analysis Results

The vulnerability analysis results are shown in Table 7-2, Table 7-3 and Figure 7-2 cover results of the networks can for open ports and traceroute and the actual vulnerability scan result. The results of the port scans for both platforms 1 and 2 illustrates that only the important ports are open. It's important to state that if ports that are not required to be open are found open, they become automatic loopholes that attackers can exploit to gain unauthorized access into a system and do damage.

From results shown in Table 7-3 only important and required ports are found to be open. For platform 1, four ports are open which are port 22tcp, port 80tcp, port 443tcp, and port 8443tcp. Port 22tcp which runs the ssh service which includes secure login and file transfer. Port 80tcp runs HTTP services which make the web application accessible and usable. Port 443tcp which also runs https and SSL service which is vital for a web application to be usable. Port 8444tcp

which runs SSL and https services as well. For platform 2, two ports are open which are port 80tcp and port 443tcp. Port 80tcp runs HTTP services which make the web application accessible and usable. Port 443tcp which also runs https and SSL service which is vital for a web application to be usable.

The results as seen in Figure 7-2 demonstrates that there are 8 vulnerabilities found in platform 1 and only one vulnerability found in platform 2. The vulnerability found in platform 2 which is a low-risk probability vulnerability which cannot be exploited to cause damage in the system. Seven of the eight vulnerabilities found in platform 1 are critical but not very high risk, nevertheless, they can be exploited by highly skilled attackers and therefore solutions to eliminating these vulnerabilities are discussed below;

- CVE-2016-6515 is a vulnerability that could allow an attacker to perform a denial of service using an unlimited length of passwords. This vulnerability can be eliminated by updating OpenSSH on port 22tcp to OpenSSH 8.0 which was released on the 17th of April 2019. The OpenSSH 8.0 does not accept password authentication request for passwords more than 1024 characters thereby eliminating that vulnerability.
- CVE-2015-8325 is a vulnerability that occurs when the UseLogin feature is enabled which can allow an attacker to gain privileged access into platform 1. Updating the Ubuntu to the most recent version will eliminate this vulnerability since UseLogin is disabled in sshd_config of Ubuntu.
- CVE-2018-15473 is an OpenSSH vulnerability that allows an attacker to determine verified usernames on platform 1. This vulnerability can be eliminated by updating OpenSSH on port 22tcp to OpenSSH 8.0 which was released on the 17th of April 2019
- CVE-2017-15906 is an OpenSSH vulnerability which allows an attacker to create zero-length files on platform 1. This vulnerability can be eliminated by updating OpenSSH on port 22tcp to OpenSSH 8.0 which was released on the 17th of April 2019
- CVE-2018-15919 is an OpenSSH vulnerability that allows valid users to be detected by remote users. This vulnerability can be eliminated by updating OpenSSH on port 22tcp to OpenSSH 8.0 which was released on the 17th of April 2019
- CVE-2016-10708 is an OpenSSH vulnerability that allows attackers to perform a denial of service on platform 1. This vulnerability can be eliminated by updating OpenSSH on port 22tcp to OpenSSH 8.0 which was released on the 17th of April 2019
- CVE-2016-6210 is an OpenSSH vulnerability that allows attackers to hash passwords in platform 1. This vulnerability can be eliminated by updating OpenSSH on port 22tcp to OpenSSH 8.0 which was released on the 17th of April 2019.

8.3.2 Web Application Analysis Results

The web application analysis results are shown in Figure 7-3 which illustrates the vulnerabilities found in the web application of platform 1 and platform 2. The results show that both platform 1 and platform 2 has 23 low-risk probability vulnerabilities and 2 critical risk level vulnerabilities for each platform. The 2 critical risk level vulnerabilities are the same for platform 1 and 2. The vulnerabilities are;

- The X-frame-Option header which serves as a defense against clickjacking attack in platform 1 and 2 are absent. This vulnerability can be eliminated by configuring the internet information services (IIS) manager to ensure that an HTTP response header is sent thereby preventing clickjacking attacks.
- The cross-site scripting (XSS)-Protection header for platform 1 and 2 are not properly defined which makes it difficult to defend the platforms against some forms of cross-site scripting. This

vulnerability can be eliminated by defining it properly in the internet information services (IIS) manager.

Other low-risk probability vulnerabilities common to both platform 1 and platform 2 are;

- The anti-sniffing header is absent on both platform 1 and 2 which can allow for sniffing with an older version of internet explorer or google chrome. This vulnerability can be eliminated by setting the anti-sniffing header properly in the internet information services (IIS) manager.
- Improper setting of the cache-control and pragma HTTP in both platform 1 and 2. This vulnerability can be eliminated by setting the cache-control and pragma HTTP properly in the internet information services (IIS) manager.
- Third-party JavaScript present in platform 1. This vulnerability can be eliminated by ensuring that only trusted third-party JavaScript is present and that the end user is unable to control or manipulate the scripts.

8.3.3 Database Analysis Results

The Figure 7-4,Figure 7-5 and Figure 7-6 shows the results of the database analysis for platform 1 and platform 2.From the results it can be seen that platform 1 and platform 2 are secured from most common database vulnerabilities because it has WAF (web application firewall) or an IPS (intrusion prevention system) which is provided by Amazon. This web application firewall detected the penetration test operation carried out in this project and prevented further intrusion. Therefore, the database of the platform 1 and platform 2 are secured.

8.4 Execution Phase Result

The execution phase results show that the exploitation of the discovered vulnerabilities were unsuccessful. This was mainly because the vulnerabilities discovered were critical but not severe or very high risk. The ease and chances of successfully exploiting these vulnerabilities are usually low and require very advanced hacking skills. The fact that the many exploitation attacks carried out in the project on the discovered vulnerabilities were unsuccessful does not guarantee that these vulnerabilities cannot be exploited by very advanced and highly sophisticated hackers, it simply indicates that the Edemso Software cannot be easily attacked and does not have high-risk vulnerabilities that can be exploited easily. To eliminate all possible risk of attacks the proposed recommendations on how to handle the discovered vulnerabilities should be fully implemented.

8.5 Overall Security Test Results

In summary from the Table 7-7, the Edemso software which is made up of two platforms 1 and 2 have a total of 36 vulnerabilities, 9 of which are network vulnerabilities and 27 are web application vulnerabilities. There were no vulnerabilities found during the database analysis. The Edemso software does not have any severe or high-risk vulnerability, but it has 9 critical risk or medium risk vulnerabilities and 27 low-risk probability vulnerabilities. The 9 critical risk vulnerabilities can be eliminated by updating the OpenSSH and Ubuntu to the most recent versions, making some specific configurations in the internet information services (IIS) manager. Many of the low-risk probabilities are very unlikely to be exploited and therefore not considered to be a risk to the Edemso software, nevertheless, some of the common low-risk vulnerabilities can be eliminated by making some changes in the internet information services (IIS) manager.

8.6 Further Research Work

Software and data security are a continuous and ongoing process that requires continuous research and innovation. During this project, only two methods of security testing were used to test the Edemso software. These methods are direct testing methods and comparative testing methods. The third method which is the test environment method is a good method that should be considered for future studies and research work. This method will give room to extensively test the Edemso software without concern of doing damage to the real system. Also, the security tester will not feel limited by the protective firewall provided by the Amazon web services. Future research and study using the test environment will give new insights into the security state of the Edemso software.

9 Conclusion

This project has demonstrated that software and data security of any system, network, software or asset of an organization should be an ongoing process and an area of continuous improvement. The Edemso software security was studied in relation to common threats, security testing methods, the security testing tools and the various software security testing phase with the core objective of knowing the security state of the software and possible recommendations for improvement.

The principal findings of the project show that the Edemso software is not at risk of very severe vulnerabilities or weaknesses with 70% of the discovered vulnerabilities classified as low-risk probability vulnerability and 25% as critical risk vulnerabilities. The 25% critical risk vulnerabilities that were discovered in this project can be exploited with attacks like denial of service, password hashing, password attacks, clickjacking and cross-site scripting can be quickly eliminated if the proposed recommendations are implemented.

During this project several attempts to exploit the discovered vulnerabilities and gain access to the Edemso software were unsuccessful, this is mostly because the discovered vulnerabilities exploited were not very high-risk vulnerabilities. This is not to say the discovered vulnerabilities cannot be exploited by more advanced and sophisticated attackers with superior hacking tools. To fully secure the Edemso software from any possible attack now and in the future, the proposed recommendations should be fully implemented. The summary of the recommendations is; the OpenSSH in port 22tcp should be updated to the most recent version, the Ubuntu used in port 22tcp, 80tcp and 440tcp should be updated to the most recent version and make several configurations in the internet information services (IIS) of the Edemso software.

The Edemso system database is generally secured as the web application firewall provided by Amazon web services detected every intrusion attempt made to inject the database of the Edemso software during the project. This is a clear indication that the database of the Edemso software is secured and safe from many common vulnerabilities and potential attacks. It is therefore recommended that the Edemso software continue to use the web application firewall service offered by Amazon.

Based on the findings of this project, the Edemso software is recommended for regular security test to stay updated on newer vulnerabilities and to protect the software from future intrusion. Also, the web application firewall by Amazon web service remains enough for the security of the Edemso software database. Also, to get more information about the security of the Edemso software, future research work can be done on Edemso using a test environment method, where the Edemso software is deployed into a local server and extensive security test is performed within the test environment. This future work will further guarantee the security of the Edemso software based on the new knowledge gathered.

10 References

[1] Cevia Solutions, “User Manual,” Jan 2018. [Online]. Available: https://www.ceviasolutions.com/edemso. [Accessed 10 Jan 2019].

[2] Cevia Solutions, “Design and Security,” Jan 2018. [Online]. Available: https://www.ceviasolutions.com/edemso. [Accessed 10 January 2018].

[3] Cevia Solutions , “Data Model,” Jan 2018. [Online]. Available: https://www.ceviasolutions.com/edemso. [Accessed 10 January 2019].

[4] D. Zhang, “Big Data Security and Privacy Protection,” in *Proceedings of the 8th International Conference on Management and Computer Science (ICMCS 2018)*, NYC, 2018.

[5] EasyTechGuides, “Malware: Types, Protection, Prevention, Detection & Removal - Ultimate Guide,” EasyTechGuides.

[6] S. Schuster, “Blocking Marketscore: Why Cornell Did It,” Cornell University, Office of Information Technologies, 205.

[7] J. Schofield, “How can I remove a ransomware infection?,” The Guardian, London, 2016.

[8] TheINQUIRER, “Shedun trojan adware is hitting the Android Accessibility Service,” TheINQUIRER, 2015.

[9] W. Stallings, Computer security: principles and practice, Boston: Pearson. ISBN 978-0-13-277506-9., 2012.

[10] J. Aycock, Computer Viruses and Malware, Springer. ISBN 978-0-387-30236-2., 206.

[11] P. V. v. d. Bussche, The EU General Data Protection Regulation (GDPR) A Practical Guide, Germany: Springer, Cham, 2017.

[12] ISO/IEC 27000, “ISO/IEC 27000 family - Information security management systems,” ISO ORG, 2013. [Online]. Available: https://www.iso.org/isoiec-27001-information-security.html. [Accessed 19 March 2019].

[13] S. O. B. H. C. J. S. K. L. N. A. Noel, “Combinatorial analysis of network security. In:AeroSense,” *International Society for Optics and Photonics*, pp. 140-149, 2002.

[14] Cabinet Office, “Cyber Security Strategy of the United Kingdom,” 2009. [Online]. Available: www.cabinetoffice.gov.uk/media/216620/css0906.pdf. [Accessed 11 FEB 2019].

[15] R. Moore, Cyber crime: Investigating High-Technology Computer Crime, Cleveland, Mississippi: Anderson Publishing, 2005.

[16] J. G. H. Warren G. Kruse, Computer forensics: incident response essentials, Addison-Wesley., 2002.

[17] North Denver News, “Cybercrime— what are the costs to victims - North Denver New,” North Denver News, 2015.

[18] Reuters, “Cyber crime costs global economy $445 billion a year: report,” Reuters, 2014.

[19] J. Lewis, “Economic Impact of Cybercrime - No Slowing Down,” McAfee, 2018.

[20] P. Foreman, “Vulnerability Management,” in *Vulnerability Management*, Taylor & Francis Group, 2010, p. 1.

[21] ISO/IEC FIDIS 27005, “"Information technology -- Security techniques-Information security risk management",” ISO/IEC FIDIS 27005, 2008.

[22] A. Kakareka, Computer and Information Security Handbook, Morgan Kaufmann Publications. Elsevier Inc., 2009.

[23] S. A. Keylogger, “Keylogger Removal,” SpyReveal Anti Keylogger, 2016.

[24] P. Eckersley and E. Portnoy, “ Intel's Management Engine is a security hazard, and users need a way to disable it,” EFF, 2017.

[25] M. Prince, “Empty DDoS Threats: Meet the Armada Collective,” CloudFlare, 2016.

[26] OWASP, “OWASP Top 10 2017,” The OWASP Foundation, United States, 2017.

[27] The OWASP™ Foundation, “The OWASP Projects,” The OWASP™ Foundation, [Online]. Available: https://www.owasp.org/index.php/Main_Page. [Accessed 12 April 2019].

[28] iphonebackupextractor.com , “How to extract data from an iCloud account with two-factor authentication activated,” 2019. [Online]. Available: www.iphonebackupextractor.com . [Accessed 2019].

[29] B. Krebs, “Security Fix – Citibank Phish Spoofs 2-Factor Authentication",” Washington Post, Washington, July 10, 2006.

[30] It's FOSS, “Best Linux Distributions for Hacking and Penetration Testing,” It's FOSS, [Online]. Available: https://itsfoss.com/linux-hacking-penetration-testing/. [Accessed 11 Feb 2019].

[31] AlternativeTo Crowdsourced Software Recommendation, “Alternatives to VirtualBox for all platforms with any license,” AlternativeTo Crowdsourced Software Recommendation, [Online]. Available: https://alternativeto.net/software/virtualbox/. [Accessed 11 Feb 2019].

[32] I. S. H. R. E. E. a. A. J. B. Konstantinos Xynos, “Penetration Testing and Vulnerability Assessments:,” International Cyber Resilience conference, United Kingdom, 2010.

[33] B. S. S. M. G. Arkin, “Software Penetration Testing,” *IEEE Security and Privacy,* vol. 3, no. 1, 2005.

[34] B. B. M. Jai Narayan Goela, “Vulnerability Assessment & Penetration Testing as a Cyber Defence,” Elsevier B.V, India, 2015.

[35] I. Krsul, “Computer vulnerability analysis,” Thesis proposal, Indiana, 1997.

[36] S. O. B. H. C. J. S. K. L. N. A. Noel, “Combinatorial analysis of network security.,” International Society for Optics and Photonics, 2002.

[37] S. K. E. K. C. J. N. Kals, “Secubat: a web vulnerability scanner.,” in *Proceedings of the 15th international conference on,* 2006.

[38] S. N. S. Jajodia, “Topological vulnerability analysis.,” in *Cyber Situational Awareness Springer;,* 2010.

[39] A. B. B. K. C. V. G. Doupe´, “Fear the ear: discovering and mitigating execution after redirect vulnerabilities,” in *Proceedings of the 18th ACM conference on Computer and communications security ACM,* 2011.

[40] Oracle Virtual Box, “Oracle Virtual Box,” [Online]. Available: https://www.virtualbox.org/. [Accessed Feb 2019].

[41] K. b. O. Security, “Kali Linux,” Offensive Security, [Online]. Available: https://www.kali.org/. [Accessed March 2019].

Appendices