IMPROVING SECURITY OF WEBSITES USING VULNERABILITY ASSESSMENT

BY

VINCENT APPIAH

PG8959213

A Thesis submitted to the Department of Computer Science

Kwame Nkrumah University of Science and

Technology

in partial fulfilment of the requirements for the degree

of

MASTER OF SCIENCE: INFORMATION TECHNOLOGY

OCTOBER, 2016

DECLARATION

“I declare that the thesis titled “Improving security of websites using vulnerability assessment” undertaken under the supervision of Dr. Michael Asante is the original work carried out by me with the exception of portions where references have been duly cited..

Date………………………………. …………………………………….

(STUDENT)

VINCENT APPIAH

(PG 8959213)

Date………………………………. …………………………………….

(SUPERVISOR)

DR. MICHAEL ASANTE

Date... ....

(HEAD OF DEPARTMENT)

ACKNOWLEGEMENTS

I would like to thank the Almighty God for His Grace and Protection.

My deepest appreciation goes to my project supervisor Dr. Michael Asante whose guidance, constructive comments and critical revision of the drafts made it possible for me to complete this thesis.

I would also like to thank Mr. Gideon Abbequaye , University of Cape Coast for his support ,advice and encouragement throughout my journey in the computer science world.

Finally I would also like to thank the Managers of the websites on which the vulnerability assessments were done. This test was useful and educational.

DEDICATION

This work is dedicated to my parents J.W. Appiah and Theresa Nettey for their support throughout my academic life. I wish them good health and long life.

ABSTRACT

The web is now an important means of transacting business. Without security, websites cannot thrive in today's complex computer ecosystem as there are new threats emerging as old ones are being tackled. Vulnerability assessment is one of the means by which security can be improved on websites. The aim of this study was to use vulnerability assessment to improve security by identifying vulnerabilities and proposing solutions to solve the security issues. Assessment was done on 5 web hosts belonging to different entities. Two of the web host had recently been compromised so this assessment was important to them. Nmap, Nikto and Nessus were the tools used for the assessment. The first stage in the vulnerability assessement was information planning which involved activities and configurations performed before the actual assessement. The second stage was information gathering which involved obtaining information about the targets necessary to help identify vulnerabilities. This was followed by vulnerability scanning to identify vulnerabilities on the target hosts. The results indicated all the five hosts had security flaws which needed to be addressed. 16 vulnerabilities were identified on host 1, 8 vulnerabilities were identified on host 2, 15 vulnerabilities on host 3, 4 vulnerabilities on host 4 and 10 vulnerabilities on host 5. After the vulnerabilities were identified, a solution was proposed to mitigate the security flaws identified. The solution involved three steps which were encryption, network monitoring and update and upgrade. At the end of the study reports were sent to the web managers of the hosts on which the assessments were done. The study was beneficial to the respective managers of the website because they discovered security flaws which they were not aware of even though there had been recent upgrade of their infrastructures.

TABLE OF CONTENTS

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

LIST OF FIGURES

LIST OF TABLES

APPENDICES

CHAPTER 1

INTRODUCTION

1. 1 Background

Computer Security is the protection of computing systems and the data that they store or access. Currently, computer security is one of the most talked about issues in computing. This is due to its importance in almost every computer system.

Computer security is now employed in every field which deals with information processing and data storage. The use of ATM cards, the use of credit cards, authentication and information access all involves computer security to safe guard the activities. In other to maintain a productive computing environment, computer security should be a priority. An organization which suffers from security breach is likely to lose customers, have a tarnished image and pay huge amounts of money to recover from the breach , not to talk about legal costs.

1.2 Justification

Cyber crime is on the increase across the globe and as such organizations should also protect their systems against such attacks. One way of ensuring protection is to identify such security flaws before the attackers do by conducting security tests and implementing solutions to mitigate such security problems. This research will help the respective officers to identify security problems within the selected systems, correct them and also improve them. This research will also provide useful information on the selected websites for further research.

1.3 Problem statement

There has been a rapid growth of Computer technology in the form of Internet and mobile technology. This has created a conducive and enabling environment for hackers and other malicious individuals to improve their attacking techniques leading to an increase in cyber crimes. Some of these attacks are also difficult to detect while others are complex.

Cybercrimes include stealing of credit cards, identity thefts, denial of service attacks, botnet attacks, sql injection attacks. These malicious attacks destroy computer systems, thereby resulting in the spending of lots of money for the repair of such computer systems. Some of these security breaches also result in the release of confidential information to the public and this has resulted in distrust among the users of internet for business transactions due the fear of their private details being stolen and sometimes spreading them across the internet for public viewing. Companies such as Sony, JP Morgan, Facebook, Microsoft, Twitter and a lot more have suffered heavy losses due to these cybercrimes.

An ever increasing cyber crime rate requires improving security by probing the computer system to see if there are any security flaws. It is therefore important that measures are put in place to ensure trust and safety when using such computer systems. A secure system will also ensure that Integrity of data, Availability of data and Confidentiality of users are protected. This research therefore seeks to improve security of selected computer systems by identifying vulnerabilities and design appropriate solutions to eliminate them.

1.4 Main Objective

The aim of this study was to improve security of selected websites using vulnerability assessments.

1.5 Objectives

The aim of this research was achieved by:

- Identifying vulnerabilities on the web hosts.
- Determine the severity of the vulnerabilities identified.
- Proposing solutions to address the security issues discovered.
- Proposing recommendations to further improve the security of the web hosts.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Computer security also known as cyber security comprises of techniques used to ensure that data or a system is well protected and available to the right personnel and involves the protection of resources and information from being accessed, modified, created or destroyed by unauthorized personnel. Computer Security also helps in the prevention and detection of unauthorized use of your computer and involves the process of safeguarding against intruders from using your computer resources for malicious intents. It is the ability of a system to protect and to ensure the availability of information and the system resources, with respect to confidentiality and integrity (Ramilli ,2012).

Nowadays information has become asset to many institutions and as a result these institutions have become targets for people with malicious intents. For example e-business use information such as usernames, user address and bank account details for business transactions another user getting hold of such information may use them for impersonation and commit other crimes in the name of the victim. Information therefore needs to be protected to prevent such things from happening. Aside information theft, malicious activities such as installation of backdoor, denial of service and unlawful access are other malicious activities that are often conducted against some of these institutions that make use of computer technology for productive activities.

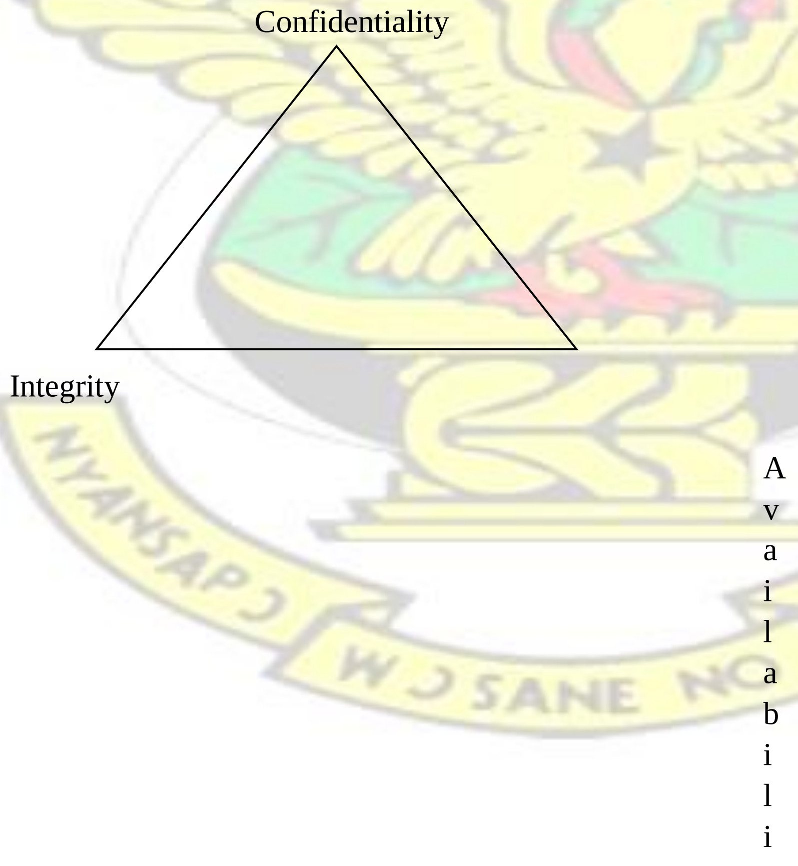
Computer security is one of the most important areas in the IT world. A lot of money and resources is invested into computer security showing its importance. Countries such as USA, Britain, China, Russia and France spend billions of dollars yearly in order to improve their cyber security. In the state of the union address in 2015, President Obama indicated that $14 billion has been set up in the 2016 budget proposal to beef up cyber security in the US (Kerr, 2015).

With the introduction of sophisticated and yet easy to use tools, network attacks are now a piece of cake for even novice attackers. As such network security personnel must be abreast with these tools and techniques in order to best defend the networks. Security officers should be able to find vulnerabilities and fix them before the attacker does any damage.

2.2 Models of Computer Security

These are principles that are used to ensure security of resources on computer systems. They are usually referred to as the C.I.A. triangle. There are three of them and they are :

1. Confidentiality.
2. Integrity.
3. Availability.



t
y

Fig. 1: Models of Computer Security

2.2.1 Confidentiality

It ensures that access to places or information are limited to the authorized personnel or facility. Confidentiality ensures that sensitive information are not given to the wrong people. Confidentiality also ensures the secrecy of information (Nemati, 2008). It is a security measure which protects against the disclosure of information to parties other than the intended recipient. For example in military, confidentiality ensures that military tactics, technology, weapons and other top secret information are not exposed to the enemy. Businesses use confidentiality to keep trade secrets safe and also to protect customer information. Confidentiality is breached if unauthorized individuals or applications can view information not intended for them (Whitman and Mattord, 2012).

Techniques that are used to ensure confidentiality include the following.

- Encryption
- Biometric verification

2.2.1.1 Encryption

This is the process of encoding data into a form called ciphertext which can only be understood by authorized people. The act of converting the data to its original meaningful form is known as decryption and this is done by the use of a secret key. Without the correct key the ciphertext cannot be decrypted. By the use of secret keys shared between only the parties involved in the data exchange, confidentiality is ensured.

There are two main encryption schemes which are:

- Asymmetric encryption
- Symmetric encryption

Asymmetric encryption involves the encryption and decryption of a message using a pair of keys. The two keys are public key and private key. Anyone with the public key and encrypt a message but only the private key holder can decrypt the message. This type of encryption is also known as public-key cryptography. In Symmetric encryption, the same key is used for both encryption and decryption of a message. This encryption scheme requires both parties involved in the exchange of data to know the secret key. For example if computer A is sending a message to computer B . Both computer A and B must know the key before they can encrypt and decrypt message being exchanged between them. Symmetric encryption is the oldest encryption technique (Ramili, 2012).

2.2.1.2 Biometric Verification

It is an authentication technique by which a person is identified using biological traits that are unique to the person only. Unique biological traits that are used in biometric verification include:

- Fingerprint.
- Iris.
- DNA.
- Hand geometry
- Face recognition.

Biometric verification is one of the most effective ways of ensuring data confidentiality due to the fact that human traits are nearly impossible to replicate unlike usernames and passwords. For example, fingerprint scanning can be used to identify two different individuals since two people cannot the same fingerprints. Fingerprint is one of the widely used biometric techniques for authentication (InfoSec Institute,2015).

2.2.2 Integrity

This ensures that data is accessed and modified by authorized users and as result makes data authentic and trustworthy. Integrity helps to prevent the changing of data in transit as this form of data is susceptible to modifications by unauthorized people. Data integrity is also a security measure that helps to ensure consistency in data by preventing its modification from unauthorized users and thereby making the data quality. Data with integrity is data which is unchanged and accurate from its source to its destination (Lehtinen and Gangemi,2011).

Attacks that can affect data integrity include man-in-the-middle attack where the attacker intercepts data in transit and makes changes to it before it reaches its destination.

It must be noted that not all data modifications are intentional. Data modifications can be accidental.

Data can be modified in the following ways.:

- Man-in-the-middle attacks.
- Modification or corruption by computer viruses and worms.
- Errors occurring during the transmission of data.
- Natural disasters such as earthquakes, floods and fires.
- Faulty hardware such as storage disks.

Techniques that can be used to ensure integrity include.

- Access control.
- Mirroring.

Access control is a security technique used to manage user access to information and resources on a computer system. This is usually done by means of identification where the user avails his/her credentials, authentication where the user confirms his/her identity and authorization where the user is granted access to the requested resources based on the permissions assigned to the user . With the aid of access control mechanisms, a user’s access can be granted or revoked depending on the security policies for the system. File permissions and data privileges are examples of access control mechanisms used to ensure data integrity (Vacca, 2009).

Mirroring involves making two or more copies of a data and storing them. The copies of the data can then be compared and if they are not the same then the integrity of the data can be doubted and appropriate measures taken. However mirroring cannot ensure complete integrity as an attacker can easily modify all copies of a data (Sivathanu *et al*, 2005).

2.2.3 Availability

This ensures that data is available to users at all times as well as preventing the loss of such data. Implementing availability also means that authorized users will always have access to their respective data even in emergency situations. Availability also means that there will be no unauthorized omission of data to legitimate users . Despite efforts being made to make data available some challenges are always encountered.

These challenges include:

- Occurrence of natural disasters such as floods and earthquakes.
- Faulty equipments.
- Software errors.
- Denial of service attacks.

Regular backup can help ensure that data is always available.

2.3 Importance of Computer Security

- Cyber security ensures that networks and computer systems are protected from cyber criminals. Having a secured network will prevent attackers from intruding and obtaining sensitive information as well as causing mayhem. An ever increasing cybercrimes requires that computer systems are well protected to prevent them from being attacked by cyber criminals. Institutions, like Facebook, Twitter and Sony Pictures have all fallen victim to cyber attacks and the results were not pleasing.

- Computer security has now made it possible to safeguard information. Most transactions that are done today on web applications involve personal and sensitive information which if not protected might be exposed to third parties. Computer Security ensures that this is possible.

- Whenever there is a security breach , lots of money is spent to repair such breaches as well as improve them . Computer security ensures that such breaches are prevented thereby saving cost.

- Computer security also helps to identify security flaws and vulnerabilities and appropriate solutions given. This is usually done through security audits, vulnerability assessments and penetration tests.

2.4 Website Security

Websites are collection of documents that can be accessed through the internet. Websites are now the primary source of information and used for a lot of activities. It is used for academic purposes, business purposes, entertainment etc.

Due to this website nowadays are always under attack. While some of these attacks are meant for stealing, others are meant to disable the services being provided by the target websites.

2.5 Website Security risks

Websites now face a great deal of security risks. These risks can affect confidentiality, integrity or availability of data. Negative impact of some of these risks is very low while others can be very devastating. Some of the security risks are:

- Buffer overflows.
- Denial of service attacks (Dos)
- OWASP Top 10

2.5.1 Buffer Overflow

This is the situation where data being written by a program to a buffer is more than the capacity of the buffer. As a result the extra data flows to the adjacent memory locations. Buffer overflows occur due to deficiency in memory management implementations in a program such as bounds checking mechanisms. Programs that are written in C usually face this issue. For example if a program allocates 20 bytes to a memory buffer and attempts are made to store 25 bytes, the extra 5 bytes will flood to the adjacent buffer and this might cause the program to crash. If a data in that adjacent space it might be overwritten. Buffer overflows can lead to the crashing of a program (denial of service) or insertion of a remote shell which can be used to execute arbitrary codes (Nemati, 2008).

2.5.2 Denial of Service

This is an attack that renders an application or network unable to function properly. This is usually performed by sending several requests to the application. If the number of requests is more than it can handle, the application hangs and users will not be able to use the service. Buffer overflow attacks can also cause denial of service by flooding the memory with data. A distributed denial of service is used to describe the situation where large numbers of computers are used to cause denial of service. (Svenhard and Radaslic,2012). Denial of service attacks can take several forms which include:

- Buffer overflow
- Smurf attack
- Tear drop attack

Buffer overflow attacks are usually performed by sending data which is larger than the allocated memory buffer. As a result the extra bytes flood to adjacent buffers and the program crashes. Example is the PING of death where oversized Internet Control Message Protocol (ICMP) packets are sent to a receiving application. This causes the application to crash (TechTarget, 2007).

Smurf attack involves the attacker sending packets to a receiving machine. The request is then sent to all hosts on the network using the broadcast address. The packet then sent to the address indicated in the packet headers. This is usually the address of the target address (IP spoofing). Because this is a broadcast, all the hosts which received the request also send their response to the same address. If the packets are overwhelmingly large, then the target address is unable to receive all other incoming traffic.

The tear drop attack involves sending large packet data to the target machine. The Internet Protocol (IP) unable to handle reassembly of the packet fragments due to a confusing offset value eventually causes the system to crash.

2.5.3 OWASP Top 10

Apart from the aforementioned risks, 10 security risks has also been identified by Open Web Application Security Project (OWASP) as the most critical security risks associated with web applications. These risks are known to be common forms of attacks. Aside that they are known to be exploitable and can have a negative impact on websites when executed hence their rank as the top 10 (0WASP, 2013). The top 10 risks as published by OWASP are:

1. Injection flaws

2. Broken authentication and session management.
3. Cross site scripting.
4. Insecure direct object references.
5. Security misconfiguration.
6. Sensitive data exposure.
7. Missing level access control.
8. Cross site request forgery(CSRF).
9. Using components with known vulnerabilities.
10. Unvalidated redirects and forwards.

2.5.3.1 Injection Flaws

SANS institute explains that injection flaws occur when an unexpected data is sent by a malicious client. Injection flaws allow an attacker to inject code into the vulnerable computer system. If the injected code is executed, the effect can be disastrous. Aside from the stealing information, injection attacks can cause denial of service or multiplication of worms in a system. Injection attacks include SQL injection, OS injections and LDAP injections. Injection flaws occur when a user input is not properly filtered for string escape characters that are often embedded in SQL statements (OWASP, 2013). In 2007 the Open Source Web Application Security Project(OWASP) graded sql injection attacks among the top ten most common attacks on web applications. With the development of automated tools such as Havij, Sqlmap and Sql ninja, SQL injection attacks have been made easier and novice hackers can easily attack websites with the technique. Several major websites have fallen victim to this type of attack showing its common usage among attackers.

In the year 2011 several injection attacks were made. The hacker groups LulzSec and

Anonymous were behind most of these attacks. They called these series of attacks Operation AntiSec (UKEssays,2015).

In April, 2011 the Sony Playstation Network was attacked. Again, SQLi was used for this attack. The hacking group LulzSec claimed responsibility. The group obtained personal information from over 77 million users from Playstation network(PSN). Hackers had access to data of about 77 million users.   (Quinn and Arthur,2011).

In June, 2011 Sony Playstation had its site compromised by the hacker group Lulzsec . Over 1,000,000 user information including password and e-mail addresses were taken from the website. (Aamoth, 2011).

Anonymous, which is another hacker group as part of the AntiSec campaign used SQL injection to attack Booz Allen Hamilton, a consulting firm in July 11 ,2011. Anonymous attacks released details on internal data including 90,000 military emails and passwords.

2.5.3.2 Broken Authentication and Session Management

This is the second most common flaw in the OWASP top 10. This stems from the fact that flaws exist in session management implementations in web applications. Misconfigurations such as storage of passwords in plain texts or weak encryption of user credentials can lead to this form of attack. According to OWASP, flaws in the implementation of password management, logout mechanism, timeout, remember me, forgot my password etc can also lead to broken authentication and session management attacks. If a website only employs a specific URL parameter setting to define authentication and an attacker obtains this Url, then this website is likely to suffer from broken authentication and session management attack (Svenhard and Radaslic,2012).

2.5.3.3 Cross-Site Scripting (XSS)

This is a type of vulnerability in which malicious code injected by a client is executed by the web application. The execution is made possible because the web application is unable to properly filter input properly. This can lead to stealing of cookies, website defacement and session hijacking. XSS is amongst the most common vulnerabilities of web applications (Acunetix,
2014).

There are three main types of XSS and these are:

- Stored XSS
- Reflected XX
- DOM based XSS

A stored XSS occurs when the malicious code is stored on the server. When the code is not properly neutralized and a victim retrieves it, the code is executed.

Reflected XSS occurs when the malicious script after it being entered into the web application is returned as an error message to a user.

A typical reflected XSS might occur when a malicious script is entered into a search box. If the response from the web application includes all or part of the user input then reflected XSS has been performed (OWASP, 2013).

In DOM based XSS , the malicious script is not stored on the server. Instead it is executed on the user browser due to inability of the client browser to filter the script. During a DOM based attack, a sanitized data is converted to executable JavaScript by the code running on the page (Acunetix, 2014).

2.5.3.4 Insecure Direct Object References

This is where unauthenticated clients are given access to restricted resources such as directories and configuration files. An example is a situation where a directory or a password file that should be available to only administrators on network is exposed to other users on the network. The absence of access control check can often result in unauthorized access to such resources through manipulation of URL parameters.

2.5.3.5 Security Misconfiguration

This flaw exists if web applications enable certain features by default. For example default passwords, default accounts, enabled directory listing, bugs in source codes and other misconfigured settings. Security misconfigurations can give way to external and internal attacks and according to OWASP can result in unauthorized access or complete system compromise. Secure configuration settings should be used to ensure use of web applications.

2.5.3.6 Sensitive data exposure

Sometimes sensitive data is left unprotected on web applications. These can be stolen or modified by attackers and used to gain access or perform unauthorized transactions. Using weak encryption schemes can also result in sensitive data exposure. Attackers can use bruteforce to obtain the plain text. Also sensitive data can be used to exploit the web application or find other exploitable vulnerabilities on the web application.

2.5.3.7 Missing Level Access Control

This occurs when users are not properly authenticated but given access to restricted resources. A web application must be able to limit and control the access to resources. If the application is unable to do this, then attackers can leverage this to gain access to restricted resources and even modify data on the server. This might affect the integrity of the data. There should be security checks to ensure that a user is properly authenticated and given the proper access rights especially if several users with different roles are exist on the web application(OWASP,2013).

2.5.3.8 Cross-Site Request Forgery (CSRF)

This is a type of attack where unauthorized HTTP requests are sent from a user's browser to a web application in which the user is currently logged on. In contrast to XSS, CSRF exploits the trust that a site has in a user's browser. Because there is trust, the web application is forced to execute these requests (Auger, 2010).

A CRSF attack usually begins when an attacker coerce a victim into clicking a malicious link. After the victim has clicked the link session cookies and other authentication information is stolen by the attacker. The information is then used to force a victim's browser to make requests to the vulnerable web application on behalf of the victim.

CRSF attack is also called one-click attack and is number eight on the OWASP 2013 top ten.

2.5.3.9 Using Components with Known Vulnerabilities

Applications with known vulnerabilities are likely to be compromised because exploits might be available. If such applications are compromised, an attacker might gain full access to the network and this will affect confidentiality.

2.5.3.10 Unvalidated redirects and forwards.

This is due to improper validation / unvalidation of user data. Attackers can leverage this to redirect victims to malicious webpages as well. Also forwards can be used to access restricted pages. This can affect confidentiality of data.

2.6 Improving security of websites.

With the advent of new technologies, new flaws are being discovered whiles attack methods are enhanced. As a result security risks for websites are also increasing. It is therefore important that steps are taken to improve security of websites.

One way of ensuring security is to regularly conduct vulnerability assessment.

2.7 Vulnerability assessment

It is the identification of anomalies in computer and network securities that will lead to weaknesses in technology. It include methodologies for prioritization and implementation of additional security measures for the fixation and protection of computer systems. It is the process of evaluating the security of a network through the identification of security flaws and vulnerabilities that might exist in a computer system also find appropriate mitigations for the identified flaws. It is also a means of identifying the vulnerabilities and exploits that might exist on a computer network.

During vulnerability assessment, manual and automated techniques are often used to. This helps in the proper assessment of the security of a computer system and also lowers the number of false positives. Security consultants and system administrators use this to discover holes before an intruder can find them.

The process of conducting vulnerability assessment involves:

- Information gathering.
- Vulnerability scanning.
- Report.

Information gathering involves the gathering of information about the target that will be helpful in the next phase which is Vulnerability Scanning. ‘It consists of collecting all possible information about the target to help perform a thorough security evaluation’ (Ramili, 2012).

Activities that are done during this phase are:

- Manual exploration of websites.
- Crawling of websites for missed or hidden content .
- Checking of metafiles for information leakage files that expose content, such as robots.txt, sitemap.xml, .DS_Store .
- Using major search engines to search for information about the target websites.
- Searching for default installation files.
- Web Application Fingerprinting.
- Identification of ports (open,closed and filtered ports)
- Identification of IP addresses.

Information gathering is followed by vulnerability scanning. This is the phase where weaknesses, flaws, vulnerabilities of the target system are identified using information gathered from earlier activities. Manual and automated methods are used in this phase to identify vulnerabilities in the target system. Vulnerabilities such as SQL injection,XSS,Buffer overflow are identified during this phase .It involves the use of both manual and automated techniques.

After the testing, it is mandatory that report be given to the customer. The report should include all findings as well as solutions (if any problem or weakness was found).

Recommendations about security management should also be given in the report.

2.8 Importance of vulnerability assessment

- It helps in the identification of vulnerabilities and threats.
- Information obtained can be used to improve security of computer systems.
- It also helps to know the architecture of the computer system.
- Security breach can be expensive. Vulnerability assessment helps prevent such breaches through identification of security flaws.

CHAPTER 3

METHODOLOGY

Vulnerability assessment was done on five (5) web sites. These web sites belonged to different institutions. Due to security reasons the web site urls ,IP addresses as well as names of the owners of the websites were omitted in this study. Instead aliases were used. The web servers were given numbers one (1) to five (5) as their names. The Assessment was carried out in 4 phases which included

- Planning
- Information Gathering Vulnerability scanning

- Analysis and Reporting.

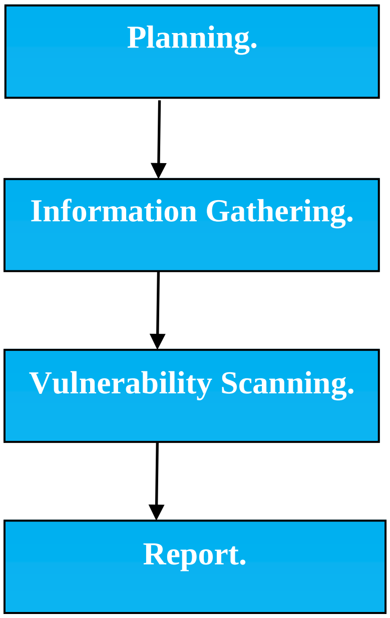


Fig. 3.1: Vulnerability assessment methodology

3.1 Planning

This phase consists of all activities that will needed to be performed before the actual vulnerability assessment is performed. In the planning phase, the scope and objectives for the activity is defined. It also involves getting Management Approvals, signing of documents and agreements, selection of tools as well as preparation of testing machines. Also the testing team prepares a strategy for the performance of the assessment based on security

policies of the requesting organization, industry standards, best practices, etc..

To perform the vulnerability assessment approvals were obtained from the respective web management in order to perform the vulnerability assessment on the selected websites.

Tools that were selected and used for the vulnerability assessment were: □

Oracle Virtual Box

- Nmap.
- Nikto.
- Nessus.

Oracle VM VirtualBox is virtualization software that allows a user to run multiple operating systems on a computer at the same time. It runs on Windows, Mac OS X, Linux and Oracle Solaris systems and is ideal for testing, developing and deploying solutions across multiple platforms on one machine. Oracle VirtualBox is also a high performance software and supports up to 32 virtual CPUs (https://www.virtualbox.org/).

NMAP (”Network Mapper”) is a free and open source utility for network scanning and security auditing. Many security professionals and network administrators use it for network inventory, managing service upgrade schedules, and monitoring host or service uptime. NMAP utilizes raw IP packets as a part of novel approaches to figure out what hosts are accessible on the network, which web services are being run on the server, what operating systems (and OS versions) they are running, what kind of packet filters/firewalls are being used, and many different security tasks. It can be used to rapidly scan large networks as well as single hosts. For beginners NMAP suite incorporates a propelled GUI and results viewer (Zenmap) for easy usage. It also has a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping). Nmap runs on several operating systems including windows and linux (https://nmap.org).

Nessus is a remote security scanning tool, which performs scans on a computer. When it discovers a vulnerability it raises an alert. Nessus performs over 1200 checks on a given computer, testing to see if any of these attacks could be used to break into the computer or otherwise harm it. The

Nessus tool works a little differently than other scanners. Rather than purporting to offer a single, all-encompassing vulnerability database that gets updated regularly, Nessus package includes the Nessus Attack Scripting Language (NASL), which allows security professionals to use a simple language to describe individual attacks. Network administrators can therefore use the NASL to develop their own customized scans. Nessus packages are available for windows, linux and mac users (http://www.cs.cmu.edu/ ~dwendlan/personal/nessus.html).

Nikto is an Open Source web server scanner which performs over 6000 tests against web servers for known security vulnerabilities and mis-configurations. Aside web servers, it can also be used to scan virtual hosts and websites. It can be used to identify potentially dangerous files, programs or scripts. Because it is not a stealthy tool it is usually captured in log files making it a suitable tool for checking how effective an intrusion detection system is (https://cirt.net/Nikto2). Some major features of Nikto include :

- SSL Support (Unix with OpenSSL or maybe Windows with ActiveState's Perl/NetSSL)
- Full HTTP proxy support
- Checks for outdated server components
- Save reports in plain text, XML, HTML, NBE or CSV
- Template engine to easily customize reports
- Scan multiple ports on a server, or multiple servers via input file (including nmap output)
- Easily updated via command line

The testing machine to be used was also set up in this phase and involved the following :

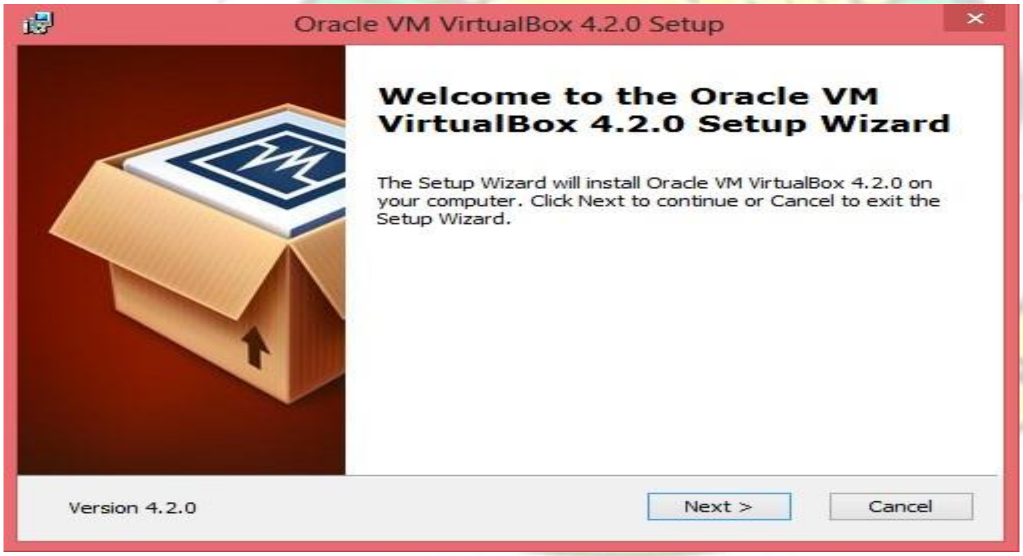
- Download and installation of Oracle Virtual box.
- Download and Installation of the Kali Linux 1.0.9a (64 bit) in a virtual environment.
- Installation and configuration of tools that were used for the vulnerability assessment.

The testing machine was a Toshiba Satellite C55-A Laptop with the following specifications

- Processor: Intel DuoCore @ 1.8GHz
- Installed RAM: 4 GB
- System type: 64 bit
- Hard disk capacity: 500 GB

3.1.1 Oracle Virtual box installation and virtual environment creation

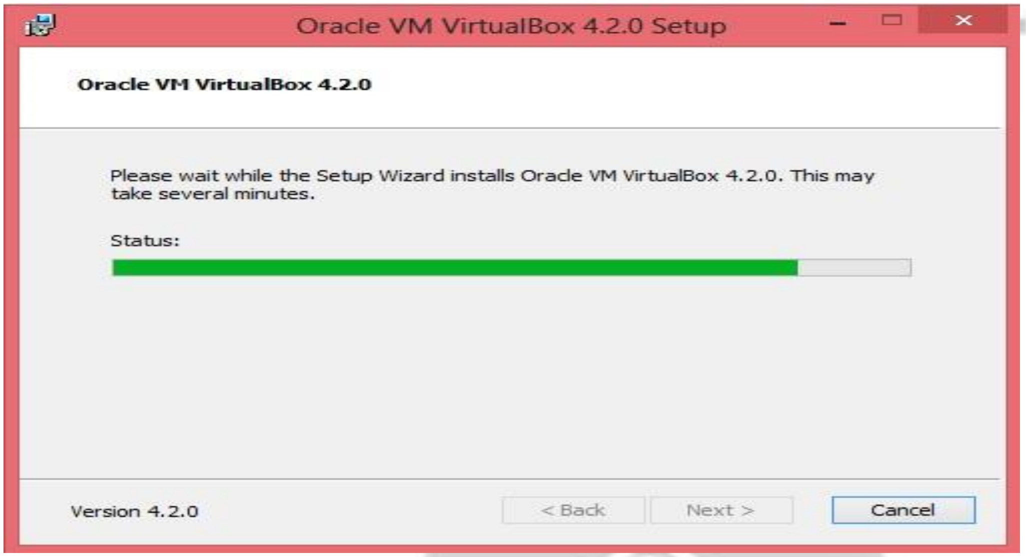
To create the virtual environment, Oracle Virtual box 4.2 was first downloaded from the official website *https://www.virtualbox.org/wiki/Downloads.* The download package was *VirtualBox 4.2.0-80737-Win.exe* . This was then installed on the testing machine after which the virtual environment was created as shown in fig. 3.2, 3.3 ,3.4 , 3.5 and 3.6.



Welcome to the Oracle VM VirtualBox 4.2.0 Setup Wizard

The Setup Wizard will install Oracle VM VirtualBox 4.2.0 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

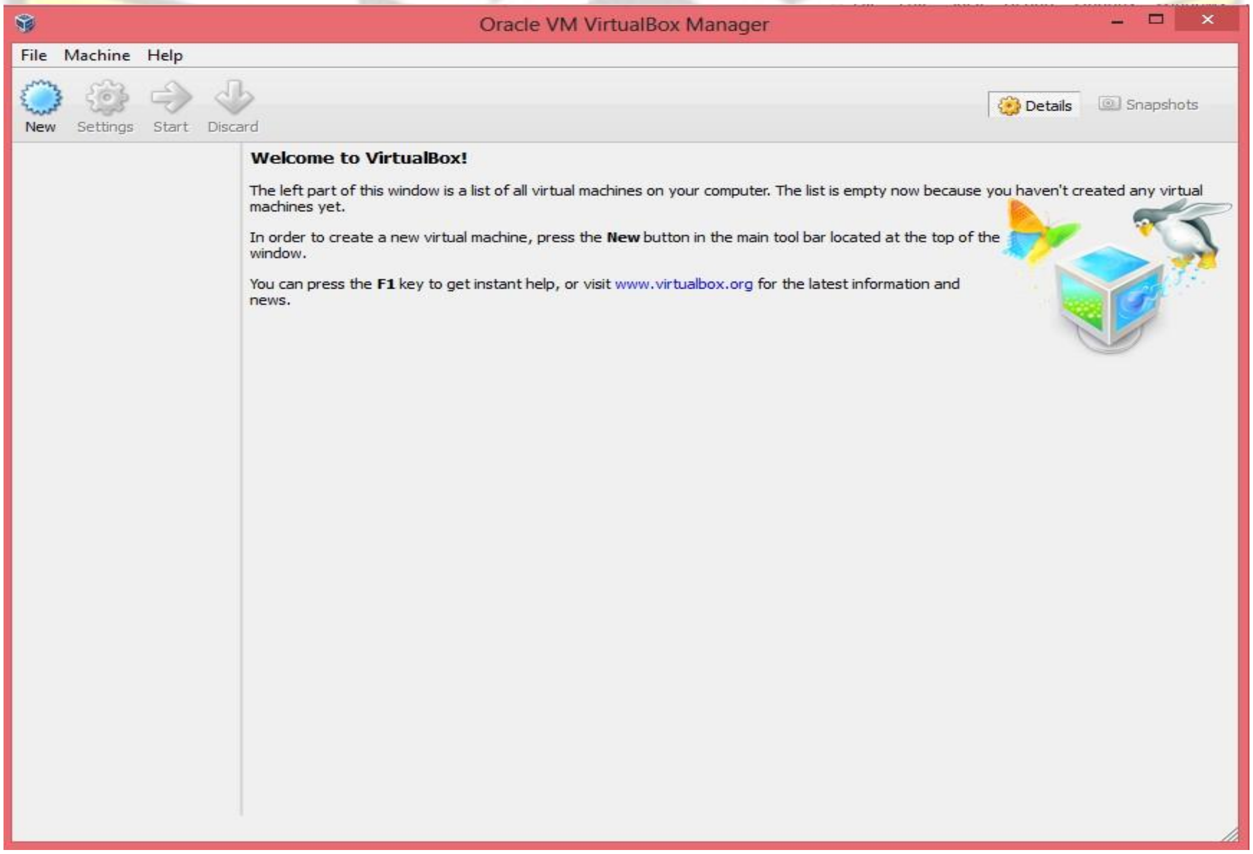
Fig. 3.2: Installing Oracle Virtual box



Please wait while the Setup Wizard installs Oracle VM VirtualBox 4.2.0. This may take several minutes.

Status:

Fig 3.3: Oracle Virtualbox Installation Progress



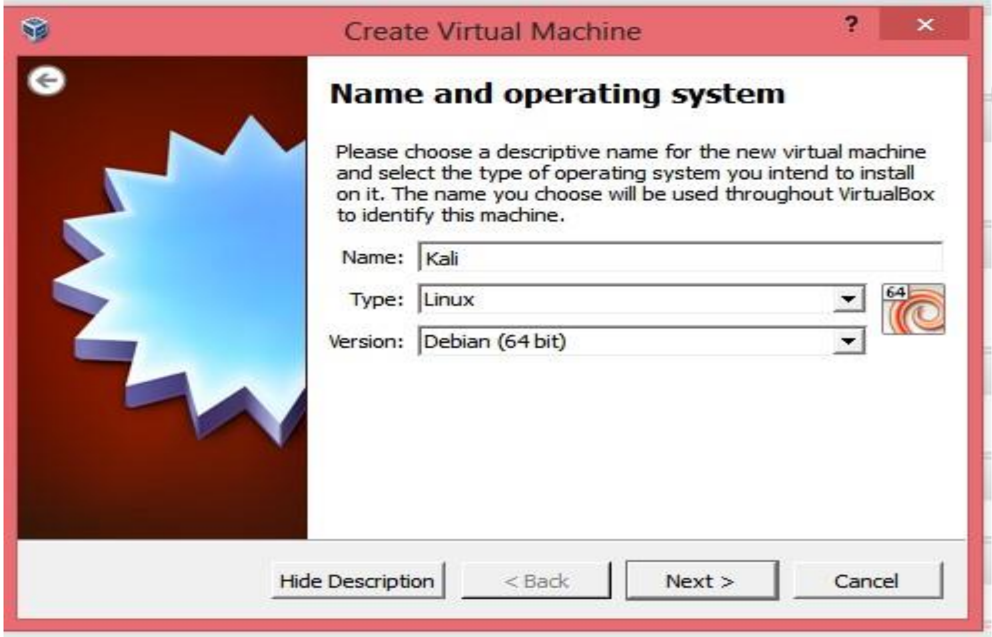
Welcome to VirtualBox!

The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for the latest information and news.

Fig. 3.4: Oracle Virtualbox Interface



Name and operating system

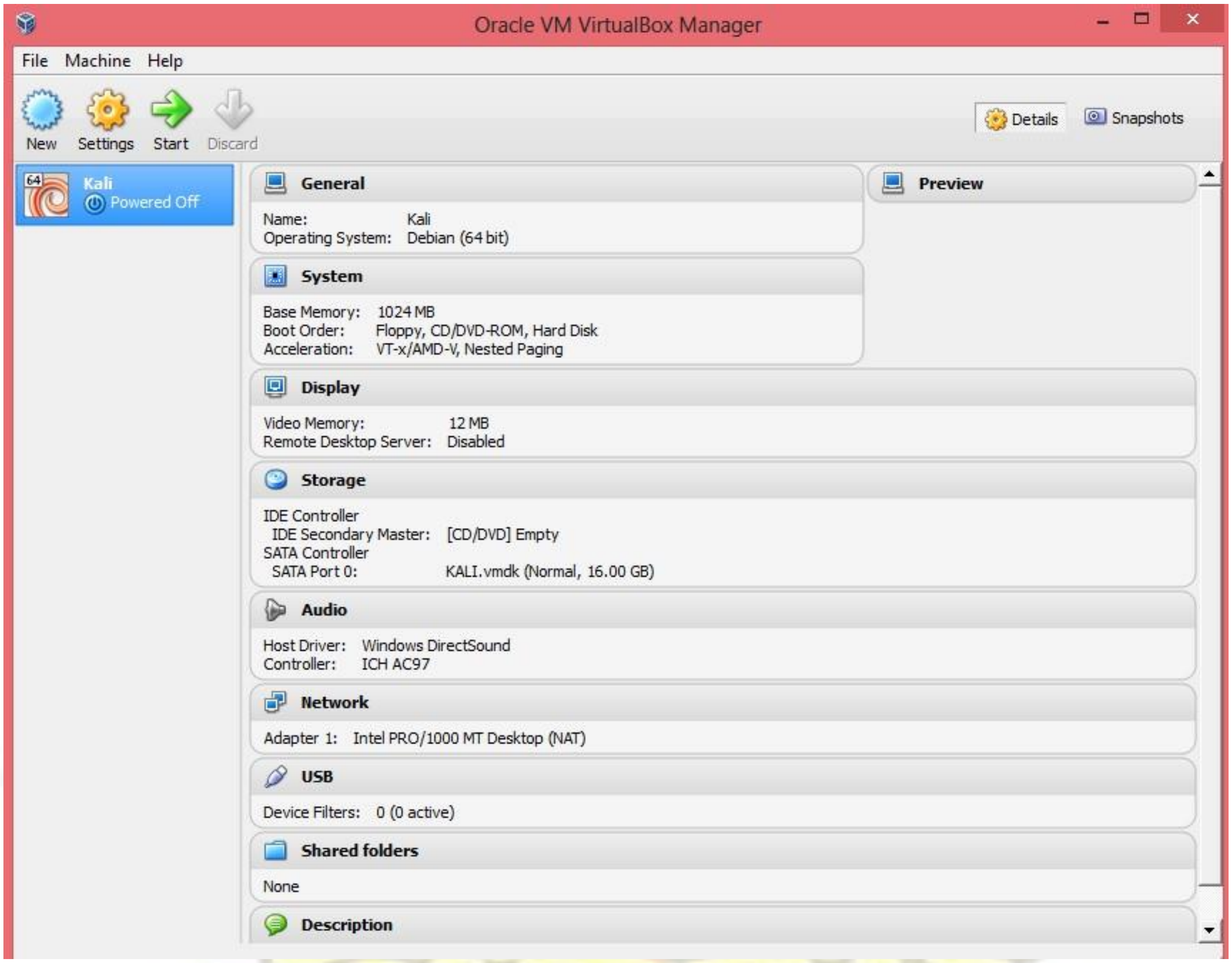
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Kali

Type: Linux

Version: Debian (64 bit)

Fig. 3.5: Creating the virtual environment using virtual box



General

Name: Kali
Operating System: Debian (64 bit)

System

Base Memory: 1024 MB
Boot Order: Floppy, CD/DVD-ROM, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

Display

Video Memory: 12 MB
Remote Desktop Server: Disabled

Storage

IDE Controller
 IDE Secondary Master: [CD/DVD] Empty
SATA Controller
 SATA Port 0: KALI.vmdk (Normal, 16.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

Device Filters: 0 (0 active)

Shared folders

None

Description

Fig. 3.6 : Successful Creation of Virtual Environment

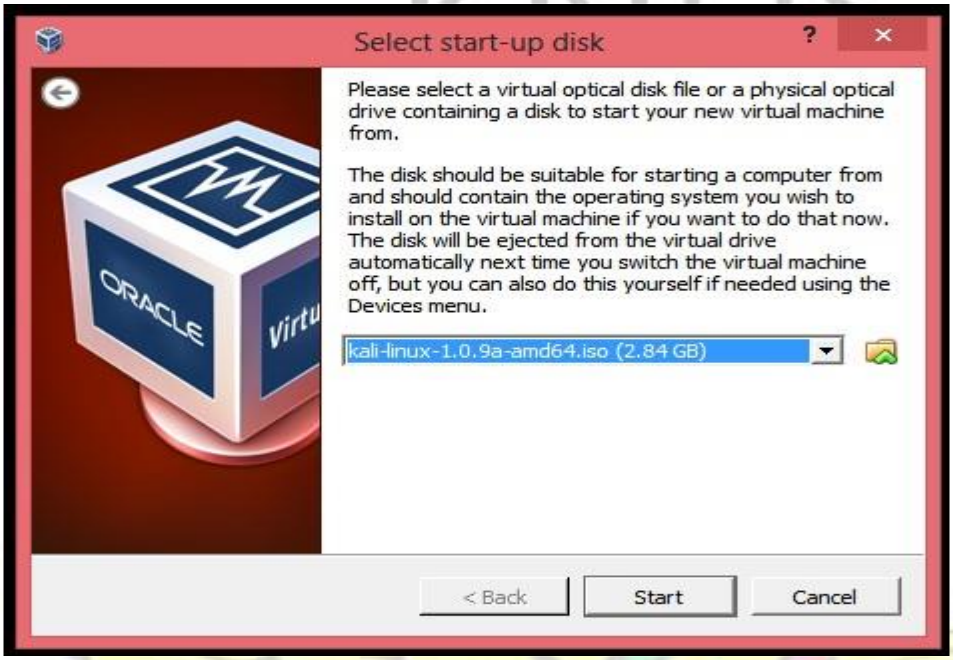
3.1.2 Kali Linux Installation

Kali Linux is a Linux distribution based on the Debian Linux. It was developed by Offensive Security which is a Computer Security firm.

Kali Linux comes with over 600 penetration tools making it an ideal candidate for performing vulnerability assessment as well as other website security tasks.

In this research Kali Linux ISO image was downloaded from the official website *https://www.kali.org/downloads* and installed on the virtual environment which was created using

Oracle Virtual box as shown in fig 3.7,3.8,3.9 and 3.10. The download package was *kalilinux-1.0.9a-amd64.iso*.

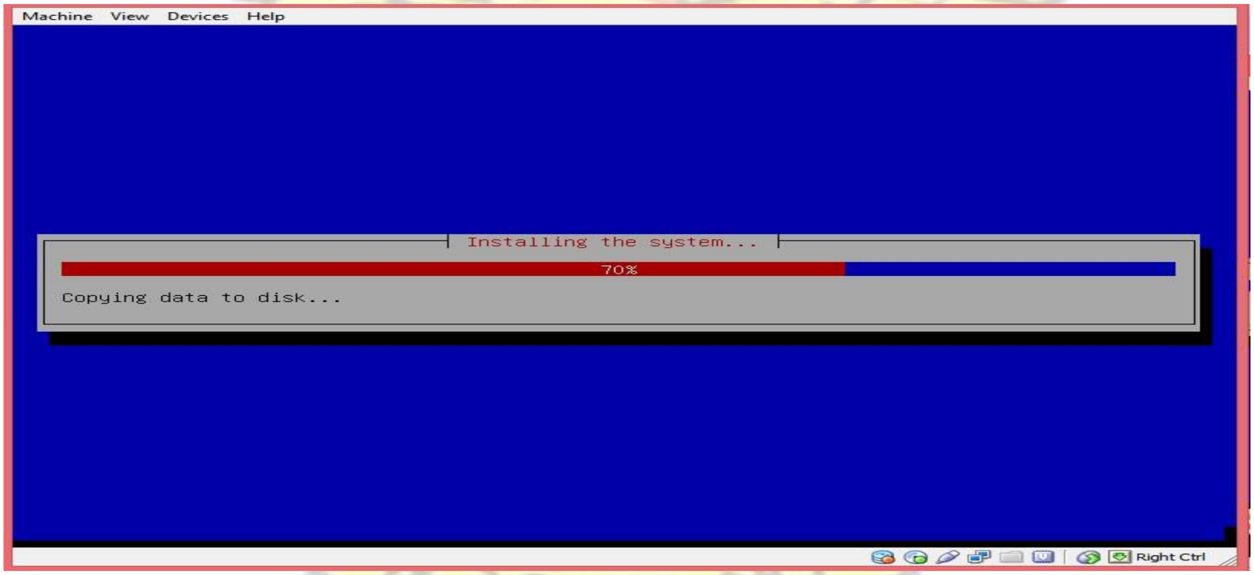


Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

kali-linux-1.0.9a-amd64.iso (2.84 GB)

Fig: 3.7: Selection of Kali Linux ISO image to be installed on the virtual environment



```
Installing the system...
70%
Copying data to disk...
```

Fig. 3.8: Installing Kali Linux on Virtual environment

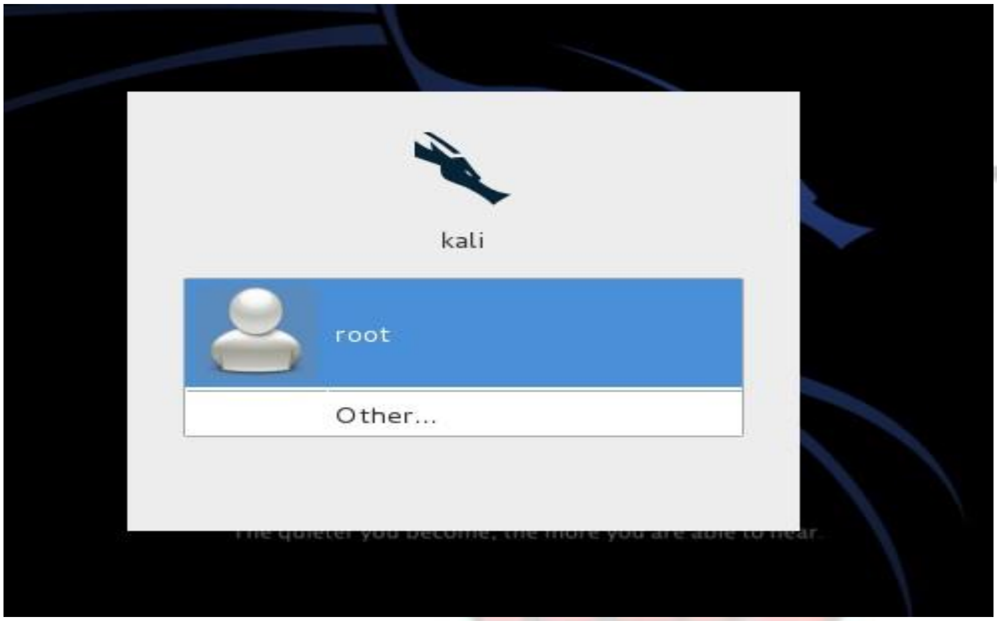


Fig. 3.9: Kali Linux Login Interface

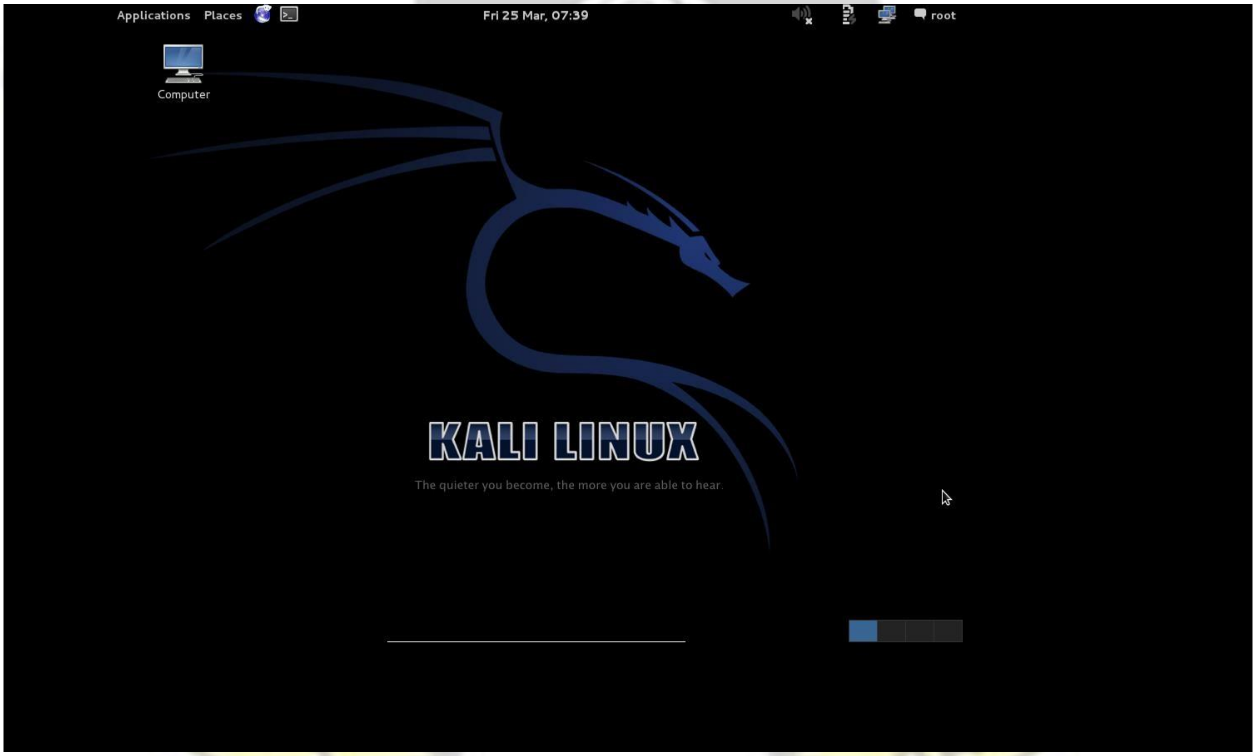


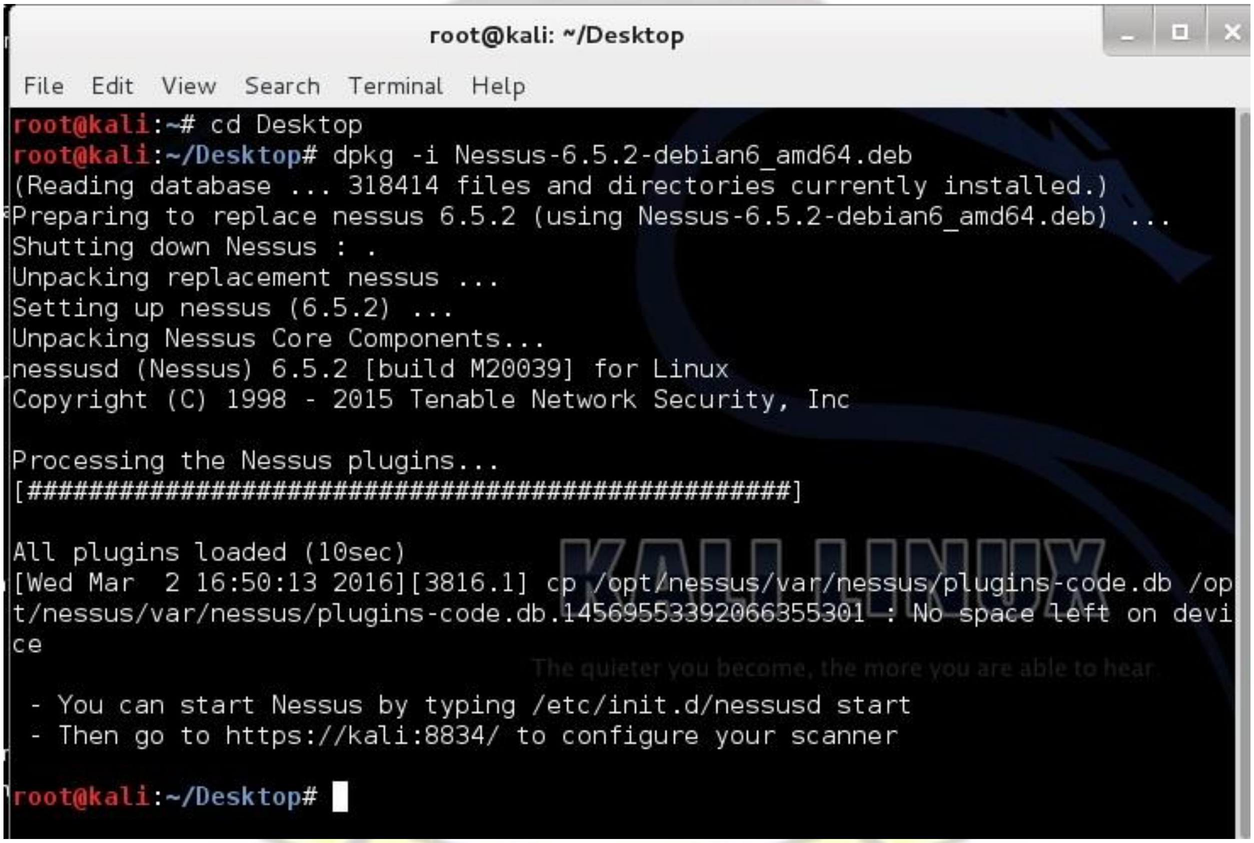
Fig 3.10 : Kali Linux interface after successful login

3.1.3 Installation and configuration of vulnerability scanning Tools

Nmap and Nikto were bundled together with the Kali Linux so no additional installation was needed. However Nessus was not installed.

Nessus was downloaded from the official website and installed on the Kali Linux.

To install the Nessus package, a terminal window was opened and the command *dpkg -i Nessus-6.5.2-debian6_amd64.deb* was used to install the package onto the operating system. This is shown in fig. 3.11



```
root@kali:~# cd Desktop
root@kali:~/Desktop# dpkg -i Nessus-6.5.2-debian6_amd64.deb
(Reading database ... 318414 files and directories currently installed.)
Preparing to replace nessus 6.5.2 (using Nessus-6.5.2-debian6_amd64.deb) ...
Shutting down Nessus : .
Unpacking replacement nessus ...
Setting up nessus (6.5.2) ...
Unpacking Nessus Core Components...
nessusd (Nessus) 6.5.2 [build M20039] for Linux
Copyright (C) 1998 - 2015 Tenable Network Security, Inc

Processing the Nessus plugins...
[##################################################]

All plugins loaded (10sec)
[Wed Mar  2 16:50:13 2016][3816.1] cp /opt/nessus/var/nessus/plugins-code.db /op
t/nessus/var/nessus/plugins-code.db.1456955339206635530l : No space left on devi
ce

 - You can start Nessus by typing /etc/init.d/nessusd start
 - Then go to https://kali:8834/ to configure your scanner

root@kali:~/Desktop# 
```

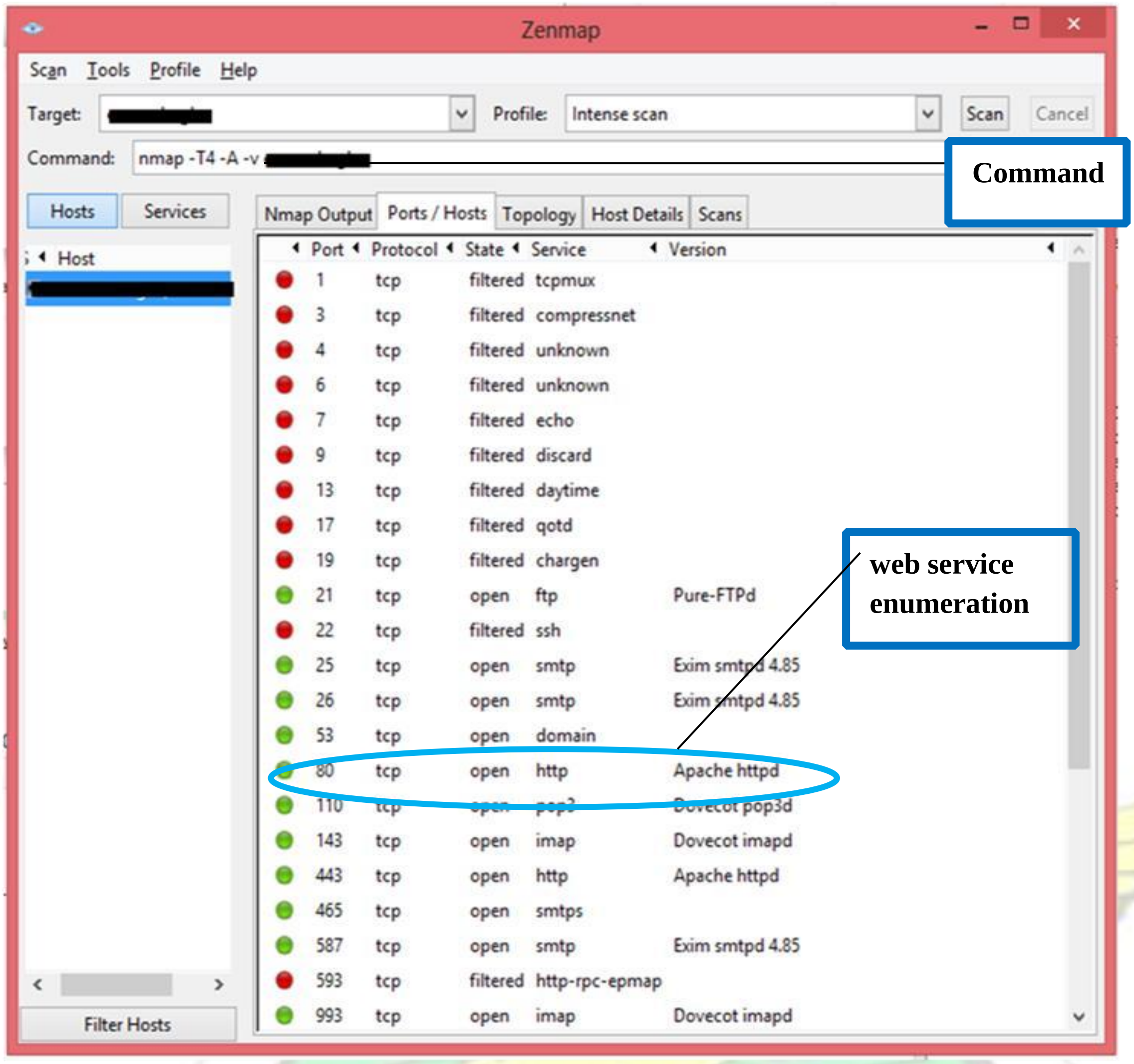
Fig. 3.11: Nessus Installation

3.2 Information Gathering

The aim of this activity was to gather as much information as possible about the target websites which will be helpful in the finding of vulnerabilities. Nmap was used in this phase. Nmap was used for IP address identification, port scanning and web application fingerprinting.

Port scanning comprises of techniques that used to probe web servers to identify states of ports as well as the types of services running on the identified ports. Web application fingerprinting is the process of identifying the version of web server as well the versions of web applications being run on the web server.

The GUI version of Nmap known as Zenmap was used to perform information gathering tasks. Zenmap was designed to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users. Zenmap includes a command creator that allows commands to be created and executed. Scans as well as custom scripts can also be created and saved as profiles to make them easy to run repeatedly. Results from the scanning activities are displayed when scanning is completed. Saved scans can be compared with one another to see how they differ. The results of recent scans are stored in a database and can be retrieved with ease. fig 3.12 shows an information gathering scan. Sensitive information such as IP addresses and web addresses have been filtered due to security reasons. Results from the information gathering can be found in table 3.1 – 3.9.



Target: [redacted]

Profile: Intense scan

Command: nmap -T4 -A -v [redacted]

Host

[redacted]

	Port	Protocol	State	Service	Version
🔴	1	tcp	filtered	tcpmux	
🔴	3	tcp	filtered	compressnet	
🔴	4	tcp	filtered	unknown	
🔴	6	tcp	filtered	unknown	
🔴	7	tcp	filtered	echo	
🔴	9	tcp	filtered	discard	
🔴	13	tcp	filtered	daytime	
🔴	17	tcp	filtered	qotd	
🔴	19	tcp	filtered	chargen	
🟢	21	tcp	open	ftp	Pure-FTPd
🔴	22	tcp	filtered	ssh	
🟢	25	tcp	open	smtp	Exim smtpd 4.85
🟢	26	tcp	open	smtp	Exim smtpd 4.85
🟢	53	tcp	open	domain	
🟢	80	tcp	open	http	Apache httpd
🟢	110	tcp	open	pop3	Dovecot pop3d
🟢	143	tcp	open	imap	Dovecot imapd
🟢	443	tcp	open	http	Apache httpd
🟢	465	tcp	open	smtps	
🟢	587	tcp	open	smtp	Exim smtpd 4.85
🔴	593	tcp	filtered	http-rpc-epmap	
🟢	993	tcp	open	imap	Dovecot imapd

Command

web service enumeration

Fig. 3.12: Nmap information gathering output

Web host	IP address

1	***.***.53.162
2	***.***.80.197
3	***.***.58.115
4	***.***.27.33
5	***.***.181.70

Table 3.1: IP Addresses of the Scanned Websites

Web host	No. of open ports	No. of filtered ports	Closed
1	3	995	2
2	3	997	0
3	14	18	968
4	6	994	0
5	14	1	985

Table 3.2 Port scanning summaries

Port	Protocol	State	Service	Version	OS

53	tcp	open	Domain	ISC BIND Not Disclosed	Linux 3.2-3.6
80	tcp	open	http	Apache httpd	
113	tcp	closed	ident		
443	tcp	open	http	Apache httpd	
3690	tcp	closed	svn		

Table 3.3: Information gathering details for Web host 1

Port	Protocol	State	Service	Version	OS
22	tcp	open	ssh	OpenSSH 5.3(protocol 2.0)	Linux 3.1.9
80	tcp	open	http	Apache httpd 2.4.16 ((Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4)	
443	tcp	open	http	Apache httpd 2.4.16 ((Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4)	

Table 3.4: Information gathering summary for Web host 2

Port	Protocol	State	Service	Version	OS

1	tcp	filtered	tcpmux		
3	tcp	filtered	compressnet		
4	tcp	filtered	unknown		
6	tcp	filtered	unknown		
7	tcp	filtered	echo		
9	tcp	filtered	discard		
13	tcp	filtered	daytime		
17	tcp	filtered	qotd		
19	tcp	filtered	chargen		
21	tcp	open	ftp	Pure-FTPd	
22	tcp	filtered	ssh		
25	tcp	open	smtp	Exim smtpd 4.85	
26	tcp	open	smtp		

Table 3.5: Information gathering summary for Web host 3

53	tcp tcp	open	domain		
80	tcp tcp	open	http pop3	Apache httpd	
110	tcp tcp	open	imap	Dovecot pop3d	
143	tcp tcp	open	http smtps	Dovecot imapd	
443		open	smtp http-	Apache httpd	
465	tcp tcp tcp tcp	open	rpcepmap		
587	tcp tcp tcp tcp	open	imap pop3	Exim smtpd 4.85	
593	tcp tcp tcp	filtered	pvuniwien EtherNet/IP-		
993		open	1 mysql dec-	Dovecot imapd	
995		open	notes	Dovecot pop3d	
1081		filtered	unknown		
2222		filtered	unknown		
3306		open	http	MySQL 5.5.42-37.1	
3333		filtered	unknown unknown		
5915		filtered			
6692		filtered			
8080		open		Apache httpd	
9575		filtered filtered			
49165					

Table 3.6: Information gathering summary for Web host 3

Port	Protocol	State	Service	Version	OS
21 25 80 443 1032 1248	tcp tcp tcp tcp tcp tcp	open open open open open open	ftp smtp http https iad3 hermes	Microsoft ESMTP 7.0.6002.18264	Microsoft Windows

Table 3.7: Information gathering summary for Web host 4

Port	Protocol	State	Service	Version	OS
21	tcp	open	ftp	Pure-FTPd	OpenBSD 4.0
25	tcp	open	smtp	Exim smtpd 4.85	
53	tcp	open	domain		
80	tcp	open	http	Apache httpd 2.2.31((Unix) mod_ssl/2.2.31 OpenSSL/1.0.1e-fips mod_bwlimited/1.4)	
110	tcp	open	pop3	Dovecot pop3d Dovecot	
143	tcp	open	imap	imapd	
406	tcp	filtered	imsp		
443	tcp	open	http	Apache httpd 2.2.31((Unix) mod_ssl/2.2.31 OpenSSL/1.0.1e-fips mod_bwlimited/1.4)	

Table 3.8: Information gathering summary for Web host 5

465	tcp tcp	open	smtp	Exim smtpd 4.85	
587	tcp tcp	open	smtp	Exim smtpd 4.85	
993	tcp	open	imap	Dovecot imap	
995		open	pop3 ssh	Dovecot pop3d	
2382	tcp tcp	open		OpenSSH 5.3	
			smtp mysql	(protocol 2.0)	
2525		open open		Exim smtpd 4.85	
3306				MySQL 5.5.42-cll	

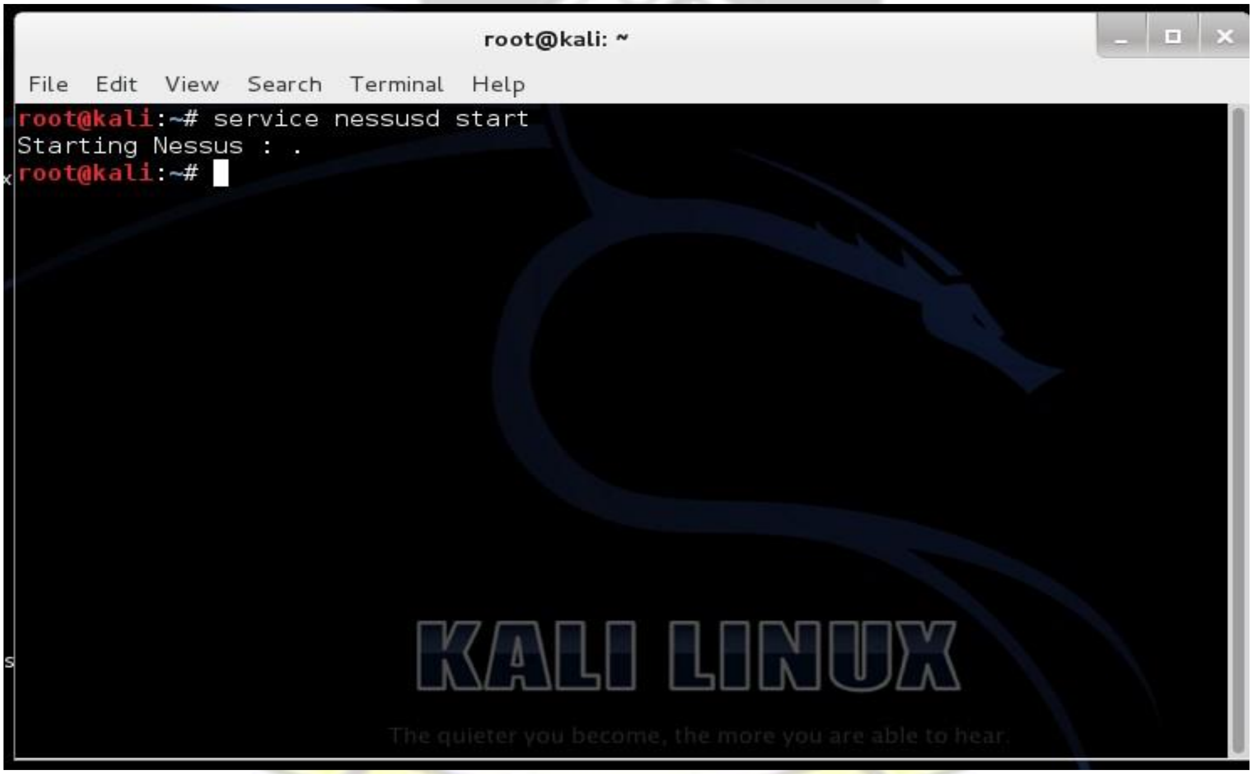
Table 3.9: Information gathering summary for web host 5

3.3 Vulnerability Scanning

This was done to identify vulnerabilities and weakness in the selected websites.

Results from the information gathering was used to initiate the vulnerability scan of the web hosts. Nessus and Nikto were used for the vulnerability scanning on all the 5 web hosts. Testing was done to identify vulnerabilities as suggested by the OWASP Guidelines. The identified vulnerabilities are given in table 3.10 and 3.11.

To start ,Nessus , a terminal was opened and the command *service nessusd start w*as used to start the Nessus service. This is shown in fig 3.13


root@kali: ~
File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to hear.

```
root@kali:~# service nessusd start
Starting Nessus : .
root@kali:~# 
```

Fig. 3.13: Starting Nessus

To access the Nessus web user interface which was used for the vulnerability scanning, the web address 127.0.0.1 was entered onto a web browser. After which user credentials were supplied to gain access as shown in the fig. 3.14. After gaining access, vulnerabilities scanning was performed by selecting plugins as well as other scanning options as shown in fig. 3.15

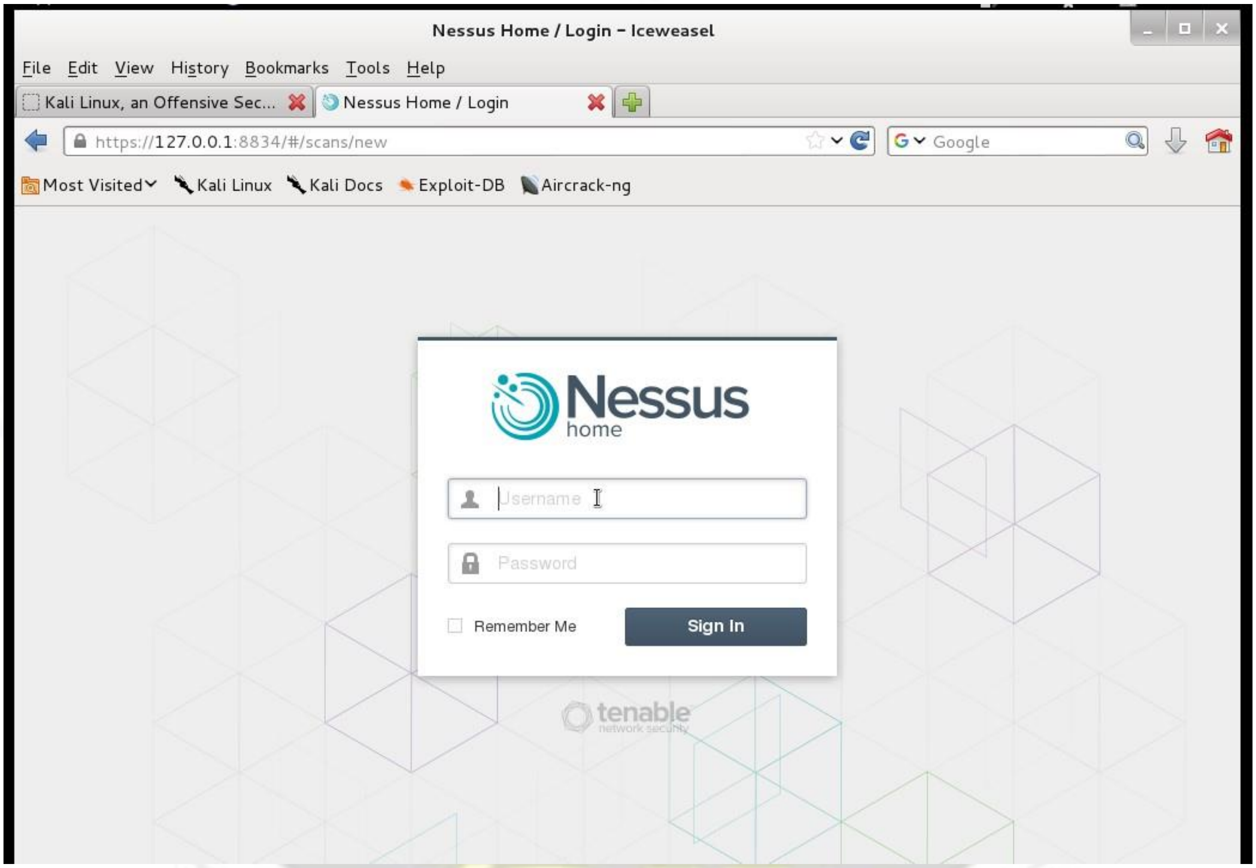
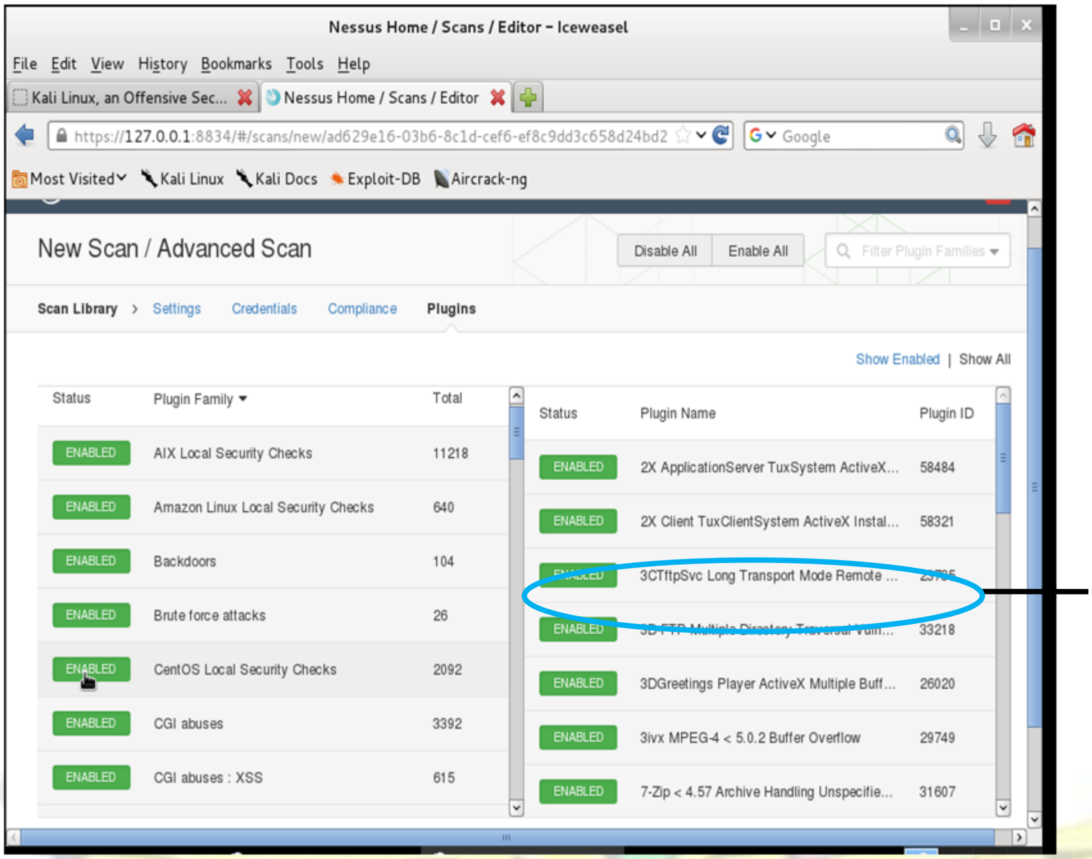


Fig. 3.14: Nessus Web User Interface Login


Nessus Home / Scans / Editor – Iceweasel
File Edit View History Bookmarks Tools Help
Kali Linux, an Offensive Sec...
Nessus Home / Scans / Editor
https://127.0.0.1:8834/#/scans/new/ad629e16-03b6-8c1d-cef6-ef8c9dd3c658d24bd2
Google
Most Visited Kali Linux Kali Docs Exploit-DB Aircrack-ng
Disable All
Enable All
Filter Plugin Families
Scan Library
Settings
Credentials
Compliance
Plugins
Show Enabled | Show All

New Scan / Advanced Scan

Status	Plugin Family	Total
ENABLED	AIX Local Security Checks	11218
ENABLED	Amazon Linux Local Security Checks	640
ENABLED	Backdoors	104
ENABLED	Brute force attacks	26
ENABLED	CentOS Local Security Checks	2092
ENABLED	CGI abuses	3392
ENABLED	CGI abuses : XSS	615

Status	Plugin Name	Plugin ID
ENABLED	2X ApplicationServer TuxSystem ActiveX...	58484
ENABLED	2X Client TuxClientSystem ActiveX Instal...	58321
ENABLED	3CTftpSvc Long Transport Mode Remote ...	23795
ENABLED	3D-FTP Multiple Directory Traversal Vuln...	33218
ENABLED	3DGreetings Player ActiveX Multiple Buff...	26020
ENABLED	3ivx MPEG-4 < 5.0.2 Buffer Overflow	29749
ENABLED	7-Zip < 4.57 Archive Handling Unspecifie...	31607

Plugin

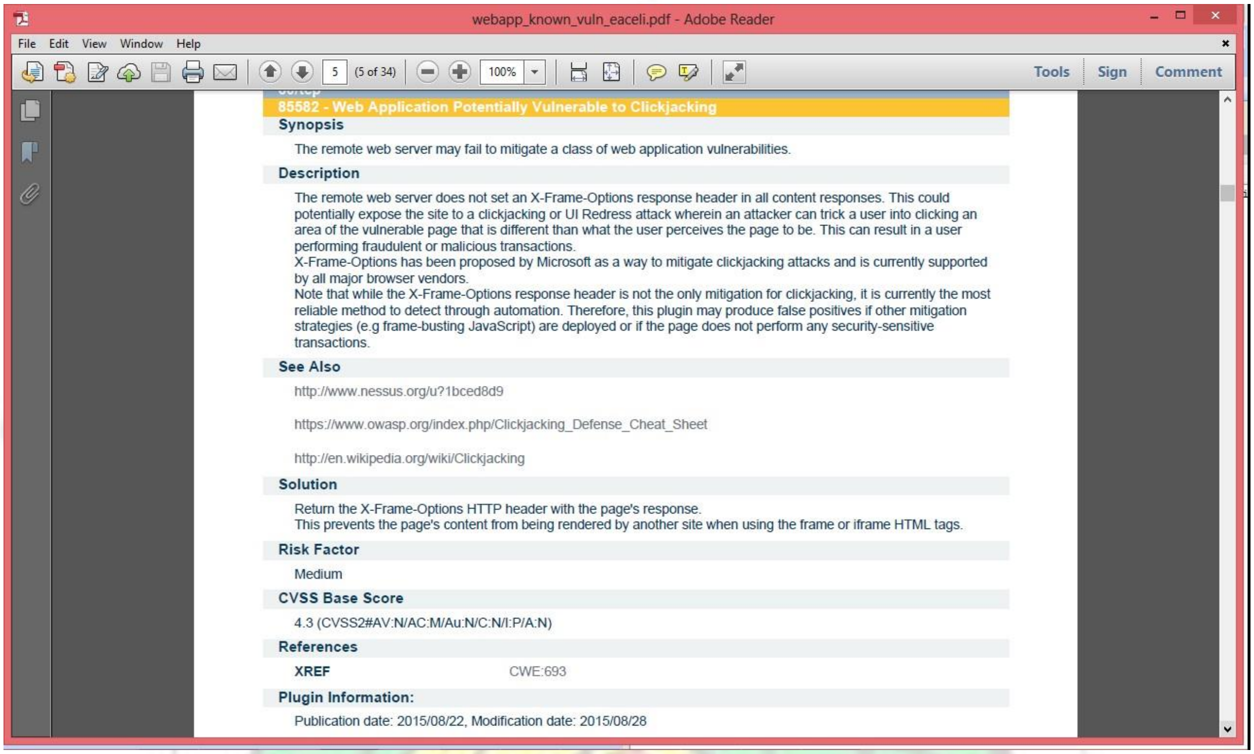
Fig: 3.15: Nessus Scanner Plugins

Reports were generated for all the vulnerability scanning performed for each host. Reports were generated in PDF format. The reported generated included the following :

- **The vulnerability name**: This helps when the tester wants to search for additional information of the vulnerability.
- **Description**: A brief description of the vulnerability is also included as well as how it can be exploited by attackers.
- **Risk factor** : This tells the severity of the identified vulnerabilities and helps in prioritization of solution for the vulnerabilities identified.

- **Reference** : This indicates the vulnerability ID and related vulnerabilities as well as which database it can be found.
- **Plugin:** This indicates which plugin was used to identify the vulnerability.

Fig. 3.16 shows a sample of the report generated by Nessus.


webapp_known_vuln_eaceli.pdf - Adobe Reader
File Edit View Window Help
5 (5 of 34)
100%
Tools
Sign
Comment

85582 - Web Application Potentially Vulnerable to Clickjacking

Synopsis

The remote web server may fail to mitigate a class of web application vulnerabilities.

Description

The remote web server does not set an X-Frame-Options response header in all content responses. This could potentially expose the site to a clickjacking or UI Redress attack wherein an attacker can trick a user into clicking an area of the vulnerable page that is different than what the user perceives the page to be. This can result in a user performing fraudulent or malicious transactions.
X-Frame-Options has been proposed by Microsoft as a way to mitigate clickjacking attacks and is currently supported by all major browser vendors.
Note that while the X-Frame-Options response header is not the only mitigation for clickjacking, it is currently the most reliable method to detect through automation. Therefore, this plugin may produce false positives if other mitigation strategies (e.g frame-busting JavaScript) are deployed or if the page does not perform any security-sensitive transactions.

See Also

http://www.nessus.org/u?1bced8d9

https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet

http://en.wikipedia.org/wiki/Clickjacking

Solution

Return the X-Frame-Options HTTP header with the page's response.
This prevents the page's content from being rendered by another site when using the frame or iframe HTML tags.

Risk Factor

Medium

CVSS Base Score

4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)

References

XREF	CWE:693

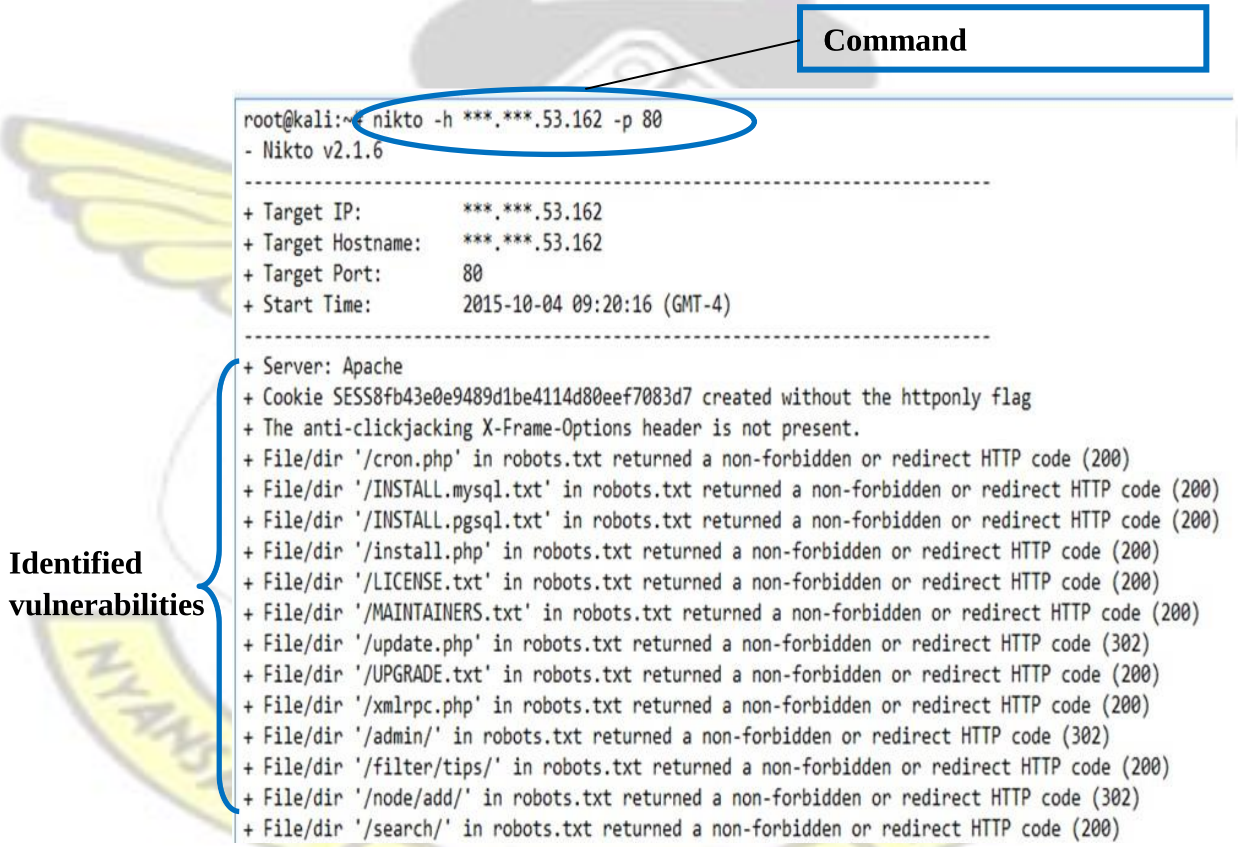
Plugin Information:

Publication date: 2015/08/22, Modification date: 2015/08/28

Fig. 3.16: Nessus scan report

Aside Nessus ,Nikto vulnerability scanner was another tool used in the vulnerability scanning. To initiate the Nikto scanning, a terminal was opened and the following command was given : *Nikto –h IP–p Port – output filename.txt* where *IP* was the ip address of the web host and additional option *Port* was used to specify which port to be scanned. –output command was used to save the scan results in a text file.

Any identified vulnerability or security issue was printed onto the terminal. Fig. 3.17 shows the Nikto vulnerability scanning output of one of the hosts. The output contains the command , target IP, target Hostname, Target Port, Starting time of scanning and the identified vulnerabilities if any. However, unlike Nessus, Nikto does not evaluate the vulnerabilities that are identified. Nikto lists all security issues and does not categorize them like what Nessus reports. So the vulnerabilities identified by Nikto were compared to what was present in the vulnerability database to obtain the general names as well as risk factors and a brief description. For example , in the output in fig. 3.17, a lot of file directory names were discovered in the robots.txt file and all these issues were labelled as a robots.txt information disclosure in vulnerability databases such as OSVDB and CVE.



```
root@kali:~# nikto -h ***.***.53.162 -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          ***.***.53.162
+ Target Hostname:    ***.***.53.162
+ Target Port:        80
+ Start Time:         2015-10-04 09:20:16 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache
+ Cookie SESS8fb43e0e9489d1be4114d80eef7083d7 created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ File/dir '/cron.php' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/INSTALL.mysql.txt' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/INSTALL.pgsql.txt' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/install.php' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/LICENSE.txt' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/MAINTAINERS.txt' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/update.php' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ File/dir '/UPGRADE.txt' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/xmlrpc.php' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/admin/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ File/dir '/filter/tips/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ File/dir '/node/add/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ File/dir '/search/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
```

Fig. 3.17: Nikto Scanner output

3.3.1 Identified Vulnerabilities

In all, 28 vulnerabilities were discovered in the 5 hosts that were scanned. The identified vulnerabilities and the associated hosts are listed in tables 3.10 and 3.11

Vulnerability	Host 1	Host 2	Host 3	Host 4	Host 5
Anonymous FTP Enabled			✓		
Apache HTTP Server User Dir Directive Username Enumeration			✓		
AutoCompletion of Password	✓				
Cleartext Transmission of Sensitive Information	✓			✓	
Clickjacking vulnerability	✓	✓	✓	✓	✓
Cookies without HTTPonly Flag Identified	✓	✓	✓		✓
Cross-Site Scripting Vulnerability			✓		
Directory Indexing Enabled					✓
DOS Amplification Vulnerability			✓		

FTP Bounce Attack Vulnerability			✓		
HTTP TRACE method Enabled.		✓			✓
Logjam vulnerability.	✓				

Table 3.10: Identified Vulnerabilities

Microsoft Internet Information Services (IIS) Flaw.				✓	
Missing HSTS	✓	✓	✓		
MoinMoin Two Unspecified XSS.		✓	✓		
Multiple Web Server Default Page Fingerprinting Weakness	✓		✓		
Multiple Web Server Interesting Web Document	✓		✓		✓
Multiple Web Server robots.txt Remote Information Disclosure	✓				
PHP expose_php Information Disclosure					✓
RC4 Algorithm Invariance-Weakness	✓	✓	✓		

RSA keys less than 2048 bits	✓			✓	
SMTP Service Supports Cleartext Login			✓		✓
SSH Protocol CBC Mode Enabled	✓	✓			✓
SSH weak Mac Algorithm enabled	✓	✓			✓
SSL v2 and SSL v3 Detection	✓		✓		
Untrusted SSL-Certificate	✓				
Usr/doc Directory Information Disclosure			✓		✓
Weak Hashing Algorithms for the Signing of SSL Certificate	✓				

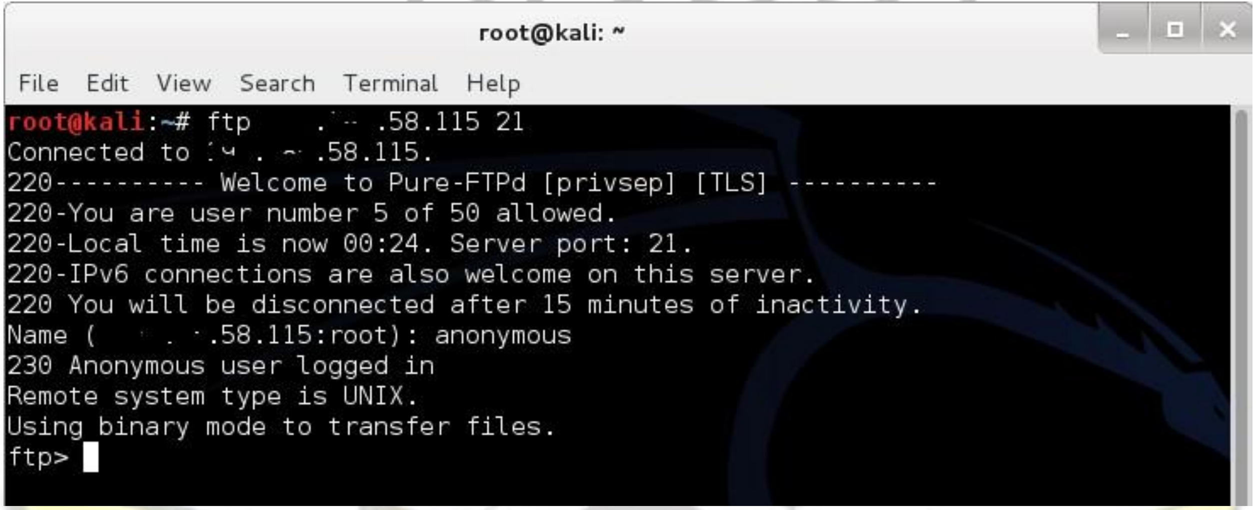
Table 3.11: Identified Vulnerabilities

3.3.1.1 Anonymous FTP Enabled

Severity: Medium

File Transfer Protocol (FTP) is a standard Internet protocol for transmitting files between two or more computers on the Internet over TCP/IP connections. Using FTP, a client can upload, download, delete, rename, move and copy files on a server. A user typically needs to log on to the FTP server, although some servers make some or all of their content available without login, also known as anonymous FTP.

During the information gathering ,host 3 was found to be running an ftp service on port 21. Anonymous FTP has been enabled and login was successful without any user credentials . This is considered risky because any user can connect without providing any credentials. A user can therefore access any file on the ftp server even if the file is intended to be private.



```
root@kali:~# ftp [redacted].58.115 21
Connected to [redacted].58.115.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 5 of 50 allowed.
220-Local time is now 00:24. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name ([redacted].58.115:root): anonymous
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

Fig. 3.18: Successful Ftp login

Solution

- Anonymous ftp login should be disabled.
- All users must be made to enter credentials before logging in to the ftp server.

3.3.1.2 Apache HTTP Server User Dir Directive Username Enumeration

Severity: Medium

Nikto scanner also reported that user enumeration on host 3 is possible by requesting ~username (responds with 'Forbidden' for users, 'not found' for non-existent users). This could be due the User Dir module being enabled. An attacker can leverage this and be able to enumerate valid user names.

Solution

According to http://osvdb.org/637 there is no workaround for this issue. However the user dir can be disabled to mitigate this issue.

3.3.1.3 Auto-Completion of Password

Severity: Info

Scanning revealed that the host 1 web server had at least one HTML form field containing an input type 'password' with auto-complete option not set to off. As a result, browsers of users may save these credentials which might lead to information leakage.

Solution

The HTML form field(s) should have the auto-completion option turned off.

3.3.1.4 Cleartext Transmission of Sensitive Information

Severity: Low

Nessus Scan reports showed several HTML form fields containing input of type 'password' transmitted information to the server in cleartext and therefore nyone eavesdropping the connections may therefore be able to obtain usernames and passwords of users. This might also lead to a compromise of the server if such information should fall in the hands of people with malicious intents. (CWE,2012).

This vulnerability was identified in host 1 and host 4.

Solution

It is recommended that such sensitive information be transmitted over a secure channel such as HTTPS and SSL.

3.3.1.5 Clickjacking vulnerability

Severity: Medium

Clickjacking attacks allow an attacker to render the contents of the webpage in a frame and this can be used to steal sensitive information from users. This vulnerability was discovered in all the five hosts that were scanned.

Nikto and Nessus both discovered the absence of anti-clickjacking x-frame options response header in some of the web pages sent by the respective hosts and therefore the web hosts were vulnerable to click jacking attacks . Details of the vulnerability is given in appendix A.

Solution

The use of X-Frame Options Responder Header will help defend against attacks conducted via this vulnerability. (OWASP, 2015). Depending on the circumstances the web server through the X-frame options response header can use the following activities to prevent clickjacking attacks.

- Denying domains from framing web page content
- Allowing certain sites to frame web content and disallowing others.

The following browser settings can also be used to mitigate clickjacking weakness (CAPEC, 2015) :

- The NoScript plug-in should be used when using Firefox browser to prevent iFrames.
- Turn off JavaScript.
- Turn off Flash
- CSS must be disabled.

3.3.1.6 Cookies without HTTPonly Flag

Severity: Info

HTTPonly is a secure flag that disallow cookies from being seen by sending all communications over HTTPs. Unlike HTTP which sends communication in plaintext, HTTPs communications are encrypted and can therefore not be seen. However cookies without this secure flag even when they are sent over HTTPs can be redirected (via man-in-the-middle attack) to an HTTP channel from which the plaintext can be recovered (InfoSec Institute, 2014).

Without HTTP only flag, a server is vulnerable to an XSS attack(OWASP, 2014).

Cookies without the httponly flag were identified in web host 1,2,3 and 5. Details for the identified cookies is given in appendix B.

Solution

This problem can be mitigated by setting the HTTP flag only on every cookie created by the web server.

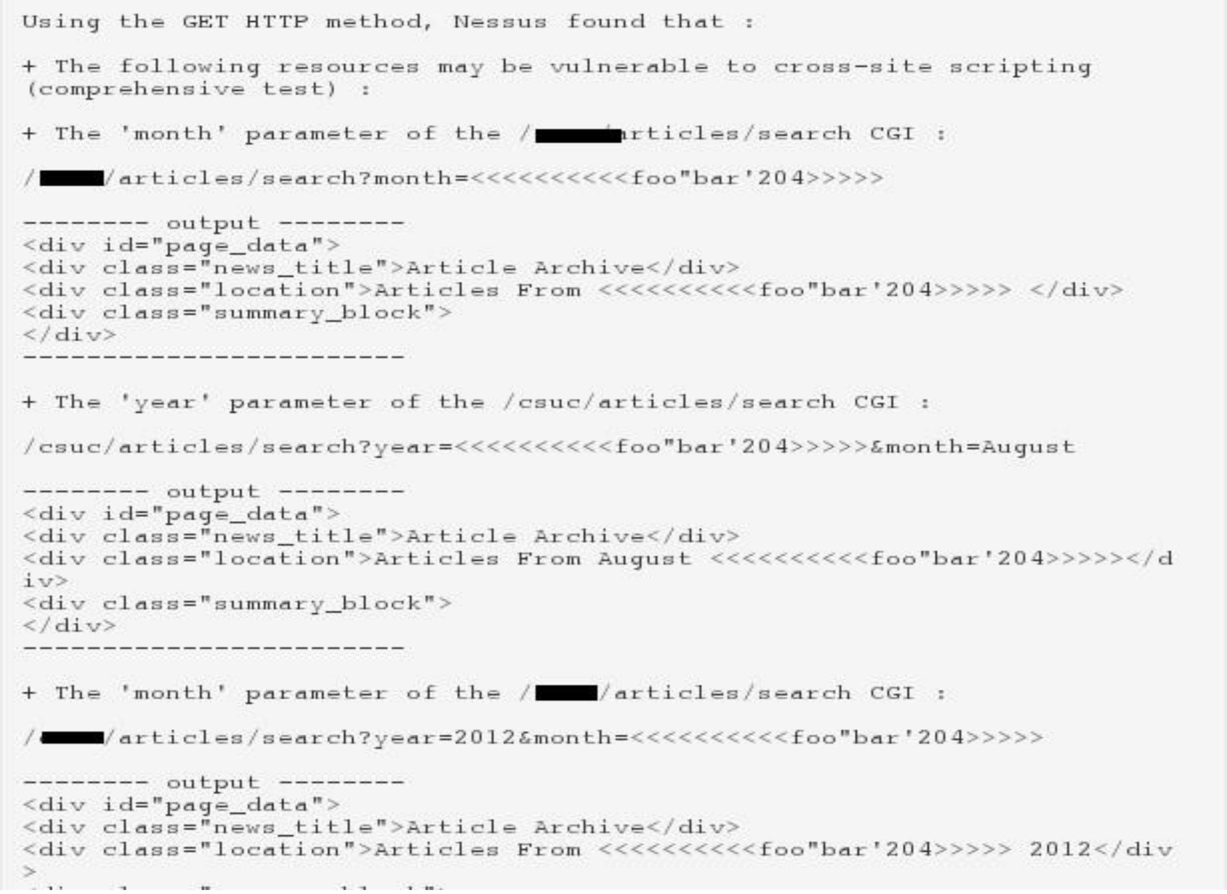
By setting the HTTPOnly flag on a cookie that a web server creates, the cookies cannot be accessed by the client's browser. If the client tries to access it, the browser will return an empty string due to the secure flag on it (OWASP, 2014).

3.3.1.7 Cross-Site Scripting Vulnerability

Severity: Medium

Cross-Site Scripting Vulnerability is a widespread vulnerability that affects many web applications. The danger behind XSS is that it allows an attacker to inject content into a website and modify how it is displayed, forcing a victim's browser to execute the code provided by the attacker while loading the page.

Nessus also reported that the CGI scripts on the host 3 server failed to sanitize malicious JavaScripts properly. Attackers could therefore leverage this to launch a cross-site scripting attacks. Nessus further reported that the XSS vulnerability is likely to be non-persistent or reflected. This is shown in Fig. 3.19



```
Using the GET HTTP method, Nessus found that :

+ The following resources may be vulnerable to cross-site scripting
(comprehensive test) :

+ The 'month' parameter of the /[redacted]rticles/search CGI :

/[redacted]/articles/search?month=<<<<<<<<<<foo"bar'204>>>>>

-------- output --------
<div id="page_data">
<div class="news_title">Article Archive</div>
<div class="location">Articles From <<<<<<<<<<foo"bar'204>>>>> </div>
<div class="summary_block">
</div>
------------------------

+ The 'year' parameter of the /csuc/articles/search CGI :

/csuc/articles/search?year=<<<<<<<<<<foo"bar'204>>>>>&month=August

-------- output --------
<div id="page_data">
<div class="news_title">Article Archive</div>
<div class="location">Articles From August <<<<<<<<<<foo"bar'204>>>>></d
iv>
<div class="summary_block">
</div>
------------------------

+ The 'month' parameter of the /[redacted]/articles/search CGI :

/[redacted]/articles/search?year=2012&month=<<<<<<<<<<foo"bar'204>>>>>

-------- output --------
<div id="page_data">
<div class="news_title">Article Archive</div>
<div class="location">Articles From <<<<<<<<<<foo"bar'204>>>>> 2012</div
>
```

Fig. 3.19: Cross-Site Scripting Vulnerability

Solution

- Access to the vulnerable application must be restricted
- Upgrade the application to a more secure version.
- Remove the vulnerable application.

3.3.1.8 Directory Indexing Enabled

Severity: Info

Directory indexing on a web server may reveal sensitive files or directories. This information can be used by attackers to compromise or access contents of the web server.

Directory index was found for the following on host 5:

- test/
- images/
- images/?pattern=/etc/*&sort=name/
- apps/
- includes/
- Lib/
- temp/
- pdf/

Solution

Directory indexing should be disabled to avoid potential leakage of sensitive information.(http://osvdb.org/3268).

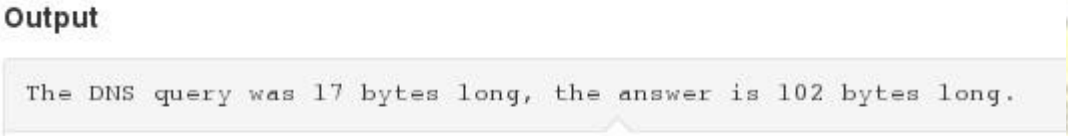
3.3.1.9 DOS Amplification Vulnerability.

Severity: Medium

This occurs when an attacker is able to directs DNS name lookup request to a vulnerable DNS server and generate large DNS response which is then used to flood a target system(usually the source address provided by attacker).

According to the Nessus scanner, the remote DNS server on host 3 was answering to any request. It was therefore possible to query the name servers (NS) of the root zone ('.') and get an answer that is bigger than the original request. These could be due to a default configuration which allows unrestricted access to the DNS server.

By spoofing the source IP address, a remote attacker can take advantage of this to launch a denial of service attack against a third-party host using the remote DNS server. This will be traced back to the web server instead of the attackers address.



Output

```
The DNS query was 17 bytes long, the answer is 102 bytes long.
```

Fig. 3.20: DNS query result

Solution

- The DNS should be configured to reject such queries.

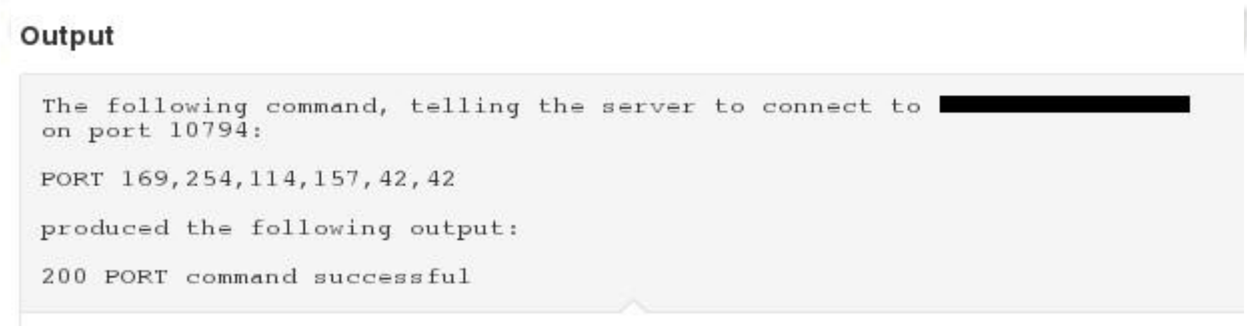
- Access to the DNS server should be restricted.

3.3.1.10 FTP Bounce Attack Vulnerability

Severity: High

This flaw allows a remote attacker to hide behind a network (running the ftp service) and perform port scan on another host via the PORT command making them think the attack comes that network. An attacker could leverage this and cause a denial of service attack to a third party host.

During the vulnerability scanning, Nessus reported that the ftp service being run on host 3 was vulnerable to the ftp bounce attack. Fig. 3.21 shows the proof of concept which was generated by Nessus.



Output

```
The following command, telling the server to connect to [redacted]
on port 10794:

PORT 169,254,114,157,42,42

produced the following output:

200 PORT command successful
```

Fig. 3.21: Port forwarding

Solution

Ensure that the FTP server cannot establish connections to machines other than the originating client.

Computer Emergency Response Team(CERT) advisory on how to deal with this issue can be found here :

- https://www.cert.org/historical/advisories/ca-1997-27.cfm

- http://web.archive.org/web/20131105191347/http://www.cert.org/tech_tips/ftp_port_attacks.html

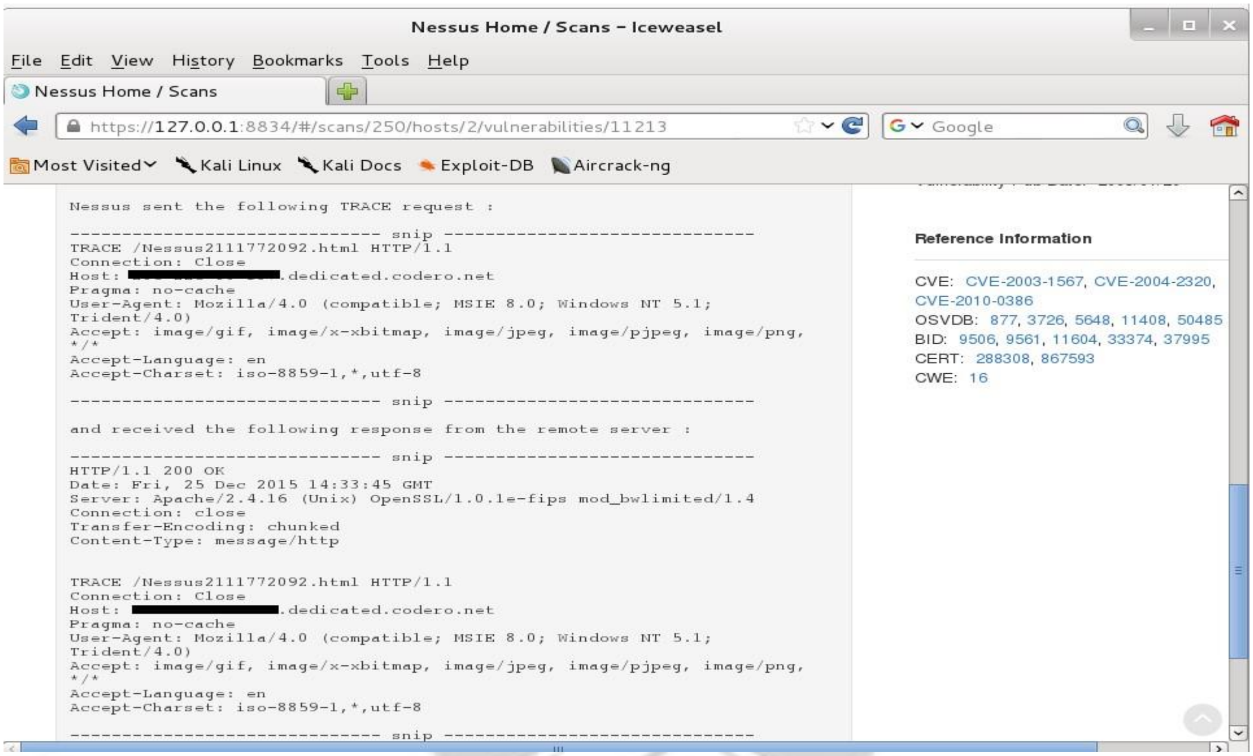
3.3.1.11 HTTP TRACE method Enabled

Severity: Medium

The HTTP TRACE method is normally used for debugging purposes by returning the HTTP request which will contain the entire message back to the client. An attacker with the help of XMLHTTP, ActiveX, or XMLDOM or scripting objects can exploit this to obtain sensitive data such as cookies and headers. Apart from TRACE method, HTTP TRACK method can also result in the same attacks (RAPID7, 2004).

Web host 2 and 5 were discovered to have enabled HTTP TRACE method making them vulnerable to Cross site tracing attack which is a form of cross site scripting attack.(OWASP,2013).

Nessus also included a proof of concept in the report. This is shown in fig 3.22 and 3.23 for host 2 and 5 respectively.

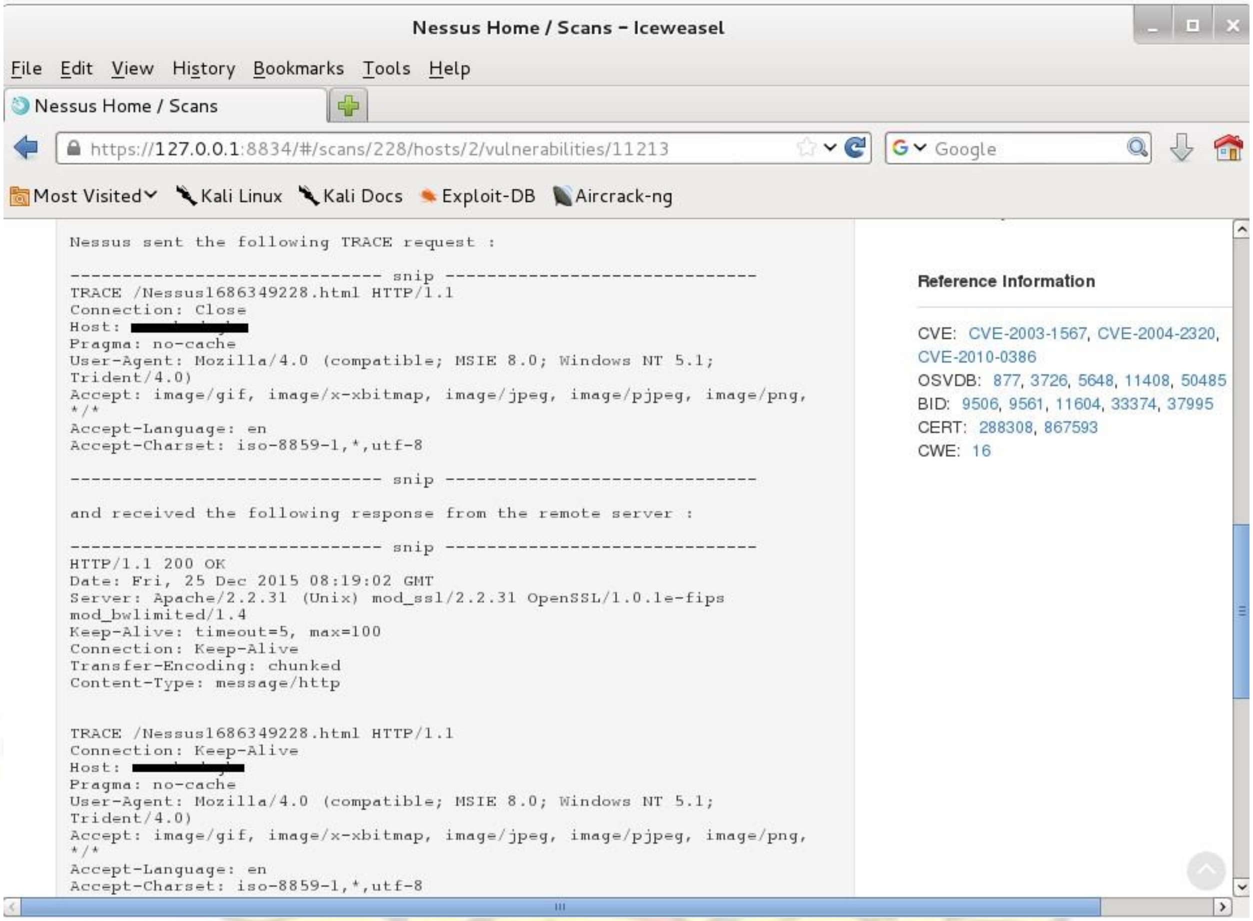

Nessus Home / Scans – Iceweasel
File Edit View History Bookmarks Tools Help
Nessus Home / Scans
https://127.0.0.1:8834/#/scans/250/hosts/2/vulnerabilities/11213
Google
Most Visited
Kali Linux
Kali Docs
Exploit-DB
Aircrack-ng

```
Nessus sent the following TRACE request :

------------------------------ snip ------------------------------
TRACE /Nessus2111772092.html HTTP/1.1
Connection: Close
Host: [redacted].dedicated.codero.net
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png,
*/*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8

------------------------------ snip ------------------------------

and received the following response from the remote server :

------------------------------ snip ------------------------------
HTTP/1.1 200 OK
Date: Fri, 25 Dec 2015 14:33:45 GMT
Server: Apache/2.4.16 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4
Connection: close
Transfer-Encoding: chunked
Content-Type: message/http


TRACE /Nessus2111772092.html HTTP/1.1
Connection: Close
Host: [redacted].dedicated.codero.net
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png,
*/*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8

------------------------------ snip ------------------------------
```

Reference Information

CVE: CVE-2003-1567, CVE-2004-2320, CVE-2010-0386
OSVDB: 877, 3726, 5648, 11408, 50485
BID: 9506, 9561, 11604, 33374, 37995
CERT: 288308, 867593
CWE: 16

Fig. 3.22 : HTTP TRACE method enabled on host 2


Nessus Home / Scans - Iceweasel
File Edit View History Bookmarks Tools Help
Nessus Home / Scans
https://127.0.0.1:8834/#/scans/228/hosts/2/vulnerabilities/11213
Google
Most Visited
Kali Linux
Kali Docs
Exploit-DB
Aircrack-ng

```
Nessus sent the following TRACE request :

------------------------------ snip ------------------------------
TRACE /Nessus1686349228.html HTTP/1.1
Connection: Close
Host: [redacted]
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png,
*/*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8

------------------------------ snip ------------------------------

and received the following response from the remote server :

------------------------------ snip ------------------------------
HTTP/1.1 200 OK
Date: Fri, 25 Dec 2015 08:19:02 GMT
Server: Apache/2.2.31 (Unix) mod_ssl/2.2.31 OpenSSL/1.0.1e-fips
mod_bwlimited/1.4
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: message/http


TRACE /Nessus1686349228.html HTTP/1.1
Connection: Keep-Alive
Host: [redacted]
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png,
*/*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

Reference Information

CVE: CVE-2003-1567, CVE-2004-2320, CVE-2010-0386
OSVDB: 877, 3726, 5648, 11408, 50485
BID: 9506, 9561, 11604, 33374, 37995
CERT: 288308, 867593
CWE: 16

Fig. 3.23 : HTTP TRACE method enabled on host 5

Solution

All HTTP TRACE and TRACK support in web servers must be disabled (CERT., 2003).

Additional solutions can be found at the following websites:

- http://msdn2.microsoft.com/en-us/library/ms533046.aspx
- http://jeremiahgrossman.blogspot.com/2007/04/xst-lives-bypassing-httponly.html
- http://httpd.apache.org/docs/1.3/mod/mod_rewrite.html
- http://www.microsoft.com/technet/security/tools/urlscan.mspx

3.3.1.12 Logjam Vulnerability.

Severity: Medium

This vulnerability allows an attacker to downgrade a Transport Layer Security (TLS) connection to use 512 bit DH export-grade cryptography via man-in-the middle attack, allowing him to read the exchanged data and inject data into the connection.. This happens when a web server allows SSL/TLS connections with Diffie-Hellman moduli less than or equal to 1024 bits. During vulnerability scanning, this vulnerability was discovered in host 1 (Schneier,2015).

Solution

The service must be reconfigured to use a unique Diffie-Hellman moduli of 2048 bits or greater. Also Elliptic-Curve Diffie Hellman Encryption should be use. This provides a stronger protection.

3.3.1.13 Microsoft Internet Information Services (IIS) Flaw

Severity: Medium

IIS (Internet Information Server) is a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft's Windows NT and Windows 2000 Server operating systems.

Web host 4 was found to be using Microsoft IIS 7.0 server. It was found to be outdated (IIS 7.0) and from the Microsoft website, IIS 7.0 has a flaw in the ftp service which makes it possible for an attacker to send specially crafted FTP commands to the server. This could allow information disclosure. (Microsoft,2012). Other vulnerabilities also associated with Microsoft IIS7.0 are given in table 3.12

Name	**ID**
Stack consumption vulnerability	CVE-2010-1899

Privilege escalation vulnerability	CVE-2008-0074
Integer overflow vulnerability	CVE-2008-1446
IIS Authentication Memory Corruption Vulnerability	CVE-2010-1256

Table 3.12: Microsoft IIS 7.0 vulnerabilities

Solution

Security patches should be installed to mitigate this issue.

3.3.1.14 Missing HSTS

Severity: Info

HSTS (HTTP Strict Transport Security) is a security mechanism that ensures that users browse over secure connections (https) even when http links are clicked on. Without HSTS all connections on a web server will be done on http which is considered insecure. (https://https.cio.gov/hsts). Web servers without HSTS are vulnerable to man-in-the-middle attacks and downgrade attacks. Also there is a high possibility of a user overriding the invalid certificate message and the issue of privacy leaks on web servers without HSTS (OWASP, 2015).

During the vulnerability scanning, it was discovered that HTTP Strict Transport Security (HSTS) was missing in host 1, 2 and 3.

Solution

HSTS should be implemented on the web server.

3.3.1.15 MoinMoin Two Unspecified XSS

Severity: Info

MoinMoin is a wiki engine implemented in Python. Moinmoin 1.1 was found to be installed on the web servers for host 2 and 3. MoinMoin 1.1 and prior contain at least two XSS vulnerabilities.

The first flaw is its inability to sanitize user input properly and as a result was vulnerable to XSS. The second flaw is the incorrect handling of group names that contain virtual group names such as "All", "Known" or "Trusted". This could result in a remote user having incorrect permissions (Ubuntu Security Notice , 2012).

Solution

Update to the current version (v1.9.3)

3.3.1.16 Multiple Web Server Default Page Fingerprinting Weakness

Severity: Info

Certain applications create files by default during their installations onto the web server. Files which are installed by default when some softwares are installed may reveal information about the web server and sometimes vulnerabilities hence it been classified as a weakness by security experts.

The following default files were found on the web host: 1:

- INSTALL.mysql.txt
- INSTALL.pgsql.txt
- Icons/README

The file ‘mailman/listinfo’ was found on the host 3 server.

There is no known exploitation for this weakness. However, files which are installed by default by certain softwares may reveal information about the web server and sometimes vulnerabilities. Attackers can use this knowledge to gain access to resources on the web host.

Solution

There should be a restricted access to such files or complete removal of such files.

3.3.1.17 Multiple Web Server Interesting Web Document

Severity: Info

This vulnerability is associated with files or directories which may be of interest. These may disclose sensitive information about the web server or aid in attack of the web server. (http://www.osvdb.org/3092) . Interesting files/directories were found on host 1 and 3 during the information gathering .

Details of the identified files/directories is given in appendix C

Solution

- File /directory should be removed from the web server.
- File / directory should be protected using password or other protection techniques.

3.3.1.18 Multiple Web Server robots.txt Remote Information Disclosure

Severity: Info

Web server robots.txt files contain instructions given to web crawlers and web robots about which areas of the website to scan or process during a web search.

A lot of sensitive information was revealed in the web server robots.txt for host 1. Directories, installation details and documents were found in the robots.txt file. This information could be used by an attacker to gain access.

Details of the information disclosure in the robots.txt file is given in appendix D:

Solution

Web manager / Administrator should ensure that no sensitive information is stored in the robots.txt file.

3.3.1.19 PHP expose_php Information Disclosure

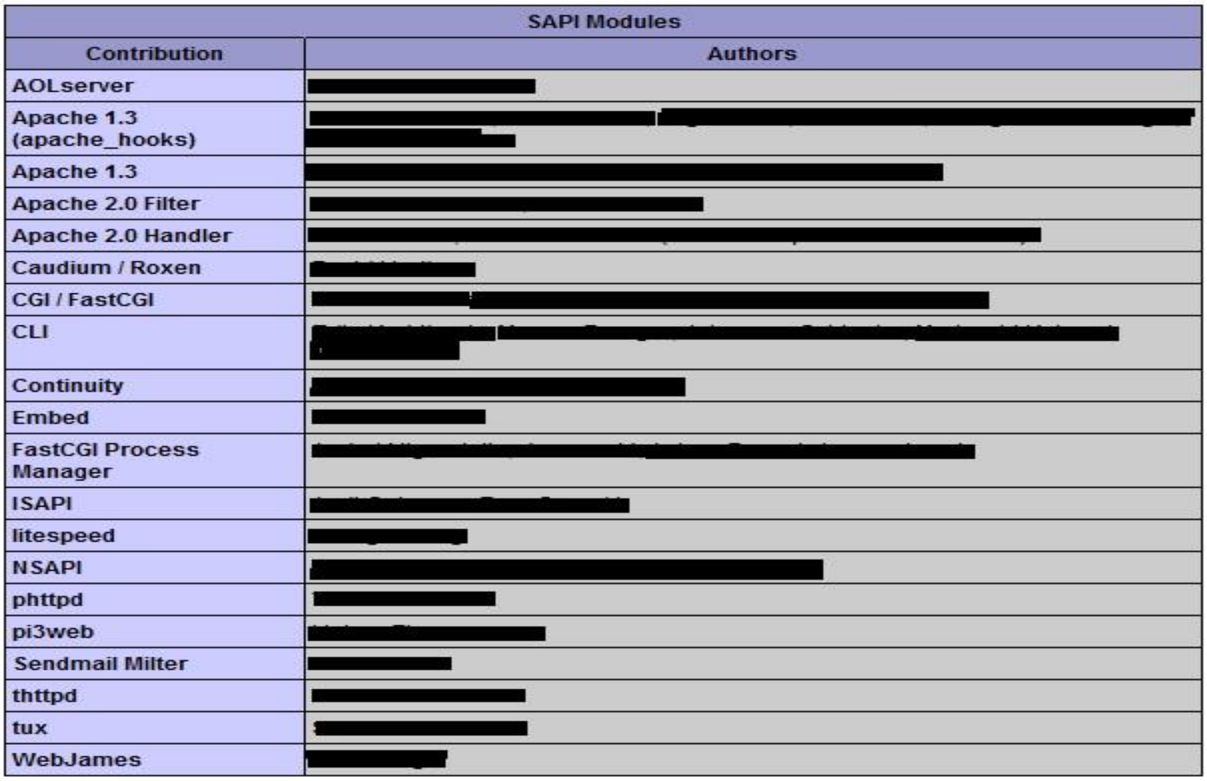
Severity: Medium

An Easter egg is a hidden message or feature, completely unrelated to normal functionality, that developers put inside software, website, or game. Even though easter eggs are harmless, they can reveal information that will be beneficial for attackers.

From the vulnerability scanning, it was discovered that a php configuration on the web host 5 server allowed disclosure of sensitive information such as software installation details through the use of specially crafted HTTP requests known as Easter eggs.

Easter egg code 'PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000' was used to obtain php development credit for the web host. The query used for the test was http://***.***.181.70/index.php/?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000

This produced a web page with the developer details. Screen shots are shown are in fig. 3.24



SAPI Modules	
Contribution	Authors
AOLserver	[redacted]
Apache 1.3 (apache_hooks)	[redacted]
Apache 1.3	[redacted]
Apache 2.0 Filter	[redacted]
Apache 2.0 Handler	[redacted]
Caudium / Roxen	[redacted]
CGI / FastCGI	[redacted]
CLI	[redacted]
Continuity	[redacted]
Embed	[redacted]
FastCGI Process Manager	[redacted]
ISAPI	[redacted]
litespeed	[redacted]
NSAPI	[redacted]
phttpd	[redacted]
pi3web	[redacted]
Sendmail Milter	[redacted]
thttpd	[redacted]
tux	[redacted]
WebJames	[redacted]

Fig. 3.24: PHP Credits

Solution

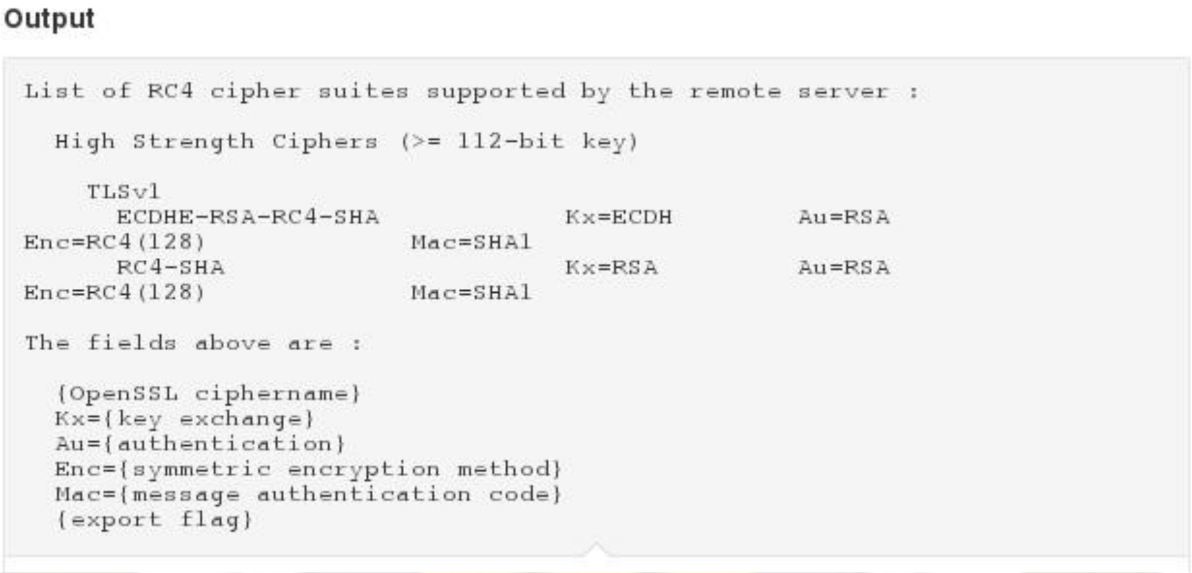
The value of expose_php in the php.ini configuration file should be set to 'off' . This will prevent access to the php easter eggs. (http://www.0php.com/php_easter_egg.php).

3.3.1.20 RC4 Algorithm Invariance-Weakness

Severity: Medium

RC4 is an encryption algorithm and involves the use of a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream. The RC4 algorithm is known to contain a flaw related to generation of pseudo-random characters which also results in the generation of a weak cryptographic key. As a result an attacker (with the use of session cookies) can recover plaintext from cipher text using brute-force and man-in-the-middle attacks. (Kovacs, 2015).

Host 1,2 and 3 were found to be supporting the use of the same set of RC4 ciphers as shown in fig.3.25



Output

```
List of RC4 cipher suites supported by the remote server :

  High Strength Ciphers (>= 112-bit key)

    TLSv1
      ECDHE-RSA-RC4-SHA            Kx=ECDH        Au=RSA
Enc=RC4(128)             Mac=SHA1
      RC4-SHA                      Kx=RSA         Au=RSA
Enc=RC4(128)             Mac=SHA1

The fields above are :

  {OpenSSL ciphername}
  Kx={key exchange}
  Au={authentication}
  Enc={symmetric encryption method}
  Mac={message authentication code}
  {export flag}
```

Fig. 3.25: RC4 Ciphers supported

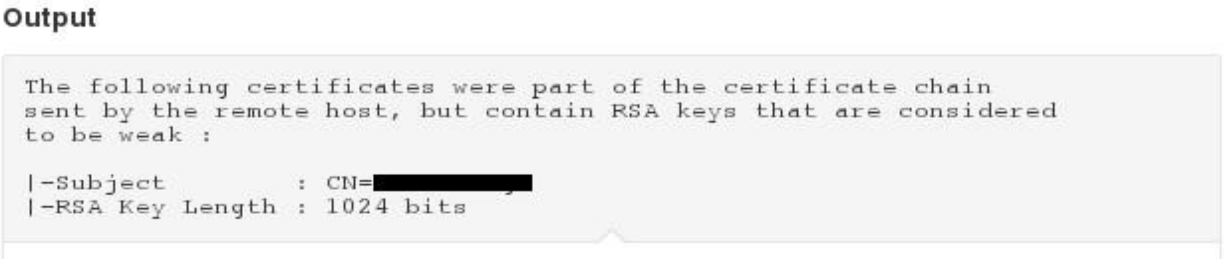
Solution

- Web administrators should disable the use of RC4 ciphers.
- Web managers / administrators should using secure ciphers.

3.3.1.21 RSA Keys less than 2048 bits.

Severity: Low

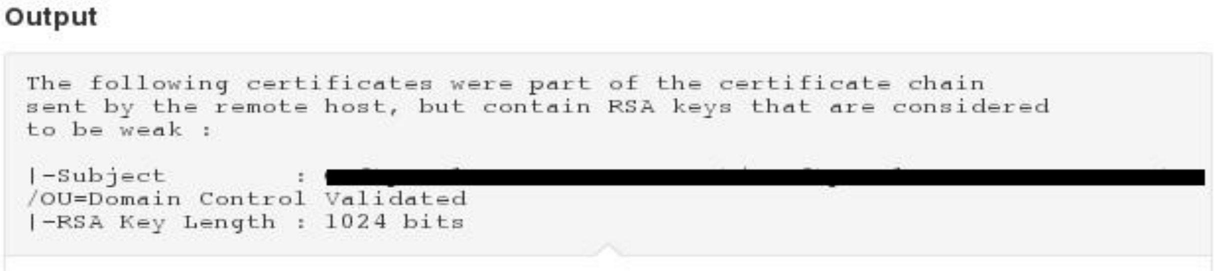
RSA(Rivest-Shamir-Adleman) is a cryptographic algorithm which uses both public and private keys for its encryption. It is found in many security protocols including SSH and SSL/TLS . The vulnerability scan revealed that host 1 and 4 had SSL certificate chains containing RSA keys less than 2048 bits. In 2011, the Certification Authority/Browser set a new directive which indicated that all SSL certificate chains which issued after 31st December, 2013 must have a length of 2048 bits. This was to ensure that such certificates are secure (CA/Browser Forum, 2011). The output generated by Nessus for the two hosts is show in fig 3.26 and 3.27.



Output

```
The following certificates were part of the certificate chain
sent by the remote host, but contain RSA keys that are considered
to be weak :

|-Subject         : CN=[redacted]
|-RSA Key Length : 1024 bits
```

Fig. 3.26 : RSA key length on host 1



Output

```
The following certificates were part of the certificate chain
sent by the remote host, but contain RSA keys that are considered
to be weak :

|-Subject         : [redacted]
/OU=Domain Control Validated
|-RSA Key Length : 1024 bits
```

Fig 3.27: RSA key length on host 4

Solution

All keys less than 2048 bits must be upgraded to 2048 bits to ensure secure chain certificates.

3.3.1.22 SMTP Service Supports Cleartext Login

Severity: Low

SMTP (Simple Mail Transfer Protocol) is a TCP/IP Protocol used in sending and receiving emails. It was also discovered that cleartext logins were supported via LOGIN and PLAIN channels on the host 3 and 5 SMTP services. These channels are unencrypted and as a result an attacker may be able to sniff user passwords.

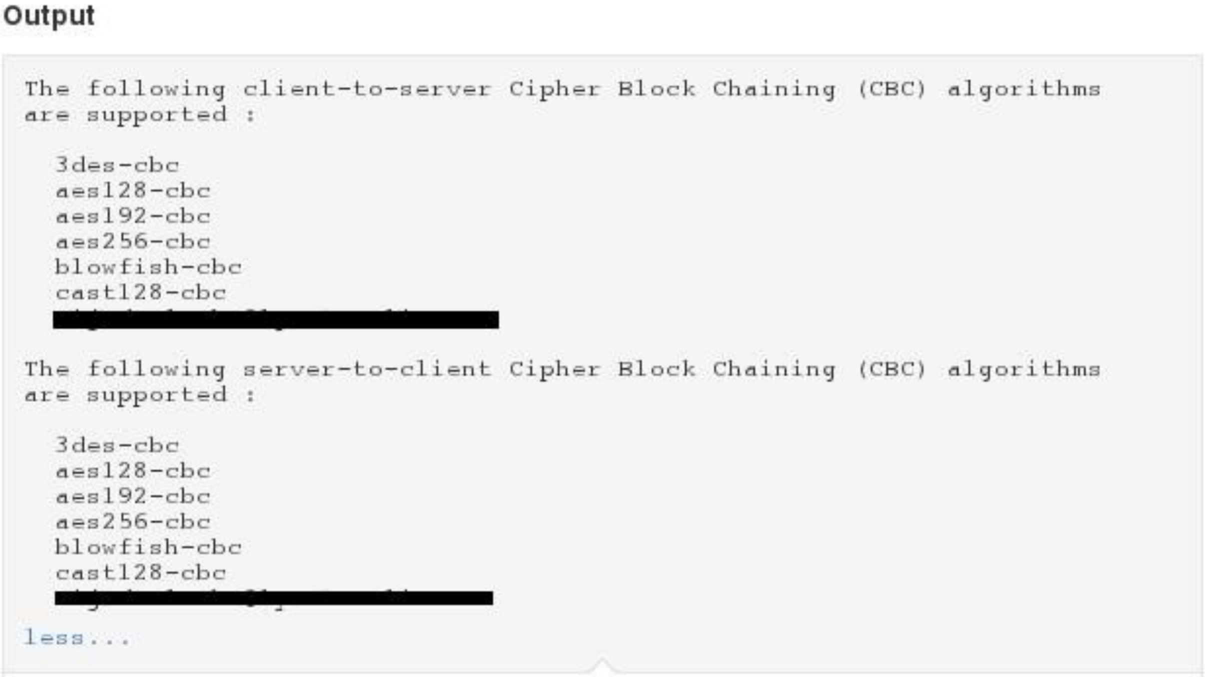
Solution

Nessus suggests that the service be configured to run communications over encrypted channels.

3.3.1.23 SSH Protocol CBC Mode Enabled

Severity: Low

CBC is an encryption algorithm and is considered weak because attackers can easily obtain plaintext data from CBC ciphers during an SSH Session via unknown vectors . Hosts 1,2 and 5 all supported the same set of CBC algorithms and this is shown in fig. 3.28



Output

```
The following client-to-server Cipher Block Chaining (CBC) algorithms
are supported :

  3des-cbc
  aes128-cbc
  aes192-cbc
  aes256-cbc
  blowfish-cbc
  cast128-cbc
  [redacted]

The following server-to-client Cipher Block Chaining (CBC) algorithms
are supported :

  3des-cbc
  aes128-cbc
  aes192-cbc
  aes256-cbc
  blowfish-cbc
  cast128-cbc
  [redacted]
less...
```

Fig. 3.28: CBC Algorithms supported

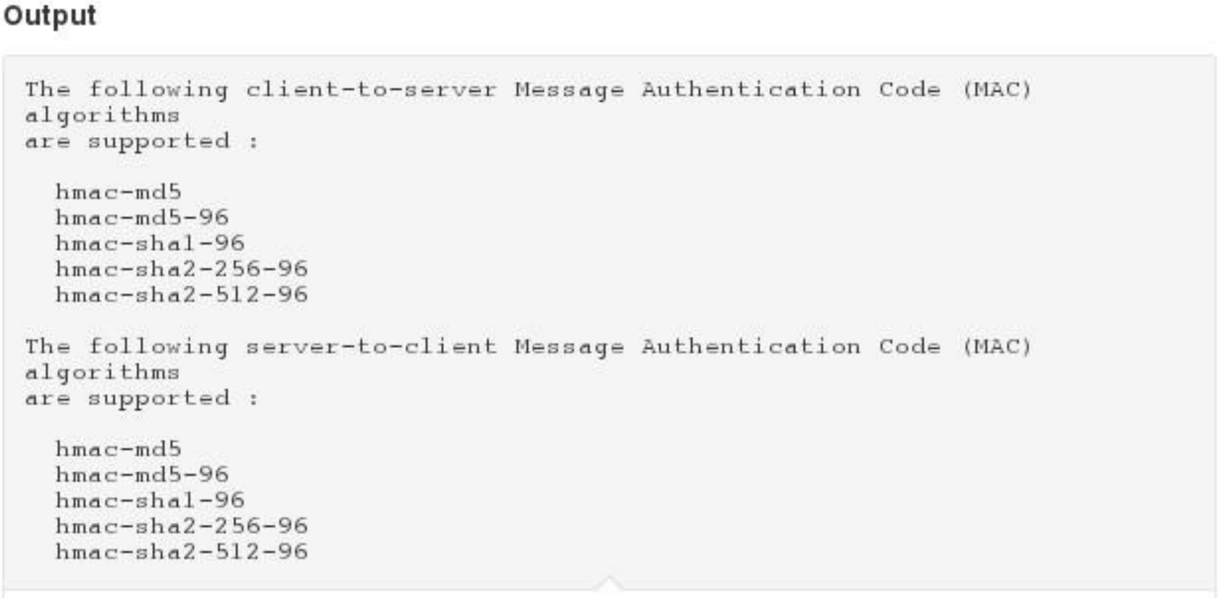
Solution

CTR mode ciphers should be used instead of CBC.

3.3.1.24 SSH weak Mac Algorithm enabled

Severity: Low

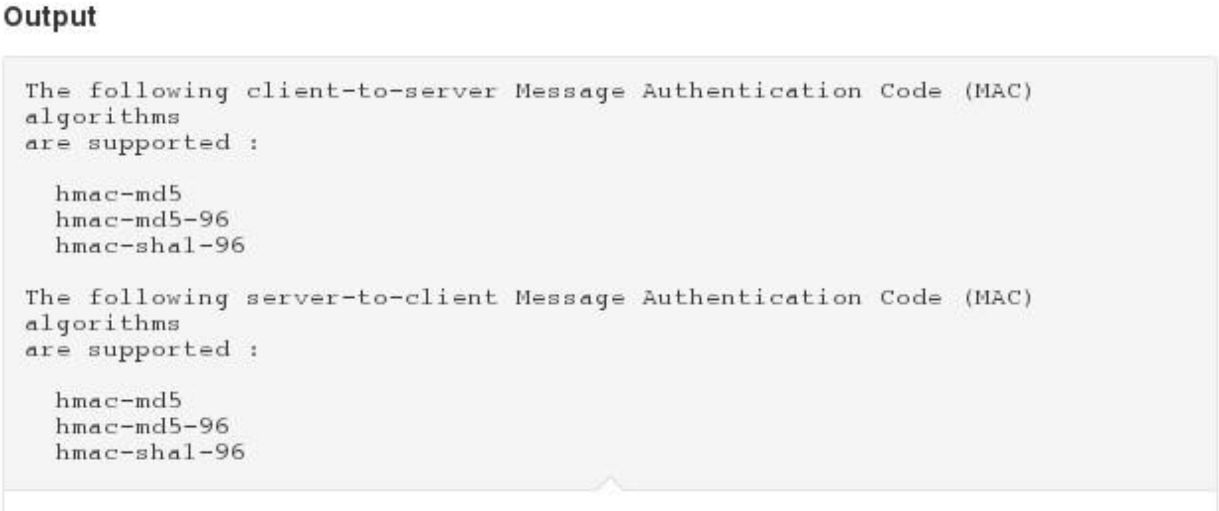
MAC algorithm is a cryptographic technique which uses a symmetric key to provide message authentication. For establishing MAC process, the sender and receiver share a symmetric key K. Hosts 1,2 and 5 were using found to be using SSH as a service. All the three hosts had their SSH configured to use a 96 bit key or MD5 encryption algorithm. These Algorithms are considered to be weak and susceptible to brute force attacks (https://www.tenable.com/plugins/index.php?view=single&id=71049). Fig. 3.29, 3.30 and 3.31 show the supported MAC algorithms on hosts 1,2 and 5 respectively.



Output

```
The following client-to-server Message Authentication Code (MAC)
algorithms
are supported :

  hmac-md5
  hmac-md5-96
  hmac-sha1-96
  hmac-sha2-256-96
  hmac-sha2-512-96

The following server-to-client Message Authentication Code (MAC)
algorithms
are supported :

  hmac-md5
  hmac-md5-96
  hmac-sha1-96
  hmac-sha2-256-96
  hmac-sha2-512-96
```

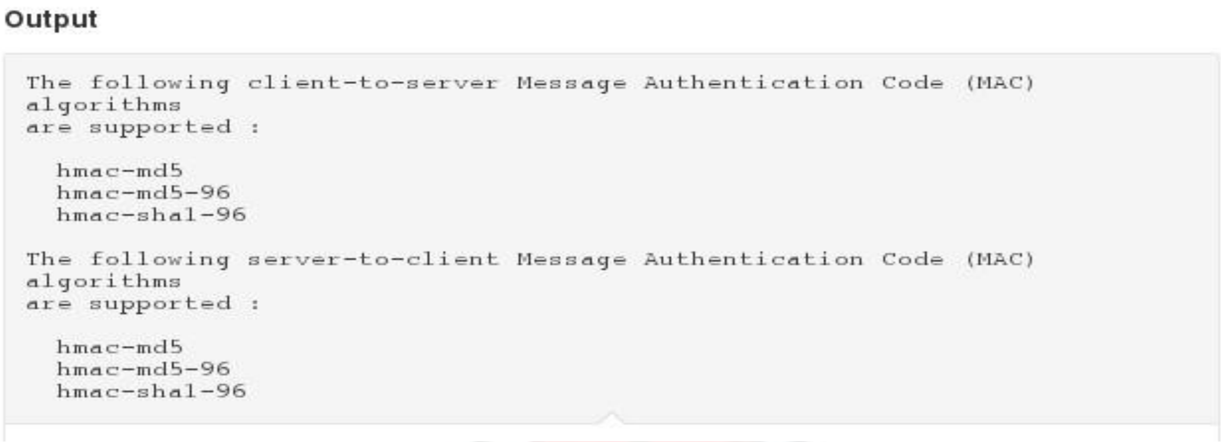
Fig. 3.29 : Weak MAC algorithm on host 1



Output

```
The following client-to-server Message Authentication Code (MAC)
algorithms
are supported :

  hmac-md5
  hmac-md5-96
  hmac-sha1-96

The following server-to-client Message Authentication Code (MAC)
algorithms
are supported :

  hmac-md5
  hmac-md5-96
  hmac-sha1-96
```

Fig. 3.30 : Weak MAC algorithm on host 2



Output

```
The following client-to-server Message Authentication Code (MAC)
algorithms
are supported :

  hmac-md5
  hmac-md5-96
  hmac-sha1-96

The following server-to-client Message Authentication Code (MAC)
algorithms
are supported :

  hmac-md5
  hmac-md5-96
  hmac-sha1-96
```

Fig. 3.31: Weak MAC algorithm on host 5

Solution:

- To mitigate this issue, any 96-bit HMAC Algorithms should be disabled.
- MD5-based HMAC Algorithms should also be disabled.

3.3.1.25 SSL v2 and SSL v3 Detection

Severity: Medium

SSL (Secure Sockets Layer) is a security protocol that transmits information and data across computer networks in an encrypted form.

Host 1 and 3 were found to be using SSLv2 and SSL v3. However these versions of SSL are known to contain several cryptographic flaws including Rivest Cipher 4 (RC4), Cipher Blocking Chain (CBC) and MAC cryptographic algorithms. Flaws in these encryption schemes have led to several successful attacks against SSL. Some of these attacks were.

- Logjam attack.
- FREAK attack.

- RC4 attacks.
- POODLE attacks.

Due to the security flaws, the Internet Engineering Task Force in 2011 prohibited the use of SSL v2 (IETF,2011) and in June, 2015, SSLv3 was also deprecated (IETF,2015).

It is therefore not recommended that SSL v2 and v3 be used in any form of communication across the network.

Solution

SSL should be disabled from the web server .

Transport Layer Security(TLS) should be used for communications across computer networks.

3.3.1.26 Untrusted SSL-Certificate

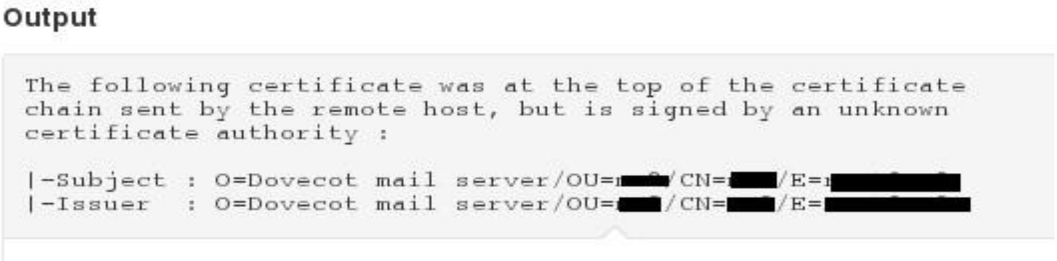
Severity: Medium

SSL certificates are used to establish secure connections between a web server and the client. The SSL connection protects sensitive data, such as credit card information which is exchanged during each session.

The scan revealed that the host 1 server's X.509 certificate did not have a signature from a known public certificate authority and as a result the certificate could not be trusted. This could be due to

- Missing intermediate certificates
- Unrecognized chains
- Self-signed certificates
- Certificate chain containing a signature that either didn't match the certificate's information or inability to verify the certificate.

According to the Nessus scan report, 'having an untrusted SSL certificate on a web host can make it easy for an attacker to carry out man-in-the-middle attacks against the remote host'. Fig 3.32 shows details of the untrusted SSL certificate.



Output

```
The following certificate was at the top of the certificate
chain sent by the remote host, but is signed by an unknown
certificate authority :

|-Subject : O=Dovecot mail server/OU=[redacted]/CN=[redacted]/E=[redacted]
|-Issuer  : O=Dovecot mail server/OU=[redacted]/CN=[redacted]/E=[redacted]
```

Fig. 3.32: Untrusted SSL Certificate

Solution

Purchase or generation and use of proper certificates.

3.3.1.27 Usr/doc Directory Information Disclosure

Severity: Medium

A browsable directory is one in which anyone can see the content. This gives room for attackers to obtain sensitive information because all files are accessible to anyone who requests them(Rapid7,2000).

Nikto reported that usr/doc directory on host 3 was browsable whiles Nessus reported that usr/doc directory on host 5 was browsable

Solution

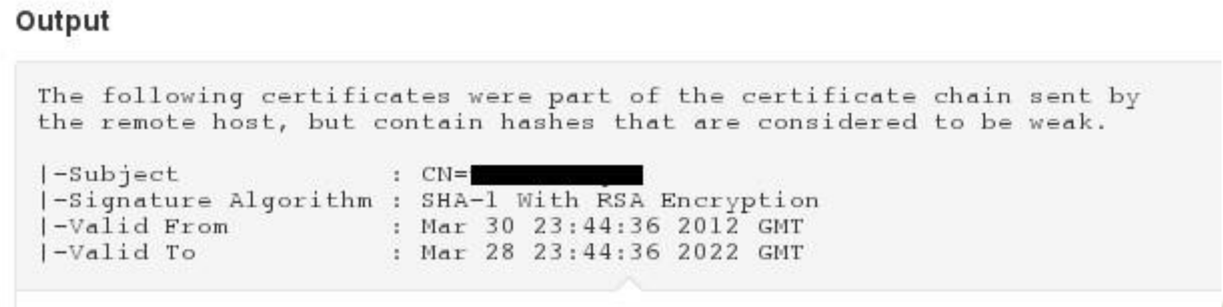
There should be a restricted access to the doc directory.

3.3.1.28 Weak Hashing Algorithms for the Signing of SSL Certificate

Severity: Medium

Scanning revealed that one or more SSL Certificates on host 1 were signed using SHA-1 hashing algorithm which is considered to be weak.

Having a weak hashing algorithm in a web host makes the host vulnerable to collision attacks where the attacker exploits the weak hashing algorithm to generate another certificate with the same digital signature, making it possible to manipulate the services on the host (Sotirov,2008).



Output

```
The following certificates were part of the certificate chain sent by
the remote host, but contain hashes that are considered to be weak.

|-Subject             : CN=[redacted]
|-Signature Algorithm : SHA-1 With RSA Encryption
|-Valid From          : Mar 30 23:44:36 2012 GMT
|-Valid To            : Mar 28 23:44:36 2022 GMT
```

Fig. 3.33 :Weak hashing algorithm on host 1 SSL certificate

Solution

The weak algorithm should not be used. Instead a stronger hashing algorithm such as SHA256 should be used.

3.3.2 Risk factor

This is a way of evaluating an identified vulnerability and its impact on the business of the associated entity.

In vulnerability scanning, risk factors of all the identified vulnerabilities were given by Nessus. Vulnerabilities were classified as Critical, High, Medium, Low or Info based on CVSS Values. CVSS stands for Common Vulnerability Scoring System and is a standard used to assign severity to identified vulnerabilities to help prioritize response and countermeasures to the vulnerabilities. CVSS Values ranges from 0 – 10. CVSS values and the associated risk factor is given in table 3.13. Priority is given to the vulnerability with the highest CVSS Base score.

Risk factor	**CVSS Base Score**
Critical	10.0
High	7.0-9.9
Medium	4.0-6.9
Low	1.0-3.9
Info	0

Table 3.13: Risk factors

3.3.2.1 Critical

Vulnerabilities that score in the critical range usually have most of the following characteristics:

- Exploitation of the vulnerability likely results in root-level compromise of servers or infrastructure devices.
- Publicly available exploits.
- Easy exploitation of vulnerability.

- Exploitation requires little or no skills

Vulnerabilities that are identified as critical should be patched or upgraded as soon as possible . If the issues cannot be resolved immediately, it is advised that such systems should be accessible from the internet.

3.3.2.2 High

Vulnerabilities with risk factor of high have the following characteristics

- Difficult to exploit.
- Exploitation could result in elevated privileges.
- Exploitation could result in a significant data loss or downtime.
- Exploitation requires skill.
- Exploitation requires specialized conditions such as social engineering methods.

3.3.2.3 Medium

Vulnerabilities that score in the medium range usually have some of the following characteristics:

- The attack requires the attacker to manipulate individual victims via social engineering tactics.
- Denial of service vulnerabilities that are difficult to set up.
- Exploits that require an attacker to reside on the same local network as the victim.
- Exploitation provides only very limited access.
- Successful exploitation requires user privileges.

3.3.2.4 Low

Vulnerabilities in the labeled as low typically have very minimal impact on an organisation's business.

Exploitation of such vulnerabilities usually requires local or physical system access.

The affected product typically requires access to a wide range of systems and users, possibly anonymous and untrusted (e.g., Internet-facing web or mail server).

Such vulnerabilities could be combined with other attack vectors to compromise a web server. Such vulnerabilities may be difficult to exploit

3.3.2.5 Info

Vulnerabilities labeled as info did not pose any threat themselves but could provide information that can be used to gain insight into how to compromise hence their inclusion in the vulnerability list.

Risk factor	**No. of Vulnerabilities**				
	Host 1	**Host 2**	**Host 3**	**Host 4**	**Host 5**
Critical	0	0	0	0	0
High	0	0	1	0	0
Medium	6	3	8	2	4
Low	4	2	1	2	3
Info	6	3	5	0	3

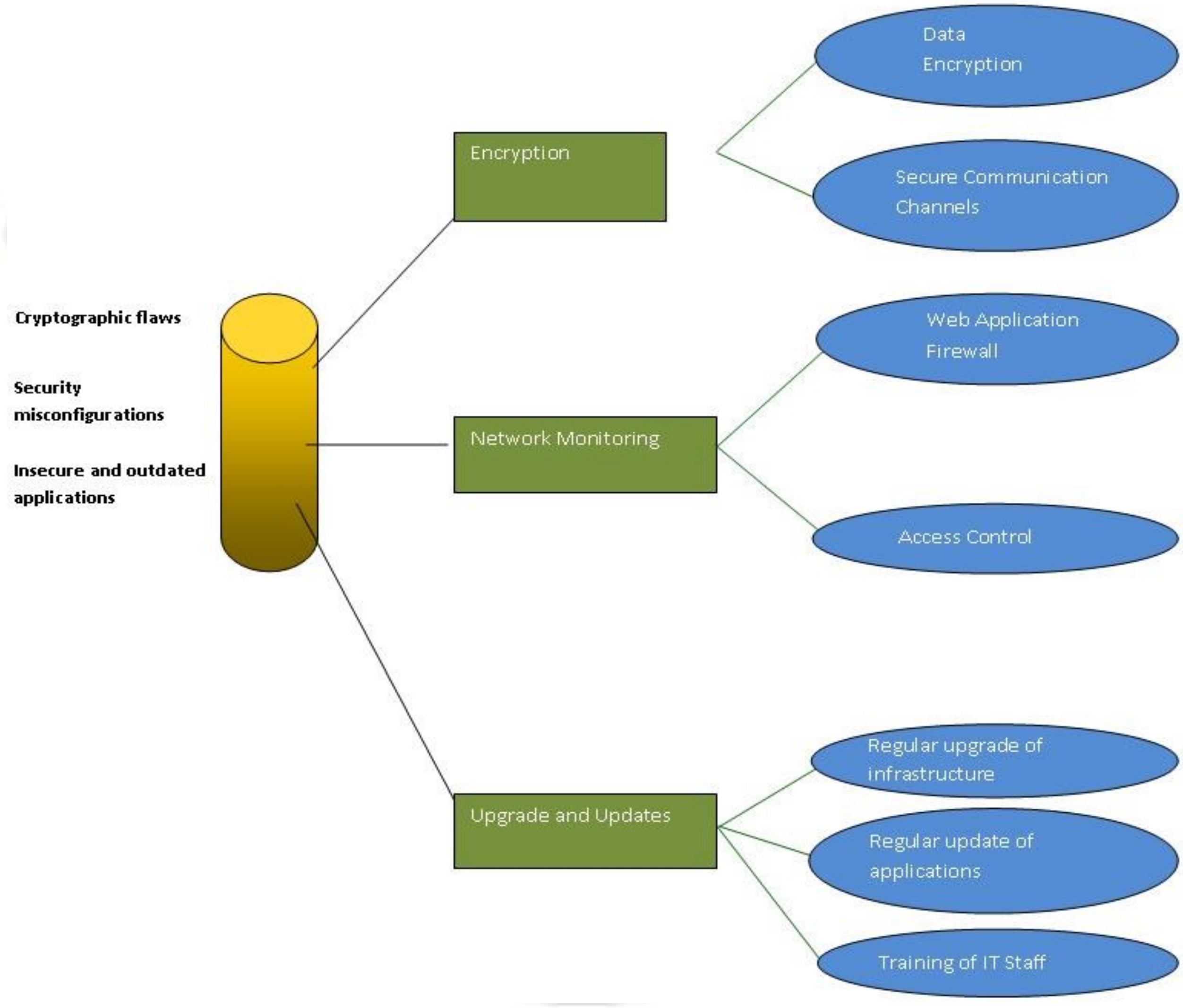
Table 3.14 Vulnerability summary

3.4 Proposed Solution

To mitigate the security issues identified, a proposed solution was designed. The solution comprises of three processes which are as follows.

- Data Encryption Network Monitoring
- Upgrade and Update.

The proposed solution is represented in fig. 3.34



Cryptographic flaws
Security misconfigurations
Insecure and outdated applications
Encryption
Network Monitoring
Upgrade and Updates
Data Encryption
Secure Communication Channels
Web Application Firewall
Access Control
Regular upgrade of infrastructure
Regular update of applications
Training of IT Staff

Fig 3.34: Solution to address the identified vulnerabilities

3.4.1 Encryption

It was discovered that all the five hosts transmitted data in clear text. It was therefore possible for any user to sniff and capture data which is being transmitted over the network. To solve this problem, the data should be encrypted before storage or transmission. Encrypting the data will make it readable for only the people who have the encryption key. A proper encryption process includes:

- Making sure the correct data is encrypted.
- Proper storage and management of encryption keys.
- Implementing secure encryption models.
- Encryption algorithms should be reviewed to ensure that they contain no vulnerabilities.

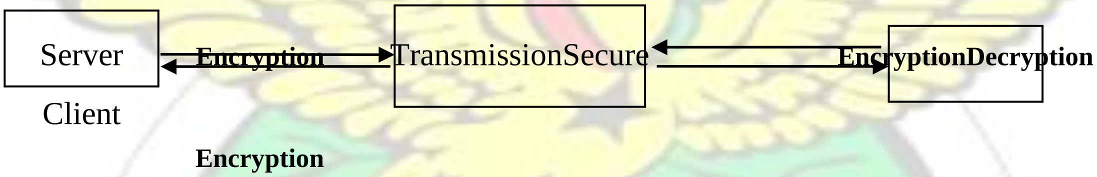


Fig.3.35 :Data encryption process

Aside the data itself being encrypted, the data should be transmitted across secure communication channels to avoid users from sniffing them. Transmitting the data in a secure channel provides additional protection to the data in transition. There are several encryption tools available. OPENSSL is an example of such tools that can be used to provide secure data transmission.

OpenSSL offers SSL and TLS encryption for data in transit and encrypts communications between Point A and Point B – the website server and browser. This encryption will prevent anyone from being able to intercept that traffic (Man in the middle attacks). OPENSSL can be used on both windows and linux environment. There is also a Solaris version available. Aside these advantages, OPENSSL is also open source hence a cheaper way of ensuring secure data transmission.

3.4.2 Network monitoring

To detect and ensure correction of network attacks and security breaches, there should be regular monitoring of the network. Measures should be put in place so that suspicious activities as well as security breaches are identified. Also any identified security issue must be corrected as soon as they are detected. Aside security breaches, all user activities should be monitored to ensure that they comply with security policies as well as user policies of the network.

Monitoring should occur on a continuous basis to assess performance of implemented controls over time and ensure that identified deficiencies are reported to senior management in a timely manner. Compliance with access authorizations should be monitored by periodically comparing authorizations to actual access activity. Access control software typically provides a means of reporting user access authorizations and access activity.

Monitoring activities may include maintenance of audit trails, continuous review of actual or attempted unauthorized, unusual, or sensitive access, investigation of and response to suspicious access activity as well as ongoing security surveillance activities.

Tools that can be used to monitor networks are Intrusion Detection and Prevention Systems, Web Application Firewalls and Access Control Systems.

Even though intrusion detection and prevention systems can be used to monitor the systems, a combined implementation of Web application firewalls and Access Control systems will provide a better security strategy.

Access control systems monitor the activity of users within the network and ensure that users do not gain access to resources which they have not been authorized to use or if they exceed the time limit allocated to the use or access of a resource. Access control systems typically will grant, limit ,prevent or revoke access to a user depending on what policies have been implemented on the system (Krishnasamy,1995).

A web application firewall (WAF) is an appliance, server plugin, or filter that applies a set of rules to an HTTP conversation to and from a web application. WAF monitors the activity of outsiders (people without any privileges and outside the network) and sends alerts if an attempt is made to gain access to the network. Apart from sending alerts web application firewalls protects web applications from attacks that Intrusion Prevention Systems cannot prevent such as SQL injection and Cross-Site Scripting attacks. This is possible because WAFs understand protocol logics such as HTTP GET, POST, HEAD, JavaScript, SQL, HTML and XML which reside at the application layer. WAFs not only detect attacks that are known to occur in web application environments, they also detect (and can prevent) new unknown types of attacks. By watching for unusual or unexpected patterns in the traffic they can alert and/or defend against unknown attacks. For example if a WAF detects that the application is returning much more data than it is expected to, the WAF can block it and alert someone (McMillan, 2009).

3.4.3 Upgrade and Update

Another way of mitigating the vulnerabilities is to regularly update the applications being run on the server. Updates and bug fixes are provided by the software vendors. So the web managers and IT staffs should obtain them to make their applications secure.

Upgrading of infrastructure should also be done. This should include the change of insecure applications to secured ones. Also outdated applications should be replaced with current ones. In most cases exploits are available for outdated applications and so attackers can easily attack the affected applications.

IT Staff should also be given periodic training and resources. This will make them abreast with good security practices as well . Training the IT Staff on how to use and configure the web applications will also be helpful in hardening the security of the web applications.

3.5 Report

A detailed report was given for each website that was tested. Copies were sent to the managers/owners of the website. The report included the following items:

Executive summary: This is a summary of the report. It contains the summary for tasks accomplished, methodology and high level findings as well as recommendations.

Scope: This section describes the scope of the work (including IP ranges of the target hosts that were tested).

Methodology: This contains details on how the vulnerability assessment was done including the tools and techniques that were used for the task.

Findings: Contains all the identified vulnerabilities as well as the level of risk they pose to the organization.

Solutions and Recommendations: Contains the proposed mitigations for the identified vulnerabilities as well as recommendations on how to improve the security of the web servers.

CHAPTER 4

4.0 RESULTS AND DISCUSSION

This section describes the results of vulnerability assessments which were performed on selected web hosts and the significance of the findings.

4.1 Results

16 vulnerabilities were identified in host 1 and 8 vulnerabilities for host 2. The number of vulnerabilities identified in host 3 and host 4 were 15 and 4 respectively. 10 vulnerabilities were identified in host 5. From the results, host 1 had the highest number of vulnerabilities. This was followed by host 3, host 5, host 2 respectively. Host 4 had the least number of vulnerabilities. A graphical representation is given in fig. 4.1.

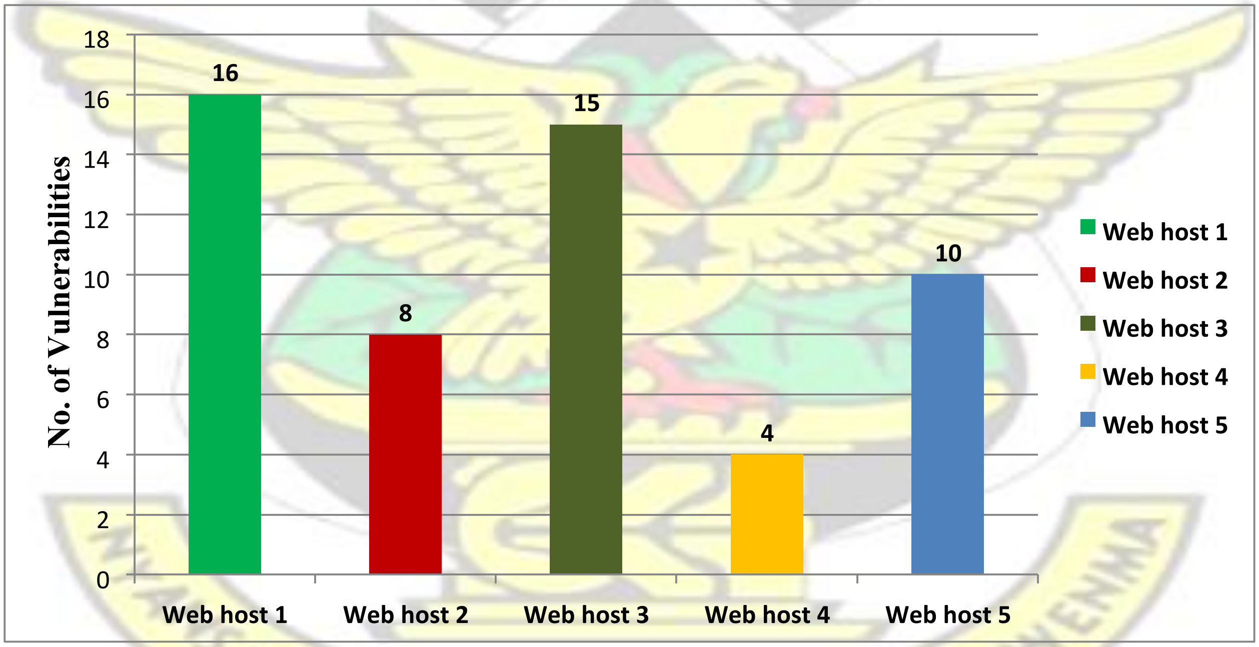


Fig.4.1: Vulnerability summary for the scanned web hosts

Host 1

Out of the 16 vulnerabilities identified, 6 of them were classified as medium, while 4 of them were low. 6 vulnerabilities were also labeled as info. No vulnerability was classified as critical or high.

The threat level for the server could be said to be medium since the most severe among the vulnerabilities were labeled medium.

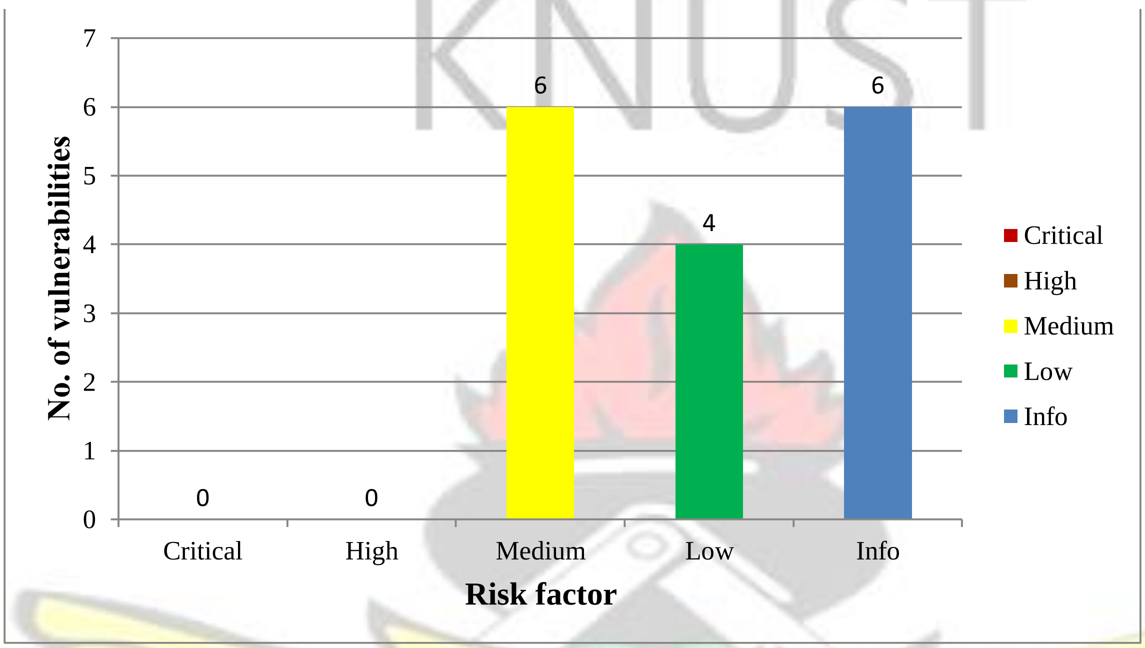


Fig. 4.2: Identified vulnerabilities on host 1

Host 2

Out of the 8 vulnerabilities identified in host 2, the risk posed by 3 of them was medium while the number of vulnerabilities with risk factors low and info were 2 and 3 respectively. There was no vulnerability associated with risk factors critical and high. Threat level for this host was also medium because the most severe among the identified vulnerabilities was medium.

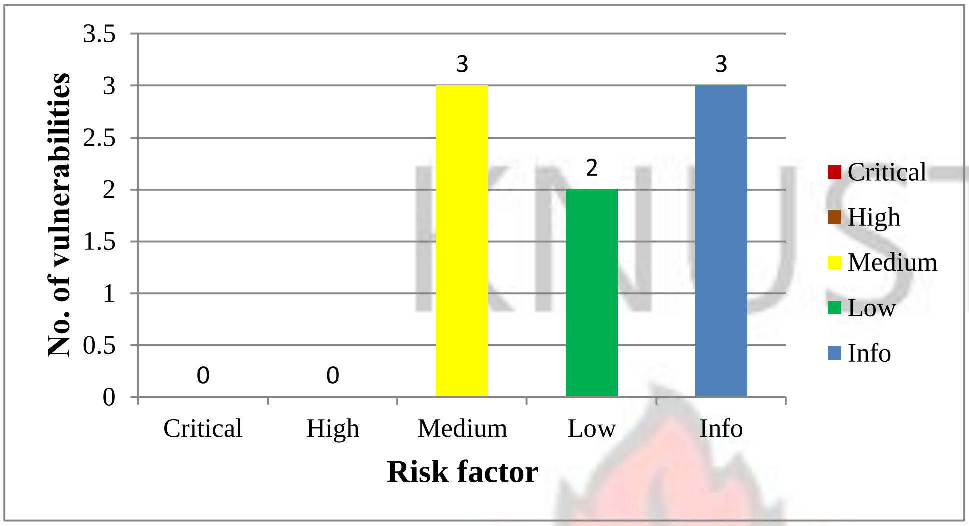


Fig. 4.3: Identified vulnerabilities on host 2

Host 3

Out of the 15 vulnerabilities identified in host 2, the severity of 1 of them was high. The risk posed by 8 of them was medium while the number of vulnerabilities with risk factors low and info were 1 and 5 respectively. No vulnerability was identified as critical. The most severe was labeled High making its thread level as High. Therefore in dealing with the identified vulnerabilities , the vulnerability labeled High should be tackled first.

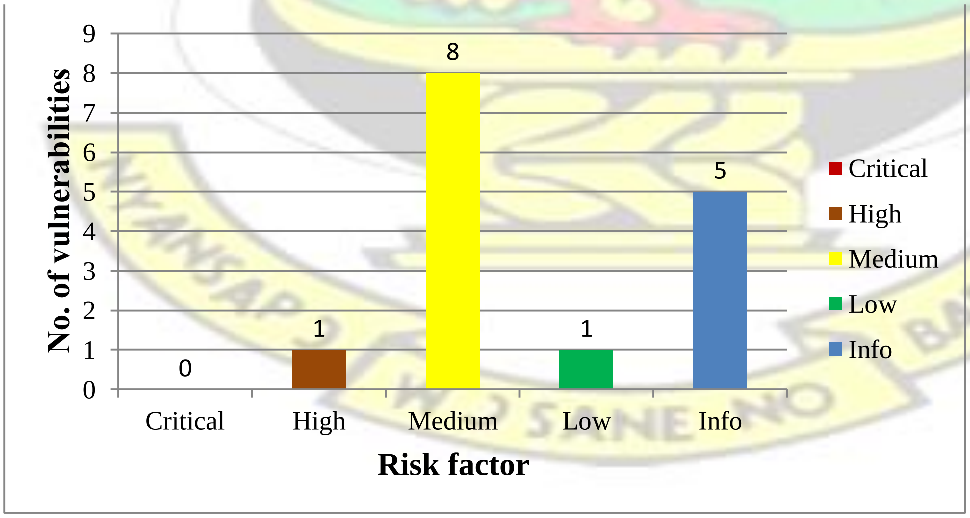


Fig. 4.4: Identified vulnerabilities on host 3

Host 4

Host 4 recorded the least number of vulnerabilities. Out of 4 vulnerabilities were identified, 2 of the vulnerabilities had a severity of medium while 2 vulnerabilities were labeled as low. No vulnerability was classified as critical, high or info. Threat level was therefore medium.

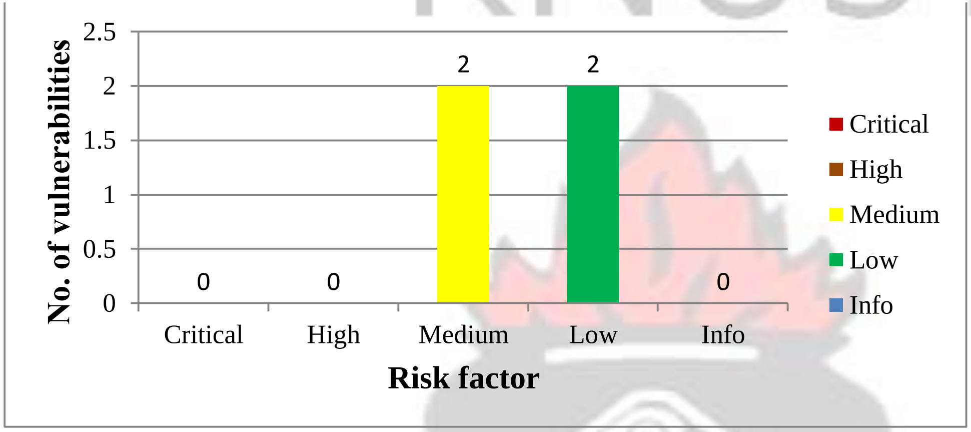


Fig 4.5: Identified vulnerabilities on host 4

Host 5

10 vulnerabilities were identified in host 5. 4 of them had a risk factor of medium. 3 vulnerabilities had risk factor low and 3 vulnerabilities were labeled as informational (info). The most severe vulnerabilities were labeled medium and so the threat level was also medium.

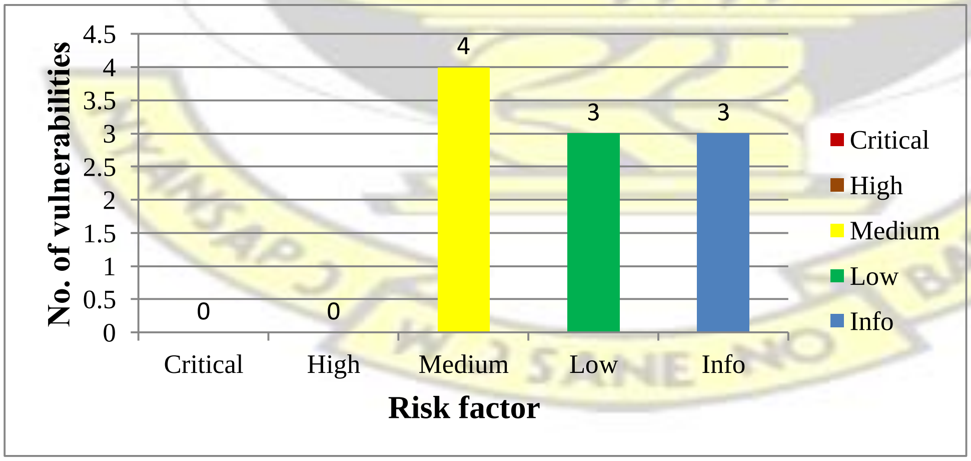


Fig. 4.6: Identified vulnerabilities on host 5

4.2 Discussion

This study was done to identify vulnerabilities in selected web hosts and a solution proposed to mitigate the identified vulnerabilities. Even though everything was done to ensure that the tests and methodology used followed standard procedures, certain factors affected the results of the tests.

One of the factors was the fact that the websites were productive environments and as such there was a high risk of disrupting the services. Due to this, Denial of service (Dos) tests as well as memory corruption tests were not performed. This affected the results because even though security issues were discovered, there was no way of investigating whether the websites were secure from Dos attacks and memory corruption attacks.

The vulnerability assessment was performed outside the network. This was to simulate how an attacker outside the network might infiltrate the network using discovered vulnerabilities. It is therefore possible that if internal tests were done, more information would have been discovered since attacks can come from within the network. However no internal assessment was done. This factor should therefore be considered when doing an analysis on the result.

This assessment was done on 5 websites belonging to different entities. At the end of the assessment security issues were discovered. The owners of the websites were not aware of these issues and so this assessment was useful to them.

Vulnerabilities were discovered in all the 5 scanned web hosts. Some of the vulnerabilities existed in more than one of the scanned hosts while others were identified on a single host. From the report generated by Nessus and Nikto, the identified vulnerabilities were due to the following reasons:

- Cryptographic flaws

- Security misconfigurations
- Applications with vulnerabilities

4.2.1 Cryptographic flaws

Cryptographic flaws occur when sensitive data is stored or transmitted in unencrypted form or encrypted using weak algorithms. Insecure Cryptographic Storage isn’t a single vulnerability, but a collection of vulnerabilities which can lead to a breach in confidentiality.

A data which is not securely encrypted can easily be obtained by unauthorized users. Fig. 4.7 shows a typical cryptographic flaw. Attacks such as logjam and FREAK were successful due to existing cryptographic flaws in the target web applications.

The number of cryptographic flaws identified on host 1 was 6 whiles host 2 and 3 had 3 and 2 flaws respectively. 1 cryptographic flaw was identified on host 4 and 3 flaws identified on host 5. The following cryptographic flaws were identified:

- Logjam vulnerability
- RC4 Algorithm Invariance-Weakness.
- RSA keys less than 2048 bits
- SSH Protocol CBC Mode Enabled.
- SSH weak Mac Algorithm enabled
- Weak Hashing Algorithms for the Signing of SSL Certificate

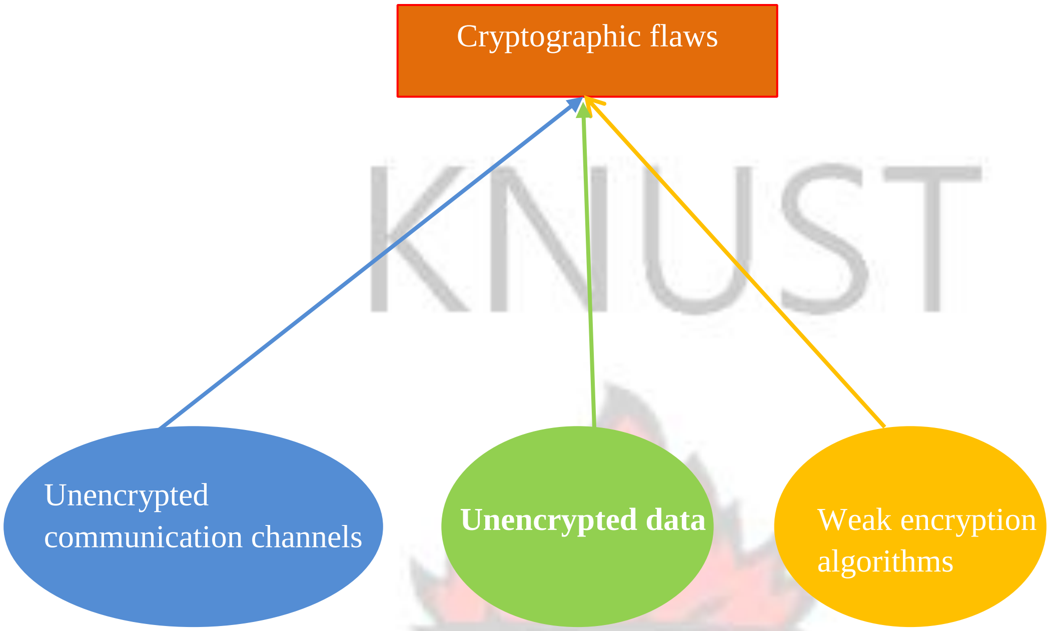

Cryptographic flaws
Unencrypted communication channels
Unencrypted data
Weak encryption algorithms

Fig. 4.7: Cryptographic flaws

4.2.2 Security misconfigurations

The infrastructure that supports a Web application comprises a complex variety of devices and software, including servers, firewalls, databases and OS and application software. All these elements need to be securely configured and maintained, with the application running with the least privileges necessary, yet many systems are never fully hardened. If security configurations are not properly done the system can be compromised. Security misconfigurations can lead to the following :

- Unauthorised access, modification and transmission of data.
- Denial of service.
- Virus attacks
- Memory corruption attacks.
- Buffer overflow attacks.

- Installation of backdoors.

Integrity, confidentiality and availability of data can be affected if web applications are not properly configured.

It was discovered that security misconfigurations existed on all the five host servers. Hosts 1,2,3 and 5 all had cookies set without httponly flag making them susceptible to man-in-the middle attacks. Host 2 and 5 had enabled HTTP TRACE method making them vulnerable to Cross-Site Scripting attacks. Host 5 had enabled directory indexing which could result in unauthorized access to information and data.

Host 3 was also discovered to be running an ftp server which was not properly configured making it possible to login anonymously and also vulnerable DOS attacks. Host 3 remote DNS server had not been properly configured so could answer any request making vulnerable to DOS attacks.

A php misconfiguration was also found on host 5 web server and this could allow information disclosure via the use of PHP Easter eggs.

Also HTTP Strict Transport Security was found missing on hosts 1,2 and 3 making them vulnerable to man-in-the middle attacks since the http transmission was insecure.

All the five hosts were also vulnerable to clickjacking due to the absence of X-Frame Options Response Header and so attackers could render the content of the web pages they generate to steal sensitive information from users.

4.2.3 Applications with vulnerabilities

Insecure applications occur due to poor application design based on the false assumption that users will always follow the application rules. For example, if a user's account ID is shown in the page

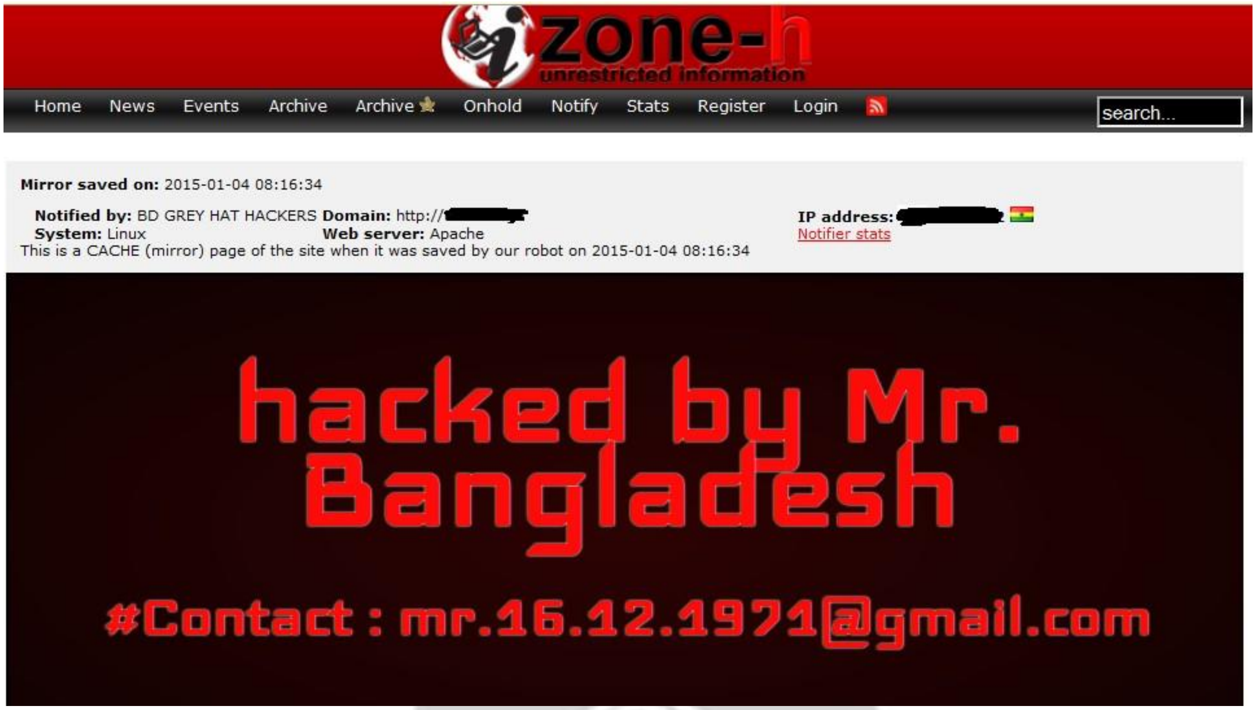
URL or in a hidden field, a malicious user may be able to guess another user's ID and resubmit the request to access their data, particularly if the ID is a predictable value. Common places where this data is incorrectly exposed are URLs and links, hidden form fields, the unprotected view state in ASP.NET, drop-down list boxes, JavaScript code and client-side objects like Java applets. Servers running applications with vulnerabilities can easily be compromised. The applications are also susceptible to attacks such as denial of service, memory corruption, buffer overflow and XSS attacks.

In this research, 3 of the web hosts were found to be running applications with known vulnerabilities.

Host 4 was found to be running Microsoft IIS 7.0 which contained vulnerabilities such as Memory Corruption, Buffer Overflow and Denial of Service. Host 2 and 3 both were running MoinMoin 2 which also contains 2 vulnerabilities relating to Cross-Site Scripting.

4.2.4 Other security issues

Aside these issues, a search from the database of http://zone-h.org revealed that web host 1 and host 2 had been recently been compromised as shown in fig. 4.7 and 4.8. This represented a serious information disclosure because the IP address as well as host operating system is publicly available. Owners of web host 1 and host 2 are therefore advised to review their security policies and fix the identified flaws to prevent another attack.


zone-h
unrestricted information
Home News Events Archive Archive Onhold Notify Stats Register Login
search...

Mirror saved on: 2015-01-04 08:16:34

Notified by: BD GREY HAT HACKERS **Domain:** http://[redacted]

System: Linux **Web server:** Apache

This is a CACHE (mirror) page of the site when it was saved by our robot on 2015-01-04 08:16:34

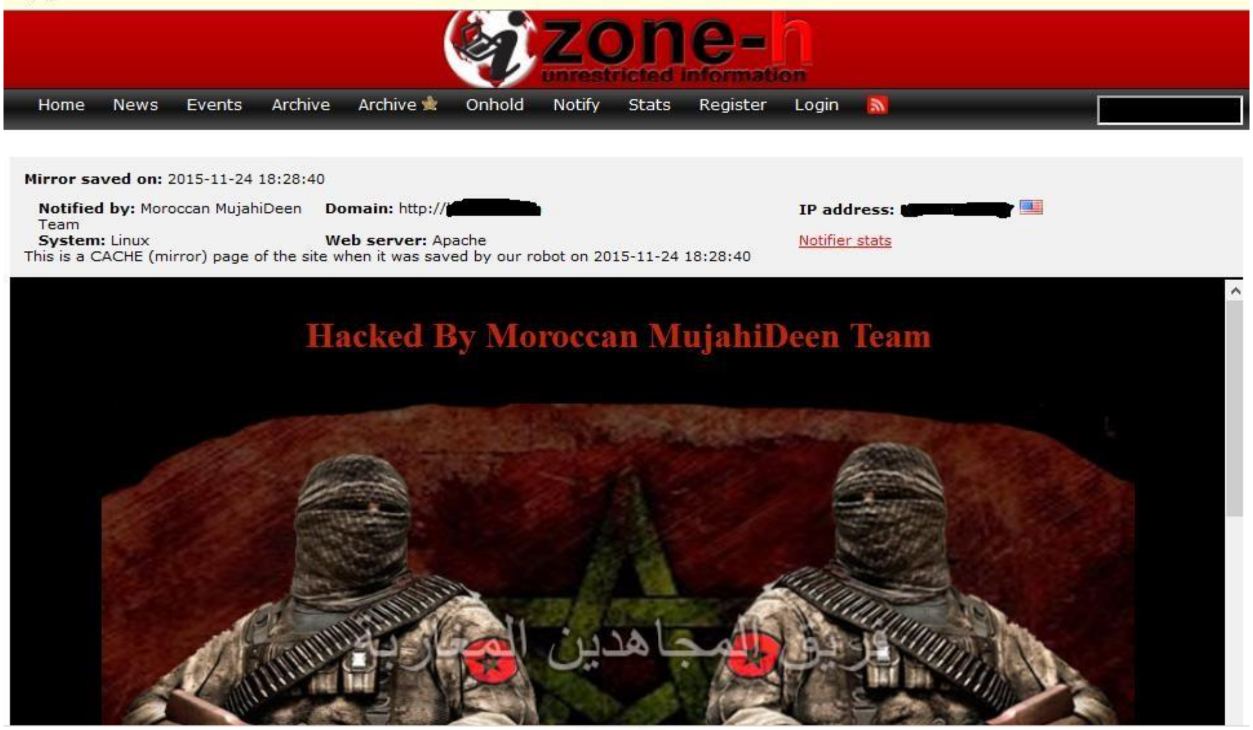
IP address: [redacted]

Notifier stats

hacked by Mr. Bangladesh

#Contact : mr.16.12.1971@gmail.com

Fig 4.8: Web defacement of host 1


zone-h
unrestricted information
Home News Events Archive Archive Onhold Notify Stats Register Login
فريق المجاهدين المغاربة

Mirror saved on: 2015-11-24 18:28:40

Notified by: Moroccan MujahiDeen Team **Domain:** http://[redacted]

System: Linux **Web server:** Apache

This is a CACHE (mirror) page of the site when it was saved by our robot on 2015-11-24 18:28:40

IP address: [redacted]

Notifier stats

Hacked By Moroccan MujahiDeen Team

Fig 4.9: Web defacement of host 2

CHAPTER 5

5.0 CONCLUSION AND RECOMMENDATION

The vulnerability assessment was helpful as it provided information about the security of the selected websites. Vulnerabilities were discovered in all the web hosts that were scanned. Some of vulnerabilities were found in all the web hosts whiles others were specific to a particular host.

These discoveries brought to light that there are security issues that need to be addressed in all the five hosts that were scanned.

As technology is evolving, new techniques are also being developed to exploit computer systems. It is therefore important to be abreast with such techniques in other to combat these security threats.

Based on the findings, it is recommended that:

1. All security issues identified should be resolved.
2. All applications being used on the respective web servers should be upgraded or changed to a more secure one.
3. All hosts within the respective networks should be checked for security flaws.
4. Regular tests should be conducted to access the security of the respective networks.
5. Personnel should be trained on how to maintain security of respective networks.
6. There should be internal vulnerability assessments for the websites.
7. Denial of service and Memory corruption tests must be done on the web servers.

REFERENCES

Aamoth ,D.(2011) ,New Sony Hack Claims Over a Million User Passwords, Available at *http://techland.time.com/2011/06/02/new-sony-hack-claims-one-million-user-passwords,* Accessed April 3, 2015.

Acunetix(2014),What is Cross Site Scripting, Available at https://www.acunetix.com/ websitesecurity/ cross-site-scripting., Accessed November 30, 2015.

Auger, R., (2010),Cross Site Request Forgery Available at *http://projects.webappsec.org/w/ page/13246919/ Cross Site Request Forgery,* Accessed January 27, 2015.

CA/Browser Forum (2011), Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, v.1.0, Available at *https://www.cabforum.org/ Baseline_Requirements_V1.pdf,* Accessed October 11,2015.

CAPEC(2015), Clickjacking , Available at *http://capec.mitre.org/data/definitions/103.html*, Accessed January 16, 2016.

CERT (2003), Web Servers Enable HTTP TRACE Method by Default, Available at *https://www.kb.cert.org/vuls/id/867593* , Accessed January 3, 2016.

CERT(1999),Problems with FTP Port Commands, Available at *http://web.archive.org/web/ 20131105191347/http://www.cert.org/tech_tips/ftp_port_attacks.html*, Accessed November 12, 2015.

CWE (2012), Cleartext transmission of Sensitive Information, Available at *https://cwe.mitre.org/data/definitions/319.html*, Accessed November 15,2015.

Hitachi (2015), Definition of Access Control, Available at *http://hitachi-id.com/concepts/access_control.html*, Accessed October15, 2015.

http://osvdb.org/3268., Accessed June 18, 2015. http://osvdb.org/637, Accessed January 10, 2016. http://osvdb.org/91162, Accessed February 17, 2015.

http://osvdb.org/show/osvdb/50012 , Accessed January 27, 2015.

http://www.0php.com/php_easter_egg.php , Accessed January 16, 2016.

http://www.osvdb.org/3092 , Accessed January 16, 2016. http://www.osvdb.org/3268 , Accessed January 16, 2016. http://www.osvdb.org/5648 , Accessed January 16, 2016.

http://zone-h.org/mirror/id/23497697 , Accessed January 7, 2016.

https//cwe.mitre.org/data/definitions/319.html , Accessed November 29, 2015.

https://https.cio.gov/hsts , Accessed January 23, 2016.

https://www.cert.org/historical/advisories/ca-1997-27.cfm. (Accessed December 14, 2015).

IETF (2011), Prohibiting Secure Sockets Layer (SSL) Version 2.0, Available at *https://tools.ietf.org/html/rfc6176,* Accessed January 2, 2015.

IETF(2015), Deprecating Secure Sockets Layer Version 3.0, Available at *https://tools.ietf.org/ html/rfc7568,* Accessed November 22, 2015.

InfoSec Institute (2015). Biometrics: Today's Choice for the Future of Authentication, Available at *http://resources.infosecinstitute.com/biometrics-todays-choice-future-authentication*, Accessed July 2, 2015

InfoSec Institute(2014), Securing Cookies with HttpOnly and secure Flags, *Application Security*, Available at *http://resources.infosecinstitute.com/securing-cookies-httponly-secure-flags* , Accessed November 28, 2015.

Kearns, G. (2007), "Information Technology Audit & Forensic Techniques" ,PowerPoint Presentation presented at the ACG 6936 Summer 2007 in St. Petersburg College of Business, Available at *http://www.usfsp.edu/gkearns/ACG6936/ppt/investigative.ppt*, Accessed June,2015.

Kerr, D.,(2015), Obama asks for $14 billion to step up cybersecurity, Available at *http://www.cnet.com/news/obama-adds-14b-to-budget-for-stepped-up-cybersecurity*, Accessed March 16, 2015.

Kizza, J. ,M ,(2014), “Computer Network Security and Cyber Ethics”, fourth edition, pp 30-32

Kovacs, E.,(2015), New Attack on RC4-Based SSL/TLS Leverages 13-Year-Old Vulnerability, Available at http://www.securityweek.com/new-attack-rc4-based-ssltls-leverages-13-yearold-vulnerability, Accessed November 16, 2015.

Lazzez A. and Slimani,T. (2015), Forensics Investigation of Web Application Security Attacks , Available at *http://www.mecs-press.org/ijcnis/ijcnis-v7-n3/IJCNIS-V7-N3-2.pdf* (Accessed June,2015).

Lehtinen ,R. and Gangemi G.,T.,Sr. (2011), Computer Security Basics, 2nd Edition, O’Reilly, pp 24-26

McMillan, J.,(2009), What is the Difference Between an IPS and a Web Application Firewall?, Available at https://www.sans.org/security-resources/idfaq/what-is-the-difference-betweenan-ips-and-a-web-application-firewall/1/25, Accessed March 20,2015.

Mell, P. , Scarfone , K. and Romanosky, S. (2007), “A complete guide to the common vulnerability system version 2.0” , Available at *https://www.first.org/cvss/cvss-v2guide.pd*f, Accessed January,2015.

Microsoft (2012), Vulnerabilities in Microsoft Internet Information Services (IIS) Could Allow Information Disclosure (2733829), Available at *https://technet.microsoft.com/ library/security/ms12-073* , Accessed January 12, 2015.

Naik, N.,A., Kurundkar,G.,D., Khamitkar, S.,D., Kalyankar, N.,V. , “Penetration Testing: A Roadmap to Network”, *Journal of Computing*, Volume 1, Issue 1, December 2009, ISSN: 2151-9617,pp 1– 4

Nemati, H.(2008),’Information Security and Ethics-Concepts, Methodologies, Tools and

Applications', Information Science Reference, pp 73-75,346,351,543

OWASP (2013), OWASP Top Ten Project, Available at https://www.owasp.org/ index.php/Category:OWASP_Top_Ten_Project. Accessed May 3rd, 2015.

OWASP (2014), HTTPonly, Available at *https://www.owasp.org/index.php/HttpOnly*, Accessed November 20, 2015.

OWASP (2015), HTTP Strict Transport Security Cheat Sheet, Available at *https://www.owasp.org/index.php/HTTP_Strict_Transport_Security*, Accessed November 16, 2015.

OWASP(2013), Types of Cross-Site Scripting, Available at *https://www.owasp.org/index.php /Types_of_Cross-Site_Scripting*, Accessed January 19, 2016.

OWASP(2013),OWASP Testing Guide 4.0, Available at *https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf*, Accessed January,18 2015.

OWASP(2014), Cross site scripting prevention cheat sheet , Available at *https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet*, . Accessed June 30, 2015.

OWASP(2015),Clickjacking Defense Cheat Sheet, Available at *https://www.owasp.org/ index.php/Clickjacking_Defense_Cheat_Sheet*, Accessed November 30, 2015.

Quinn B.and Arthur C.(2011) , "playstation-network-hackers-data",*The Guardian,* Available at *http://www.theguardian.com/technology/2011/apr/26/playstation-network-hackers-data*, Accessed March 15, 2015.

Ramilli M.,(2012) "A Design Methodology for Computer Security Testing", Available at *http://amsdottorato.unibo.it/4438/4/Marco_Ramilli_Dissertation.pdf*, Accessed May 27,2015

RAPID7 (2000), Browsable Web Directory, Available at *https://www.rapid7.com/ db/vulnerabilities/http-generic-browsable-dir*, Accessed June 20,2015.

RAPID7 (2004), http trace method enabled, Available at *https://www.rapid7.com/db/ vulnerabilities/http-trace-method-enabled,* Accessed November 27, 2015.

Rouse, M.(2008) Biometric Verfication , Available at http://searchsecurity.techtarget.com/ definition/biometric-verification , Accessed December 14, 2015.

Saumi, S. ,Shreeraj, S.,Stuart, M. and Addison W. (2002), "Attacks and Defenses", Security,*Journal of Computing*, Volume 1, Issue 1, December 2009, ISSN: 2151-9617,pp 1-5

Schneier, B., (2015), Logjam (and Another) Vulnerability against Diffie-Hellman Key Exchange, Available at *https://www.schneier.com/blog/archives/2015/05/ the_logjam_ and_.html* , Accessed June 14,2015.

Sivathanu , G., Wright, C.P. and Zadok E. (2005) , "Ensuring Data Integrity in Storage: Techniques and Applications", Stony Brook University, Available at *https://www.fsl.cs.sunysb.edu/docs/integrity-storagess05/integrity.html* ,Accessed October 15, 2015.

Sotirov, A. (2008), Creating a rogue CA certificate , Available at , *http://www.phreedom.org/ research/rogue-ca*., Accessed November 30, 2015.

Svenhard, P. and Radaslic, A. (2012), "A penetration test of an Internet service provider", School of Information Science, Computer and Electrical Engineering, pp 5-25.

TechTarget(2007), Denial of Service , Available at *http://searchsoftwarequality.techtarget.com/ definition/denial-of-service*, Accessed May 28, 2015.

Ubuntu Security Notice (2012), USN-1604-1: MoinMoin vulnerabilities, Available at *http://www.ubuntu.com/usn/usn-1604-1*, Accessed August 6, 2015.

UKEssays(2015), "Sql Injection Attacks In 2011" ,*Computer Science Essay* ,Available at *http://www.ukessays.com/essays/computer-science/sql-injection-attacks-in-2011-computerscience-essay.php*,Accessed March 15, 2015.

Vacca J.R(2009), Computer and Information Security Handbook, Elsevier Inc., pp 63-70

Whitman M.,E. and Mattord ,H.J.(2012), Principles of Information Security, Fourth Edition, Cengage,pp 20-40.

APPENDICES

APPENDIX A

Clickjacking vulnerability details

The following pages sent by host 1 did not use an X-Frame-Options response header : -

http://41.204.53.162/aboutus/publications/lecture-series/2/2011-02-22/wheel-immutableconceptdialecticalhermeneutical-dileneatio

- http://***.***.53.162/aboutus/vcoffice/letters-writings/2006

- http:// ***.***.53.162/aboutus/vcoffice/letters-writings/2006/14-12-0

-http://***.***.53.162/aboutus/publications/lecture-series/1/2011-05-16/music-vitalingredienteducation-child

- http:// ***.***.53.162/aboutus/publications/lecture-series/1/2011-05-16

-http://***.***.53.162/aboutus/publications/lecture-series/1/2011-05-16/reflections-povertyandwealth-creation-ghana

- http://***.***.53.162/aboutus/publications/lecture-series/1

- http://***.***.53.162/aboutus/publications/lecture-series/1/2011-08-31

-http://***.***.53.162/aboutus/publications/lecture-series/0/2012-05-14/drivers-pedestriansandmechanics-interrogating-road-carnag

- http://***.***.53.162/aboutus/publications/2014-09-24

-http://***.***.53.162/aboutus/publications/2014-09-24/university-cape-coast-goldenjubileemessage

- http://***.***.53.162/events/publications/

- http://***.***.53.162/events/vcoffice/letters-writings/2010/30-11

- http://***.***.53.162/events/vcoffice/letters-writings/2008/6-11

- http://***.***.53.162/events/vcoffice/letters-writings/2008

- http://***.***.53.162/events/vcoffice/letters-writings/2008/5-11

- http://***.***.53.162/events/vcoffice/letters-writings/2007/9-11 -
  http://***.***.53.162/events/vcoffice/letters-writings/2007/8-11

For host 2, Nessus reported that the following pages did not use an X-Frame-Options response header :

- http://***-***.80.197.dedicated.codero.net/pipermail/
- http://***.***.80.197.dedicated.codero.net/

For host 3, the following pages did not use an X-Frame Options response header

- http://***.***.58.115/csuc/p/faculty/faculty-profiles/faculty-of-humanities
- http://***.***.58.115/csuc/p/academics/university-library/online-resources
- http://***.***.58.115/csuc/p/academics/research-conferences
- http://***.***.58.115/csuc/p/csuc-chaplaincy/from-the-chaplain

-http://***.***.58.115/csuc/p/academics/center-for-leadership-professional-development/ourvision-mission-objectives

- http://***.***.58.115/csuc/p/csuc-chaplaincy/the-faith-and-practice-programme-fapp
- http://192.185.58.115/csuc/p/academics/center-for-leadership-professional-development/introduction
- http://192.185.58.115/csuc/p/news-articles/downloads
- http://192.185.58.115/csuc/p/news-articles/downloads/all-application-forms
- http://192.185.58.115/csuc/p/academics/center-for-leadership-professional-development
- http://192.185.58.115/csuc/p/academics/school-of-graduate-studies/welcome-message-from-thedean
- http://192.185.58.115/csuc/articles/archived
- http://192.185.58.115/csuc/articles/archived/csuc-an-ideal-place-of-learning

- http://192.185.58.115/csuc/articles/news/fake-certificates-who-wants-them
- http://192.185.58.115/csuc/p/academics/school-of-business/department-of-marketing-logisticscorporatestrategy
- http://192.185.58.115/csuc/p/academics/school-of-business
- http://192.185.58.115/csuc/p/academics/school-of-business-studies
- http://192.185.58.115/csuc/p/academics/school-of-business-studies/department-of-accountingfinance
- http://192.185.58.115/csuc/p/academics/faculty-of-health-and-applied-sciences/department-ofnursing
- http://192.185.58.115/csuc/p/academics/faculty-of-health-and-applied-sciences/department-ofcomputerscience
- http://192.185.58.115/csuc/p/academics/faculty-of-humanities/department-of-theology
- http://192.185.58.115/csuc/p/academics/faculty-of-humanities
- http://192.185.58.115/csuc/p/academics/faculty-of-humanities/department-of-communicationstudies

Host 4 had the following web pages missing the x-frame options response header

- http://***.***.27.33/dotnetnuke/register.aspx
- http://***.***.27.33/dotnetnuke/AboutUs.aspx
- http://***.***.27.33/dotnetnuke/login.aspx
- http://***.***.27.33/dotnetnuke/Home.aspx
- http://***.***.27.33/dotnetnuke/
- http://***.***.27.33/dotnetnuke/AboutUs/StyleGuide.aspx
- http://***.***.27.33/dotnetnuke/AboutUs
- http://***.***.27.33/dotnetnuke/OurServices.aspx
- http://***.***.27.33/dotnetnuke/ContactUs.aspx
- http://***.***.27.33/dotnetnuke/terms.aspx

- http://***.***.27.33/dotnetnuke/privacy.aspx

For host 5 these were the pages missing the x-frame options response header

The following pages did not use an X-Frame-Options response header :

- http://******/images/prettyPhoto/light_square/
- http:// ******/images/thumbnails/
- http:// ******/uploads/gallery/gall_watermark/
- http:// ******/images/prettyPhoto/
- http:// ******/images/innerpage_banner/
- http:// ******/images/header.php
- http:// ******/css/aboutus.php
- http:// ******/css/_notes/
- http:// ******/includes/old_header.php
- http:// ******/apps/cgi-bin/
- http:// ******/includes/old_footer.php
- http:// ******/includes/index_demo.html
- http:// ******/lib/class.wsdl.php
- http:// ******/includes/header_19.06.14.php
- http:// ******/lib/class.soapclient.php
- http:// ******/includes/header.php
- http:// ******/lib/class.soap_server.php
- http:// ******/lib/class.soap_parser.php
- http:// ******/lib/class.soap_fault.php
- http:// ******/includes/client-17-02-2014.php
- http:// ******/js/accordion_js/

- http:// ******/uploads/pages/
- http:// ******/uploads/news/
- http:// ******/uploads/members/
- http:// ******/uploads/home/
- http:// ******/uploads/gallery/
- http:// ******/apps/
- http:// ******/test/
- http:// ******/admin/
- http:// ******/doc/
- http:// ******/includes/
- http:// ******/lib/
- http:// ******/webmail
- http:// ******/controlpanel
- http:// ******/temp/
- http:// ******/uploads/
- http:// ******/js/
- http:// ******/_notes/
- http:// ******/css/
- http:// ******/images/
- http:// ******/uploads/adv/
- http:// ******/uploads/adv_banners/
- http:// ******/uploads/articles/
- http:// ******/uploads/personals/
- http:// ******/uploads/policy/
- http:// ******/uploads/procurement/

- http:// ******/includes/ad_banner.php
- http:// ******/lib/class.nusoap_base.php
- http:// ******/includes/footer.php
- http:// ******/includes/client1.php

APPENDIX B Cookies without HTTPonly flag

The following cookies did not set the HttpOnly cookie flag for host 1 :

```
Name : PHPSESSID

Path : /

Value : d3b9puhbil4ce0oi187i6dvsl0

Domain    :

Version : 1

Expires :

Comment :

Secure : 0 Httponly

: 0

Port :


Name : SESSb3c2ee1763a81a06b6a81418730addd0

Path : /

Value : d5s2ha06hbkl2ugrcpvr0q4p81

Domain :

Version : 1

Expires : Wed, 18-Nov-2015 23:39:16 GMT

Comment :

Secure : 0

Httponly : 0

Port :


Name : SESScc1da4ccd9a696c0ac5218e51872f72b

Path : /
```

```
Value : qdvuubpjlodhok13lr985mo3s5

Domain :

Version : 1

Expires : Wed, 18-Nov-2015 23:39:09 GMT

Comment :

Secure : 0 Httponly

: 0

Port :


Name : SESS6d0f3ee2a9f83a7e3e20e72d1581be88

Path : /

Value : pt6778akm42v4c2li8mq7961q7

Domain :

Version : 1

Expires : Wed, 18-Nov-2015 23:37:38 GMT

Comment :

Secure : 0 Httponly

: 0

Port :

Name : SESSa3027455b5be37d365a480d59dd774b0

Path : /

Value : t2tfu3vbjvc2ajdm4ajacudth4

Domain :
```

```
Version : 1
Expires : Wed, 18-Nov-2015 23:37:41 GMT
Comment :
Secure : 0 Httponly
: 0
Port :
```

For host 2 only Nikto identified the presence of 1 cookie without the httponly flag. Nikto indicated that a cookie with the name PHPSSESID was created without the httponly flag.

The following cookies generated by host 3 web server did not set the HttpOnly cookie flag :

```
Name : port
Path : / Value
: 2082
Domain :
Version : 1
Expires :
Comment :
Secure : 0
Httponly : 0
Port :5602 – W
```

For host 5 only 1 cookie was identified by nessus as not having the httponly flag and it is shown below

Name : PHPSESSID

Path : /

Value : 0dfa5b191dd98a7dad2120afe2984bcc

Domain   :

Version : 1

Expires :

Comment :

Secure : 0 Httponly

: 0

Port :

APPENDIX C

Multiple web server interesting web documents

The following interesting files/directories were found on host 1 web server.

- sitemap.xml:
- downloads/:

- install/:
- library/
- /install/install.php
- UPGRADE.txt
- install.php
- install.php: install.php file.
- LICENSE.txt: License file.
- xmlrpc.php: xmlrpc.php.

For host 3 , the interesting files/directories were :

- css/
- downloads/
- file/
- js/
- new/
- img-sys/
- java-sys/

Interesting files for host 5 were :

- test/
- test.html
- temp/
- includes/
- apps/
- admin/

APPENDIX D

Multiple Web Server robots.txt Remote Information Disclosure

Details of the information disclosure in the robots.txt file on host 1 were:

Directories

Disallow: /utils/

Disallow: /includes/

```
Disallow: /misc/

Disallow: /modules/

Disallow: /profiles/

Disallow: /scripts/

Disallow: /themes/
```

Files

```
Disallow: /CHANGELOG.txt

Disallow: /cron.php

Disallow: /INSTALL.mysql.txt

Disallow: /INSTALL.pgsql.txt

Disallow: /install.php

Disallow: /INSTALL.txt Disallow:

/LICENSE.txt Disallow:

/MAINTAINERS.txt

Disallow: /update.php

Disallow: /UPGRADE.txt

Disallow: /xmlrpc.php
```

Paths

```
Disallow: /admin/

Disallow: /comment/reply/
```

```
Disallow: /filter/tips/
Disallow: /logout/
Disallow: /node/add/
Disallow: /search/
Disallow: /user/register/
Disallow: /user/password/
Disallow: /user/login/
Disallow: /?q=admin/
Disallow: /?q=comment/reply/
Disallow: /?q=filter/tips/
Disallow: /?q=logout/
Disallow: /?q=node/add/
Disallow: /?q=search/
Disallow: /?q=user/password/
Disallow: /?q=user/register/
```