

2021

INFORME DE PENTESTING



ÍNDICE

- 01** Introducción
- 02** Metodología
- 03** Análisis General de Siscap2
- 04** Plan de ataque
- 05** Pentesting
- 06** Recomendaciones

INTRODUCCIÓN

Los objetivos de la presente práctica, es aplicar los conocimientos del primer parcial de la materia Seguridad de Software.

Dicho semestre abarcó los principios básicos del "Hacking Ético", por lo cual en este trabajo se estará analizando una IP de una página web, con la cual trataremos de buscar y encontrar vulnerabilidades, además de dar un informe con recomendaciones para mejorar la seguridad de la misma.

METODOLOGÍA

Para la realización de esta práctica, es necesario tener las herramientas necesarias «valga la redundancia», por lo que se hizo uso de un software de virtualización, en este caso "Virtualbox", en la cual se montó una la imagen de Kali Linux,

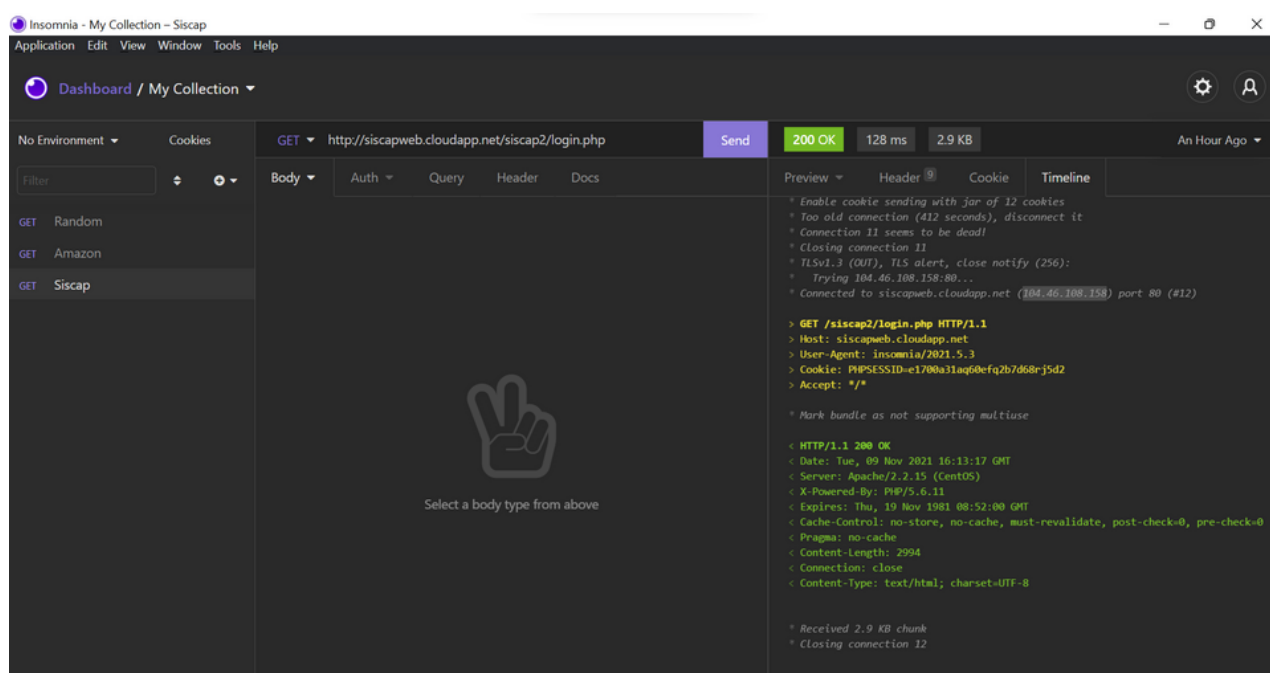
Una vez teniendo Kali Linux, también se hará uso de la aplicación Insomnia para saber la IP de la página web seleccionada, además de otra información que será de ayuda.

Una vez teniendo todas las herramientas necesarias, se procedió a tomar la página web de "Siscap2", la cual es un aula virtual, y analizar el URL en Insomnia, dándonos como resultado la IP de donde se conecta.

Ya teniendo la IP, se procedió a usar el comando "nmap" en la consola de comandos de Kali Linux, dándonos como resultado numerosos puertos abiertos, además de las versiones de sus respectivas tecnologías utilizadas tanto para la Base de Datos, como para el uso del protocolo SSH.

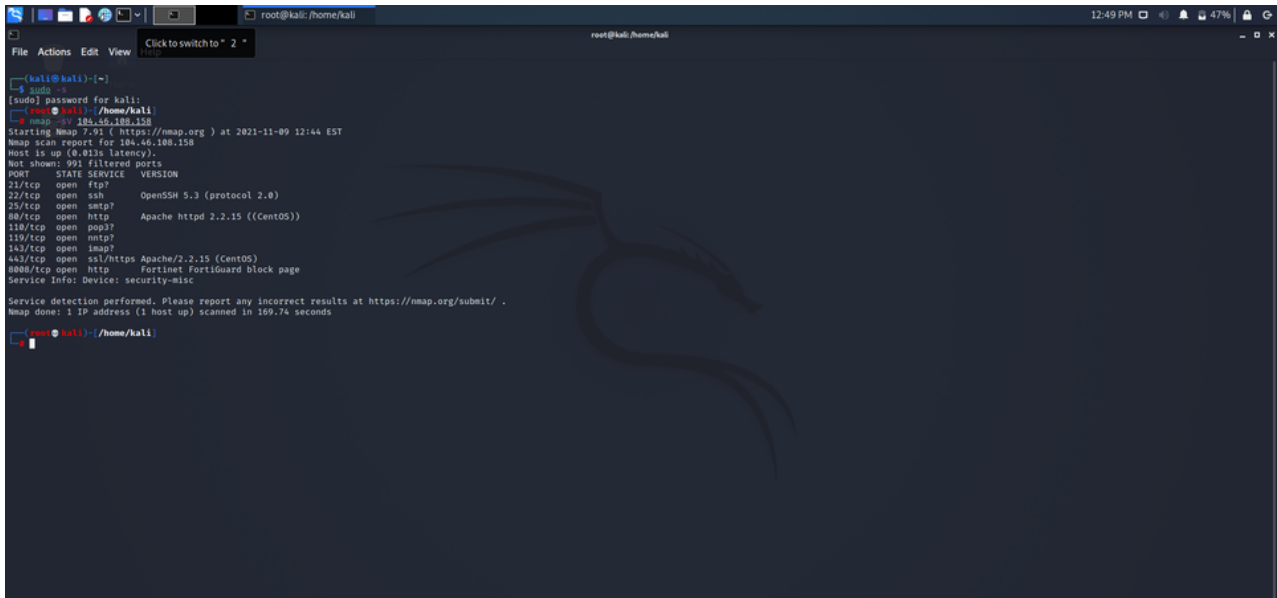
ANÁLISIS GENERAL

Como análisis general del portal Siscap2, tenemos que analizar la información arrojada tanto de Insomnia como de la consola de comandos de Kali Linux.



Con respecto a la información dada de Insomnia, se puede observar que desde el primero momento o primera parte del análisis, ya tenemos de primera mano, toda la información del servidor, desde que se está usando Apache en su versión 2.2.15 con un Sistema Operativo CentOS, y con solo esta información ya podemos empezar con las vulnerabilidades, pero es importante tener toda la información posible y es por eso que de igual forma usamos "nmap" en Kali Linux.

ANÁLISIS GENERAL



```
(kali@kali)~$ sudo -s
(sudo) password for kali:
(sudo) # kali ~ /home/kali
# nmap -sV 104.46.188.158
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-09 12:44 EST
Nmap scan report for 104.46.188.158
Host is up (0.013s latency).
Not shown: 991 filtered ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp?
22/tcp    open  ssh          OpenSSH 5.3 (protocol 2.0)
25/tcp    open  smtp?
80/tcp    open  http         Apache httpd 2.2.15 ((CentOS))
110/tcp   open  pop3?
119/tcp   open  nntp?
143/tcp   open  imap?
443/tcp   open  ssl/http     Apache/2.2.15 (CentOS)
8080/tcp   open  http         Fortinet FortiGuard block page
Service Info: Device: security-misc

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 169.74 seconds
# kali ~ /home/kali
```

Al hacer el mismo tipo de análisis con las herramientas de Kali Linux, que en este caso es el comando "`nmap -sV <IP>`" en el cual se le agrega `-sV` para que de esta forma no solo analice los puertos, sino las versiones de los mismos. Es por esto que ahora tenemos más información además de los diferentes puertos que están expuestos, sino que incluso tenemos el conjunto de aplicaciones que su utilizan para el protocolo SSH, además de la versión del mismo.

PLAN DE ATAQUE

De forma muy simple y breve, ya teniendo la información anteriormente expuesta, es posible empezar con buscar las diferentes vulnerabilidades tanto de la versión del protocolo SSH, como de la Base de datos de Apache.

En este caso tendremos como plan de ataque, empezar con la Base de Datos, que como se mencionó anteriormente, es simplemente una búsqueda por Google acerca de las vulnerabilidades de Apache en su versión 2.2.15, y si es posible, haciendo incapié en el sistema operativo CentOS. Teniendo como resultado la siguiente lista con las diferentes vulnerabilidades, junto con la puntuación de la gravedad del mismo, como el nivel de complejidad para hacer uso de esta vulnerabilidad.

https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-459994/Apache-Http-Server-2.2.15.html

Pasando al protocolo, tenemos lo siguiente:

<https://vulmon.com/searchpage?page=1&q=Openbsd%20Openssh%205.3&sortby=byrelevance&scoretype=cvssv2>

PENTESTING

De forma complementaria al testing inicial en el plan de ataque, ya que cabe aclarar que en el punto anterior se implmentó gran parte de la práctica del pentesting, en el mismo SO de Kali Linux, tiene un comando expecíficamente para encontrar las diferentes vulnerabilidades existentes directamente en la consola de comandos.

Este comando es el siguiente:

`$ searchsploit < VERSION >`

Aplicando dicho comando a lo encontrado anteriormente con "nmap", tenemos los siguiente:

```
kali@kali: ~  
$ nmap -sV 10.10.10.10  
Nmap done: 1 IP address (1 host up) scanned in 104.91 seconds  
--(kali@kali):~--  
$ searchsploit OpenSSH 5.3  
Exploit Title | Path  
-----  
OpenSSH 2.3 < 7.7 - Username Enumeration | linux/remote/45213.py  
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC) | linux/remote/45218.py  
OpenSSH < 6.6 SFTP (x64) - Command Execution | linux_x86_64/remote/45000.c  
OpenSSH < 6.6 SFTP - Command Execution | linux/remote/45001.py  
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation | linux/local/40962.txt  
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading | linux/remote/40963.txt  
OpenSSH < 7.7 - User Enumeration (2) | linux/remote/45009.py  
Shellcodes: No Results  
--(kali@kali):~--  
$ searchsploit Apache 2.2.15  
Exploit Title | Path  
-----  
Apache * PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution | php/remote/29290.c  
Apache * PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner | php/remote/29310.py  
Apache 2.0.18 mod_proxy - Reverse Proxy Security Bypass | linux/remote/36663.txt  
Apache < 2.0.64 / < 2.2.21 mod_setenvif - Integer Overflow | linux/dos/41769.txt  
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak | linux/webapps/42745.py  
Apache CAF < 2.5.10/2.6.7/2.7.4 - Denial of Service | multiple/dos/26718.txt  
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow | unix/remote/764.c  
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1) | unix/remote/47080.c  
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2) | linux/webapps/39642.txt  
Apache OpenMeetings 1.9.x < 3.1.0 - 'ZIP' File Directory Traversal | multiple/webapps/18329.txt  
Apache Struts 2 < 2.3.1 - Multiple Vulnerabilities | multiple/remote/44556.py  
Apache Struts 2.0.1 < 2.3.33 / 2.5 < 2.5.10 - Arbitrary Code Execution | multiple/remote/41698.rb  
Apache Struts2 2.0.0 < 2.3.15 - Prefixed Parameters OGNL Injection | multiple/webapps/44583.txt  
Apache Tomcat < 5.5.17 - Remote Directory Listing | multiple/remote/2061.txt  
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal | unix/remote/14489.c  
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC) | multiple/remote/6229.txt  
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1) | windows/webapps/42953.txt  
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2) | jsp/webapps/42966.py  
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC) | linux/dos/26906.txt  
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution | linux/remote/74.pl  
Shellcodes: No Results  
--(kali@kali):~--  
$
```


RECOMENDACIONES

La primer recomendación para el portal de Siscap2, es hacer uso de un certificado SSL, ya que lo primero que se puede notar al entrar a la página, es que no tiene este certificado, el cual es de vital importancia y más para un portal en el que se hace uso de información personal, ya no solo de alumnos, si no igual de los docentes.

En cuanto a los puertos, se sabe que no deberían de estar expuestos de esa forma, ya que no solo están dando información de qué numerosos puertos son "accesibles", sino que incluso dan la información de qué tecnologías están usando para dichos puertos, aumentando así los riesgos de cualquier tipo de robo de información, mediante exploits/vulnerabilidades.