EJERCICIO 1

1: Ataque a HTTP

python3 -m http.server y conectarse desde un navegador a http://127.0.0.1:8080.

Capturar las tramas usando wireshark.

2. Ataque a FTP ("File Transfer Protocol")

Desde una ventana de la terminal hacer:

1) "sudo su" y "msfdb init && msfconsole"

2) use auxiliary/server/capture/ftp

3)set srvhost 127.0.0.1

4)set banner Servidor FTP de [Poner aquí vuestro nombre y Apellidos]

5) exploit

Desde otra ventana hacemos:

1) ftp 127.0.0.1

2) Introducimos usuario y contraseña y desde la otra ventana podremos ver la información. Además capturamos las tramas con Wireshark para ver que podemos verlo todo en claro.

3. Ataque a TELNET

Desde una ventana de la terminal teniendo iniciado metasploit como se indica en el anterior ejercicio hacemos:

1) use auxiliary/server/capture/telnet

2) set srvhost 127.0.0.1

3) set banner Servidor TELNET de [Poner aquí vuestro nombre y Apellidos]

4) exploit

Desde otra ventana hacemos:

1) nc 127.0.0.1 23

2) Pulsamos ENTER

3) Introducimos los datos de inicio de sesión y capturamos todo el tráfico con Wireshark

EJERCICIO 2

Desde una ventana de la terminal:

1) "sudo su" y "msfdb init && msfconsole"

2) Seguir los cinco primeros python3pasos del ejercicio 2

Desde otra ventana de la terminal:

3) python3 -m http.server

Por último desde otra ventana ejecutamos el comando nmap:

4) sudo nmap -v -A -O 127.0.0.1

EJERCICIO 3

a) 81dc9bdb52d04dc20036dbd8313ed055

1) Necesitamos averiguar el formato del hash por lo que hacemos hash-identifier del hash y nos devuelve MD5.

2) Hacemos john --list=formats | grep -i "MD5" y como nos dan una pista que es subtipo raw tenemos dos opciones entre las que nos salen: Raw-MD5 o Raw-MD5u, así que probaremos con ambas para ver cuál de ellas es.

3) Empezamos con Raw-MD5. Guardamos en un archivo hash.txt el hash: echo 81dc9bdb52d04dc20036dbd8313ed055 > hash.txt.

4) john --wordlist=' /usr/share/wordlists/rockyou.txt' --format=Raw-MD5 hash.txt

5) Obtenemos la contraseña: 1234

6) Si hacemos lo mismo con Raw-MD5u obtenemos la contraseña: markinho..*7¡Vamos!

7) Pero si observamos en el primer caso nos sale 1g que indica una contraseña descifrada y en el segundo 0g lo que indica que no se ha descifrado ninguna contraseña.

b) a77eb3defefc90c462a8d7cf63b950c3a73e350a

Igual que el anterior pero debemos añadir la regla =0oo00 al fichero /etc/john/john.conf justo debajo de List.Rules:Wordlist

1) Al hacer hash-identifier obtenemos dos posibles hash: SHA-1 o MySQL5 - SHA-1(SHA-1($pass))

2) Probamos con SHA-1: john --list=formats | grep -i "SHA1"

3) john --wordlist=' /usr/share/wordlists/rockyou.txt' --format=Raw-SHA1 --rules hash.txt

4)Obtenemos la contraseña: 0wen11..0smara

c)6YmAFQjzlBmpUlNS1$ui9s3a7UO/eKK7BEhEeH9zc9VVKiG4QsLE45uJxjgPgagl7RJEAaUfnBwUC/tjTuOuOMTKJCy2GhBXhv7qUPa/

1) Igual que el 1 pero no podemos susar hash-identifier: Para saber que un hash es **SHA-512**, puedes observar su formato y la forma en que está estructurado. Los hashes de contraseñas almacenadas en sistemas Linux con **SHA-512** suelen seguir este formato:

6<salt>$<hash>

En este caso, el 6 indica que el algoritmo utilizado es **SHA-512** (el 6 es el código para SHA-512 en el sistema de **crypt(3)**). El **salt** es una cadena aleatoria que se utiliza para hacer el hash más seguro, y el **hash** es la parte que representa la contraseña cifrada. Es decir es sha512crypt

2) john --list=formats | grep -i "SHA512"

3) john --wordlist=’ /usr/share/wordlists/rockyou.txt’ --format=sha512crypt hash.txt

4) La contraseña es: ladyluck1..jm4ever