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. ¿Qué utilidad tiene esto?

Para el caso de la criptografía asimétrica serviría para demostrar que eres un usuario (autentificación) ya que te pedirían que les cifrases un texto con la clave privada y el otro usuario debería descifrarlo con la pública para poder comprobar que lo has cifrado correctamente. El realizar el md5 es útil para comprobar que el texto no ha sido modificado a la hora de enviarlo ya que si modificas un poco el texto cifrado a la hora de descifrarlo obtienes una solución muy distinta.

(autentificación. Se aplica tanto a datos como a usuarios. En el caso de los datos, p.e. un mensaje, nos permite saber si éste ha sido alterado o no, mientras que en el caso de los usuarios se trata de verificar su identidad. En este sentido, el mecanismo más básico aplicado a datos es el digest que consiste en obtener un código que siempre es el mismo si se aplica sobre los mismos datos (una alteración de los datos implica un código distinto). Los algoritmos clásicos son MD5 y SHA-1. Si al digest se le aplica una clave entonces tenemos un MAC (message authentication code). Si se protege con un esquema asimétrico entoces obtenemos una firma digital, cuyo objeto es asegurar la procedencia de los datos (puesto que se se calcula con una clave privada y se verifica con la clave pública correspodiente) y también que los datos no han sido alterados (ya que lo que se encripta/desencripta es un digest) Internet

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?

Un carácter x indica que la contraseña cifrada se almacena en el archivo /etc/shadow. Tenga en cuenta que debe usar el comando passwd para calcular el hash de una contraseña escrita en la CLI o para almacenar / actualizar el hash de la contraseña en el archivo /etc/shadow.

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?

Lo que haríamos es introducir en el archivo de hosts conocidos nuestra clave pública, gracias a esto podríamos acceder al servidor a través de ssh con nuestra clave privada sin necesidad de introducir contraseña.

Comando:
$ssh <usuario>@><host> -i "ruta donde tenemos la clave privada"

Pregunta 4:

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

RSA

DES

MD5

BASE64

AES

SHA1

SHA256

HTTPS

BLOWFISH

RUBBERDUCKIE

FTP

OPENSSL

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’
```

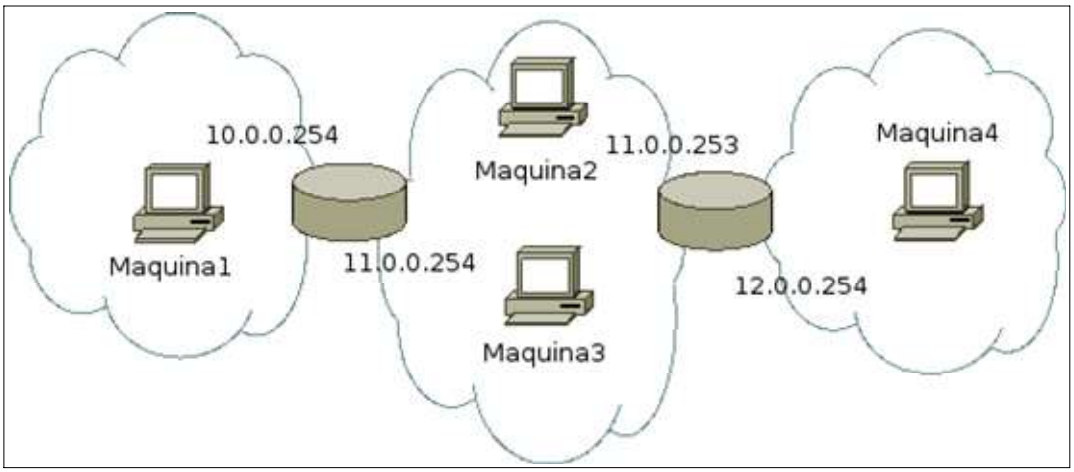
En caso afirmativo explique cómo se compondría y por qué funcionaría.

Al estar obteniendo la variable con post no podríamos realizar una inyección de código con la url, esto lo podríamos realizar en el caso de obtener los parámetros con get. Sin embargo si pudiésemos insertar la consulta sql ‘ “lalala”) or 1==1 or (1’ en la caja de donde se obtuviese el valor con el post estaríamos realizando la inyección. Esto ocurre porque en el código php no se está haciendo ninguna validación de entrada.

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
5         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.

Para poder realizar esto con el tráfico cifrado vamos a trabajar con túneles SSH.

Para poder trabajar con esto vamos a tener que modificar los iptables.

En la maquina 3 modificamos la iptable de tal forma que el Gateway de los paquetes con destino 12.0.0.0 mask 255.255.255.0 sea 11.0.0.253:
En PC3

$route del -n 12.0.0.0 netmask 255.255.255.0 gw 11.0.0.254
$route add -n 12.0.0.0 netmask 255.255.255.0 gw 11.0.0.253

De esta forma ya podríamos ir del PC1 al PC4

Para poder ir de PC4 a PC1 haremos que los paquetes que vayan con dirección 10.0.0.0 mask 255.255.255.0 tengan un Gateway a 11.0.0.254:
En PC3

$route add -n 10.0.0.0 netmask 255.255.255.0 gw 11.0.0.254

Una vez tenemos configuradas las tablas de rutas abrimos los túneles:

Abrimos un túnel entre la maquina1 y la maquina3:
En PC1

$ ssh tlm@10.0.0.10 -L 5555:11.0.0.10:6666

Abrimos un túnel entre la maquina3 y la maquina4:
En PC3

$ ssh tlm@11.0.0.10 -L 6666:12.0.0.10:80

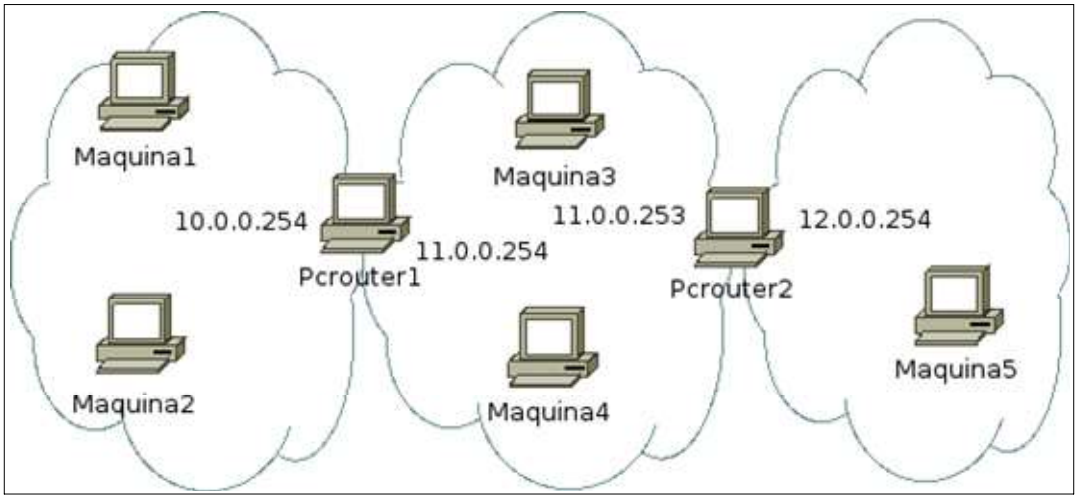
Por último, en deberíamos modificar el navegador web de PC1 cambiando el proxy a puerto 7777 con 127.0.0.1 (localhost)

Y en PC1 conectarnos con ssh -D 7777 tlm@10.0.0.10

Y en el navegador escribiríamos en url 12.0.0.10 y así tendríamos acceso a la web desde el PC1

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 está trabajando mientras pasa por detrás su jefe. Eso sí, 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 está 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 sí, 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.

Opcion1 (modificamos las iptables):

Modifico desde PC1 la iptable del router 1 para poder recibir y enviar paquetes desde la maquina 1.

$ iptables -A input -s 10.0.0.10/255.255.255.0 -d any -p tcp – dport 443 -j ACCEPT
$ iptables -A input -s 10.0.0.10/255.255.255.0 -d any -p tcp – dport 80 -j ACCEPT

$ iptables -A output -s any -d 10.0.0.10 -p tcp – dport 443 -j ACCEPT
$ iptables -A output -s any -d 10.0.0.10 -p tcp – dport 80 -j ACCEPT

Modifico desde PC1 la iptable del router 2 para poder recibir y enviar paquetes desde la maquina 1.

$ iptables -A input -s 10.0.0.10/255.255.255.0 -d any -p tcp – dport 443 -j ACCEPT
$ iptables -A input -s 10.0.0.10/255.255.255.0 -d any -p tcp – dport 80 -j ACCEPT

$ iptables -A output -s any -d 10.0.0.10 -p tcp – dport 443 -j ACCEPT
$ iptables -A output -s any -d 10.0.0.10 -p tcp – dport 80 -j ACCEPT

Hacemos un túnel de la maquina1 a la maquina3:

$ ssh tlm@10.0.0.254 -L 5555:11.0.0.10:80

Ponemos en el navegador la url:

http://10.0.0.10:5555

Como sabemos que este servidor tiene un programa llamado visualiza.php que obtiene datos por get en la url escribiríamos:

http://10.0.0.10:5555/visualiza.php?maquina= www@11.0.0.253 -L 80:12.0.0.10:80;

Opcion2 (no modificamos las iptables):

Como en la maquina 2 hay un operario poniendo las credenciales de usuario cada 5 mins, hacemos un SSLstrip para poder tener los paquetes y ettercap para poder sniffar los paquetes y así tener las cookies de sesión de la maquina2.

Ahora que estamos en la maquina2 realizamos los comando para crear un túnel de esta a

maquina3 o 4.

Posibles cosas para realizar desde la maquina4:

Por url poner http://127.0.0.1/ejecuta.php?cmd= Comandos posibles ;

Comandos posibles:

Posibles cosas para realizar desde la maquina3:

Con la clave publica de la maq3 la añadimos por visualiza.php en la maquina5 en el fichero de knownhost.

Otra opción:

Modificación de iptables desde maquina1.

SSLStrip y ettercap para snifar las cookies de sesión de la maquina2 y entonces hacer la modificación de las iptables desde la maquina2.

GLOBAL:
Creamos 2 tuneles desde la maquina1, uno a la maquina3 y otro a la maquina4.

Una vez que tenemos estos dos túneles descargamos la clave publica de la maquina3 (clavepub.pub).

Desde la maquina1 nos conectamos a la maquina4 y en la url ponemos http://127.0.0.1/ejecuta.php?cmd= clavepub.pub >> /var/mykeys/clavepub

Al hacer esto ya tenemos la clave publica de la maquina3 en la maquina5.

Como desde la maquina3 podemos ejecutar ssh podemos meternos desde la maquina1 a la 5.

En la url introudicriamos http://11.0.0.10/visualiza.php?maquina=tlm@12.0.0.10; curl /home/tlm/notasdeexamen.xls