Tunel de 2 gatos

Enunciado

> Hemos detectado una comunicación sospechosa. ¿Serías capaz de averiguar que se han entregado?

Files

- pingMe.pcapng

Solución

Si abrimos la captura con la herramienta Wireshark podemos observar que la mayor parte del tráfico es tráfico DNS, haciendo peticiones un tanto sospechosas. Estas peticiones parecen estar en hexadecimal por tanto puede ser un gran comienzo para averiguar algo.

Para extraer estas consultas utilizamos **TShark**, un analizador de tráfico de red de línea de comando que puede instalarse mediante apt.

```
$ tshark -T fields -e dns.qry.name -r tunel2gatos.pcap

60aa01caf2adc06da8.sugusuva.es
60aa01caf2adc06da8.sugusuva.es
f36301caf2adc06da8.sugusuva.es
f36301caf2adc06da8.sugusuva.es
ab3401caf2adc06da8.sugusuva.es
ab3401caf2adc06da8.sugusuva.es
<snip>
```

Podemos observar que los paquetes están repetidos ya que hay consulta y respuesta además de tener ***.sugusuga.es*** que tampoco queremos y algun ***'.'*** a mayores que hay entre medias de algún dominio, por tanto vamos a eliminarlo de la salida con bash.

```
$ tshark -T fields -e dns.qry.name -r tunel2gatos.pcap | sed 's/.sugusuva.es//g'
| sed 's/\.//g' | uniq
```

```
60aa01caf2adc06da8
f36301caf2adc06da8
ab3401caf2adc06da8
072e01caf2adc06da8
92fd01caf2adc06da8
a84f01caf2adc06da8
b9a801caf2adc06da8
<snip>
```

Si tratamos de convertir estas salida vemos que mucha parte de ella no es ASCII, pero si observamos información que parece ser salida de comandos.

Investigando un poco esta forma sospechosa de comunicación vemos que se trata de un C2 llamado dnscat2 que también es información que da la pista.

Si investigamos el funcionamiento de éste y como envía la información vemos que los primeros 9 bytes de cada paquete son información del C2 que no son útiles y por eso aparecen cosas ilegibles. Procedemos a eliminarlo:

```
$ tshark -T fields -e dns.qry.name -r tunel2gatos.pcap | sed 's/.sugusuva.es//g'
| sed 's/\.//g' | uniq | cut -c 19- | xxd -r -p
sh (kali)��5kali
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20>
(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),
100(users),101(netdev),112(wireshark),114(bluetooth),123(scanner),130(vboxsf),13
1(kaboxer)
/home/kali
colaboradores.txt
credenciales.txt
usuarios.txt
zip -e -P OracleVMVirtualBox! confidencial.zip credenciales.txt usuarios.txt
colaboradores.txt
  adding: credenciales.txt (deflated 72%)
  adding: usuarios.txt (deflated 72%)
  adding: colaboradores.txt (deflated 73%)
```

Vemos que lo que se ha hecho ha sido ejecutar una serie de comandos como por ejemplo zip a 3 ficheros y protegerlo con contraseña.

A mayores en la captura puede verse que sigue habiendo datos binarios que tienen una firma de ***PK*** lo que da lugar a un archivo ***ZIP (50 4B 03 04)***.

Vamos a eliminar los primeros bytes que contienen la información de los comandos y quedarnos solo con los datos del ZIP, que guardaremos como un fichero .zip.

Para ello me guarde la salida anterior en un fichero de texto, elimine las líneas que contenian la información de los comandos y lo utilice para la entrada del siguiente comando:

```
$ xxd -r -p fichero.txt > out.zip
```

Ahora intentamos descomprimirlo con unzip

```
$ unzip out.zip
```

Nos pide una contraseña, pero ya la sabemos porque es la que hemos visto anteriormente, la introducimos y nos genera los 3 ficheros que vimos en los comandos. En total tienen miles de líneas con nombres de usuario y contraseñas por tanto como sabemos el formato de la flag vamos a tratar de encontrarlo utilizando ***grep***.

```
$ grep -r SUGUS .
```

Esto nos devuelve la flag.

> ***SUGUS{1_l0v3_Ma61c_NumB3r5_aNd_DNS!}***