Universidad Nacional Autónoma de México

Plan de Becarios en Seguridad de la Información

14° Generación

Programación con PERL

Reporte del proyecto - Log Watcher

Equipo 2

-

Rodolfo Baruch Guerra

Edgar Hernández Vásquez

Alexis Brayan López Matías

Descripción del servicio

El servicio desarrollado, cumple con la funcionalidad de revisar constantemente el log de un servicio web (Apache o NGINX), con la finalidad de identificar y mitigar ataques de fuerza bruta que se pudieran llegar a presentar en algún recurso del servidor.

Cuando se identifica un intento de este ataque, el servicio hace un bloqueo temporal a la IP atacante por medio de IPtables. Los tiempos de bloqueo, número de intentos fallidos máximos y demás configuraciones del servicio, son explicadas en el siguiente punto del documento con más detalle.

Todas las dependencias necesarias por el servicio están dentro del archivo “install.sh”, por lo que el único requisito por parte del equipo donde se va a instalar, es tener una conexión a internet para poder descargar el servicio y las dependencias necesarias para su correcto funcionamiento.

Cómo instalar el servicio dentro del sistema

Para instalar el servicio generado por el equipo, es necesario clonar el siguiente repositorio: https://github.com/barvch/proyecto_logs_perl

```
git clone https://github.com/barvch/proyecto_logs_perl.git
```

Una vez clonado debemos de contar con lo siguientes archivos:

```
kali@kali:~/Documents/perl/proyecto_logs_perl$ ls
apache_eq2.conf  apache_eq2.pl  apache_eq2.service  BlockedIPs  install.sh  IPBlocker.pm  README.md  UnblockIP.pl
kali@kali:~/Documents/perl/proyecto_logs_perl$
```

Ahora, es necesario ejecutar el binario “install.sh”, para realizar esta operación, damos los permisos de ejecución pertinentes al binario ejecutando:

```
chmod u+x install.sh
```

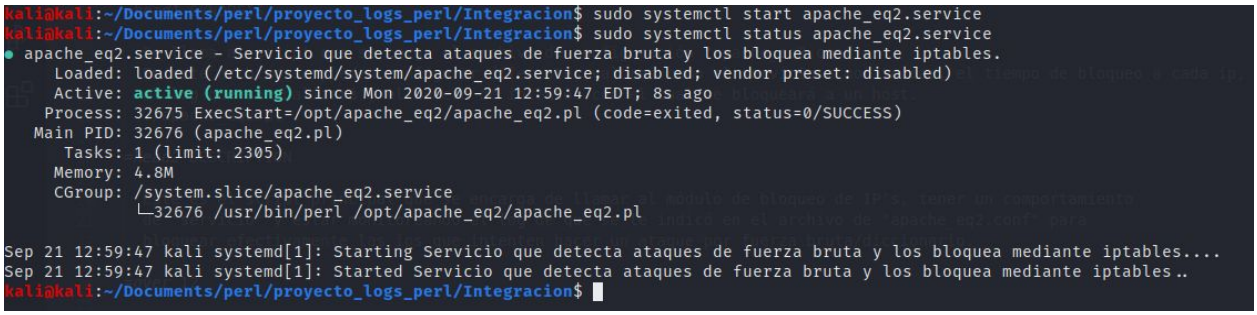
Y ahora, ejecutamos el binario como super usuario:

```
sudo ./install.sh
```

Una vez terminada la ejecución del comando anterior, el servicio debería de iniciar automáticamente, pero podemos revisar su estado ejecutando:

```
systemctl status apache_eq2.service
```

Y podemos revisar que el estado del servicio es activo:



Se ha creado una carpeta que tiene el mismo nombre del servicio dentro de /opt; ahí es donde viven los archivos necesarios para el servicio y los detalles del servicio a vigilar:

Estos archivos, son necesarios para el correcto funcionamiento del servicio. Si se desea hacer algún cambio sobre el servicio/log a revisar, puede ser modificado en el archivo de configuración para el servicio (apache_eq2.conf):

```
# Configuraciones generales del programa en Perl (Servicio de monitoreo)
[ apache ]
#logs del servicio
log = /var/log/apache2/apache2_eq2.log
# Configuraciones del servicio a monitorear
[ apache2 ]
enable = yes
log = /var/log/apache2/access.log
attempts = 3
time = 3
```

Dentro del archivo, se puede indicar el servicio a monitorear y el log que se vigila para detectar los ataques, así como la ruta en donde se guardarán los logs creados por el propio servicio.

Adicionalmente, se pueden indicar el número de intentos fallidos permitidos y tiempo de baneo antes de que se dispare éste para la IP que está intentando hacer el ataque de fuerza bruta.

Si se hace algún cambio dentro del archivo de configuración, es importante reiniciar el servicio para que tengan efecto los cambios realizados:

```
sudo systemctl restart apache_eq2.service
```

De manera adicional, si se desea obtener más documentación sobre los archivos que ocupa el servicio, todos los archivos cuentan con POD y puede ser consultado ejecutando:

```
perldoc <archivoPorRevisar>.pl
```

Demo del servicio

Para probar el funcionamiento del servicio, se puede ejecutar a modo de prueba, el archivo “testIPBloquer.pl”, el cual bloqueará de manera temporal unas cuantas IP’s.

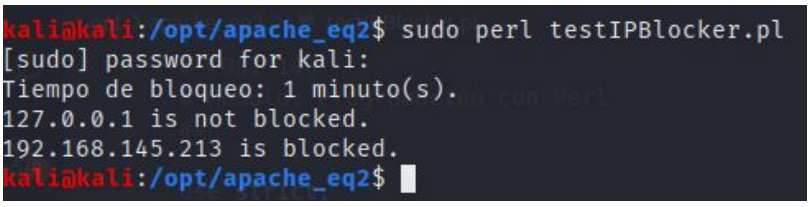
Revisamos el estado actual de las reglas de IPtables antes de ejecutar el test:

```
sudo iptables -L INPUT -v -n
```

Y ahora, ejecutamos el test:

```
sudo perl testIPBlocker.pl
```

Y nos da la siguiente información:



Ahora, si revisamos de nuevo el estado actual de las reglas de bloqueo, encontramos que han sido baneadas de manera correcta las IP's indicadas dentro del archivo de prueba:

```
kali@kali:~/Documents/perl/proyecto_logs_perl/Integracion$ sudo iptables -L INPUT -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       192.168.145.211      0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.145.212      0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.145.213      0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.145.214      0.0.0.0/0
kali@kali:~/Documents/perl/proyecto_logs_perl/Integracion$ 
```

Después de un minuto, revisamos de nuevo el estado y podemos ver que ya no se encuentran baneadas las direcciones:

```
kali@kali:~/Documents/perl/proyecto_logs_perl/Integracion$ sudo iptables -L INPUT -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       192.168.145.211      0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.145.212      0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.145.213      0.0.0.0/0
    0     0 DROP       all  --  *      *       192.168.145.214      0.0.0.0/0
kali@kali:~/Documents/perl/proyecto_logs_perl/Integracion$ sudo iptables -L INPUT -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
kali@kali:~/Documents/perl/proyecto_logs_perl/Integracion$ 
```

Ahora, para probar el funcionamiento del servicio en tiempo real, vamos a generar tráfico hacia el servidor web como si fuéramos un atacante ejecutando el siguiente comando:

```
for i in {1..20}; do curl 192.168.166.1; done;
```

Lo que hará, es enviar 20 curls a la dirección IP que sea ingresada, en este caso, es necesario indicar la dirección IP del servidor que se está revisando y ahora, si revisamos el log de apache asociado al server, podemos ver que se han generado las siguientes peticiones:

```
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
192.168.166.133 - - [21/Sep/2020:12:57:34 -0500] "GET / HTTP/1.1" 200 10956 "-" "curl/7.72.0"
```

Se puede ver que **se han registrado múltiples peticiones en el mismo segundo al mismo recurso del mismo host**. Acto seguido, el servicio lee las nuevas líneas que han sido agregadas en el log desde la última vez que se revisó y debería de notar las peticiones mencionadas, por lo que procederá a banear la IP que ha hecho todas la peticiones:

```
Every 1.0s: sudo iptables -L INPUT -v -n                    wkgp: Mon Sep 21 12:57:49 2020

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   240 DROP       all  --  *      *       192.168.166.133      0.0.0.0/0
```

Podemos ver, que la dirección IP ha sido baneada de manera correcta y que ya no se están escuchando más peticiones que sean solicitadas por ese host.

Si ahora, intentamos hacer una nueva petición al servidor cuando nos encontramos baneados, nos aparece lo siguiente:

```
curl: (28) Failed to connect to 192.168.166.1 port 80: Connection timed out
root@kali:~#
```

Podemos verificar el baneo de la IP, revisando el estado del log del servicio de monitoreo:

```
Every 1.0s: sudo iptables -L INPUT -v -n                    wkgp: Mon Sep 21 13:00:52 2020

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Every 1.0s: cat apache2_eq2.log                             wkgp: Mon Sep 21 13:00:51 2020

INICIO DEL SERVICIO - 2020-09-21 12:53:48
BLOCKED - 192.168.166.133 - 2020-09-21 12:57:34
UNBLOCKED - 192.168.166.133 - 2020-09-21 13:00:34
```

Lo sombreado en verde, indica que se ha bloqueado la dirección IP y la hora a la que ha sucedido el ban, entonces queda corroborado que se ha bloqueado la IP.

Lo sombreado den amarillo, indica que al paso de 3 minutos desde que inició el bloqueo, se desbloquea la IP y se pueden volver a hacer peticiones al servidor.

Si deseamos parar el servicio de monitoreo, sólo es necesario ejecutar:

```
sudo systemctl stop apache_eq2.service
```

Y si revisamos nuevamente el log del servicio, podemos ver que se indica fecha y hora de cuando se inicia o apaga el servicio:

```
Every 1.0s: cat apache2_eq2.log                             wkgp: Mon Sep 21 13:01:29 2020

INICIO DEL SERVICIO - 2020-09-21 12:53:48
BLOCKED - 192.168.166.133 - 2020-09-21 12:57:34
UNBLOCKED - 192.168.166.133 - 2020-09-21 13:00:34
SERVICIO DETENIDO 2020-09-21 13:01:05
```