



SSH GUARD

SSHGuard protects hosts from brute-force attacks by:

- ✳ Monitoring system logs
- ⚠ Detecting attacks
- 📄 Blocking attackers using a firewall

Started for SSH, now protects a wide range of services out of the box!

Ioar Crespo y Guillermo Berasategui

SSHGUARD

¿Qué es sshguard?

sshguard es un programa que protege a hosts frente a ataques de fuerza bruta contra SSH y otros servicios. El programa se encarga de analizar los logs del sistema y, en base a ataques reconocidos, detecta ataques y procede a bloquear a los atacantes mediante backends de cortafuegos como iptables o pf.

Es una opción gratuita y muy robusta para proteger servidores de diferentes tipos, frente los ataques de fuerza bruta que intentan o buscar un login o denegar el servicio.

Los sistemas de log que soporta son:

- cockpit
- Common Log Format
- macOS log
- metalog
- multilog
- raw log files
- syslog
- syslog-ng
- systemd journal

Puede detectar ataques contra:

- SSH
- FTP
- IMAP, POP
- ...

Soporta firewalls como:

- FirewallD
- Ipfw
- Iptables (Netfilter)
- PF
- ...

¿Cómo funciona sshguard?

Sshguard puede leer mensajes de log mediante la entrada estándar, así como monitorizar diferentes ficheros de log. Estos mensajes se analizan en busca de patrones reconocidos y en caso de detectarse un ataque la IP del atacante procede a ser bloqueada. Al paso de un tiempo que puede ajustarse, la IP del supuesto atacante vuelve a ser desbloqueada (a no ser que esté en una lista negra prefijada).

Se tienen en cuenta:

- El umbral de puntuación para ser bloqueado. Por defecto es 30. La mayoría de los ataques generan una puntuación de 10.
- Tiempo de bloqueo aplicado al atacante. Por defecto es de 120 segundos.
- Tiempo de detección. Es el espacio de tiempo en el que se guarda la puntuación del atacante. Si ese espacio de tiempo expira sin haber habido ninguna interacción nueva, la puntuación de la IP del atacante se reseteará a 0 o, lo que es lo mismo, se olvidará como atacante. Por defecto es de 1800 segundos.
- Lista negra. Cuando un atacante excede el umbral puede introducirse en una lista negra para ser recordado.
- Lista blanca. Lista que contiene las IPs o hostnames para los que no se aplicará sshguard.

La forma en la que sshguard funciona es sencilla. En el firewall del sistema se añaden reglas en las que se seleccionan los puertos que se van a bloquear y se le añade como parámetro de bloqueo a sshguard. Es decir, todo lo que sshguard diga que es un ataque, se bloqueará (la IP) en el firewall.

¿Cómo se configura sshguard?

Para configurar sshguard el primer paso es instalarlo:

- Para Linux (basado en Debian): `apt-get install sshguard`
- Para OSX (necesario tener instalado homebrew): `brew install sshguard`
- Para otros sistemas operativos consultar la documentación [1].

El segundo paso es configurar las reglas en el firewall correspondiente. En este caso sólo explicaremos los pasos para Linux (basado en Debian) y OSX, pero el resto de información para otros sistemas operativos puede encontrarse en la documentación.

Para Linux:

1. Crear una cadena que permita a sshguard bloquear atacantes:

```
$ iptables -N sshguard
```

2. En caso de querer bloquear cualquier trafico de atacantes, es decir, bloquear ataques en todos los puertos, utilizar este comando (si no, saltar al paso 3):

```
$ iptables -A INPUT -j sshguard
```

Esto lo que hace es concatenar la regla INPUT a la regla sshguard creada en el anterior paso, permitiendo a sshguard bloquear el tráfico en todos los puertos.

3. Si lo que queremos es bloquear unos puertos específicos, utilizaremos:

```
$ iptables -A INPUT -m multiport -p tcp --destination-ports  
21,22,80,110,143 -j sshguard
```

Donde seleccionamos el módulo “multiport” (con -m) de iptables, el protocolo tcp (con -p) y los puertos que queremos bloquear (con --destination-ports y los puertos separados por comas).

4. Finalmente, reiniciamos el firewall con uno de los siguientes comandos (según la distro):

```
$ sudo service iptables restart  
$ sudo /etc/init.d/iptables restart  
$ sudo service networking restart
```

Para OSX:

1. Abrimos el configfile del firewall PF:

```
sudo vim /etc/pf.conf
```

2. Escribimos al final:

```
table <sshguard> persist
block in quick on en0 proto tcp from <sshguard> to any port 22 label
"sshguard"
block in quick on en1 proto tcp from <sshguard> to any port 22 label
"sshguard"
```

Con esto estaríamos diciendo que tanto en la interfaz ethernet (línea uno) como en la interfaz de wifi (línea 2), se bloquee todo lo que diga sshguard en el puerto 22. Si quisiésemos gestionar más puertos habría que añadir en el lugar de 22 los puertos separados por comas. Por ejemplo:

... to any port 21,22,80 ...

En caso de querer gestionar todos los puertos, habría que omitir "port puerto1,puerto2,...". De este modo:

... to any label "sshguard"

3. Guardamos y cerramos el archivo (en vim con :wq).

4. Reiniciamos PF con:

```
$ sudo pfctl -f /etc/pf.conf
```

5. En caso de querer que sshguard se inicie al principio, copiar un archivo suyo en el directorio LaunchDaemons:

```
$ cp -fv /usr/local/opt/sshguard/*.plist /Library/LaunchDaemons
```

6. Iniciar sshguard:

```
$ brew services start sshguard
$ launchctl load /Library/LaunchDaemons/homebrew.mxcl.sshguard.plist
```

Llegados a este punto, ya tendríamos configurado sshguard en sus parámetros por defecto. En caso de cambiar los parámetros que explicábamos en el apartado anterior:

- Cambiar umbral de puntuación de bloqueo:

```
$ sshguard -a puntos # Donde "puntos" es el número de puntos
```

- Cambiar tiempo de bloqueo:

```
$ sshguard -p segundos # Donde "segundos" es el número de segundos
```

- Cambiar tiempo de detección:

```
$ sshguard -s segundos # Donde "segundos" es el número de segundos
```

- Añadir IP o hostname a la lista blanca:

```
$ sshguard -w host # Donde "host" es una IP, un rango de IPs o un hostname.
```

Lamentablemente, parece que estos comandos no funcionan (al ejecutarse, o se queda congelado o imprime la tabla de rutas y no se cierra) y sshguard dejó de actualizarse hace un par de años. La poca documentación nos ha impedido solucionar el problema.

Ejecución

En esta captura de pantalla podemos ver, en primer lugar un login correcto a la máquina servidor. Después haremos intentos fallidos (intencionados) de login para ver cómo sshguard, tras un número de intentos, nos bloquea la conexión. Esto puede verse en el último intento de introducción de contraseña (al que hemos tenido que dar ctrl+c porque ya no respondía) y en el intento de acceso posterior en el que el tiempo de intento de conexión expira (porque el servidor nos bloquea el acceso).

```
[GhostMac:~ ioar$ ssh anon@192.168.1.43
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-93-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 699 paquetes.
434 actualizaciones son de seguridad.

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Nov 21 22:18:03 2019 from 192.168.1.39
[anon@anon-pc:~]$ exit
logout
Connection to 192.168.1.43 closed.
[GhostMac:~ ioar$ ssh anon@192.168.1.43
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
[anon@192.168.1.43's password:
Permission denied, please try again.
[anon@192.168.1.43's password:
Permission denied, please try again.
[anon@192.168.1.43's password:
anon@192.168.1.43: Permission denied (publickey,password).
[GhostMac:~ ioar$ ssh anon@192.168.1.43
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
[Enter passphrase for key '/Users/ioar/.ssh/id_rsa':
[anon@192.168.1.43's password:
Permission denied, please try again.
[anon@192.168.1.43's password:
Permission denied, please try again.
[anon@192.168.1.43's password:
^C
[GhostMac:~ ioar$ ssh anon@192.168.1.43
ssh: connect to host 192.168.1.43 port 22: Operation timed out
GhostMac:~ ioar$ |
```

Ventajas

El programa es muy robusto y funciona muy bien en sus parámetros por defecto. En funcionamiento, pues, no da ningún problema. Además es gratuito y de código abierto [2].

Tiene opciones para añadir direcciones a listas blancas, así como para seleccionar los parámetros de bloqueo: puntuación, tiempo de bloqueo, etc.

Desventajas

La poca y nefasta documentación, no deja muy fácil la configuración por parte del usuario, más allá de simples tutoriales que hay por algunas páginas web.

El hecho de que se mantenga con poca regularidad nos deja dudas sobre su nivel de seguridad y no sabemos hasta qué punto sería capaz de detectar los nuevos ataques que hayan podido surgir estos últimos años.

No entendemos bien por qué las opciones que se exponen en su manual no funcionan bien. Se supone que debería ejecutarse como un programa normal de Linux/Unix, donde se pone el nombre del comando y sus opciones, pero al ejecutarlo se queda congelado, a veces, o muestra la tabla de rutas y se queda congelado, otras veces. Esto limita bastante su configuración, al menos hasta que lo arreglen o documenten si esto es su funcionamiento normal (y como lidiar con él) o, por el contrario, se debe a un fallo.

Script en Python

Debido a la falta de documentación y dificultad de configuración que mencionábamos antes, hemos decidido crear un script en Python que facilite esta configuración. El script está pensado tanto para Linux (basado en Debian) como para OSX.

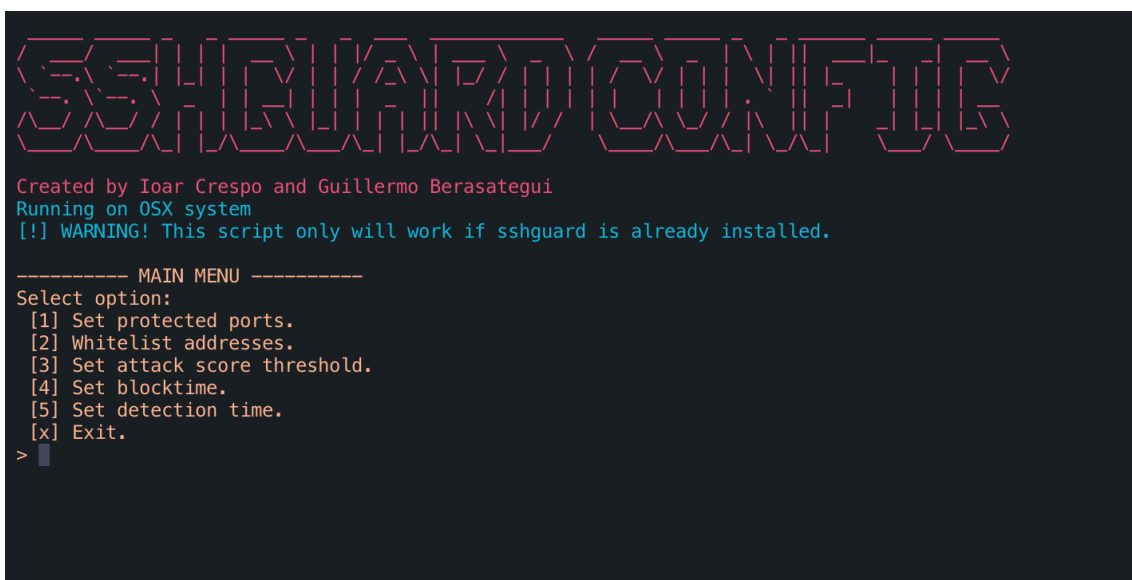
El código puede encontrarse en GitHub [3].

Basta hacer un git clone, entrar en el directorio y ejecutar:

```
$ python3 sshguard-config
```

El programa es intuitivo y permite hacer de forma fácil la configuración descrita en los apartados anteriores. El mismo programa se encarga también de poner en marcha sshguard y todas las acciones necesarias para su funcionamiento. Por desgracia y como ya hemos dicho antes, hay comandos de sshguard que no funcionan y que no hemos podido solucionar debido a la escasa documentación. Aún así, hemos creado esas opciones que, hasta que no sean hipotéticamente arregladas por los desarrolladores de sshguard, están limitadas a mostrar un mensaje de advertencia de este incidente, aunque los comandos para realizar esas acciones ya están escritos en el código y bastaría con descomentarlos.

Aquí dejamos algunas capturas del script:



```
SSHGUARD CONFIG
Created by Ioar Crespo and Guillermo Berasategui
Running on OSX system
[!] WARNING! This script only will work if sshguard is already installed.

----- MAIN MENU -----
Select option:
[1] Set protected ports.
[2] Whitelist addresses.
[3] Set attack score threshold.
[4] Set blocktime.
[5] Set detection time.
[x] Exit.
>
```


SSHGUARD CONFIG

Created by Ioar Crespo and Guillermo Berasategui

Running on OSX system

[!] WARNING! This script only will work if sshguard is already installed.

----- MAIN MENU -----

Select option:

- [1] Set protected ports.
- [2] Whitelist addresses.
- [3] Set attack score threshold.
- [4] Set blocktime.
- [5] Set detection time.
- [x] Exit.

> 1

Input ports to be secured by sshguard, separated by spaces (f.e.: 22 80 25).

Type 'all' to secure all ports. Type 'x' to go back to menu.

> 22, 80

SSHGUARD CONFIG

Created by Ioar Crespo and Guillermo Berasategui

Running on OSX system

[!] WARNING! This script only will work if sshguard is already installed.

----- MAIN MENU -----

Select option:

- [1] Set protected ports.
- [2] Whitelist addresses.
- [3] Set attack score threshold.
- [4] Set blocktime.
- [5] Set detection time.
- [x] Exit.

> 2

Sorry. Not working until SSHGUARD fixes it.

----- MAIN MENU -----

Select option:

- [1] Set protected ports.
- [2] Whitelist addresses.
- [3] Set attack score threshold.
- [4] Set blocktime.
- [5] Set detection time.
- [x] Exit.

> 3

Sorry. Not working until SSHGUARD fixes it.

----- MAIN MENU -----

Select option:

- [1] Set protected ports.
- [2] Whitelist addresses.
- [3] Set attack score threshold.
- [4] Set blocktime.
- [5] Set detection time.
- [x] Exit.

> x

Bye :)

NOTAS

- [1]. <https://sshguard.net/docs.html>
- [2]. <https://bitbucket.org/sshguard/sshguard/>
- [3]. <https://github.com/commodore1917/sshguard-config>