

# Seguridad en servidores

¿Por qué este tema?

# Seguridad en servidores

## Objetivos

- Ataques comunes y ejemplos
- Kali linux
- Fail2ban
- Otras áreas

# Seguridad en servidores

## Inyección SQL

Una inyección SQL es cuando el atacante consigue "inyectar" o insertar código SQL invasor dentro del código SQL programado, tratando de alterar el funcionamiento normal de la consulta SQL para lograr que se ejecute el código inyectado.

```
consulta := "SELECT * FROM usuarios WHERE nombre = " + nombreUsuario + ";
```

Si el parámetro fuera este:

```
"Alicia"; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE "%"
```

La consulta que se ejecuta resultaría esta:

```
SELECT * FROM usuarios WHERE nombre = 'Alicia';  
DROP TABLE usuarios;  
SELECT * FROM datos WHERE nombre LIKE '%';
```

Tratar de evitar conectarse a la base de datos como superusuario, usar usuarios o grupos con los mínimos permisos necesarios.  
Usar funciones que saniticen la entrada de usuario, por ejemplo en PHP tenemos la función `mysql_real_escape_string`.

# Seguridad en servidores

## Ataques exitosos

Un grupo de hackers (TeamBerserk) consiguió llevarse 100 000 \$ tras un ataque de inyección SQL a un proveedor de servicio de telefonía, TV e internet. Consiguieron una Hoja de cálculo con los nombres de usuario y passwords en texto plano. Después, se aprovecharon de que la gente suele usar la misma clave en varias webs para ir al sitio web de PayPal e incluso CitiBank

Un caso muy famoso fue la exposición de 134 millones de tarjetas de crédito debido a una inyección SQL que permitió instalar spyware indetectable por los antivirus en los sistemas de datos de la compañía Heartland Payment Systems. Los costes estimados de este ataque se estiman que fueron sobre los 200 millones de dolares.

# Seguridad en servidores

## DDoS

Uno de los ataques más comunes es el DDoS, un ataque de denegación de servicios distribuido usando una botnet.

Es un tipo de ciberataque donde se trata de congestionar el acceso a una red o sobrecarga de los recursos computacionales de la víctima mediante muchísimas peticiones y finalmente haciéndola inservible de modo que nadie pueda acceder a ella.

Las botnet son ordenadores zombies, ordenadores infectados por malware que están a la espera de recibir órdenes y pueden utilizar técnicas para amplificar sus ataques, como **DNS reflection**: Los ataques con DNS reflection se aprovechan de que hay servidores DNS indebidamente configurados que aceptan peticiones de cualquier máquina en internet. Utilizan generalmente UDP, por lo que no hay handshake y no se sabe el origen de un paquete, un atacante puede forjar un paquete diciendo que viene de una IP cualquiera. Esto significa que podemos forzar a que el servidor DNS mande una respuesta grande mediante peticiones pequeñas a una máquina concreta, ampliando el ataque.

Página web con ataques en directo:

<http://map.ipviking.com/>

# Seguridad en servidores

## Protección contra DDoS: Cloudflare

Cloudflare nos obliga a poner sus DNS como los autoritativos para el dominio web. Cloudflare usa anycast, básicamente múltiples máquinas tienen la misma IP, cuando se envía una petición los routers redirigirán a la máquina que esté más cercana. Esto es la base para parar un ataque DDoS, ya que el ataque a una red que use unicast todos los zombies de la botnet atacarían a la misma máquina, en cambio en una red anycast el ataque no sería efectivo porque el ancho de banda se distribuiría entre los diferentes centros de datos, cada porción del ataque sería absorbida por el centro de datos más cercano. Además, si por cualquier motivo el sitio web original deja de responder, CloudFlare cachea el sitio web y mostrará de forma casi transparente la última copia del sitio web de la que dispongan.

### Problema

De nada nos sirve utilizar Cloudflare si dejamos expuesta la IP final de nuestra máquina donde está alojada nuestra página web.

### Ejemplo

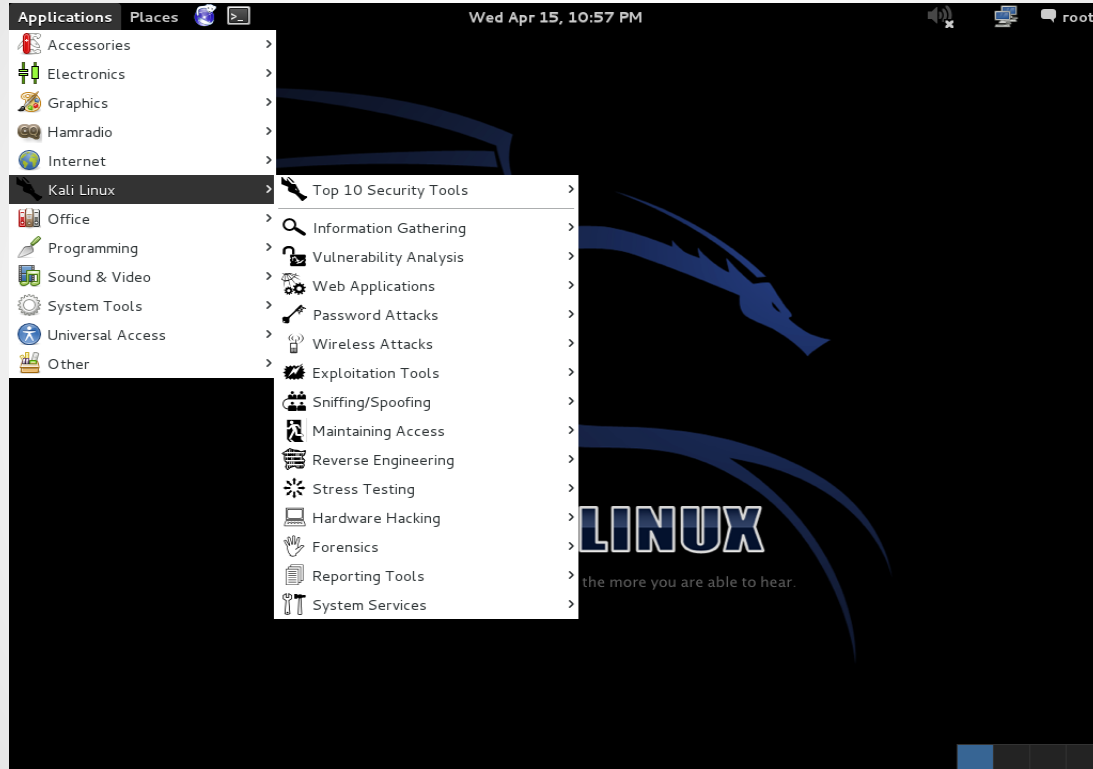
Spamhaus, un servicio líder en la lucha contra el spam estaba recibiendo ataques DDoS, y se puso en contacto con Cloudflare para tratar de mitigar estos ataques, de aproximadamente 10Gb/s. A partir de ese momento, CloudFlare fue quien recibió las peticiones dirigidas a Spamhaus pudiendo mitigar el ataque inicial. Los atacantes, vieron que el ataque no estaba haciendo efecto, y aumentaron la escala del ataque. El ataque inicial contra Cloudflare fueron de 75 Gb/s, y el día 22 llegó hasta 120 Gb/s y finalmente la cifra que consiguieron fueron 300 Gb/s, llegando a notarse las consecuencias incluso en ISP de nivel 2. Además, conforme aumentaban de intensidad, el ataque diversificaba sus objetivos apuntando a los puntos neutros (puntos de interconexión entre distintas redes). Los atacantes lograron congestionar el punto neutro de Londres, al parecer debido a una configuración demasiado permisiva del router. Los atacantes usaron la anterior técnica mencionada como DNS reflection

# Seguridad en servidores

## Kali linux

- Es una distribución basada en Debian la cual trae preinstaladas numerosas herramientas para auditoria y seguridad informática.
- Un solo usuario, acceso root por diseño, debido a la naturaleza de las auditorias de seguridad, la mayoría de las aplicaciones requieren acceso privilegiado, y sería incómodo estar siempre activandolos.
- Servicios de red desactivados por defecto, minimizando la exposición del SO, si se instala un servicio de red no persistirá una vez se reinicie el ordenador.
  - Un conjunto de repositorios mínimos y confiables para mantener la integridad del sistema

# Seguridad en servidores

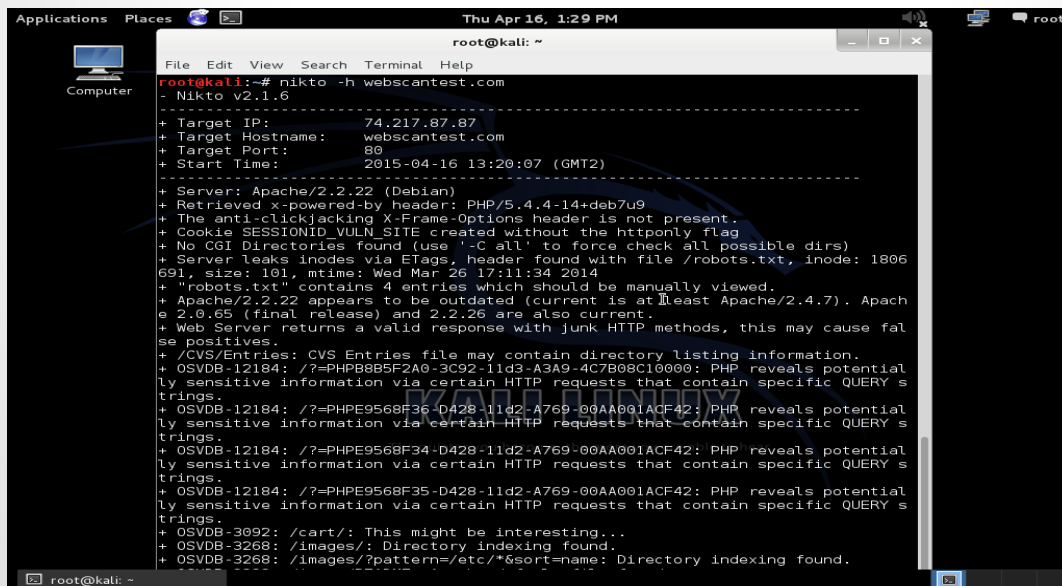


Menú herramientas kali linux



# Seguridad en servidores

Por ejemplo, si tenemos una página web podemos analizarla para ver que vulnerabilidades conocidas podríamos tener, un programa sencillo para ello seria Nikto. Su uso es muy sencillo, simplemente escribimos en la terminal **nikto -h ip/dirección** de la página web



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nikto -h webscantest.com  
- Nikto v2.1.6  
-----  
+ Target IP: 74.217.87.87  
+ Target Hostname: webscantest.com  
+ Target Port: 80  
+ Start Time: 2015-04-16 13:20:07 (GMT2)  
-----  
+ Server: Apache/2.2.22 (Debian)  
+ Retrieved x-powered-by header: PHP/5.4.4-14+deb7u9  
+ The anti-clickjacking X-Frame-Options header is not present.  
+ Cookie SESSIONID VULN SITE created without the httponly flag  
+ No CGI Directories found (use '-C all' to force check all possible dirs)  
+ Server leaks inodes via ETags, header found with file /robots.txt, inode: 1806  
691, size: 101, mtime: Wed Mar 26 17:11:34 2014  
+ "robots.txt" contains 4 entries which should be manually viewed.  
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.7). Apache  
2.0.65 (final release) and 2.2.26 are also current.  
+ Web Server returns a valid response with junk HTTP methods, this may cause false  
positives.  
+ /CVS/Entries: CVS Entries file may contain directory listing information.  
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potential  
ly sensitive information via certain HTTP requests that contain specific QUERY s  
trings.  
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potential  
ly sensitive information via certain HTTP requests that contain specific QUERY s  
trings.  
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potential  
ly sensitive information via certain HTTP requests that contain specific QUERY s  
trings.  
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potential  
ly sensitive information via certain HTTP requests that contain specific QUERY s  
trings.  
+ OSVDB-3992: /cart/: This might be interesting...  
+ OSVDB-3268: /images/: Directory indexing found.  
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.  
-----
```

# Seguridad en servidores

Lo que hace es identificar cual es el servidor web sobre el que esta funcionando la web.

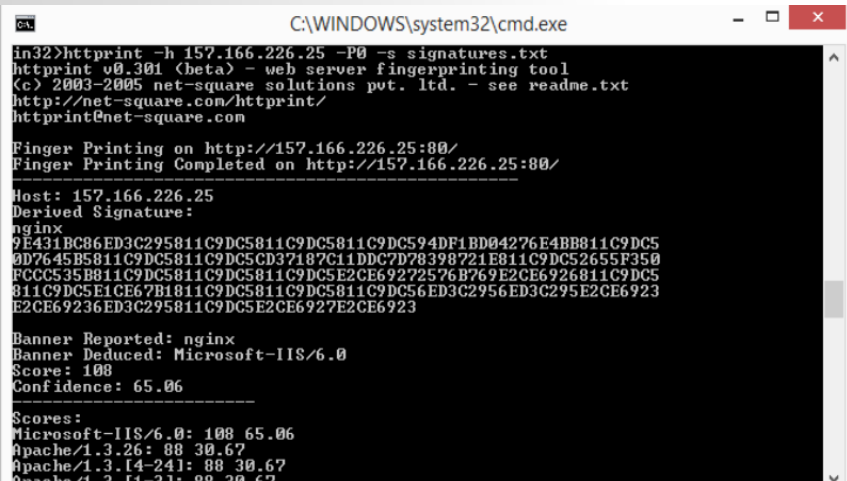
¿Pero esto no lo reportan ya las cabeceras de respuesta del protocolo HTTP?

Sí, pero un administrador web puede camuflarlo y responder por ejemplo que se está usando Nginx en vez de Microsoft-IIS/6.0

Y esto es útil para despistar al atacante, porque las vulnerabilidades de estos dos programas son obviamente distintas. Httpprint nos devuelve el servidor web que es reportado, y el servidor web que el cree que es el que de verdad se está usando, con un % de confianza determinado. Ejemplo de uso:

**httpprint -h ip/direccion web -P0 -s signatures.txt**

-P0 es una opción para deshabilitar el ping ya que muchos servidores web lo tienen bloqueado.



```
C:\WINDOWS\system32\cmd.exe
in32>httpprint -h 157.166.226.25 -P0 -s signatures.txt
httpprint v0.301 (beta) - web server fingerprinting tool
(c) 2003-2005 net-square solutions pvt. ltd. - see readme.txt
http://net-square.com/httpprint/
httpprint@net-square.com

Finger Printing on http://157.166.226.25:80/
Finger Printing Completed on http://157.166.226.25:80/

Host: 157.166.226.25
Derived Signature:
nginx
9E431BC86ED3C295811C9DC5811C9DC5811C9DC594DF1BD04276E4BB811C9DC5
9D7645B5811C9DC5811C9DC5CD37187C11DDC7D78398721E811C9DC52655R350
RCC535B811C9DC5811C9DC5811C9DC5E2CE69272576B769E2CE6926811C9DC5
811C9DC5E1CE67B1811C9DC5811C9DC5811C9DC56ED3C295ED3C295E2CE6923
E2CE69236ED3C295811C9DC5E2CE6927E2CE6923

Banner Reported: nginx
Banner Deduced: Microsoft-IIS/6.0
Score: 100
Confidence: 65.06

Scores:
Microsoft-IIS/6.0: 100 65.06
Apache/1.3.26: 88 30.67
Apache/1.3.14-241: 88 30.67
Apache/1.3.14-21: 88 28.67
```

# Seguridad en servidores

## Fail2ban

Cuando tenemos un servicio (ftp, web, etc...) uno de los ataques que mejores resultados da es el ataque a fuerza bruta, scripts automáticos que prueban combinaciones de usuario/contraseña (ya sea mediante fuerza bruta o mediante diccionarios). Fail2ban nos provee con la manera de aliviar estos ataques. Fail2ban es un **parseador de logs**, esto es, analiza los logs que producen nuestros servicios (por ejemplo, el servicio SSH) y ejecuta acciones cuando se cumplan unas "condiciones" definidas por nosotros, básicamente una expresión regular.

```
apt-get install fail2ban | yum install fail2ban
```

Uso En fail2ban tenemos:

- Filtros, una expresión regular.
- Acciones, comandos que son ejecutados
- Jail, es la combinación de un filtro más una acción
- Cliente
- Servidor, El cliente actúa como front-end y el servidor como back end.

Para iniciar el servicio, después de instalar tenemos que ejecutar: `sudo service fail2ban start`

# Seguridad en servidores

Tenemos un fichero de configuración general llamado fail2ban.conf. El fichero que nos interesa es jail.conf, que contiene la configuración de las jails, pero no debemos modificar este fichero, deberemos crear un jail.local con las modificaciones necesarias que serán las que se apliquen. Este fichero contiene las jails, un ejemplo de una seria:

```
[ssh-iptables]
    enabled = true # Activa la jaula
    filter = sshd # Filtro que esta en el directorio filter.d
action = iptables[name=SSH, port=ssh, protocol=tcp] # acción que esta en el directorio action.d, fichero iptables.conf
    logpath = /var/log/auth.log # el log que ha de escanear
    findtime = 300 # 300 segundos
    maxretry = 5 # numero de intentos maximos permitidos antes de aplicar accion.
```

Si el filtro devuelve "true" la acción descrita en iptables.conf se llevará a cabo.

## Filtros

Los filtros son expresiones regulares.

Los filtros basicamente aplican las expresiones regulares al log del servicio que estemos monitorizando, por ejemplo si tenemos un servicio que cuando alguien pone una contraseña incorrecta escribe en el log una linea parecida a :

*Failed password for invalid user recruit from 81.74.87.66*

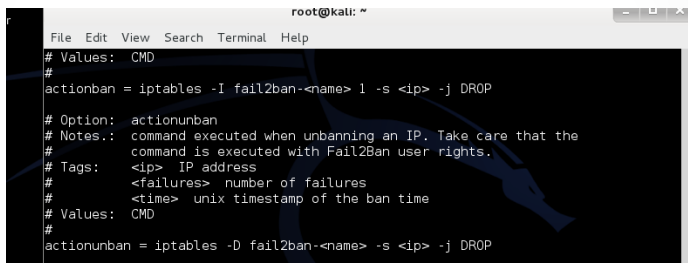
Podemos crear una expresion regular para esta linea.

# Seguridad en servidores

## Acciones

Los comandos a ejecutar.

Son ejecutadas en ciertos eventos, por ejemplo, al iniciar/parar una jail o al banear un host.



```
root@kali: ~  
File Edit View Search Terminal Help  
# Values: CMD  
#  
actionban = iptables -I fail2ban-<name> 1 -s <ip> -j DROP  
# Option: actionunban  
# Notes.: command executed when unbanning an IP. Take care that the  
#         command is executed with Fail2Ban user rights.  
# Tags:   <ip> IP address  
#         <failures> number of failures  
#         <time> unix timestamp of the ban time  
# Values: CMD  
#  
actionunban = iptables -D fail2ban-<name> -s <ip> -j DROP
```

Cuando ocurra queremos banear una IP, el programa llama automaticamente a iptables y hace que se banee la IP, impidiendole por ejemplo acceder por SSH, podemos modificar la acción por defecto y poner lo que nosotros queramos, simplemente despues de actionban se ejecuta el comando que nosotros hayamos puesto.

# Seguridad en servidores

```
[ssh]

enabled = true
port    = ssh
filter  = sshd
action  = iptables[name=SSH, port=ssh, protocol=tcp]
logpath = /var/log/auth.log
maxretry = 1_
```

```
ubuntu server2 [Corriendo] - Oracle VM VirtualBox
pablo@ubuntu:~$ ssh pablo@192.168.1.100
pablo@192.168.1.100's password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Wed Apr 22 16:39:22 CEST 2015

System load:  0.0      Processes:      80
Usage of /:   23.6% of 5.60GB   Users logged in:  1
Memory usage: 23%      IP address for eth0: 10.0.2.15
Swap usage:   0%         IP address for eth1: 192.168.1.100

Graph this data and manage this system at:
https://landscape.canonical.com/

Last login: Wed Apr 22 16:39:22 2015 from 192.168.1.101
pablo@ubuntu:~$ exit
logout
Connection to 192.168.1.100 closed.
pablo@ubuntu:~$ _
```

```
ubuntu server [Corriendo] - Oracle VM VirtualBox
pablo@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
get prot opt source destination
Zban-SSH tcp -- anywhere anywhere tcp dpt:ssh

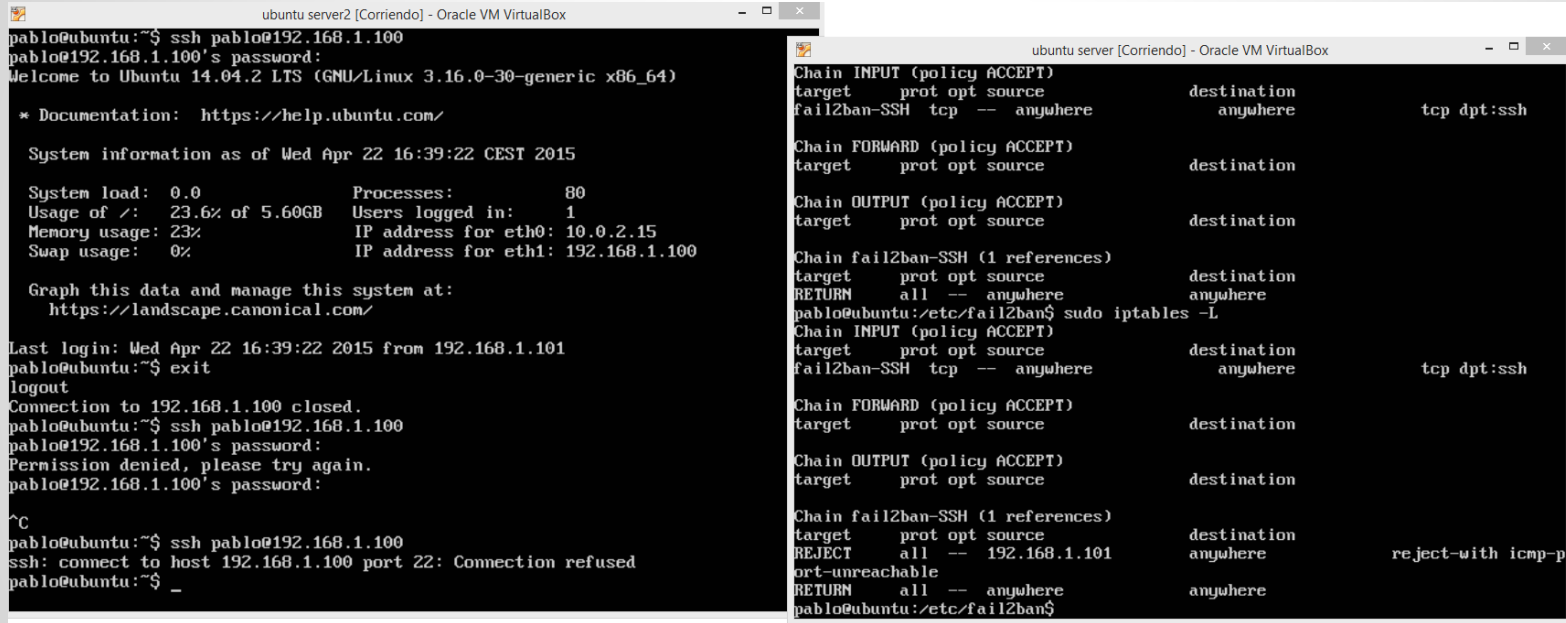
Chain FORWARD (policy ACCEPT)
get prot opt source destination

Chain OUTPUT (policy ACCEPT)
get prot opt source destination

Chain Zban-SSH (1 references)
get prot opt source destination
DEN all -- anywhere anywhere
pablo@ubuntu:~$
```

configuracion de fail2ban  
iptables sin reglas  
entrar por ssh funciona.

# Seguridad en servidores



The image displays two terminal windows from an Oracle VM VirtualBox environment. The left window, titled 'ubuntu server2 [Corriendo] - Oracle VM VirtualBox', shows a user 'pablo' logging into 'pablo@192.168.1.100'. It displays the Ubuntu 14.04.2 LTS welcome message, system information (load, memory, processes), and a failed SSH connection attempt. The right window, titled 'ubuntu server [Corriendo] - Oracle VM VirtualBox', shows the output of the 'iptables -L' command, displaying the current firewall rules for INPUT, FORWARD, and OUTPUT chains, including a rule to reject connections to 192.168.1.101.

```
ubuntu server2 [Corriendo] - Oracle VM VirtualBox
pablo@ubuntu:~$ ssh pablo@192.168.1.100
pablo@192.168.1.100's password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Wed Apr 22 16:39:22 CEST 2015

System load:  0.0      Processes:      80
Usage of /:   23.6% of 5.60GB    Users logged in: 1
Memory usage: 23%      IP address for eth0: 10.0.2.15
Swap usage:  0%          IP address for eth1: 192.168.1.100

Graph this data and manage this system at:
  https://landscape.canonical.com/

Last login: Wed Apr 22 16:39:22 2015 from 192.168.1.101
pablo@ubuntu:~$ exit
logout
Connection to 192.168.1.100 closed.
pablo@ubuntu:~$ ssh pablo@192.168.1.100
pablo@192.168.1.100's password:
Permission denied, please try again.
pablo@192.168.1.100's password:
^C
pablo@ubuntu:~$ ssh pablo@192.168.1.100
ssh: connect to host 192.168.1.100 port 22: Connection refused
pablo@ubuntu:~$ _

ubuntu server [Corriendo] - Oracle VM VirtualBox
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain fail2ban-SSH (1 references)
target     prot opt source                destination
RETURN    all  --  anywhere             anywhere
pablo@ubuntu:/etc/fail2ban$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            tcp dpt:ssh
fail2ban-SSH tcp  --  anywhere             anywhere                tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain fail2ban-SSH (1 references)
target     prot opt source                destination
REJECT    all  --  192.168.1.101        anywhere               reject-with icmp-p
ort-unreachable
RETURN    all  --  anywhere             anywhere
pablo@ubuntu:/etc/fail2ban$
```

Baneo automático, reglas de iptables  
cambiadas automáticamente

# Seguridad en servidores

```
Last login: Wed Apr 22 16:39:22 2015 from 192.168.1.101
pablo@ubuntu:~$ exit
logout
Connection to 192.168.1.100 closed.
pablo@ubuntu:~$ ssh pablo@192.168.1.100
pablo@192.168.1.100's password:
Permission denied, please try again.
pablo@192.168.1.100's password:
^C
pablo@ubuntu:~$ ssh pablo@192.168.1.100
ssh: connect to host 192.168.1.100 port 22: Connection refused
pablo@ubuntu:~$ ssh pablo@192.168.1.100
pablo@192.168.1.100's password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Wed Apr 22 16:39:42 CEST 2015

System load:  0.0           Processes:      81
Usage of /:   23.6% of 5.60GB Users logged in:  1
Memory usage: 23%          IP address for eth0: 10.0.2.15
Swap usage:  0%            IP address for eth1: 192.168.1.100

Graph this data and manage this system at:
https://landscape.canonical.com/

Last login: Wed Apr 22 16:39:43 2015 from 192.168.1.101
pablo@ubuntu:~$
```

```
ubuntu server [Corriendo] - Oracle VM VirtualBox
in FORWARD (policy ACCEPT)
get prot opt source destination

in OUTPUT (policy ACCEPT)
get prot opt source destination

in fail2ban-SSH (1 references)
get prot opt source destination reject-with icmp-p
ECT all -- 192.168.1.101 anywhere
- unreachable
URN all -- anywhere anywhere
lo@ubuntu:/etc/fail2ban$ fail2ban-client set ssh unbanip 192.168.1.101
DR Unable to contact server. Is it running?
lo@ubuntu:/etc/fail2ban$ sudo fail2ban-client set ssh unbanip 192.168.1.101
lo@ubuntu:/etc/fail2ban$ sudo iptables -L
in INPUT (policy ACCEPT)
get prot opt source destination
fail2ban-SSH tcp -- anywhere anywhere tcp dpt:ssh

in FORWARD (policy ACCEPT)
get prot opt source destination

in OUTPUT (policy ACCEPT)
get prot opt source destination

in fail2ban-SSH (1 references)
get prot opt source destination
URN all -- anywhere anywhere
lo@ubuntu:/etc/fail2ban$
```

## Desbaneo manual



# Seguridad en servidores

## ¿Que más hay?

AppArmor: Módulo de seguridad de linux configurable, alternativa a SELinux.

SELinux: Módulo de seguridad, una alternativa a AppArmor.

Malware: Código maligno, virus, gusanos, troyanos, rootkits, ransomware...

Seguridad movil: Infección móviles Android, protección...

Criptografía: Uso de cifrados para comunicaciones seguras en internet.

Ofuscación de código: Hacer más difícil de leer el código de un programa.

Auditoría Informática: Proceso de recoger, agrupar y evaluar evidencias

Informática forense: Tecnicas para identificar, preservar, analizar etc... datos que sean válidos dentro de un proceso legal.

Criptografía, La base de datos de Adobe fue accedida, los password estaban cifrados indebidamente, usando el modo ECB que cifra bloques de 8 bytes separadamente, por lo que dos bloques de 8 bytes iguales darían el mismo cifrado.

Cryptolocker, un ransomware que secuestra tus archivos del ordenador cifrándolos, haciendo imposible acceder a ellos sin la clave de descifrado, la cual si quieres tienes que pagar en bitcoins una cantidad equivalente a 500 €, con un límite de tiempo

# Seguridad en servidores

Adobe password data		Password hint
110edf2294fb8bf4	->	numbers 123456
110edf2294fb8bf4	->	==123456 ① 123456
110edf2294fb8bf4	->	c'est "123456"
8fda7e1f0b56593f e2a311ba09ab4707	->	numbers
8fda7e1f0b56593f e2a311ba09ab4707	->	1-8 ② 12345678
8fda7e1f0b56593f e2a311ba09ab4707	->	8digit
2fca9b003de39778 e2a311ba09ab4707	->	the password is password
2fca9b003de39778 e2a311ba09ab4707	->	password ③ password
2fca9b003de39778 e2a311ba09ab4707	->	rhymes with assword
e5d8efed9088db0b	->	q w e r t y
e5d8efed9088db0b	->	ytrewq tagurpidi ④ qwerty
e5d8efed9088db0b	->	6 long qwert
ecba98cca55eabc2	->	sixxone
ecba98cca55eabc2	->	1+6 ⑤ 111111
ecba98cca55eabc2	->	sixones



**Equipo especializado para informática forense.**