

Pablo Casado Arenas  
75165433F  
wide12@gmail.com

UNIVERSIDAD DE GRANADA  
E.T.S.I. INFORMÁTICA Y TELECOMUNICACIÓN

# Servidores Webs de Altas Prestaciones

Seguridad en servidores

Curso 2014-2015

Tercer Curso del Grado en Ingeniería Informática

## Índice

- 1- Introducción, objetivos (pp. 3)
- 2- Ataques: A base de datos, a la red (pp. 4-7)
- 3- Kali Linux (pp. 7 – 8)
- 4- Vulnerabilidades web ( pp 8-11)
- 5- Fail2ban (pp 11-15)
- 6- Otros (pp 16)
- 7- Bibliografía (pp 17)

# Introducción

La seguridad hoy en día es esencial en Internet. Desde cifrados para mantener comunicaciones seguras, hasta firewalls para proteger nuestros servidores de ataques externos. Existen multitud de atacantes que podrían comprometer la seguridad de nuestro servidor web si no seguimos una serie de pasos y/o metodologías, pudiendo robar datos sensibles de nuestros clientes (en caso de ser una tienda web) o hacer que nos quedemos sin servicio (perdiendo dinero por lo tanto) por un determinado periodo de tiempo.

En el contexto en el que nos encontramos, resulta imposible garantizar al cien por cien la seguridad, debido a esto hay tantos ataques exitosos. Se podría decir que los atacantes van un paso por delante con respecto a los que se defienden.

## Objetivos

Los objetivos de este trabajo son:

- Dar unas pequeñas pinceladas sobre lo importante que es la seguridad
- Ver los ataques mas comunes y como poder protegerse ante ellos
- Ver una serie de herramientas que nos ayudaran a administrar la seguridad en nuestro servidor
- Conocer Kali linux como herramienta pentesting
- Ejemplos de como se han usado algunos ataques
- Mencionar algunas áreas que no se han podido ver en este trabajo

## Seguridad en base de datos

Lo más seguro es que nuestro servidor web tenga una aplicación web que interactúe con nuestra base de datos. Además de proteger el perímetro de red mediante firewalls, IDS / IPS y antivirus, es necesario ciertas prácticas o herramientas para asegurar también la base de datos.

### Inyección SQL

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

Un ejemplo del funcionamiento sería tener esta consulta que toma como parámetro un nombre de usuario.

```
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 '%';
```

Comprometiendo la seguridad de la base de datos.

### Proteccion

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`.

## **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 dólares.

## **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.

No es necesario poseer grandes conocimientos para realizar este tipo de ataques, y no es tan arriesgado como realizar un ataque directo al servidor; este tipo de ataque utiliza otros equipos intermedios.

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.

Algunos tipos de ataques DDoS:

Mail bombing: Básicamente enviar enormes cantidades de email a una dirección en un intento de congestionar el servidor o el buzón de correo.

Smurfing: Se transmite una trama ICMP de ping, usando IP spoofing lleva la dirección de origen de la víctima y como dirección de destino la dirección de broadcast de la red atacada. De esta forma, todos los equipos contestarán a la dirección IP de origen el ping, saturándola.

SYN Flood: Utiliza una IP inexistente y envia muchas tramas SYN a la víctima, como no puede contestar al ser una IP inexistente, las peticiones llenan la cola y las solicitudes reales no pueden ser atendidas.

Página web con ataques en directo:

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

No son ataques a compañías reales, pues la compañía usa "honeypots" para monitorizar esos ataques y mostrarlos.

## **Protección contra DDoS: Cloudflare**

Cloudflare es un servicio web que nos permite protegernos contra los ataques DdoS (además de otras utilidades). Tiene varias modalidades, siendo la más básica gratuita.

### **Funcionamiento**

Cloudflare nos obliga a poner sus DNS como los autoritativos para el dominio web. Cloudflare usa anycast, basicamente multiples máquinas tienen la misma IP, cuando se envia una petición los routers redirigiran a la máquina que este 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 atacarian a la misma máquina, en cambio en una red anycast el ataque no seria efectivo porque el ancho de banda se distribuiria entre los diferentes centros de datos, cada porción del ataque sera 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 ultima copia del sitio web de la que dispongan.

### **Problema**

Esto funciona, pero tenemos que tomar medidas. De nada nos sirve utilizar Cloudflare si dejamos expuesta la IP final de nuestra máquina donde esta alojada nuestra página web.

### **Ejemplo**

En el año 2013, el 18 de marzo, Spamhaus, un servicio lider 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, en vista de que el ataque no estaba haciendo efecto, decidieron aumentar 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 (tier-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

## **Kali Linux**

### **¿Qué es?**

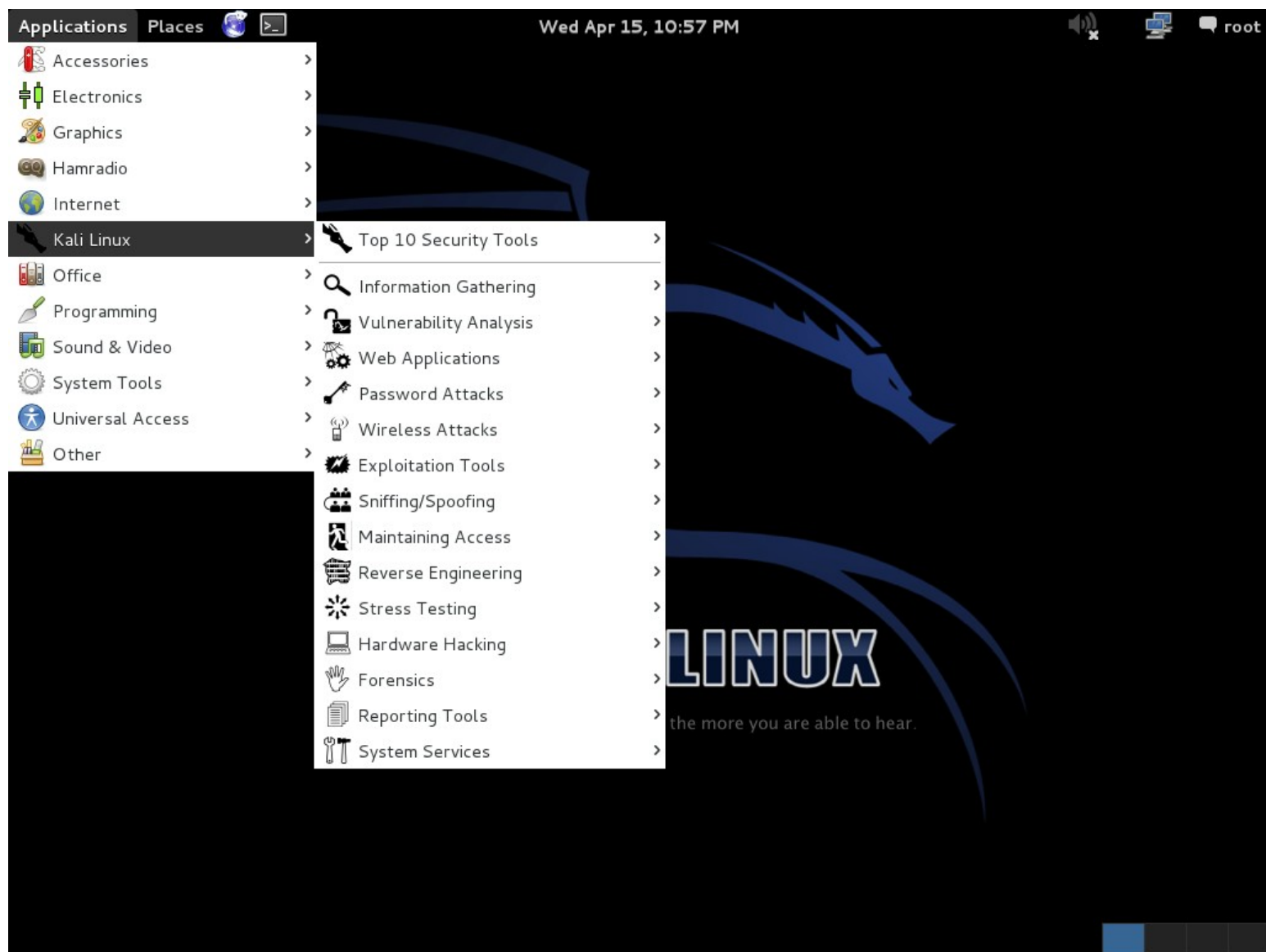
Es una distribución basada en Debian la cual trae preinstaladas numerosas herramientas para auditoría y seguridad informática, alrededor de unas 600 herramientas están disponibles, tales como nmap (escaneador de puertos) Wireshark (análizador de paquetes de red), John the Ripper (un crackeador de passwords), etc...

### **¿Por qué usarlo?**

Un solo usuario, acceso root por diseño, debido a la naturaleza de las auditorías de seguridad, la mayoría de las aplicaciones requieren acceso privilegiado, y sería incómodo estar siempre activándolos.

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



Captura de pantalla de Kali linux funcionando en una máquina virtual y mostrando las categorías de herramientas de seguridad que proporciona.

Podemos ver en la imagen las categorías en las que las herramientas están organizadas, a nosotros nos interesará "Vulnerability Analysis" y "Web applications". Aunque si deseamos testear si nuestros passwords son seguros podríamos usar las herramientas de Password Attacks.

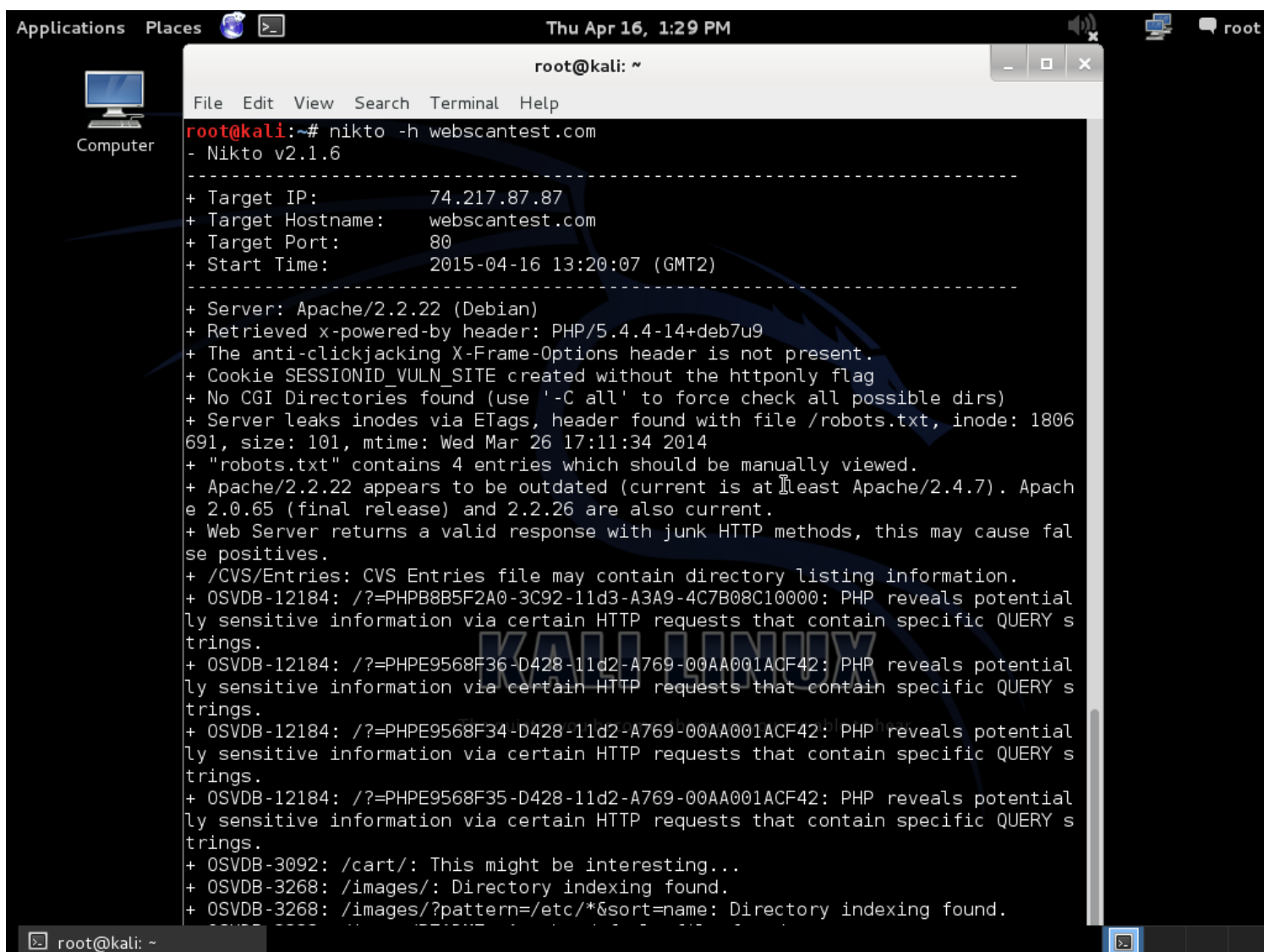


## Analizadores de vulnerabilidades web

Por ejemplo, si tenemos una página web podemos analizarla para ver que vulnerabilidades conocidas podríamos tener, un programa sencillo para ello sería Nikto. Su uso es muy sencillo, simplemente escribimos en la terminal

nikto -h ip/dirección de la página web

Ejemplo:



```
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: 1806691, 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 potentially sensitive information via certain HTTP requests that contain specific QUERY strings.  
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.  
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.  
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.  
+ OSVDB-3092: /cart/: This might be interesting...  
+ OSVDB-3268: /images/: Directory indexing found.  
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
```

Podemos ver varias vulnerabilidades y su referencia (por ejemplo OSVDB 12184 , buscandola en google podemos encontrar más información sobre esta vulnerabilidad y cómo aprovecharnos de ella).

Para Windows también hay herramientas, por ejemplo httpprint (que también se encuentra en linux).

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

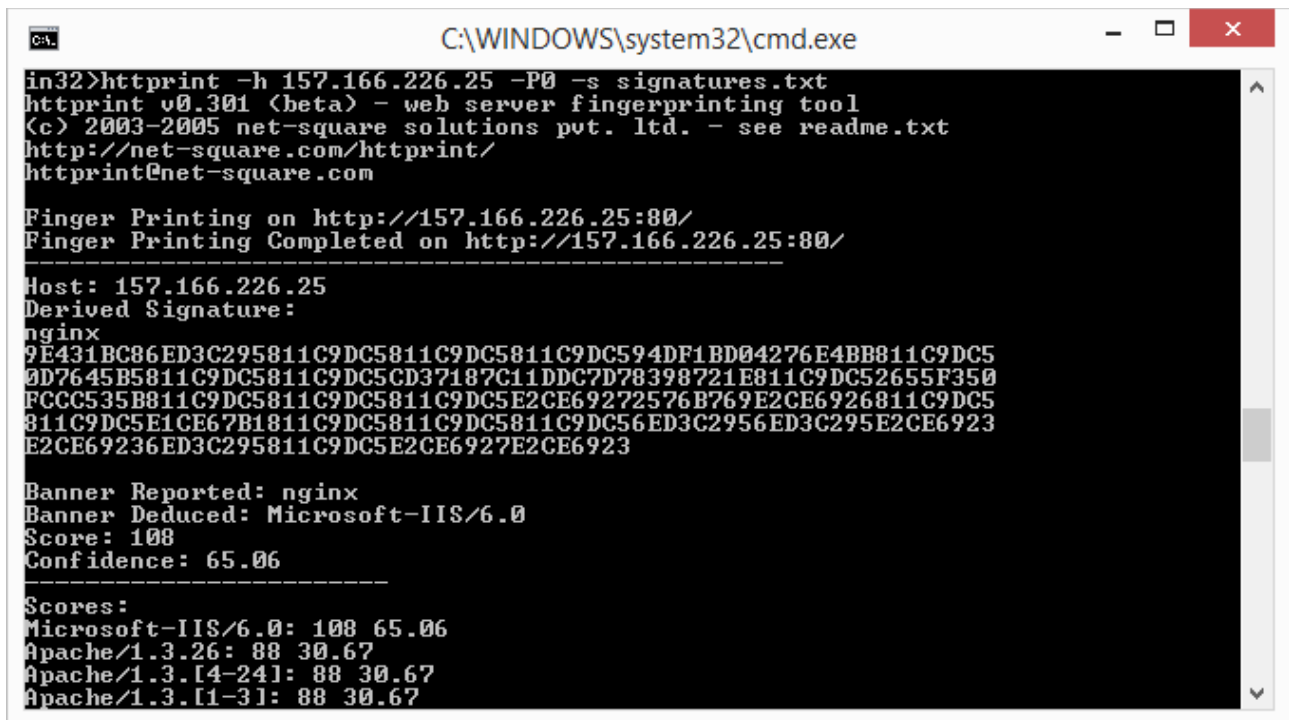
Y esto es útil para despistar al atacante, porque las vulnerabilidades de estos dos programas son obviamente distintas.

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
0D7645B5811C9DC5811C9DC5CD37187C11DDC7D78398721E811C9DC52655F350
FCCC535B811C9DC5811C9DC5811C9DC5E2CE69272576B769E2CE6926811C9DC5
811C9DC5E1CE67B1811C9DC5811C9DC5811C9DC56ED3C2956ED3C295E2CE6923
E2CE69236ED3C295811C9DC5E2CE6927E2CE6923

Banner Reported: nginx
Banner Deduced: Microsoft-IIS/6.0
Score: 108
Confidence: 65.06
-----
Scores:
Microsoft-IIS/6.0: 108 65.06
Apache/1.3.26: 88 30.67
Apache/1.3.[4-24]: 88 30.67
Apache/1.3.[1-3]: 88 30.67
```

Podemos ver como el programa cree que esta corriendo sobre un IIS 6.0 pero el servidor web se anuncia como un nginx, con un 65 % de confianza.

## **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, basicamente una expresión regular.

### **Instalación**

Lo podemos instalar desde el fuente en cualquier sistema operativo y está disponible para instalar desde nuestros gestores de paquetes más comunes (apt-get install fail2ban | yum install fail2ban ).

### **Uso**

En fail2ban tenemos:

- Filtros, una expresión regular.
- Acciones, comandos que son ejecutados
- Jail, es la combinacion de un filtro más una acción
- Cliente
- Servidor

El cliente actua como front-end y el servidor como back end.

Para iniciar el servicio, despues de instalar tenemos que ejecutar:

```
sudo service fail2ban start
```

### **Configuración**

Como la mayoría de los servicios, tendremos que configurarlo modificando un fichero de texto. La ruta de los ficheros de configuración es /etc/fail2ban/

Aquí nos encontraremos con varios archivos :

```

root@kali:~/Desktop# ls -R /etc/fail2ban/
/etc/fail2ban/:
action.d  fail2ban.conf  filter.d  jail.conf

/etc/fail2ban/action.d:
complain.conf      iptables-multiport.conf      mynetwatchman.conf
dshield.conf       iptables-multiport-log.conf  sendmail-buffered.conf
dummy.conf         iptables-new.conf            sendmail.conf
hostsdeny.conf     iptables-xt_recent-echo.conf sendmail-whois.conf
ipfilter.conf      mail-buffered.conf           sendmail-whois-lines.conf
ipfw.conf          mail.conf                    shorewall.conf
iptables-allports.conf mail-whois.conf
iptables.conf      mail-whois-lines.conf

/etc/fail2ban/filter.d:
apache-auth.conf      dovecot.conf                pure-ftp.conf
apache-badbots.conf   dropbear.conf               qmail.conf
apache-common.conf    exim.conf                   sasl.conf
apache-nohome.conf    gssftp.conf                 sieve.conf
apache-noscript.conf  lighttpd-fastcgi.conf       sshd.conf
apache-overflows.conf named-refused.conf          sshd-ddos.conf
common.conf           pam-generic.conf            vsftpd.conf
courierlogin.conf     php-url-fopen.conf          webmin-auth.conf
couriersmtp.conf      postfix.conf                wuftp.conf

```

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 sería:

```

[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 línea parecida a :

*Failed password for invalid user recruit from 81.74.87.66*

Podemos crear una expresion regular para esta linea.

## Acciones

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

Ejemplo:

```
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
```

Es decir, 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.

Vamos a probarlo con el SSH:

```
GNU nano 2.2.6 Archivo: jail.conf

#
# Optionally you may override any other parameter (e.g. banaction,
# action, port, logpath, etc) in that section within jail.local

[ssh]

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

[dropbear]

enabled = false
port = ssh
filter = dropbear
logpath = /var/log/auth.log
maxretry = 6

# Generic filter for pam. Has to be used with action which bans all ports
# such as iptables-allports, shorewall
[pam-generic]

[ 484 líneas escritas ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Repág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía
```

Nos conectamos por ssh y comprobamos que funciona:

```
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 5.6GB 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:~$ _

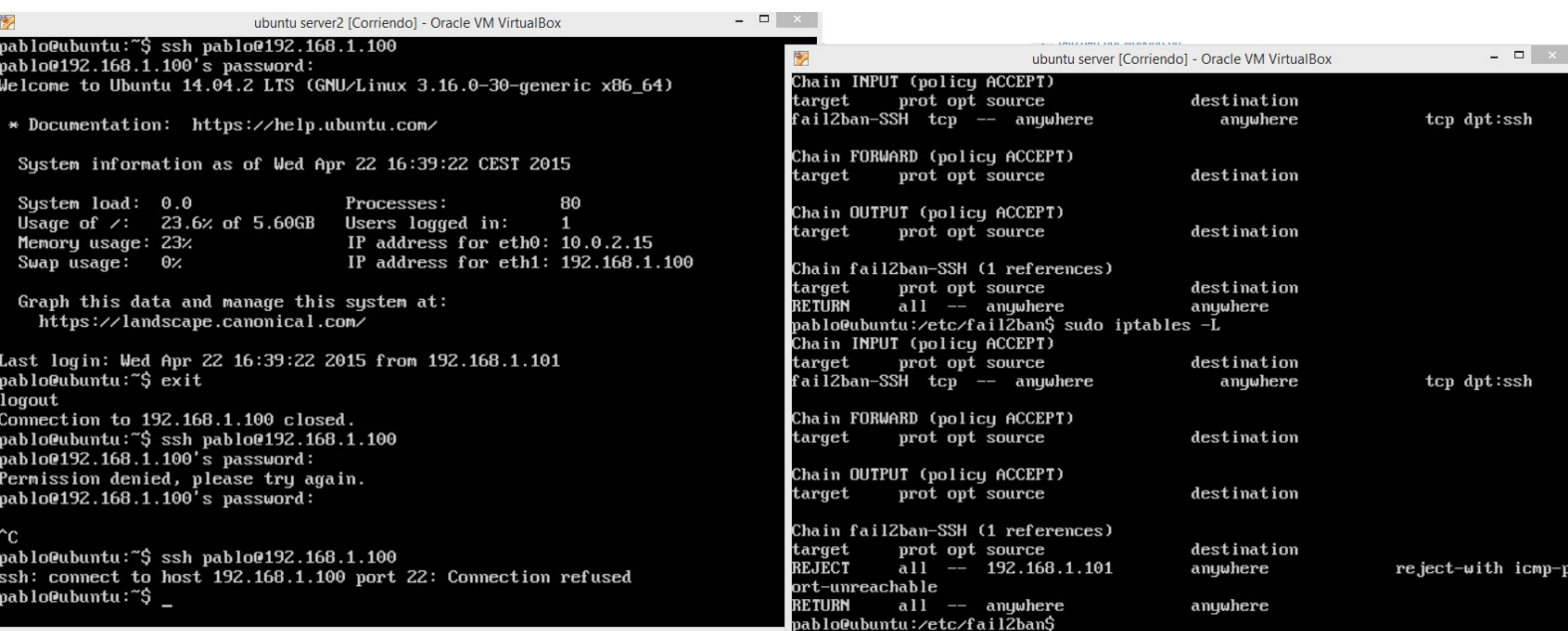
ubuntu server [Corriendo] - Oracle VM VirtualBox
pablo@ubuntu:/etc/fail2ban$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
fail2ban-SSH tcp -- anywhere anywhere tcp dpt:ssh

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
DROP all -- anywhere anywhere
pablo@ubuntu:/etc/fail2ban$
```

Fallamos el login varias veces y vemos como automaticamente nuestra IP esta baneada en el iptables del servidor y no nos deja acceder:



The image shows two terminal windows from an Oracle VM VirtualBox. The left window shows a user 'pablo' attempting to SSH into '192.168.1.100'. After three failed password attempts, the connection is refused. The right window shows the iptables configuration on the target machine. It displays the INPUT, FORWARD, and OUTPUT chains, all with a policy of ACCEPT. A specific rule in the INPUT chain, named 'fail2ban-SSH', is shown with the following configuration: target prot opt source destination anywhere tcp dpt:ssh. This rule is referenced by the fail2ban service.

```
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
fail2ban-SSH tcp -- anywhere tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

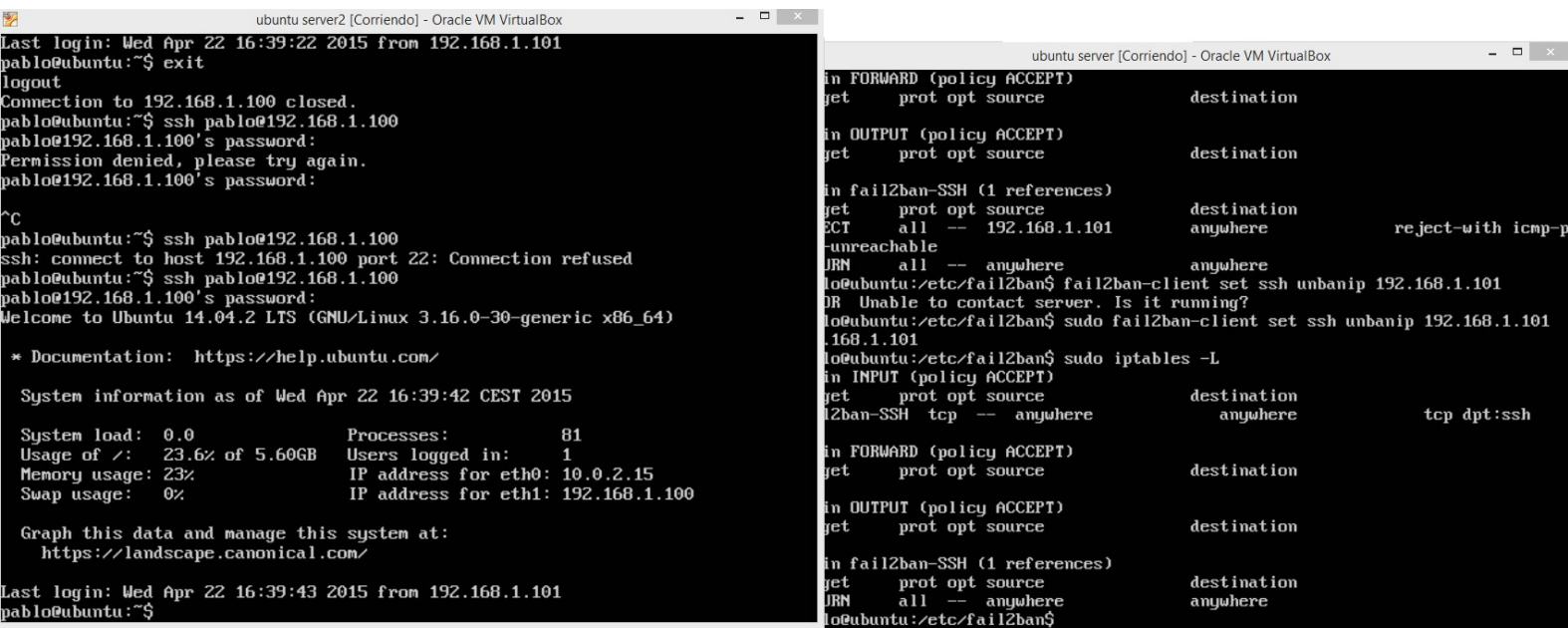
Chain fail2ban-SSH (1 references)
target prot opt source destination
RETURN all -- anywhere anywhere
pablo@ubuntu:/etc/fail2ban$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
fail2ban-SSH tcp -- anywhere anywhere tcp dpt:ssh

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$
```

En este caso desbaneamos la IP manualmente



The image shows two terminal windows. The left window shows the user 'pablo' attempting to SSH into '192.168.1.100' again, but the connection is still refused. The right window shows the user running the command 'fail2ban-client set ssh unbanip 192.168.1.101' to manually unban the IP. The output of the command is 'OR Unable to contact server. Is it running?'. The user then runs 'sudo fail2ban-client set ssh unbanip 192.168.1.101' and 'sudo iptables -L' to verify the configuration. The iptables configuration is the same as in the previous screenshot, but the user has manually unbaned the IP.

```
ubuntu server2 [Corriendo] - Oracle VM VirtualBox
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
Chain INPUT (policy ACCEPT)
target prot opt source destination
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
RETURN all -- anywhere anywhere
pablo@ubuntu:/etc/fail2ban$
pablo@ubuntu:/etc/fail2ban$ fail2ban-client set ssh unbanip 192.168.1.101
OR Unable to contact server. Is it running?
pablo@ubuntu:/etc/fail2ban$ sudo fail2ban-client set ssh unbanip 192.168.1.101
pablo@ubuntu:/etc/fail2ban$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
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
RETURN all -- anywhere anywhere
pablo@ubuntu:/etc/fail2ban$
```

Pero en general podemos definir un tiempo de baneo, que por defecto es de 600 segundos, por lo tanto despues de 600 segundos podriamos probar otra vez a acceder mediante password sin necesidad de desbanear la IP manualmente.



## Qué 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 móvil: 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: Técnicas para identificar, preservar, analizar etc... datos que sean válidos dentro de un proceso legal.

Criptografía, La base de datos de Adobe fue accedida, los passwords 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.

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

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





**Equipo especializado para informática forense.**

## **Bibliografía**

<http://www.scmagazine.com/hacker-group-claims-to-have-looted-100k-via-sql-injection-attack/article/317412/>

<http://www.csoonline.com/article/2130877/data-protection/data-protection-the-15-worst-data-security-breaches-of-the-21st-century.html>

<https://securosis.com/blog/heartland-hackers-caught-answers-and-questions/>

<http://www.symantec.com/connect/articles/five-common-web-application-vulnerabilities>

<http://www.cloudflare.com/ddos>

<http://docencia.ac.upc.es/FIB/CASO/seminaris/2q0304/M8.pdf>

[http://www.fail2ban.org/wiki/index.php/MANUAL\\_0\\_8](http://www.fail2ban.org/wiki/index.php/MANUAL_0_8)

Manpage de Nikto.

<http://docs.kali.org/introduction/should-i-use-kali-linux>

