ARTICULOS

Maxima seguridad con dsniff. El sniffer total.

Por carcoco (http://bulmalug.net/~carcoco/) creado el << 19/10/2001 18:22 >> y modificado por última vez el << 19/10/2001 18:22 >>

__Dsniff__ nos demuestra lo inseguras que son nuestras redes, sobretodo si nos empeñamos en enviar contraseñas en formato __texto plano__. Con este __sniffer__, nos daremos cuenta de los realmente importante que puede llegar a ser la utilizacion de la encriptacion en nuestras comunicaciones diarias ...

Tal y como dice el autor del programa *Dug Song*, él desarrollo esta potentísima aplicación para **auditar** sus propias redes y para demostrar la necesidad de usar **encriptación** de un modo habitual. *"Please do not abuse this software"*

Gracias a **dsniff**, tenemos un motivo mas para usar diariamente herramientas como **ssh** (la version 2, porque la 1 tiene algunos problemas de seguridad y es vulnerable) y **gpg** (Gnu pgp)

Para haceros una idea de las posibilidades del dsniff, conectaros a Internet como lo haceis habitualmente, en otra sesion como root teclear:

```
# dsniff -i ppp0
```

Ahora bajaros el correo, entrad en algun servidor/servicio que os pida contraseña y vereís como *por arte de magia* vais capturando los pares **usuario:contraseña.**

Entrad ahora usando **ssh** y vereís como en este caso nuestro sniffer **no** captura la contraseña. **;-)**

dsniff esta formado por una serie de programas que son:

- **dsniff**: simple password sniffer. (Yo realmente no lo consideria nada *simple*)
- **arpspoof**: redirect packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies.
- **dnsspoof**: forge replies to arbitrary DNS address / pointer queries on the LAN.
- **filesnarf**: saves selected files sniffed from NFS traffic in the current working directory.
- **macof**: flood the local network with random MAC addresses.
- **mailsnarf**: a fast and easy way to violate the Electronic Communications Privacy Act of 1986 (18 USC 2701-2711), be careful.
- **msgsnarf**: record selected messages from sniffed AOL Instant Messenger, ICQ 2000, IRC, and Yahoo! Messenger chat sessions.
- **sshmitm**: SSH monkey-in-the-middle.
- **tcpkill**: kills specified in-progress TCP connections.
- **tcpnice**: slow down specified TCP connections via "active" traffic shaping. (Se puede usar para evitar virus/gusanos tipo NIMDA). Os recomiendo que os paseis por : http://bulmalug.net/body.phtml?nIdNoticia=865
- **urlsnarf**: output all requested URLs sniffed from HTTP traffic in CLF (Common Log Format, used by almost all web servers), suitable for offline post-processing

- **webmitm**: HTTP / HTTPS monkey-in-the-middle.
- **webspy**: sends URLs sniffed from a client to your local Netscape browser for display, a fun party trick

Detección de sniffers usando Linux

Por carcoco (http://bulmalug.net/~carcoco/) creado el << 23/11/2001 16:39 >> y modificado por última vez el << 23/11/2001 16:39 >>

*Un **sniffer** es un programa que captura todo el tráfico que pasa por la red, de forma que ejecutado en una red local, permiten obtener pares (**usuario:contrasña**) rápidamente.*
Suele funcionar de forma pasiva, siendo muy dificiles de detectar, aunque existen algunas técnicas que nos permitiran averiguar si tenemos espias en nuestra red ...

Advertencia: **Detectar un sniffer** es sumamente dificil, por no decir ,que si esta correctamente configurado y oculto usando otras técnicas, es **practicamente imposible** detectarlos. Aqui intentaré dar algunas ideas y consejos para que conozcais de que va el tema.

Se dan 2 situaciones distintas:

- Consulta directa de las interfaces de red.
- NO es posible la consulta directa de las interfaces de red.

Consulta directa de las interfaces de red.

En el primer caso lo que tendremos que hacer es mirar el estado de las diferentes interfaces de redes que tengamos en dicho equipo. La forma más habitual e utilizar el comando *ifconfig* (paquete net-tools), aunque podemos usar otros como *ifstatus* o *cpm* (check for network interfaces in promiscuous mode).

Aqui os muestro como el resultado del comando ifconfig antes y después de ejecutar el sniffer en una máquina **FreeBSD**:

```
$ ifconfig
fxp0: flags=8843<UP,BROADCAST,
      RUNNING,SIMPLEX,MULTICAST> mtu 1500
```

Estando el sniffer en ejecución, podemos ver en la primera línea la palabra "**PROMISC**", que nos revela el estado de la tarjeta de red:

```
$ ifconfig
fxp0: flags=8943<UP,BROADCAST,
      RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
```

Normalmente cuando la interface pasa a modo promiscuo, queda reflejado en el fichero de logs, tal y como podemos ver aqui.

```
# cat /var/log/messages
.
Nov 20 08:51:20 maquineta
         /kernel: fxp0: promiscuous mode enabled
.
```

Aunque es la forma más sencilla y directa de detectar un sniffer, tampoco es infalible, puesto que aun estando en marcha el sniffer puede que no aparezca la interfaz como *promiscuos* sobretodo si han crackeado la maquina y le han metido un LKM del estilo del **RhideS v1.0** (rhides.c en 7a69#12):

"Is usualy to install a sniffer when you hack some system, but if you do it, the net device is established to promisc mode and if the admin is inteligent must to discover the sniffer. Using RhideS you can to hide some promisc mode interface easily. Inserting the module you can specify magic words."

Otras posibles medidas para detectar el sniffer son:

- Controlar y detectar los **logs** que genera el sniffer.
- Controlar las **conexiones al exterior**, por ejemplo, el envío sospechoso de e-mail a cuentas extrañas.
- Utilizar la herramienta **lsof** (LiSt Open Files), de forma que tengamos monitorizados los programas que acceden al dispositivo de red.

NO es posible la consulta directa de las interfaces de red.

En caso de que no podamos acceder y consultar el estado de las interfaces de red, puesto que el sniffer no esta en nuestra máquina sino que se encuentra en alguna otra máquina de la red. Lo que tendremos que hacer, es utilizar algun defecto en la implementación concreta del protocolo TCP/IP por algun programa/comando (tal y como hace el programa **neped** respecto a el *arp*) o ingeniarnoslas para averiguar de alguna forma si tenemos algun sniffer corriendo en la red:

"Una de las posibles técnicas, consiste en enviar paquetes a una máquina inexistente y cuya dirección no está dada de alta en el servidor de nombres. Sabremos que tenemos un sniffer en nuestra red si posteriormente detectamos cualquier intento de acceso a la máquina ficticia".

Antisniff, del que tenemos incluso el código fuentes en la version **Unix**, es una de las mejores herramientas de detección de sniffer de forma remota, aunque quizás este un poquitín obsoleto, sobretodo porque no contempla la nueva generación de sniffers.
AntiSniff is a new class of proactive security monitoring tool. It has the ability to scan a network and detect whether or not any computers are in promiscuous mode. This is often a sign that a computer has been compromised. With AntiSniff, administrators and security teams can finally get a handle on who is watching network traffic at their site. Antisniff was designed to detect compromized machines with IP stacks that a remote attacker could utilize to sniff network traffic. It was not designed to detect hardware based network probes or special purpose network analyzers which an attacker would need physical access to install.

Sentinel es otra interesante herramienta, cuyo objetivo principal es la detección remota de sniffers. Utiliza las librerias **libcap** y **libnet** y tenemos el código fuente disponible.
The sentinel project is an implementation of effective remote promiscuous detection techniques.For portability purposes, the sentinel application uses the libpcap and libnet libraries.

Por último comentar la existencia de una curiosa herramienta: **AntiAntiSniffer Sniffer**, cuyo objetivo es detectar la ejecución en la red del **Antisniff**, evitando ser detectado por el mismo.

Conclusión: Recordar (una vez más) la necesidad de usar encriptación a diario en **TODAS** nuestras comunicaciones: S/key, gpg, SSH, SSL, Firewall, VPNs, etc...

Enlaces:

- Pagina web del **dsniff**: http://www.monkey.org/~dugsong/dsniff/
  Ademas de la version actual la 2.3, podemos encontrar una beta de la nueva version que esta en desarrollo con nuevas características. http://www.monkey.org/~dugsong/dsniff/beta/dsniff-2.4b1.tar.gz
- **sniffit**: http://reptile.rug.ac.be/~coder/sniffit/sniffit.html
- **net-tools**:http://www.tazenda.demon.co.uk/phil/net-tools/
- **neped.c**:http://www.securityfocus.com/data/tools/neped.c
- **Ifstatus**: http://www.ja.net/CERT/Software/ifstatus/ifstatus2.2.tar.gz
- **cpm, ifsolstat**: http://www.ja.net/CERT/Software/sniffdetect/
- **7a69ezine**: http://www.7a69ezine.org/
- **lsof**: http://freshmeat.net/redir/lsof/6029/url_changelog/
- **Antisniff**: http://www.l0pht.com/antisniff/
  http://www.securityfocus.com/data/tools/anti_sniff_researchv1-1-2.tar.gz

- **Sentinel**: http://www.packetfactory.net/Projects/sentinel/
- **libnet**: http://www.packetfactory.net/Projects/libnet
- **libpcap**: http://www.tcpdump.org
- **Anti Antisniff**: http://www.securityfocus.com/data/tools/aass.c
- **sniffing-faq**: http://www.robertgraham.com/pubs/sniffing-faq.html
- **Sniffing (network wiretap, sniffer) FAQ**: http://cs.baylor.edu/~donahoo/tools/sniffer/sniffingFAQ.htm

Más informacion:
ftp://www6.software.ibm.com/software/developer/library/s-sniff.pdf
ftp://www6.software.ibm.com/software/developer/library/s-sniff2.pdf

--
Carlos (aka carcoco)
http://bulmalug.net/todos.phtml?id_autor=132

E-mail del autor: carcoco@grupobbva.net
Podrás encontrar este artículo e información adicional en:
http://bulmalug.net/body.phtml?nIdNoticia=928