UCRL-MA-118605 Rev 1

Department of Energy

C I A C

Computer Incident Advisory Capability

Unix Incident Guide: How to Detect an Intrusion

CIAC-2305 R.1

by Karyn Pichnarczyk, Steve Weeber, & Richard Feingold

and the other members of the CIAC team

December, 1994

Lawrence Livermore National Laboratory

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.
Springfield, VA 22161

CIAC is the U.S. Department of Energy's Computer Incident Advisory Capability. Established in 1989, shortly after the Internet Worm, CIAC provides various computer security services free of charge to employees and contractors of the DOE, such as:

- Incident Handling consulting
- Computer Security Information
- On-site Workshops
- White-hat Audits

CIAC is located at Lawrence Livermore National Laboratory and is a part of its Computer Security Technology Center. CIAC is also a founding member of FIRST, the Forum of Incident Response and Security Teams, a global organization established to foster cooperation and coordination among computer security teams worldwide.

Reference to any specific commercial product does not necessarily constitute or imply its endorsement, recommendation or favoring by CIAC, the University of California, the United States Department of Energy, or the United States Government.

Work performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Table of Contents

Introduction

Overview

If you suspect or have been notified that your computer system has been or is under attack, you must determine:

- if there really is or was an attack
- if the attack was successful
- and, to what degree the attack compromised the system

This can be routine, quite challenging, or extremely difficult. Modern operating systems are large, complex, and imperfect dynamic systems, with many places for attackers to hide and many opportunities for them to cover their tracks.

CIAC has collected and developed techniques to discover traces of an attack. Almost all attacks leave detectable remnants that may be uncovered and used in an investigation.

What's in This Guide

This document contains step-by-step instructions to follow if you are investigating an actual security incident. It can also be used as a tutorial in general techniques for use if an attack occurs.

This guide helps you with these security scenarios...	By providing you with detailed information on these topics...
A person's system is linked to the Internet; there is "a feeling" that something is wrong. A security problem might exist, but you can't be sure.	• displaying the users logged in to your system • displaying active processes • finding the footprints left by an intruder • detecting a sniffer • finding files and other intrusions left by an intruder
You are notified by CIAC that someone from another site that had an intruder found your site's name in an intruder's log file. You know that an intruder has at least "touched" your system. The extent of the contact is unknown.	
An incident response team informs you that an intruder was located, and the team's log files indicate the intruder came from *your* site.	
You get a call that someone is performing an illegal action (either breaking into another system, or breaking into that particular system) right NOW. Action must be swift in order to minimize damage.	
You suspect you have a sniffer on your system, but don't have the slightest idea where to start looking for it.	

Other Situations

This guide will help you investigate the situations described above. However, additional scenarios can occur. If your experience doesn't quite fit any of the situations listed, call CIAC at (510) 422-8193 for assistance.

Displaying the Users Logged in to Your System

Overview

If you suspect that there is an active intruder on your system, first determine where they are and what they are doing. This section shows you how to use these commands to find out who is on your system:

- the “w” command
- the “finger” command
- the “who” command

> ☞ **These commands are only useful when a suspected intruder is logged in to your system.**

The “w” Command

The “**w**” command gives you a general overview of all users and their active programs on the system. A sample output is shown here.

```
Prompt %  w

  3:47pm  up 18 days,  3:02,  7 users,  load average: 0.02, 0.00, 0.00
User     tty      login@     idle     JCPU     PCPU      what
user1    ttyp0    25Mar94    2:08     39:15    4         -tcsh
user2    ttyp1    5Apr94     8        5:51     5:28      emacs
user2    ttyp2    3:46pm                                w
user3    ttyp3    Mon 2pm              2:04     1         -csh
user3    ttyp4    Mon 3pm    41        21                 -csh
user2    ttyp6    5Apr94     3        1:38     6         -tcsh
user2    ttyp7    Wed 2pm    5:31     17       1         -tcsh
Prompt %
```

The first line displayed, the status line, gives general information: the present time, how long the system has been running, and the load on the system for various periods of time. The rest of the output from the “w” command shows you who is currently logged in to the system, which TTY they are using, and what each user is currently doing.

What to Look For

Verify that:

- all users are valid
- users have not been logged in for an abnormal length of time
- users are not running suspicious software

Vulnerabilities

The output listing from the "w" command can be easily modified to hide a skilled intruder's existence on the system.

The "finger" Command

Another command that displays who is on the system is the **"finger"** command. A sample output is shown here.

```
Prompt %  finger
Login    Name         TTY     Idle       When      Where
user1    user name     p0       26    Fri 11:46    host1.sub.domain
user2    user name     p1       34    Tue 10:42    host2.sub.domain
user4    user name     p2             Mon 14:04    host3.sub.domain
user3    user name     p3       44    Mon 14:06    host5.sub.domain
user2    user name     p4             Mon 16:43    host4.sub.domain
user2    user name     p6     3:45    Tue 11:06    host2.sub.domain
user2    user name     p7        1    Wed 14:47    host2.sub.domain
user3    user name     p8     3:04    Thu 11:04    host5.sub.domain
user3    user name     p9     1:02    Fri 13:52    host5.sub.domain
Prompt %
```

The "finger" command shows you who is currently logged in to the system, which TTY they are using, the time they logged in, and where they are logged in from.

What to Look For

Verify that:

- all users are valid
- users have not been logged in for an abnormal length of time
- the login location of each user is valid

Vulnerabilities

The output from the "finger" command can easily be modified to hide a skilled intruder's existence.

Displaying the Users Logged in to Your System, Continued

The "who" Command

The **"who"** command lists information about the users currently on the system. This information is retrieved from the /etc/utmp file. A sample output is shown here.

```
Prompt %  who
user1    ttyp0    Mar    25    11:46    (host1.sub.domain)
user2    ttyp1    Apr    5     10:42    (host2.sub.domain)
user4    ttyp2    Apr    18    14:04    (host3.sub.domain)
user3    ttyp3    Apr    11    14:06    (host5.sub.domain)
user2    ttyp4    Apr    18    16:43    (host4.sub.domain)
user2    ttyp6    Apr    5     11:06    (host2.sub.domain)
user2    ttyp7    Apr    6     14:47    (host2.sub.domain)
user3    ttyp8    Apr    14    11:04    (host5.sub.domain)
user3    ttyp9    Apr    15    13:52    (host5.sub.domain)
Prompt %
```

This command lists who is currently logged in to the system, which TTY they are using, login time, and where they are logged in from.

What to Look For

Verify that:

- all users are valid
- users have not been logged in for an abnormal length of time
- the login location of each user is valid

Vulnerabilities

The output from the "who" command can easily be modified to hide a skilled intruder's existence, as the command gets its information from the /etc/utmp file.

Displaying Active Processes

Overview

Even if an intruder is no longer logged in to a (potentially) penetrated system, a process may have been left running by the intruder to continue performing tasks. This section shows you how to use these commands to display the active processes running on your system:

- the “ps” command
- the “crash” command

The “ps” Command

The **“ps -agux”** command lists the processes that are executing on your system.

- The command’s “a” parameter displays all processes running on the system, not just those owned by you.
- The command’s “g” parameter displays all processes, as opposed to those which “ps” decides are simply “interesting” (refer to the “ps” man page for the definition of “interesting”).
- The “u” parameter displays user-oriented output.
- The “x” parameter includes processes without control terminals.

The “ps” command is a reliable way to see what programs are being executed on the system. A shortened sample output is shown here.

```
Prompt %  ps -agux
USER       PID   %CPU   %MEM    SZ   RSS    TT   STAT  START   TIME  COMMAND
user5    28206    8.1    0.4    48   280    p4   S     13:55   0:00  man inetd.conf
user5    28208    3.9    0.5    56   312    p4   S     13:55   0:00  more -s /usr/man/cat5/in
root         2    0.0    0.0     0     0    ?    D     Mar 25  0:02  pagedaemon
root        87    0.0    0.0   176     0    ?    IW    Mar 25  0:16  sendmail: accepting conn
root         1    0.0    0.0    56     0    ?    IW    Mar 25  0:04  /sbin/init -
user3    15547    0.0    0.0    88     0    ?    IW    Apr 5   0:00  selection_svc
user1      184    0.0    0.0   192     0    p0   IW    Mar 25  0:06  -tcsh (tcsh)
user2    28209    0.0    0.8   208   520    p5   R     13:55   0:00  ps -agux
user2    21674    0.0    0.4   112   248    p5   S     16:24   0:00  -tcsh (tcsh)
user3    16834    0.0    0.0    88     0    ?    IW    Apr 5   0:00  selection_svc
user3    27350    0.0    0.0   112     0    p3   IW    Apr 11  0:01  -csh (csh)
user4    23846    0.0    0.0    80     0    pa   IW    11:12   0:00  -csh (csh)
user3    23801    0.0    0.0    80     0    p8   IW    11:04   0:00  -csh (csh)
user2    18590    0.0    0.0   120     0    p7   IW    Apr 6   0:01  -tcsh (tcsh)
user2    15591    0.0    0.0   120     0    p6   IW    Apr 5   0:06  -tcsh (tcsh)
user2    15588    0.0    0.1  9288    72    p1   I     Apr 5   7:08  emacs
user2    15496    0.0    0.0   112     0    p1   IW    Apr 5   0:00  -tcsh (tcsh)
Prompt %
```

Displaying Active Processes, Continued

What to Look For

The following may indicate undesired activity:

- processes that take a long time
- unusual start times (such as 3:00 a.m.)
- unusual names
- a process that has an extremely high percentage of CPU (this may indicate a sniffer process)
- processes without a controlling terminal (a "?" in the TT column) that are executing unusual programs

Vulnerabilities

In some cases, compromised systems have been found to contain a Trojaned version of "ps" which does not display intruder processes. Also, if an invalid process is running but has a valid process name, it may be difficult to distinguish the suspicious process. For example, intruders often run sniffer processes under names such as "sendmail" or "inetd".

The "crash" Command

You can use the "**crash**" command to list all processes. This functions as a cross-check against the "ps" command. That is, finding a process with "crash" output that does not appear in "ps" output (matching pids). Once you execute "crash," you will receive a ">" prompt. Type **`proc`** in response and **`quit`** when you are finished running "crash".

```
Prompt %  crash
dumpfile = /dev/mem,  namelist = /vmunix,  outfile = stdout
> proc
PROC TABLE SIZE = 522
SLOT ST PID   PPID  PGRP   UID PRI CPU   EVENT     NAME        FLAGS
   0 s     0     0     0     0   0   0 f8172698                load sys
   1 s     1     0     0     0  30   0 f82b5494 init           load pagi
   2 s     2     0     0     0   1   0 f82b5550                load sys
   3 s   965   141   965     0  26   0 f81721f8 in.rlogind     swapped pagi
   4 s    56     1    56     0  26   0 f81721f8 portmap        swapped pagi
   6 s    59     1    42     0  26   0 f81721f8 keyserv        swapped pagi
   7 s 11039     1 11039     0  28   0 ff12a2d0 getty          swapped pagi
   8 s    73     1    73     0  26   0 f81721f8 in.named       load pagi
   9 s    76     1    75     0  26   0 f8152d2c biod           load pagi
  10 s    77     1    75     0  26   0 f8152d2c biod           load pagi
  11 s    78     1    75     0  26   0 f8152d2c biod           load pagi
  12 s    79     1    75     0  26   0 f8152d2c biod           load pagi
  13 s    90     1    90     0  26   0 f81721f8 syslogd        load pagi
  14 s    98     1    98     0  26   0 ff648d2e sendmail       load pagi
> quit
Prompt %
```

Displaying Active Processes, Continued

What to Look For	The following may indicate undesired activity: • processes that do not appear in the ps list (use the PID column to identify) • a high value in the CPU column • unusual commands in the NAME column
Vulnerabilities	Names can be faked. Like any command, “crash” can be Trojaned.

Finding the Footprints Left by an Intruder

Overview

If you suspect that an intruder has been on your system but is gone, use the commands and files described in this section to find the "footprints" the intruder may have left behind. This section shows you how to use these commands and files:

- the "last" command
- the "lastcomm" command
- the "/var/log/syslog" file
- the "netstat" command

The "last" Command

The **"last"** command displays information about logins and logouts on the system from the /var/adm/wtmp file. If you can determine the username the intruder used to log in, this command can show you how long the intruder was logged in and where they logged in from.

- The command's "-*n*" parameter is used to display the last *n* entries in the /var/adm/wtmp file.

A sample output is shown here.

```
Prompt %  last -20
user1    ftp     host1.sub.domain    Fri Apr15 15:09 - 15:10 (00:00)
user3    ttyp9   host5.sub.domain    Fri Apr 15 13:52 still logged in
user6    ttyp2   host7.sub.domain    Fri Apr 15 13:45 - 14:1 (00:26)
user6    ttyp2   host7.sub.domain    Fri Apr 15 10:34 - 10:34 (00:00)
user6    ftp     host7.sub.domain    Fri Apr 15 10:32 - 10:33  (00:01)
user4    ttyp4   host3.sub.domain    Fri Apr 15 10:17 still logged in
user5    ttyp2   host6.sub.domain    Fri Apr 15 09:20 - 10:29 (01:09)
user1    ttyh1                       Thu Apr 14 20:33 - 22:00 (01:26)
user4    ftp     host3.sub.domain    Thu Apr 14 14:21 - 14:22 (00:01)
user4    ttyp2   host3.sub.domain    Thu Apr 14 14:01 - 16:36 (02:35)
user4    ftp     host3.sub.domain    Thu Apr 14 13:43 - 13:44 (00:00)
user5    ttyp4   host6.sub.domain    Thu Apr 14 13:38 - 14:56 (01:18)
user4    ttyp2   host3.sub.domain    Thu Apr 14 13:37 - 13:47 (00:10)
user4    ftp     host3.sub.domain    Thu Apr 14 13:16 - 13:18 (00:01)
user4    ttyp2   host3.sub.domain    Thu Apr 14 13:12 - 13:18 (00:05)
user4    ttypa   host3.sub.domain    Thu Apr 14 11:13 - 15:05 (03:52)
user4    ttyp9   host3.sub.domain    Thu Apr 14 11:12 - 13:08 (01:55)
user3    ttyp8   host5.sub.domain    Thu Apr 14 11:04  still logged in
user1    ftp     host1.sub.domain    Thu Apr 14 11:01 - 11:02 (00:00)
Prompt %
```

The first column contains the username, followed by the terminal device the user is connected to. If the connection used a network device, the name of a remote system is displayed in the next column. For serial devices such as dial-up modems, the column will be blank. This is followed by the login and logout time and an indication of the length of the session.

What to Look For

- examine the log entries made around the time of the suspected attack for ones that appear to be out of the ordinary, including logins to accounts that had previously been dormant, logins from unexpected locations, logins at unusual times, and short login times
- a missing /var/adm/wtmp file or one with gaps in the output (this may indicate that an intruder attempted to hide their existence)

As a general rule, many system administrators never delete this file. Therefore, it can be quite large and include activity from when the system was first loaded.

Vulnerabilities

An intruder who breaks into a system can hide their tracks by deleting or modifying the /var/adm/wtmp file.

The "lastcomm" Command

The **"lastcomm"** command displays the last commands executed. This command is only available if you have process accounting turned on. With this command, you can see every command issued by anyone on the system. A sample output is shown here.

```
Prompt %  lastcomm
nroff            user1    ttyp1      0.39 secs Thu Sep  8 12:31
man              user1    ttyp1      0.00 secs Thu Sep  8 12:31
sh               user1    ttyp1      0.00 secs Thu Sep  8 12:31
page             user1    ttyp1      0.03 secs Thu Sep  8 12:31
col              user1    ttyp1      0.02 secs Thu Sep  8 12:31
tbl              user1    ttyp1      0.02 secs Thu Sep  8 12:31
head             user1    ttyp1      0.00 secs Thu Sep  8 12:31
lastcomm       X user1    ttyp1      0.06 secs Thu Sep  8 12:31
lastcomm       X user1    ttyp1      0.05 secs Thu Sep  8 12:31
csh          F   user1    ttyp1      0.00 secs Thu Sep  8 12:31
lastcomm       X user1    ttyp1      2.97 secs Thu Sep  8 12:28
sh               root     __         0.00 secs Thu Sep  8 12:30
atrun            root     __         0.00 secs Thu Sep  8 12:30
cron         F   root     __         0.00 secs Thu Sep  8 12:30
sh               root     __         0.00 secs Thu Sep  8 12:15
atrun            root     __         0.00 secs Thu Sep  8 12:15
cron         F   root     __         0.00 secs Thu Sep  8 12:15
sh               root     __         0.00 secs Thu Sep  8 12:00
atrun            root     __         0.00 secs Thu Sep  8 12:00
cron         F   root     __         0.00 secs Thu Sep  8 12:00
Prompt %
```

What to Look For

This command is an excellent way of seeing what a user did while on your system because it lists all commands executed by all users.

Vulnerabilities

This command produces a file that tends to get quite large very quickly as it saves the data needed to track the commands issued by every user. You should periodically rename it so that you can manage smaller files.

The “lastcomm” command only tracks the command that ran a program, but not what actions were taken after the program started (for example, it may show the editor being run, but not which files were opened after the initialization of the editor).

Many times, attacks are not discovered until days after the actual event. And in these cases, the accounting logs may have been purged by the time the attack is discovered. The biggest potential intruder-style vulnerability is that the data is kept in the file /var/adm/pacct, which the intruder can easily delete and perhaps modify if the proper privileges are obtained.

The /var/log/ syslog File

The **/var/log/syslog** file is a file that contains messages relating to various types of connections to your system. The content of this file is defined by the /etc/syslog.conf file. The results of this command contain extremely long lines; a shortened sample of this file is shown here.

```
Prompt %  more /var/log/syslog
Apr 20 13:04:22 host8 sendmail[15026]:
NAA15025:to=user8@sub.domain,
user7@sub.domain,user3@sub.domain, delay=00:00:02, mailer=smtp,
relay=computer.sub.domain. [128.xxx.xx.xx], stat=Sent (Mail
accepted)

Apr 20 13:04:23 host8 sendmail[15026]: NAA15025:
to=user5,user2, delay=00:00:03, mailer=local, stat=Sent

Apr 20 13:04:23 host8 sendmail[15026]: NAA15025:
to=user1@host1.sub.domain, delay=00:00:03, mailer=smtp,
relay=host1.sub.domain. [198.128.36.1], stat=Sent (Ok)

Apr 20 13:06:20 host8 in.telnetd[15032]: connect from
computer.sub.domain (198.xxx.xx.xx)
Prompt %
```

Most messages are from the sendmail program, and display the status of messages sent and received by your system. This file may also contain in.telnetd connection messages and other previously defined messages.

What to Look For

- Since this file saves data on incoming as well as outgoing information, especially sendmail information, one of the things to look for is outbound E-mail to suspicious hosts. This may indicate that an intruder sent out information from your system to a remote system.
- Telnet connections, both incoming and outgoing, should be examined.
- A short file may be suspicious, as it may indicate that this file has been edited or deleted. A 'hole' in the file (a large chunk of time when no messages occur) may indicate that an intruder deleted the messages related to their time on the system. Note that this 'hole' may be useful in tracking down when the intruder used the system.
- In general, look for things that may appear out of the ordinary.

Vulnerabilities

In many cases, the /var/log/syslog file is world writable and must remain so for operational reasons. Therefore, its data may be suspect and untrustworthy.

This file tends to be very long. Investigating all connections, especially sendmail messages, can be difficult. This is because at least one line is written to the /var/log/syslog file for each mail message. In addition, users tend to delete messages and forget exactly who sent them the messages, when they were received, and what they were about.

Finding the Footprints Left by an Intruder, Continued

The /var/adm/messages File

The **/var/adm/messages** file usually contains a listing of all messages that are sent to the console. The actual content of this file is defined in the /etc/syslog.conf file. A sample of this file is shown here.

```
Prompt %  more /var/adm/messages
Mar 21 10:36:04 host8 su: 'su root' failed for user1 on
/dev/ttyp2
Mar 21 10:36:08 host8 su: 'su aaa' succeeded for user1 on
/dev/ttyp2
Mar 21 16:00:59 host8 xntpd[121]: Previous time adjustment
didn't complete
Mar 24 15:01:44 host8 login: REPEATED LOGIN FAILURES ON
console, user3
Mar 25 11:42:51 host8 shutdown: reboot by user1
Mar 25 11:42:53 host8 syslogd: going down on signal 15
Mar 25 11:48:04 host8 su: 'su aaa' succeeded for user1 on
/dev/ttyp0
Mar 28 15:47:19 host8 login: ROOT LOGIN REFUSED ON ttyp3 FROM
machine.sub.domain
Mar 28 16:12:12 host8 login: ROOT LOGIN console
Apr 13 15:58:35 host8 su: 'su aaa' failed for user1 on
/dev/ttyp0
Apr 13 15:58:55 host8 su: 'su aaa' succeeded for user1 on
/dev/ttyp0
Apr 15 08:48:22 host8 named[2682]: starting. named 4.9.2 Wed
Nov 17 13:17:49 PST 1993
Apr 15 08:48:22 host8 named[2683]: Ready to answer queries.
Prompt %
```

What to Look For

The following may indicate undesired activity:

- an unauthorized user logging into the root directory
- attempts to "su" to root or a privileged account
- failed login attempts may be from a valid user making mistakes or from an intruder

In the sample file above, you would make sure that "user1" is a valid user logging into the aaa root privileged account.

Vulnerabilities

Once an intruder obtains root access, this file can be modified or deleted quite easily. Also, if the syslog.conf file is compromised, logging to this file may be discontinued.

Finding the Footprints Left by an Intruder, Continued

The "netstat" Command

The **"netstat"** command displays listening and connected processes. You should compare the output from this command with the output from the "last -*n*" command.

- The command's "-a" parameter is used to display the status of all sockets.

A sample output is shown here.

```
Prompt %  netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp        0      0  host6.sub.domain.pop   host1.sub.domain.1809  TIME_WAIT
tcp        0     78  host6.sub.domain.telne host9.sub.domain.54641 ESTABLISHED
tcp        0      0  host6.sub.domain.telne host7.sub.domain.1434  ESTABLISHED
tcp        0      0  host6.sub.domain.login host3.sub.domain.1022  ESTABLISHED
tcp        0      0  host6.sub.domain.login host3.sub.domain.1023  ESTABLISHED
tcp        0      0  host6.sub.domain.login host5.sub.domain.1021  ESTABLISHED
tcp        0      0  host6.sub.domain.telne host5.sub.domain.1957  ESTABLISHED
tcp        0      0  host6.sub.domain.login host2.sub.domain.1023  ESTABLISHED
tcp        0      0  *.printer              *.*                    LISTEN
tcp        0      0  *.731                  *.*                    LISTEN
tcp        0      0  *.pop                  *.*                    LISTEN
tcp        0      0  *.chargen              *.*                    LISTEN
tcp        0      0  *.daytime              *.*                    LISTEN
tcp        0      0  *.discard              *.*                    LISTEN
tcp        0      0  *.echo                 *.*                    LISTEN
tcp        0      0  *.time                 *.*                    LISTEN
tcp        0      0  *.finger               *.*                    LISTEN
udp        0      0  *.1022                 *.*
udp        0      0  *.1023                 *.*
udp        0      0  *.16517                *.*
udp        0      0  *.16516                *.*
udp        0      0  *.16515                *.*
udp        0      0  *.772                  *.*
udp        0      0  *.16514                *.*
udp        0      0  *.16513                *.*
Active UNIX domain sockets
Address  Type   Recv-Q Send-Q    Vnode     Conn     Refs  Nextref Addr
ff65340c dgram       0      0        0        0        0        0
ff653e8c dgram       0      0        0        0        0        0
ff64978c dgram       0      0        0        0        0        0
ff648d8c dgram       0      0 ff151508        0        0        0 /dev/log
ff64920c dgram       0      0        0        0        0        0
ff64808c dgram       0      0        0        0        0        0
Prompt %
```

What to Look For

The following may indicate undesired activity:

- you have a telnet connection that does not correlate with the output from the "who" or "w" commands
- other network connections

Vulnerabilities

In some cases, compromised systems have been found to contain a Trojaned version of "netstat" that does not show connections to or from the source of the intrusion.

Detecting a Sniffer

Overview

Sniffers are a major source of contemporary attacks. This section shows you how to use the "ifconfig" command to determine if a sniffer has been installed.

The "ifconfig" Command

The **"ifconfig"** command displays the current configuration of your network interface. Most Ethernet adaptors are (and should be) configured to accept only messages intended for themselves. An attacker must set a computer's adaptor to "promiscuous mode," in order to listen to (and record) everything on its segment of the Ethernet.

A sample output of a system in promiscuous mode is shown here.

```
Prompt %  ifconfig -a
ie0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING,PROMISC>
        inet 987.654.32.1 netmask ffffff00 broadcast 987.654.32.255
lo0: flags=49<UP,LOOPBACK,RUNNING>
        inet 127.0.0.1 netmask ff000000
Prompt %
```

Note "PROMISC" is the last parameter of the flag's description.

What to Look For

In conjunction with positive results from the above command, the following may indicate undesired activity:

- newly created files
- a process that has an extremely high percentage of CPU

Vulnerabilities

Like any command, "ifconfig" can be Trojaned. If you suspect that a sniffer has been installed, obtain "cpm" from CIAC or CERT and run it. The cpm tool will test the network interface directly and report if it is in promiscuous mode.

Finding Files and Other Evidence Left by an Intruder

Overview

When an intruder breaks into a system, information related to the attack is occasionally left behind. This information includes, but is not limited to directories, shell scripts, programs, and text files.

This section describes various files that have been found on compromised systems. Because file names can be easily changed, the actual name of the file may be different than the file names listed in this section. Many times, intruders try to hide files; methods for achieving and detecting this will be also be described.

What to Look For

When you look for files left behind by an intruder, you should:

- obtain a baseline of what your normal operating system looks like
- find files and file and directory names commonly used by intruders
- examine system logs
- inspect log files
- inspect processes
- inspect targeted files
- look for X windows intrusions

Each of these tasks is described on the following pages.

Obtaining a Baseline of What Your Normal Operating System Looks Like

To obtain a baseline of your normal operating system, you should periodically run the commands described in this document. Record and become familiar with the output from these commands. Also, obtain and periodically use SPI and Tripwire.

Finding Files and File and Directory Names Commonly Used by Intruders

The file names given in this section are commonly used by intruders. Start by looking for these file names, but realize that, as intruders learn that their bogus file names are discovered, they will change them. You must ultimately look for a name or names that do not belong.

Suspicious Files

Often, the best indication of whether or not a system has been compromised comes from a thorough examination of its file systems. The creation or modification of files is often a strong indication of intruder activity on a system.

Occasionally, the intruder will modify ("Trojan") system programs to hide the intrusion. Some system administrators have discovered that a command such as "ps" will be Trojaned to ignore the intruder's processes. Keep this in mind when running **any** command, because if a command has been Trojaned, the results of the command will be questionable.

The **"find"** command, run preferably as root, will list all files that have been modified in the previous *n* days:

```
Prompt #  find / -mtime -ndays -ls
```

Note that many intruders routinely change file modification times to hide changes made to the system. Many of these modifications may still be detected by examining a file's inode change time, which is more difficult for an intruder to forge. The following command will locate all files with inode change times that have changed in the last *n* days:

```
Prompt #  find / -ctime -ndays -ls
```

Finding Files and Other Evidence Left by an Intruder, Continued

While examining the results generated by the above commands, consider the hidden files and directories often used by attackers described in the next section, "Hidden Files and Directories."

Hidden Files and Directories

Intruders often attempt to conceal their presence on a system by using hidden files or directories; that is, those with names that begin with a "." (period). They are not displayed by the "ls" command, unless the "-a" parameter is used. The following names are commonly used by intruders:

"..." (period period period)
".. " (period period space)
".. " (period period space space)
".hushlogin"
".sh"
".xx"
".test"

Password Files and Crack

In many cases, intruders use compromised hosts to store and crack password files from other systems. Finding files that contain password entries from other systems or finding password cracking software (such as Crack) probably indicates intruder activity on your system.

Setuid Files

Unix systems allow users to temporarily elevate their privileges through a mechanism called setuid. When a file with the setuid attribute is executed by a user, the program is given the effective access rights of the owner of the file. For example, the "login" program is typically a setuid file owned by root. When a user invokes "login", the program is able to access the system with super-user privileges instead of the user's normal privileges.

Intruders often create setuid files that enable them to quickly gain root access during later visits to the system. Often, the file is placed in a hidden directory or has a hidden filename (e.g., ".sh").

Setuid files appear in directory lists with an "s" in place of the "x" in the execute bit position. For example, the output of the **"ls -l .sh"** command would display output similar to the following:

```
-r-sr-xr-x   1 root    other      86012 Jun  2 01:09 .sh
```

Note that a typical Unix system contains dozens of legitimate setuid programs necessary for normal operation of the system. Setuid files that should be suspected include:

- files that appear to have been modified recently
- unfamiliar files
- files stored in user or temporary directories

To list all setuid files on your system, use the following command:

```
Prompt #  find / -user root -perm -4000 -print
```

Examining System Logs

All Unix systems provide some level of accounting, recording the actions of both users and system processes. The amount of information recorded can vary significantly depending on both the version of Unix and its configuration. The default for many systems is to record little more than login/logout times for users. At the other end of the spectrum, systems running at an Orange Book C2 level of assurance can easily generate several megabytes of log information per hour.

To detect an intrusion, begin by examining whatever logs are available on your system. Bear in mind, however, that if an intruder gained access to your system, the information stored in the logs may have been modified to hide the intruder's tracks. Use the "last" and/or "lastcomm" commands discussed in the next two sections (and previously described above) to help you examine the logs.

The "last" Command

The **"last"** command, available on almost every version of Unix, displays login and logout activity for the system. This can be a useful place to begin an investigation. Check the login times and locations for all users and compare them to expected norms. Refer to the previous discussion of the "last" command for more information and a sample output.

The "lastcomm" Command (Accounting)

On systems with process level accounting enabled, the **"lastcomm"** command will generate a detailed list of all commands executed by each user on the system. Unusual or inappropriate system activity can often be discovered in the results from this command. For example, "lastcomm" output indicating repeated executions of the "tftp" program might indicate attempts to steal password files using TFTP.

For information on enabling process accounting on a specific Unix system, refer to the man page for "acct". Refer to the previous discussion of the "lastcomm" command for more information and a sample output.

Inspecting Log Files

Many system process events generate messages. For example, the "su" utility often makes a log entry when a user attempts to become the "super-user." These messages may prove useful in discovering unusual activity possibly caused by an intruder.

These messages are often archived in log files for later examination. Commonly used files include /var/log/syslog and /var/adm/messages; however, the file names may vary from system to system. Refer to the sections about these files in this guide or to the man page for "syslog" for more information.

~/.history

Some shells, tcsh for example, keep a record of the most recently executed commands for each user. This information is usually stored in a file in the user's home directory and is often called ".history". Examining this file may allow the reconstruction of the recent activities of a specific user.

Inspecting Processes

Look for:

- process names that are unfamiliar
- processes using an unusual amount of CPU time
- processes with names such as Crack or ypsnarf
- an unusually large number of processes

Keep in mind that process names can be changed.

Finding Files and Other Evidence Left by an Intruder, Continued

Inspecting Targeted Files	**/etc/passwd**	Look for: • new accounts • changed uid • no password • a “+::” entry
	~/.forward	The ~/.forward file is used to manipulate E-mail forwarding. When examining this file, look for any suspicious entries (that is, would it make sense for a legitimate user to manipulate his or her E-mail in that manner?).
	~/.rhosts and hosts.equiv	The ~/.rhosts file can be used to allow remote access to a system and is sometimes used by intruders to create easy backdoors into a system. If this file has recently been modified, examine it for evidence of tampering. Initially and periodically verify that the remote host and user names in the files are consistent with local user access requirements. View with extreme caution a “+” entry; this allows users from any host to access the local system. An older vulnerability is systems set up with a single “+” in the /etc/hosts.equiv file. This allows any other system to log in to your system. The “+” should be replaced with specific system names. Note, however, that an intruder cannot gain **root** access through /etc/rhosts entries.
	~/ftp Files	Directories which can be written to by anonymous FTP users are commonly used for storing and exchanging intruder files. Do not allow the user “ftp” to own any directories or files.

Finding Files and Other Evidence Left by an Intruder, Continued

System Executables in User Directories

Copies of what may appear to be system executables in user directories may actually be an attempt to conceal malicious software. For example, recent attacks have made use of binaries called “vi” and “sed”, two commonly used Unix utilities. However, these particular binaries were actually renamed intrusion software files, designed to scan systems for weaknesses.

System binaries found in unusual locations may be compared to the actual executable using the **“cmp”** command:

```
Prompt %  cmp /home/jdoe/sed /usr/bin/sed
```

Determining if System Executables Have Been Trojaned

SPI or Tripwire must be set up before an exposure in order to determine if your system executables have been Trojaned.

Use your CD-ROM to make sure you have a good copy of all your system executables, then run the above mentioned products according to the instructions that accompany them to create a basis for later comparison. Periodically, run SPI or Tripwire to detect any modification of the system executables.

/etc/inetd.conf

- Print a baseline listing of this file for comparison.
- Look for new services.

/etc/aliases

- Look for unusual aliases and those that redirect E-mail to unlikely places.
- Look for suspicious commands.

cron

- Look for new entries in cron tab, especially root’s.
- Look at each user’s table.

Finding Files and Other Evidence Left by an Intruder, Continued

	/etc/rc*	• Look for additions to install or reinstall backdoors or sniffer programs. • Use SPI or Tripwire to detect changes to files.
	NFS Exports	• Use the **"showmount -a"** command to find users that have file systems mounted. • Check the /etc/exports (or equivalent) file for modifications. • Run SPI or Tripwire to detect changes.
	Changes to Critical Binaries	• Run SPI or Tripwire initially and then periodically. • Use the **"ls -lc"** command to determine if there have been inappropriate changes to these files. Note that the change time displayed by the "ls -lc" command can be changed and the command itself can be Trojaned.
Looking for X windows Intrusions	X windows attacks are very difficult to detect because of a lack of security features. We suggest securing X systems before an attack occurs. Refer to the CIAC publication *Security for Unix Systems* (UCRL-ID-118615 or CIAC-2306) for additional information.	

Appendix A: Software Tools

Introduction

What's in This Appendix

This appendix summarizes the features and locations of several software packages mentioned in this document.

Software Tools

Check Promiscuous Mode (CPM)

The CPM tool provides an alternative means for testing the status of a network interface and determining if it is in promiscuous mode, a symptom of possible sniffer activity on the system. The CPM tool is of most use in cases where it is suspected that the standard ifconfig command has been Trojaned.

CPM is available via anonymous FTP from info.cert.org in the directory /pub/tools/cpm/.

Security Profile Inspector (SPI)

The Security Profile Inspector (SPI) tool performs security assessments of Unix- and VMS-based systems, reporting system configuration vulnerabilities, bad passwords, and violations of system file integrity. The SPI tool will also maintain a database of secure checksums for specified system directories and will alert the administrator to any changes to the contents of those directories.

For further information, call the SPI project lead at (510) 422-3881.

Tripwire

The Tripwire tool will monitor the integrity of a set of user-selected files and directories on a Unix system. The tool will detect and report to the system administrator any changes, additions, or deletions to these files.

Tripwire is available via anonymous FTP from coast.cs.purdue.edu in the directory /pub/tools/unix/Tripwire.

Appendix B: Contacting CIAC

Contacting CIAC

Phone	(510) 422-8193
Fax	(510) 423-8002
STU-III	(510) 423-2604
Electronic mail	ciac@llnl.gov
Emergency SKYPAGE	800-SKYPAGE pin# 855-0070
Anonymous FTP server	ciac.llnl.gov (IP 128.115.19.53)
BBS	(510) 423-3331 (9600 Baud) (510) 423-4753 (2400 Baud)

Reader Comments

CIAC updates and enhances the documentation it produces. If you find errors in or have suggestions to improve this document, please fill out this form. Mail it to CIAC, Lawrence Livermore National Laboratory, P.O. Box 808, Mail Stop L-303, Livermore, CA, 94551-9900. Thank you.

List errors you find here. Please include page numbers.

_____

_____

_____

_____

_____

_____

_____

_____

List suggestions for improvement here.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Optional:

Name _____ Phone _____

Stamp

Computer Incident Advisory Capability
Lawrence Livermore National Laboratory
P.O. Box 808, L-303
Livermore, CA 94551

Department of Energy

CIAC

Computer Incident Advisory Capability

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551