Intruders and UNIX Security

Matt Bishop
Department of Computer Science
University of California
Davis, CA 95616-8562

email: bishop@cs.ucdavis.edu
phone: (916) 752-8060
fax: (916) 752-4767

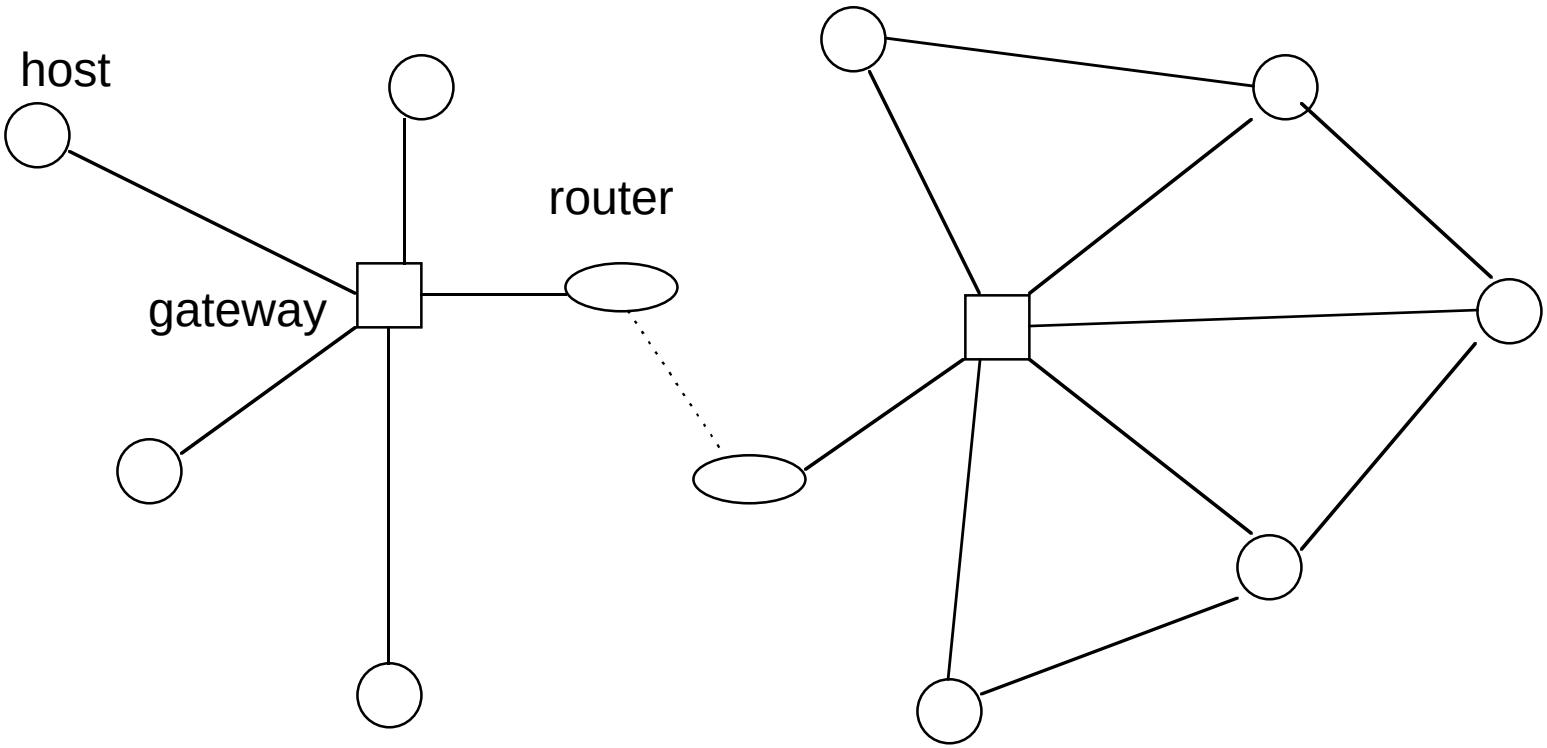
Goals

- Give you an idea of how attackers get in
- Suggest some places to look for possible detritus
- A couple of case studies

Overview of Talk

- Some Background on UNIX and the Network
- How attackers work
- How you find this out
- Case studies
- Suggestions and conclusions

The Internet



Communication Between Hosts

- Hosts appear to be connected (infrastructure, such as routers and gateways, are not visible to the applications)
- Communications broken into smaller chunks, called *packets*

  Packet size varies; can be up to 65536 characters
- Originator of request is called a *client*
- Recipient of request is a *server*

  The client sends a message to the server; the server processes it. The server may, but need not, reply.

Sending a Message

Host names are like “nob.cs.ucdavis.edu”

Network works in IP addresses

Translation done by “Directory Name Servers” or DNS

- Host sends address to local DNS, asks for associated IP address
- Local DNS returns IP address if known; if not, it contacts a DNS server that can figure it out
- DNS servers keep host names and their associted IP addresses in the same database

Key Points

- Network traffic by default is not encrypted

  Some protocols, such as Privacy-Enhanced Electronic Mail (RFCs 1421-1425) and the proposed Secure Telnet, do; they are the exceptions

- Weak origin authentication provided

  The numerical address (called “IP address”) is placed in each message; but you can supply *any* IP address (faking it is called *IP spoofing*)

- Weak data authentication (integrity) provided

  Can alter contents of packets undetectably

Network Sniffing

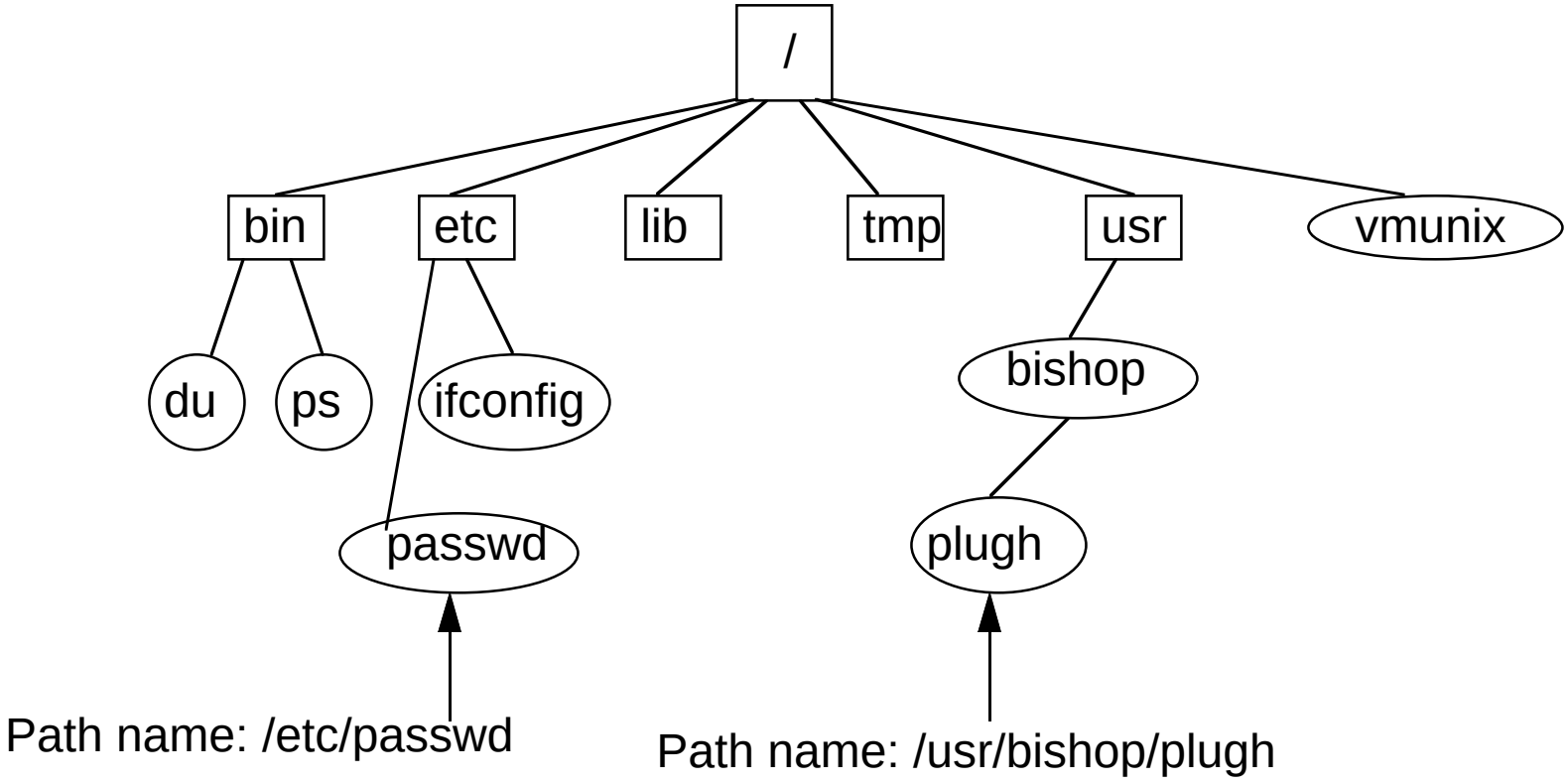
- If you can connect to the network, your host can “see” messages flowing past it, or through it, to other hosts; so you can record them
- Very common to grab account names, host names, and passwords this way

  About a year ago, done in a big way: dropped some of those programs on at least two regional servce providers, through which huge amounts of traffic flow; one estimate is the attackers go 10,000 accounts, hosts, and passwords

The UNIX System

- Developed in a research environment in 1970s by Ken Thompson and Dennis Ritchie
- Very clean, simple, elegant
  - Became popular in universities and from there spread to industry, government, *etc.*
- Designed by programmers for programmers
- Many systems on the Internet are UNIX-based

The File System



Useful Commands

- login

  Access system; must supply an account name and a password. If logging in over network and remote host trusts current host, password not used

- ps

  List information about processes, including command and owner

- ls

  List information about files, including owner and when last changed; information about files whose names begin with "." must be specifically requested

More Useful Commands

- ifconfig

  Reconfigure state of network interface; shows if network interface will present only those packets destined for this host, or if it will present all packets it sees (this is *promiscuous* mode)

- netstat

  Report status of network connections and interface, including names of hosts and services being accessed

Useful Facts

- *root* user

  Like *operator* or *wheel* user on other systems; a privileged user to whom no access control is applied. Get access to this user and you can do anything you like on the computer

- /etc/utmp, /etc/wtmp

  Files recording who is currently logged in (utmp), and who is or has been logged in in the past (wtmp)

- /etc/passwd

  Contains a list of all accounts and passwords (stored in an encrypted form); common attack is to grab this file and try to guess passwords

Useful Facts

- script

  A command file which may run other programs

- shell

  Command interpreter

- checksum

  A mathematical function which produces a number from a file, such that it is highly unlikely (say, the odds are about 70,000,000,000,000,000 to 1) that another file can be found to produce the same value for the checksum

Trust, UNIX, and the Network

- /etc/hosts.equiv, /usr/*user*/.rhosts

  List of trusted hosts; in the first file, any user on any host listed may log into the current host without giving a password; in the second, the named user from the named host may log into *user's* account without giving a password

- NIS (Network Information Service)

  Allows one central host (the *NIS server*) to contain the password and trusted host lists, and other hosts (*NIS clients*) to ask it for that information

- NFS (Network File System)

  Allows a host to share files with other hosts, like NIS but for files

Electronic Mail

On UNIX, server is called *sendmail*

Notorious for security problems, among them two special keywords sent by a client:

wiz gives you control of the remote system

debug allows you to execute commands on the remote system

Allowed to send mail to commands (which treat the letter body as input to the command)

Some versions allow appending to files

Basic Rule: *Plus ca change …*

- In past, attackers used keyboards
- Now use scripts
- Holes exploited are (usually) old ones

  For example, *rootkit* uses password sniffing; some scripts check debug and wiz command on sendmail

- Scripts do clean-up and hiding

  Rootkit generates fake du, ps, ifconfig, etc. to conceal its use on a system

What This Means

- Can't depend on catching attackers in the act
- Look for what they do
  - What was their goal?
  - What changes did they make to realize the goal?

Example: *rootkit*

- Distributed set of binary tools
  - patched versions of: netstat, ps, login, ls, (BSD and SYS V), du (BSD and SYS V), ifconfig
- Distributed set of concealment tools:
  - fixer: installs rigged programs, tries to fix up permissions and make new checksums
  - zapper: deletes /etc/utmp, /etc/wtmp entries for this user
  - sunsniffer: used to sniff network

So What to Do?

Look for anything that feels wrong

- Someone caught this because "ps" kept dumping core. If you look in the *rootkit* ps, this means a file in /dev is misconfigured and contains a carriage return
- Conclusion: the hacking tools aren't too robust in general

Emphasize This

- You rarely find successful attacks by looking for them; you usually find them when
  - you get a call from someone who is under attack from your site, or who found your site in the remnants of an attack at their site
  - something breaks and you can't figure out why
  - a very obvious attack is made

Goals May Dictate Visible Points

Example: Goal is to establish a site for pornographic pictures or pirated software

- takes up disk space, and if disk full write errors occur
- du may not show this (ex. in *rootkit*)
- df may show it (ex. not in not in *rootkit*)

One Experience

- Research/WWW host (used for editing, preparing, and posting documents, mainly) showed the following over 2 days
  - 50% decrease in free space on one disk
  - network connections increased by 100%
- What they did
  - Used "find" to scan file system; directories named "/usr/ftp/pub/inbound/..<sp><^h>" found

One Experience (2)

- Left it in place to see who came in from where
  - Used a clean version of netstat and tcp_wrappers
  - Stored logs on trusted system (another one!)

Results

- Found little of interest, "usual collection of suspects"
- Notified site administrators
- Deleted files (since they needed the space)

Another Obvious Attack

- Superuser tries to log in, but root password changed
- Got in by rebooting, resetting /etc/passwd
- Found attacker by checking syslog

  There was a recrd of a letter from "/dev/null" to "/bin/sed"

Logs and Such

- Attack scripts generally do not change logs
  - /etc/utmp, /etc/wtmp are exceptions
  - don't know of any syslog changer
  - could always edit by hand, of course …

How Common is Such Editing?

Really rare

- Attackers use out-of-the-box scripts
- You can still see attacks using sendmail debug, wiz
  - and yes, they still do work some places!!!

How Common is This?

But possible

- if to remain on host undetected, must do this
- if this happens, will need to take over machine completely to be undetected
- very likely one of a myriad of things will go wrong
- you will see unexplained problems

Example: A Subtle Change

- Not typical now, in that it doesn't involve a network and the attackers were there to stay
- Typical in the evasion attempt and subtlety of the attack
- Also, a good example of an insider attack

What Was Noticed

- Operating system crashed
  - Reason unknown; it had been stable for some months.
  - Core image taken, system rebooted
  - PM analysis begun
- Fault occurred in accounting section
  - Not changed recently
  - Source looked okay

The Clues

- On disassembly, found two UIDs not to be entered into accounting records
- Looked more, found same two UIDs to be treated as 0 (root)
- Froze users' accounts, looked in them ... nothing

The Answer

- Checked user backup tapes for those users, and found (in a dump made 6 months ago) a program to spoof a login banner
- Also found file of names and passwords; last one was root's

The Lesson

- Something not too rare (a system crash) led to the discovery of a major compromise
- Pay attention to the little things
- Ask, "Why?"

Depends on Goal

- Use system
  - leave backdoors, rig software to hide attack
- Pass-through
  - Hide trail in attacking other sites
- Access Network
  - As in "use system," change network programs too and install sniffers (use disk space)
- Infrastructure Attacks
  - Network access, changes to DNS, taking over firewalls

Rarely Only You

- College X got call from University Y
  - someone from a host at X was sweeping Y's hosts for vulnerable ports
  - reverse finger showed 2 users, one local, one from Z.edu
  - reverse finger to Z.edu showed one user, guest

This Meant …

- Conclusion: X had a compromised account
  - User's jobs terminated
  - Account password changed

Had Y not called, X would not have known this

A Remote Attack

- Goal is to get trust information
  - NIS server map is "hosts.byname" or "hosts.byaddr"
- Run program to get domain name
  - need to know NIS server name, and any client that boots from it
- Get copy of desired map
  - Can get other information like password file

Characteristics

- No presence on host machine
- Used network, authorized features of system
- Very hard to trace
  - One shot: if you notice the information going out, or the request coming in, during the attack

More Likely ...

- Notice what will follow when attacker tries to use that information

Routing Attack

- Send records to routers "updating" route to target saying that you have a much better route
- Router will happily send everything to you, which you read and forward on to target

Other Infrastructure Attacks

- Router: another computer
  - Can attack it like any other machine
  - Put a sniffer on it and read information
- Firewalls: these are trusted to keep things out
  - If it runs *sendmail* or NIS or NFS, or is not kept up-to-date, it's a good target
  - One reason why a firewall should *never* be seen as a solution, but only as a barrier

Again, This Means

- Attacks need not imply a presence on machine
- Attacks can be incredibly subtle, involving only a few modified files or programs
- Attacks usually involve more than one host
- Infrastructure attacks becoming more popular

Lessons from above

- Know your system's characteristics!
  - More formally, anomaly detection
- Have a clean set of tools handy
  - So you can trust these
- Know what files user interaction affects
  - Attackers sometimes miss things
- Check against a clean copy
  - Either an integrity-checking database or a clean copy of the system

Know your system's characteristics

- Compiles on a gateway are not cool
- There's lots of room on a disk one night, none on it the next

Have a clean set of tools handy

- Directory and file examining tools (ls, find, du, df, od, cat, lsof)
- Program examining tools (nm, strings, od, disassemblers, debuggers, core image viewers)
- Process examining tools (w, ps, iostat, vmstat)

More Handy Tools

- Network examining tools (ifconfig, netstat, cpm)
- Packet examining tools (tcpdump, snoop, etherfind, etc.)
- Log examining tools (depends on the log; cat is often good)

Where to Look

- passwd, group files

  accounts or groups added or changed; in particular, look for UIDs of 0 or system accounts, additions to group wheel
- utmp, wtmp files
- accounting files, if you use them
- syslog files

  unusual source, destination addresses for mail
- check for unexpected modules being loaded into operating system

  On Solaris, ttywatch is a common one

Other Good Things to Check

- sulog

  check for unusual locations or times
- all log files

  if they shrink, you're in trouble (unless you prune)
- root (or system) account startup files
- system directories may have files, subdirectories with funny names
- status info of system files, binaries
- environment of running processes

  ps -wwe

Some Network-Related Stuff

- users' .rhosts and /etc/host.equiv

  root is a user
- if running NIS, check your yp files
- check network interface
  - is it promiscuous and not supposed to be?
  - check connections, especially if trust is involved; lots of half-open ones may mean you are seeing an IP spoof in progress (***real*** long shot)
- ident is a starting point ***only*** in this context

  too easy to fool

Clean Copy

- attackers have tools to modify system binaries to hide their presence
- have been known to recompile system with modifications and reinstall it

  one got caught because netstat -l dumped core unexpectedly, and they found the real tree under "..<sp><^h>" in /usr

- can modify kernel on the fly; use adb
- conclusion: can't trust binaries on a compromised system

Must Detect This: Integrity Checker

- on an isolated host, build an integrity checking database from a clean copy
  - clean is important
- store database, checking software on write-only media
  - if you need to change it, rebuild the disk after changing it on a clean system
- periodically, run this check

Problem: kernel could be modified to "read" file incorrectly

- when asked for executable's contents by a program, present correct data from hidden tree
- when executing, use rigged binary

  similar things are routinely done to defeat virus scanners on PCs

Solution: Supply Your own Kernel

- have a spare disk with a clean copy of the system, including kernel
- to check existing system, reboot it from the clean disk (with no network connections or users, of course)
- run the integrity checks from it; this could simply be a compare

It's All Relative

- Each attack has its own unique characteristics; need to roll with them
- Letting them come in and watching can teach you a lot
- The only rule: there are no rules

Conclusion

- This is still an art and not a science
  - It probably always will be, because the attackers are people
- As attacks are less interactive and more script-driven, detecting attacks from logs becomes more important
- We're in a very employable field!

Words of Wisdom

When it seems hopeless, remember Dorothy Parker's words:

```
Razors pain you;
Rivers are damp;
Acids stain you;
And drugs cause cramp.
Guns aren't lawful;
Nooses give.
Gas smells awful;
You might as well live.
```