Adventures in Hackery

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

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

Goals of This Talk

- To describe incidents that happened
- To describe incidents that should have happened
- To describe incidents that I’d like to see happen

Why?

To illustrate the following themes:

- Security problems arise from many sources, including attempts to secure!
- Know the capabilities of security mechanisms you add
- Know when to stop adding security mechanisms
- Know whom to listen to, and when

Slammer

- Attack tool that depends on Secure RPC being enabled
- Requires that there be no entry for the server
  - *root* uses this entry
  - Without it, *root* is remapped to *nobody*
  - *nobody* has a well-known key

Attack

- Request a NIS map update and provide *nobody*’s credentials
  - “| cp /bin/sh /usr/etc/in.telnetd” is a good one
- Server tries to get local site (root) key
  - Not there, so substitutes *nobody*’s key
- Attacking process authenticates itself properly
  - Then update is pushed out

Why Does It Work?

- Secure RPC does improve security but only when properly configured
  - Mis-configuration causes a security problem
- Other programs have problems too
  - Mechanism to push maps out should detect the bogus name
  - Secure RPC should fail, not substitute *nobody* key

Point

Security mechanisms can be security vulnerabilities

- Be sure they do what you expect them to do
- Be sure they are installed and configured correctly
  - Read the manuals
  - Try them out
- Be sure they are appropriate for your environment

Firewalls and Fun

- Proxy firewalls and filtering firewalls are different
- Proxy Firewall
  - act on your behalf, like a staging area
- Filtering Firewall
  - filter packets with respect to port number

Assume someone on inside wants to violate policy

Tunnelling with a Filtering Firewall

- Put illicit servers on authorized ports
  - A telnet server on port 25, for example
- Use authorized services in unauthorized ways
  - Install an old version of *sendmail* and use the debug or shell commands
  - If filtered with respect to origin, wait for a good connection to pen and then steal it

Tunnelling with a Proxy Firewall

- Encode forbidden protocol in an allowed one
  - Allow email but not ftp? No problem; do ftp over email
  - Telnet through email is painfully slow but can be done
- Proxies can filter on content
  - So encode the lower-level protocol into an innocuous message …

Porous Firewalls

- Some protocols are so non-secure no firewall will help you
  - Think CGI, Java and Active-X
- Firewalls can do little
  - Can block applets, but this also blocks WWW

Point

Be sure you know what crosses the perimeter

- If external data (code) crosses the security perimeter, and something then uses that data (executes that code), the strongest perimeter mechanisms can’t help you
- Firewalls are not a solution; they are a tool used in support of a solution

The Wayward Modem

- US-based company built a very good firewall (to my knowledge, it hasn’t been broken yet)
- Company then ordered all external access points blocked, so all connections had to go through the firewall(s)
- The Singapore sales office didn’t disconnect their modem ...

Point

- If you use a firewall, the connections have to go through it before it can work
- All it takes is one way to bypass a firewall

Question

Do firewalls cause complacency and relaxed internal security?

Security by Experts

- Management became security-conscious
- Read Cheswick & Bellovin and told firewall administrator to block all ports except 25 (email)
- Site used NTP to sync clocks with external time
  - This was a policy requirement

Point

- What an expert says in a book may not apply to your particular situation
- Security is not a cookbook exercise; you have to think

The (censored) Attack

- Asked to do a vulnerability analysis of a particular site
- Given user account on one system on network
- Goal: get access to some sensitive data on a separate system

First Step

- Scan system looking for configuration vulnerabilities
- External scanning via SATAN, ISS, nfsbug showed system secured
- Identified DNS server for that host
- Decided against changing those records for the moment

Next Step

- Grabbed password file, downloaded it and ran *crack*
- Found 6 hits in that file
- Tried accounts on target and found one that worked
- Once on there, repeated above process
  - This sucker was ***really*** secured!

Last Step

- Noticed OS was Solaris 2.3
- Tried pushing mouse interpretation module on top of keyboard stream
  - It worked! They were two patches out of date!
- Now had root.

Just For Fun

- Looked for *snoop*; didn’t find it
  - Not a problem; they downloaded their version
- Ran it looking at the ftp, telnet and rlogin ports
- Found *root*’s password very quickly
  - They promptly got out and reported their actions and results

Point

- Keep up to date on security patches
- Deleting security-sensitive software (like a network sniffer) from your system won’t prevent it from being used

The Unrestricted Restricted Shell

Guest account shell *chroot*ed to /usr/home/guest

- Home directory was writable by user (*guest*)
- Subdirectory “bin” had link to standard executables (in /bin, /usr/bin, *etc.*)

The Attack

- Create /etc, then /etc/passwd
  - Single line was

```
meroot::0:0:got it:/:/bin/sh
```

- Ran *su*
  - Used /etc/passwd above, as *chroot* inherited

The Break-Out

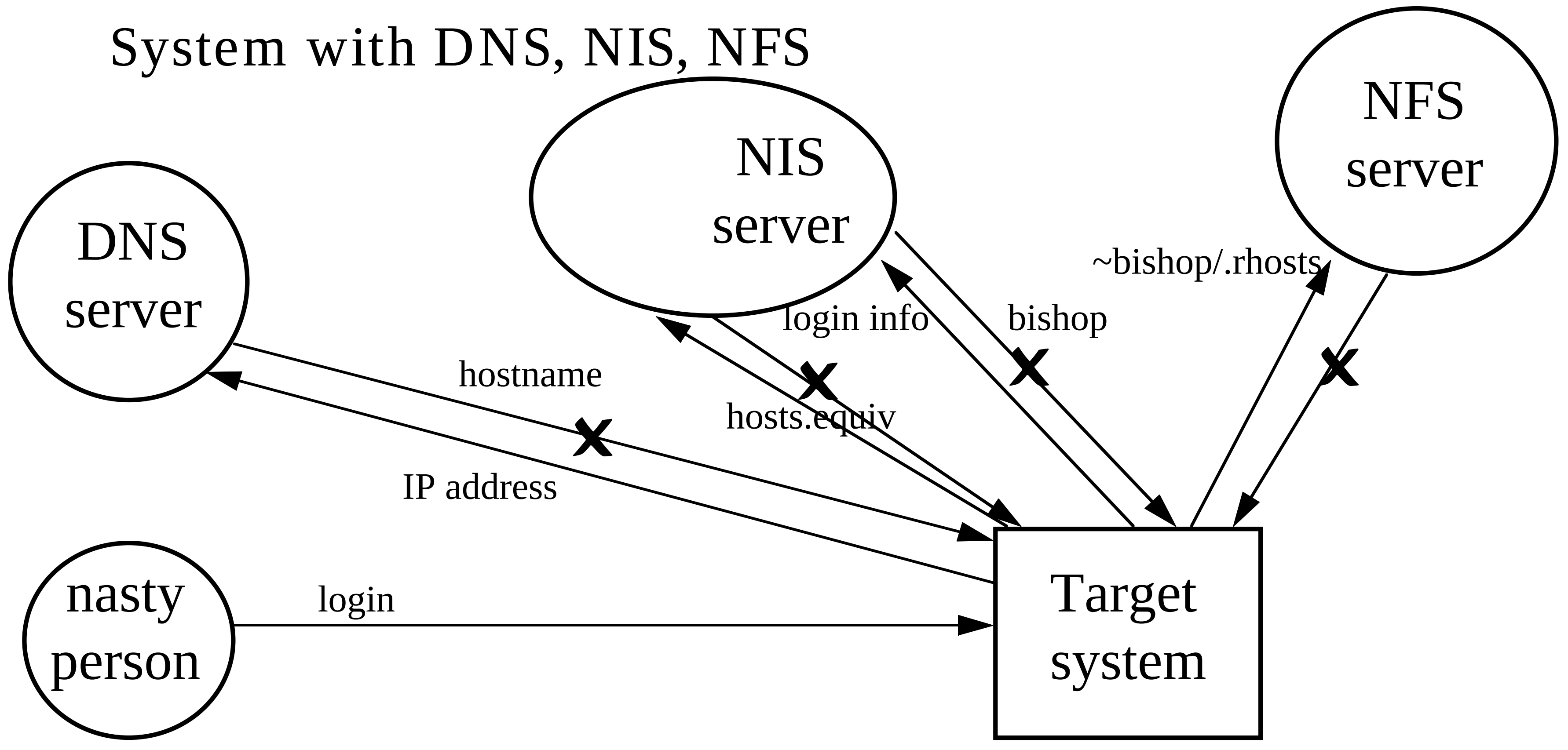
- As *root*, attacker wrote a program to create a device corresponding to the kernel memory
- Wrote into that device at the proper location to update the *rootdir* field of the *guest* shell entry in the process table
- Presto! Out of the restricted area

Point

- You must control *everything* in a restricted account
  - *chroot*ing is not enough
  - no setuid / setgid programs should be available in that area
- Don’t try to restrict *root*

Network Attack

System with DNS, NIS, NFS



✗ Where nasty person can attack

Matt Bishop
Dept. of Computer Science
University of California, Davis

Point

- Know your system’s design
- You can’t rely on non-secured external resources
- Relying on information from secured internal sources but sent over a non-secure medium is questionable at best

Apocryphal Story

- Company has *root* account without a password
- Company has modems
- Modems have an 800 number for dial-in

Did They Know?

- Ann Nonymous heard about this and dialed in
- Logged in as *root* without supplying a password
- Immediately disconnected and telephoned company

Did They Care?

- Company got Ann’s name, address, phone number
- Company filed criminal complaint to have her prosecuted for breaking into the company’s computers

Did They Win?

- After bouncing around for 3 years, an attorney in the prosecutor’s office called a friend who happened to be a computer security expert
- After discussion, decided to drop all charges

The Aftermath

- Others tried to dial into the company’s modem bank using the 800 number
- Others got root, again without supplying a password
- They did not report it

Point

- Stupidity and security are contradictions
- If you get attacked, close the doors ***before*** you prosecute ...

Conclusion

- Security requires:
  - Policy and understanding your goals
  - Planning, design and implementation
  - Tools, correctly configured and installed, that support your plan
  - Procedures that support your plan
- It takes care, planning, and maintenance