Practical Computer Security

What every grad student should know about computer security

Ethan L. Miller
elm@cs.ucsc.edu

Who needs computer security?

- We trust computers: machines don’t lie
- People can subvert computers in many ways
  - Obtain data they shouldn’t be able to get
  - Plant false information
  - Deny service: prevent others from using the system
- Vigilance is key!
  - Prevention is much easier than fixing the problem
  - Knowing the risks is crucial for prevention
- A little prevention goes a long way....
Passwords

- Passwords are the way you show the computer who you are
  - Anyone with your password can pretend they’re you
- Take precautions
  - Pick a “good” password
    - Easy to remember
    - Unrelated to any personal information
    - Not in the dictionary!
  - Don’t share your password with other people (even unintentionally!)
    - Don’t leave your password lying around
    - Be careful typing your password with others looking over your shoulder

Good and bad passwords

- Bad passwords
  - Szczurowska (yes, it’s hard to spell, but it’s your wife’s last name!)
  - SantaCruz (pretty obvious thing to check)
  - marbles (there are a limited # of words in the dictionary)
  - Jeremiah (same goes for names)
  - Pxq (hard to guess, but only 3 letters long)

- Mediocre passwords
  - b00tbL0ck (substituting 0’s for O’s, weird capitalization)
  - &soItGoes
  - #cheDDar (using symbols for words)

- Better passwords: use the initials of a phrase
  - IwIwaOMw (I wish I were an Oscar Meyer wiener)
  - SWE2:AotC (?)
  - GHWB:Rml-nnt (?)

- Don’t use any of these specific passwords!
  - The slides may be on the Web, making it easy to guess
  - Pick your own using these heuristics
Logging in

- Don’t use telnet or rlogin!
  - Password sent unencrypted
  - Entire session visible over the network
  - Most SoE machines require encrypted connections, but outside computers (unix.ic!) may not require it
- Instead, use secure shell (ssh)
  - Available on cse, unix.ic and other Unix machines
  - Free clients available for Windows, Mac
- Secure shell provides
  - End-to-end encryption: no way to eavesdrop
  - Password sent encrypted
  - Ability to encrypt other network connections (more about that later…)

Why are networks insecure?

- Any information sent over the network may be visible to everyone on the network
  - Standard programs for “sniffing” network packets
  - Standard programs for sifting through those packets
  - Wireless networks only make matters worse
- If you don’t want someone to see it, use encryption
  - Secure shell
  - Secure HTTP (https)
  - Secure sockets layer (SSL)
- Particularly true for shells, copying files, mail
Verifying the other end

- SSL & SSH encrypt traffic between your computer and the other end
  - This is *always* the case
  - Nobody can read the traffic in transit
- How do you know the other end is *really* who it claims to be?
- Solution: present a *certificate* to set up SSL or SSH
- Two ways of verifying certificate’s authenticity
  - Certificate is *signed* by someone you trust (SSL)
  - Certificate *fingerprint* is verified by the user (SSH)
- Our SSL certificates aren’t signed properly!
  - Find the fingerprint and remember it
  - First use SSL within SoE (low chance of attack) and have your computer remember the certificate

Copying files

- Avoid using ftp
  - OK to do anonymous ftp (no login involved)
  - Password for ftp sent in the clear!
- Use scp (part of ssh) instead
  - scp file1 file2 unix.ic:
    - Copies file1 and file2 to unix.ic using same account on both
  - scp file1 file2 elm@unix.ic:files/bar
    - Copies file1 and file2 to unix.ic using elm account on unix.ic, and placing the files into files/bar
- Ssh/scp clients exist for Unix, Mac, Windows
  - http://www.ece.nwu.edu/~mack23/ssh-clients.html
  - Google search for ssh clients
- Alternative: secure FTP (sftp)
  - Sundance has an sftp server
Securely reading your email

- Read email using pine over ssh
  - Just as secure as anything else over ssh
  - Often limited to text-only
- Use POP or IMAP for email
  - Send password encrypted (APOP, etc.)—mail itself transferred in the clear
  - Use IMAP over SSL (supported by SoE)—all information is encrypted (password & data)
- Security is useful for two reasons
  - Nobody can read your passwords & mail
  - Nobody can insert fake messages…

Securely sending email

- Mail servers may reject mail if they don’t trust its source
  - Reduce spam
- No problem if sending mail from SoE network
- Sending mail from off-campus
  - Use your ISP’s mail server (ask them for instructions)
  - Use SoE’s server after authenticating yourself
    - **IMPORTANT**: use SSL to make sure your password isn’t sent in the clear!
    - Most PC / Mac programs have support for this
Email isn’t always what it seems

- Email relies on trust
  - Servers will accept mail from anywhere
  - There’s no authentication done on the “From” field!
  - Email can be modified by anyone *en route* from sender to receiver
- Spoofing email is (very) easy!
  - Telnet to port 25 on a mail server
  - Type the appropriate commands to it
  - Provide the requested (fake) data
  - Send the mail off
- Spoofs are undetectable if you’re logged into a machine that the spoofed sender would use!
  - Anyone with a CATS account can spoof anyone else with a CATS account!

Authenticating email

- Use cryptographic mechanisms if it’s important
  - PGP (Pretty Good Privacy) allows you to encrypt or sign email
  - Signed email is very difficult to forge, but is readable by anyone
  - Encrypted email is only readable by the recipient
- If it seems strange, ask in person or by email
  - Extremely difficult to forge face-to-face contact!
  - Email exchange is harder to spoof than a single message
- Consider whether someone could gain by forging email
  - Fake message that office hours are cancelled does no good
  - Fake message that project deadline has been extended…
Security at home (Linux & otherwise)

- Many people now have Linux boxes at home
  - Flexible, fast, cheap
  - Really easy to break into!
- Security with a Linux box is much better if you
  - Turn off all Internet daemons except ssh
    - Do you really need ftp, http, etc?
    - Major security holes if misconfigured
  - Check system logs on a regular basis
  - Use ssh to “tunnel” X Windows or VNC connections to other hosts
    - Encrypts all traffic, including passwords
    - Makes it much more difficult for an intruder to watch what you’re doing
- Better approach: buy a hardware firewall
- Threat comes primarily from outside UCSC!

Protecting your files & data

- Previous techniques protect data “in transit”
- Also necessary to protect data while it’s sitting on the computer
  - These “attacks” are primarily from UCSC insiders
- Use Unix permissions and/or ACLs to protect your data
- Don’t run untrusted programs in your account
  - This includes precompiled student binaries!
- Use a virus checker on your PC/Mac
  - Available free for UCSC students, faculty, staff
  - [http://www2.ucsc.edu/cats/sc/software/](http://www2.ucsc.edu/cats/sc/software/)
Using Unix permissions

- Three permission bits (r, w, x)
- Three sets of people (user, group, other)
  - W should be enabled only for user
  - R should be enabled for whoever’s going to read the file
  - Only enable x for executable files
- Directories have similar bits
  - R allows people to list the directory
  - X allows people to access a file if they know the name
    - Very useful for public directories such as Web stuff…
  - Disable w for all but user
- Default policy should be to disable access for everyone but user
  - Particularly true for stuff you’re doing as a TA

Setting Unix permissions

- View current permissions with “ls -l”
- Set permissions with chmod
  - `chmod g-w` removes write permission from the group
  - `chmod g+r` adds read permission for group & others
  - `chmod g+rx` adds read permission and execute permission for group & others
    - Execute only given if user can already execute the file
- Use the -R flag to set permissions recursively
  - Be very careful with this option!
  - Useful for making a directory tree accessible
Permissions on IC systems

- IC systems use AFS
- AFS supports access control lists (ACLs)
  - Grant permissions to specific users or groups
  - Ordinary users can create permission groups
- Permissions in AFS cover directories (not individual files)
  - All files in the directory have the same permissions
  - AFS permissions override Unix permissions
- Read the man pages for the `fs` command

Dealing with students’ programs

- If possible, have the student log in
  - Do `ssh student@unix.ic` and have the student type in her password
  - Don’t run the programs logged in as yourself!
  - Don’t let students see your password
- Compile the files yourself when grading
  - Precompiled binaries may have trojan horses
- If possible, look over the code before running it in your account
Things to remember

- An ounce of prevention is worth a pound of cure
  - Cleaning up from intrusions is very difficult
  - Doing things securely isn’t tough once you’ve got security set up
- Don’t send your password in the clear
  - Use secure shell (ssh) instead of rlogin, telnet, ftp
  - Use secure email (over ssh or SSL)
- Pick a good password
  - None of these techniques are any good if someone can easily guess your password!
- Take care to protect your files