FSEM 1111 Computer Security – from a Free Software Perspective

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

Welcome to Debian GNU/Linux

- Debian – Distribution
- GNU – Free Software Organization (essential applications)
- Linux – Operating System

KDE

- KDE uses Qt
- Qt uses X11
- X11 uses libc
- libc uses Linux

Layering

- Computers are too complex for anyone to understand all the details
- Abstraction $\equiv$ generalization by removing detail
- Layering $\equiv$ abstraction in multiple, explicit steps

"All problems in computer science can be solved by another level of indirection." – Butler Lampson

So what do these layers do?

- KDE: Desktop environment
- Qt: Widgets
- X11: Graphics
- libc: Primitive operations, common abstractions
- Linux: Hardware virtualization, networking, access control

The Shell

- Simple (interpreted) programming language
- Uses simple commands: cat, echo, expr, find, grep
- Provides simple & powerful operations for combining commands
- Take COMP 2400 for a course on the shell and shell commands

The File System

- /bin – essential **bin**aries ≡ programs!
- /lib – essential **lib**raries
- /sbin – essential super-user binaries
- /boot – system startup
- /etc – system configuration
- /media – removable media (CDROM, floppy, USB sticks)

The File System

- /dev – devices $\equiv$ hardware!
- /proc – **proc**esses (operating system)
- /root – home of the super-user
- /usr – non-essential applications
- /opt – commercial (non-Debian) applications
- /home – user data

The `/usr` File System

- /usr/bin/, /usr/games/ – binaries
- /usr/lib/ – libraries
- /usr/src/, /usr/include/ – source code
- /usr/local/ – like /usr, just for non-Debian applications
- /usr/share/ – data
- /usr/share/doc/ – documentation

UNIX File Permissions

- Standard permissions: Read (4), Write (2), eXecute (1)
- Differentiation by: User, Group, Others
- `man chmod`, `man chown`
- Default permissions are $arg\& \ mask$ where `arg` is specified by the application. For `mask`, see `man umask`

Environment Variables

- HOME – where is the user's data?
- PATH – where are the binaries?
- USER – who is the user?
- HOSTNAME – what machine?
- `set` lists your current environment

Editing Text

- There are many editors out there
- Computer Science students must learn `vi` or `(x)emacs`
- `kwrite` is a simplistic editor
- I use `vi` and `xemacs`, you can also use `kwrite` or `kile`

UNIX Philosophy

"Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface." (Doug McIlroy, UNIX pipes)

In UNIX, everything is a file – this keeps the number of system calls down by providing access to APIs via files.

Add complexity (in particular visual interfaces) at a higher level. The OS presents things at the lowest level possible: files and streams of bytes.

UNIX Philosophy...

... according to Mike Gancarz:

1. Small is beautiful.
2. Make each program do one thing well.
3. Choose portability over efficiency.
4. Store data in flat text files.
5. Avoid captive user interfaces.
6. Make every program a filter.

Filters and Redirection

- Filters – a filter is a program that accepts input, transforms it, and outputs the transformed data.
- With input/output redirection, a program can take input or send output someplace other than standard input or output – to a file, for instance. Redirection of STDIN is accomplished using the $<$ symbol, redirection of STDOUT by $>$.

Pipes

- The pipe (|) is a junction that allows us to connect the standard output of one program with the standard input of another.
- Example: `ls -l | less`

Filters and Pipes

- `tr` is a filter that replaces letters in the input matching its' first argument with letters in the input matching its' second argument
- Example:
  `echo 'Hello World' | tr l '\n' | wc -l`

UNIX?

- UNIX as I use the term refers to a family of operating systems
- This family includes FreeBSD, OpenBSD, NetBSD, OS X and GNU/Linux
- They share certain common features – like, a shell
- We will learn more about UNIX in the rest of the course

Oh, and GNU stands for "GNU is Not Unix".

Questions

?

Homework

- Learn to use an editor – use it for shell programming!
- Read the tutorials on shell programming
- Play with the shell
- Write a shell script to compute $n!$
- Remember: practice makes perfect, spent time using the shell in the lab this week!