LIS900

Lecture 0

Introductory Lecture

Thomas Krichel

2002–05–13

Structure

1. Philosophy and background
2. Free software
3. *student comments and discussion*
4. elementary computer security
5. meet wotan

personal background

- trained as academic economist
- interest making research resources for economists available on the Internet: "leisure librarian"
- opened NetEc in 1993
- opened RePEc in 1997
- co-founded Open Archives Initiative in 1999

now very tired from all this. . .

traditional approach

- resource awareness
- resource usage
- resource evaluation

non-traditional approach

- resource creation
- resource description
- non-resource description (Arms' critique)

discussion of student background

- What background do you have?
- What are your long-run goal?
- How does this course fit in?

Software anatomy

Basically, software can be distributed in two ways.

- “binary” code
- “source” code

Binary code

looks like this, for example

```
^?ELF^A^A^A^@^@^@^@^@^@^@^@^B^@^C^@^A^@^@^@p<9A>^4^@^@^@X=
^G^@^@^@^@4^@ ^@^F^@(^@^_^@^\^@^F^@^@^@4^@^@^@4<80>^4<80>^
^@^@^@^@^@^@E^@^@^@D^@^@^@C^@^@^@^@^@^@<80>^<80>^^S^
@^@^@S^@^@^@D^@^@^@A^@^@^@A^@^@^@^@^@^@^@^@<80>^^@<80>^
^B^@^B^@^E^@^@^@^@P^@^@^A^@^@^@^B^@n^n^^K^@^@^@^@
^F^@^@^@^@P^@^@^B^@^@^@$^B^@$x^$x^^@^@^@^@^@^@^F^@^@^@^D
```

It will run on a compute with one OS, may not run on a computer with another.

It can not be modified.

It is difficult to find out what it does.

Source code

```
/* For now, don't try to include termcap.h.  On some systems,
   configure finds a non-standard termcap.h that the main build
   won't find.  */

#if defined HAVE_TERMCAP_H && 0
#include <termcap.h>
#else
extern void tputs P_ ((const char *, int, int (*)(int)));
extern int tgetflag P_ ((char *id));
extern int tgetent P_ ((char *, const char *));
#endif
```

This is human (geek) readable code. May be understood by humans. Can be changed. Needs a compiler software to translate it to translate it to binary software.

free speech and free beer, according to FSF.

Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1).
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. (freedom 3).

Access to the source code is a precondition for freedom 1 and 3. For that reason, some people refer to free software as open source software.

free speech and free beer

- "Public domain software", not copyrighted, but modified copies may not be free.
- "Copy-lefted free software", comes with the permission to use and modify, but prohibits adding further restrictions to the distribution.

The GNU Public License is the most famous and widely used copyleft implementing software license.

Debian GNU Linux

Is a whole operating system that consists of free software. Some non-free software is added in. Compared to an MS system, a Debian system is

- less GUI in orientation
- more configurable
- much more secure
- much more stable

It is expert friendly rather than user friendly. Since we want to be experts, we will use this stuff.

Most importantly, it comes with a free web server apache.

But we need secure methods to communicate with this machine.

Cryptography

In olden days, used by military, diplomatic corps, diarists and lovers. For the military, difficulty of encryption in the battlefield was main problem.

Original data, called "plaintext" is transformed by a function parameterized by a "key", to get cyphertext. The art of cracking the code is called cryptanalysis.

We should assume that the cryptanalyst knows the general method, but does not know the key. But the main problem in communication is keeping the key secret.

Diffie and Hellman (1976)

Let P be the plaintext, C be the cyphertext. $E()$ is the encryption key and $D()$ is the decryption key.

Then we have public key cryptography if

1. $D(E(P)) = P$.

2. E can not be broken by chosen plaintext attack.

3. Knowing E will give you no clue about D.

E can then be made public and is referred to as the public key, D is the private key.

It is possible to find key pairs that have these properties.

implementation

The secure shell (ssh) is an implementation of these principles. The machine that you want to connect to run an ssh server.

The initial connection still has a problem. Can we be sure that we are talking really to the host that we want to talk to, rather than to a bogus host that mimics to be the desired host?

There is no method to completely circumvent this problem. Host keys are used to make sure that, after the first connection onwards we are talking to a machine that we trust. It isssues a public server key.

I want to securely login to an ssh server. I take the public key of the machine. I send the machine an encrypted message “hey, I am Thomas, my public key is blahblahblah.”

Machine then knows how to send me messages that other people can not decode. But it can not be sure that it was me who sent the message. An intruder Trudi may have done that since me public key is public.

It therefore encodes a random number, and challenges me to decode that number. Trudi attack would be foiled at this point.

putty

a free (GNU public license) ssh client.

Download and use at home.

Junk hardware

wotan.liu.edu is 148.4.16.231

Wotan is the chief god of the German legend.

It is a pentium with 300 MHz (I think). It has 250 Mega of RAM. 2 times 1.7 Giga of disk space.

But it has Linux software that does not require the resources that Mickeysoft products require.

users

- SuperUser, account name “root”
- special users
- ordinary users, such as you and me. They have a home directory **/home/***username*.

now create accounts for the students, Thomas

listing files and making directories

The command to list files is **ls**. It lists the files in the current directory, but not all of them.

ls -a lists all the files in the current directory.

ls -l makes a long listing.

ls -la makes a long listing of all the files.

ls --help | more shows more things that **ls** can do.

mkdir *directory* allows you to create the directory *directory*.

changing directories

The directory separator is **/**.

cd brings you to your home directory

cd / brings you to the top directory of the machine.

cd *directory* allows you to enter the directory *directory*.

cd ~*user* allows you to enter the home directory of user *user*.

cd .. allows you to move up one level in the directory hierachy.

Task: create a directory called **public_html**

editing files

`nano` *file destination*

edits a file using nano. The most important are displayed on the lower part of the screen.

Nano is a free version of pico, the pine composer.

Pine is not elm.

copying and removing files

`cp` *origin destination* copies file *origin* to file *destination*

`cp -r` *origin destination* copies directory *origin* to directory *destination*, including all subdirectories

`rm` *file* removes the file *file*

`rm -r` *directory* remove directory *directory* including all subdirectories, be careful

Apache

apache will server a url

`http://`*server*`/~`*user*`/`

out of the subdirectory `public_html` of the user *user* on the server machine *server*. The trailing slash is optional.

If only the directory is given in the URL—as above—it will serve the file `index.html`.

Note that this is a default configuration that can be changed in the apache configuration files, that are usually called `httpd.conf` and `srm.conf`, and usually found in the directory `/etc/apache`.