Basic User Security

I) Passwords

1) Change your pass word frequently. Use "yppasswd" to change your pass word at least once a month.

2) Choose your pass word carefully.

Avoid the following:

> i) words that are contained in any dictionary of words of any language;
>
> ii) proper nouns, including names of famous real or fictitious characters;
>
> iii) acronyms that are commonly used by computer professionals;
>
> vi) simple variations of first, last or login names, etc.

A good heuristic for choosing a password is to choose an easy-to-remember phrase, such as By The Dawn's Early Light, and to take the first letters to form a password. Throw in some punctuation or mixed case letters as well. For the above phrase, one example password might be: bt}Del{.

Here are some other examples of Good Passwords:

```
no1home!
uRnot4me
prAy4Me?
I8-thErE
```

(please do not use the control characters)

If intruders can get a password file, they will usually take it to another machine and run a password-guessing program on it. These programs involve large dictionaries searches and run quickly even on slow machines. The passwords that are "cracked" are then propagated out on the net and shared.

Your pass word will be scanned weekly. If your pass word has been cracked by our guessing program, you will receive an e-mail to urge you to change your pass word. If you do not change your pass word within a week, your account will be suspended.

You as an owner of a UNIX account have the responsibility to help keep the system and other accounts secure by making it difficult for someone to guess your password.

Advice from Peter.Schmitt@Dartmouth.EDU and shlam@ie.cuhk.hk

II) No .rhosts file in your home directory

If you really need to have the .rhosts file to trust remote hosts, do not include any "+" character in your ~/.rhosts. Either putting the "+" in the first or second field in your ~/.rhosts, your account can be easily broken by the remote hosts.

e.g.

```
+ yourname
```

Any host with a user "yourname" can rlogin your account without supplying your pass word.

```
yourhost   +
```

Any users on yourhost can "rlogin yourhost -l yourname " without supplying your pass word

The correct format of the .rhosts is

```
yourhost   yourname
```

III) Protect your data

1) File Access:

Three levels of access:
User (owner)
Group (member)
Other (not user and not group)

rwx	rwx	rwx	r - read access
user	group	other	w - write access
7	7	7	x - execute (search) access

e.g.

```
csh>   ls -lg README

-rw-r----- 1 shlam  staff   8114  Mar 25 21:20 README
```

That is the README file can be read/written by the owner (shlam) and can be read by "staff" group. Other has no access to this file.

You can use "chmod" to change your file access:

Turn off all access	chmod 0 file
Turn on just user read, write & execute	chmod 700 file chmod u+rwx file
Turn off all access to other	chmod o-rwx file

You can use "umask" to set the default file/directory permissions in your .login file.

umask 0	no restrictions (not secure)
umask 2	no write for others (minimum security)
umask 22	no write for group and others (moderate)
umask 27	no group write, no other permissions (strong)
umask 77	no permissions for group or others (strongest)

-x- On directories allows access only if exact file name is known.

e.g.

```
csh>  ls -lgd $HOME

drwx--x--x 95 shlam staff              4608 Aug 25 16:50 /data/staff/shlam
```

Group and Other can only read my home files (with chmod og+r) if they know the exact name of my files.

Never chmod your $HOME, .login, .cshrc, .profile to be 666 or 777.

IV) Do not use setuid shell scripts

If the script begins "#!/bin/sh" and a link (symbolic or otherwise) can be made to it with the name "-i", a setuid shell can be immediately obtained because the script will be invoked: "#!/bin/sh -i", ie: an interactive shell.

e.g. Suppose you have a setuid shell script (myscript) to update a logfile in your home directory.

```
#!/bin/sh

myhome=/data/student/ie90/foo

echo ‘date‘ >> $myhome/logfile
```

You set it to setuid

```
csh>  chmod u+s myscript
```

Then another user softlink your myscript with "-i".

csh> ln -s /data/student/ie90/foo/myscript "-i"

When this user run "-i", he/she can get the interactive shell with your privilege. Then he/she can read your mail and delete your files.

You can use the following command to use your setuid programs under you home directory:

csh> find $HOME -perm -4000 -print

V) Always put the current directory "." at the end of your search path

Let's say you put the current directory "." before the /bin in your search path and there is a program "ls" in /tmp. The "ls" program is this:

rm -rf $HOME ← **Never try this command !!**

Then, what happen to your directory if you

csh> cd /tmp

csh> ls

In order to avoid this kind of Trojan Horse attack, make sure you have a good path environment.

VI) Try to use Magic cookie rather than xhost command

When you run a X application at a remote host, try not to use xhost at your local host. If you login the remote host with the same account of your local host, there is no need to set xhost because the Magic cookie in your ~/.Xauthority file will give the permission to the remote host to set display.

If you "xhost +", everyone can get your display. Then an intruder can get your window and keyboard information through the X-window protocol.

If you "xhost remote_host", any user at remote_host can get your display too.

If you have an account in another dept. or CSC. The X clients running from those systems cannot access your local ~/.Xauthority file. Then you can use "xauth" program to propagate the magic cookie from one host to another. You still do not need to execute any xhost command.

For example, you are at ieug34 and you want to have a remote display from hp9000.csc.

1) extract your display magic key at ieug34

```
ieug34:csh> xauth extract myxkey $DISPLAY
```

2) binary ftp the myxkey file to hp9000.csc

3) At hp9000.csc, merge the magic key to your ~/.Xauthority

```
hp9000:csh>  /usr/bin/X11/xauth merge myxkey
```

Then you can run your X application at hp9000.csc and set the display back to ieug34.

VII) Encrypt your confidential information

You can use "des" command (Data Encryption Standard) to en/decrypt your private data. In such case, an intruder cannot steal your information even he/she has broken into your account.

SYNOPSIS des (-e | -E) | (-d | -D) [-bfs] [-k key] [input-file [output-file]]

Unlike your account pass word, we cannot decrypt your files back if you forget your key unless we feed your encrypt files to mpp.

VIII) Never leave your console unattended

When you leave your console, make sure you log out your account.

IX) Do not release your pass word to anybody

Do not release your pass word to anybody; otherwise your account will be suspended.