Modem : 3654900.

Prentice Centre Bulletin Special Edition

THE UNIVERSITY OF QUEENSLAND

Queensland 4072 Australia
Telephone (07) 365 4116
International +61 7 365 4116
Facsimile (07) 365 4477
Email help@cc.uq.edu.au

For enquiries contact:
Customer Service Counter
Room 207 Prentice Building

NUMBER 29 JUNE 1992

Protect your account from hackers!

In recent months, there has been an increase in attempts to break into computer systems, both at Prentice and at other departments. System administrators keep a close watch on such attempts and report these immediately to their Head of Department. Even so, it could be too late if your data is destroyed or confidential material accessed before the intrusion is discovered.

Whilst this bulletin is not the canonical guide to computer system security, it is intended to help you protect yourself against such attacks.

There are many reasons why people attempt to illegitimately log into computers. These reasons include (but are not restricted to):

- idle curiosity;
- attempts to "beat" or "defeat" security and system managers to gain personal satisfaction and/or peer group status;
- interest in work that is being carried out on a particular system;
- games;
- destruction of data and systems in order to satisfy personal grudges;
- more serious crimes in which computer hacking has a role to play.

A good example of this is the infamous case set out in "The Cuckoo's Egg". In 1986/87, Clifford Stoll, an American astronomer working at Lawrence Berkeley Laboratory set out to solve a mystery surrounding a 75 cent discrepancy in a particular system's accounting records. By the time this mystery had been solved, his work had directly lead to the arrest of people in Germany who had been breaking into systems in order to steal military information, for subsequent sale.[1]

Whilst this is a notable case, the fact is that computer crime, particularly in the form of hacking, occurs quite often around the world.

Closer to home, the University of Queensland has not been isolated from such incidents.

Two recent incidents (both of which occurred on the St Lucia campus in 1992) illustrate this:

(i) A general student account that was expressly available only for Second Semester 1991 was illegitimately accessed by at least one, and possibly more, people. The intruder(s) viewed confidential information. A particular student was identified as one who improperly used this account.

(ii) A person logged into a local machine from overseas and caused significant damage. This person stole information, modified accounts, password files and system software. He/she also installed games and password detection software and interrupted certain processes, which had the potential to cause up to half a million dollars worth of damage.

Why do we need security on computer systems?

As an information processing tool, computers provide a reliable and efficient way in which to gather and manipulate large stores of information. An organisation's information base is considered a valuable asset — in the commercial world, information is seen as crucial to success. Information has value, in terms of the time and money expended to gather and process it, and in terms of the opportunities and insights it provides. Clearly, such an asset must be protected in both the commercial and academic worlds.

Computers are also used to control other devices. Examples include computer driven analysis equipment in medical and other scientific fields, automated production lines, engineering research labs, and power stations. A system crash or interruption to particular processes on systems controlling such devices can (and does) lead to irreparable damage to these devices, and potentially injury and death.

Such scenarios do not simply belong to the "outside world". The University does use such systems and devices in research every day. A malicious computer attack can have serious consequences.

Within AARNet, security is host-based rather than network-based. This means that the protection of hardware, software and data is the responsibility of individual system managers and users, rather than the network provider. There are sound reasons for this.

Using networking technology there are two potential sources of attack: the computer network itself (i.e. AARNet) and the public telephone network. This means that attacks may originate from over 800,000 computer systems on the

Internet (the international community of which AARNet is part), in addition to any other systems that may be connected to the public telephone network.

On the AARNet network, it is possible to prevent all communications traffic to and from other particular sites. However, when an attack is carried out from a legitimate site, it is not possible to filter the attack from otherwise legitimate connections from the same site.

There is an old adage that "the only system that is truly secure is one which is switched off and unplugged, locked in a titanium lined safe, buried in a concrete bunker, and is surrounded by nerve gas and very highly paid armed guards. Even then, I wouldn't stake my life on it."[2]

Clearly, this is not a viable solution for most sites! The solution is therefore sensible host-based security (ie security at the individual system level).

Hacking — The Law and the Penalties.

"Hacking" (or more appropriately, "cracking") is a generic term for the practice of (illegitimately) accessing or persistently attempting to access a computer system[2]. The act of gaining entry to a computer system without authorisation is unlawful, and The University of Queensland views such behaviour seriously.

Breaking into a computer system without authorisation may lead to a range of disciplinary actions by the University and may be pursued under Commonwealth Law.

Commonwealth Law

Computer offences are listed under the Commonwealth Crimes Act 1914, Part VIA, Sections 76A to 76F inclusive.

These offences are categorised, with severities ranging from unlawful access to a computer (carrying a maximum penalty of 6 months imprisonment) to destroying, altering or inserting data into a computer (carrying a maximum penalty of 10 years imprisonment).

What Can Users Do To Protect Machines?

This section outlines what users can do to protect their software and data from attack. Although this and the following section are aimed primarily at VMS and Unix systems, the general principles are applicable to any computer system.

Passwords

Your best line of defence against attack is a secure password. A password is like a key, and any entry point that allows access by default is not secure. A bad password is like leaving your front door unlocked. A safe password is considered the primary defence mechanism.

There are several principles involved in selecting a safe password.

Simple passwords that are easy to remember are typically not safe. Examples of such passwords are:

- a word which can be associated with you (your car make or number, your child's name, etc);
- a word which someone watching could easily spot (qwertyuiop);
- a dictionary word which a hacker with a PC and an on-line dictionary could discover by exhaustive trial.

Neither is it sufficient to add a number or change O to 0 and I or L to 1 when spelling such words or to spell them backwards.

Techniques for generating secure passwords include using two or three unrelated words, always including some non-alphabetics, purposely mis-spelling, or taking the first letter from each word of a phrase.

Other basic hints for generating secure passwords are:

- Use a **minimum** (not maximum!) of 8 or more characters (system permitting).
- A password on Unix systems should be mixed case, but do **not** choose only the first letter as uppercase. (e.g. Mich37bo is not as good as MicH37Bo.)
- Include at least one digit or punctuation character. Do **not** replace o with 0 and I or L with 1. (e.g. fl0pp1mp is not as good as fL0$p*Mp.)
- They should be changed frequently, and NOT reused. Password cracking algorithms have been around for quite a while now. By using computationally intensive processes, a password can be broken in time.

Applying the techniques outlined above make the length of time required to break a password prohibitively long. However, the time required to break a password drops significantly as each letter is guessed, or other information is known about a password. A process that runs for one month may guess many passwords. Therefore, a password should be changed regularly, so that even if it is finally broken, it will be long out of date. A password should never be reused.

Do not:

- leave an account without a password.
- use the same password on multiple accounts. If you have many accounts, then do not use the same password on each account. If one is broken, then all are broken. As well, do not just change one character in the password as this may be easily spotted if one of the passwords is compromised.
- use any word from a dictionary as most forms of password attack use dictionaries as a basis for password guessing.
- use birthdays, car registration numbers, room numbers, department names, machine names, locations, wife/husband's names, pet's names, children's names and so on. These may be determined as most of this information is not confidential.
- use keyboard patterns, or duplicating characters such as qwerty or aabbccdd.

After reading all of that, you may ask "well, what is a good password? What can I use?". One technique would be to use a two or three word phrase, and replace the 1st character of the 1st word with a <shift>-1, the 2nd character of the 2nd word with a <shift>-2, etc, and uppercase every second character except punctuation. For example:

!Yc@rSm$lLs (my car smells)

Note, though, that this example should NOT be used as it is now published widely!

On VMS systems you may use any of the characters $, _, A- Z and 0-9. Passwords can be up to 32 characters in length, a minimum (usually six or eight) has been set for your account. The command to set a new password is SET PASSWORD. SET PASSWORD/GENERATE will produce a list of pronounceable non-words from which to choose a

password. VMS will not allow you to use easily guessable words or passwords you've already used. In VMS, passwords are not case sensitive.

On Unix systems the case of alphabetics is significant and most of the characters on the keyboard may be used, but passwords can only be up to eight characters long.

On VM systems such as UQVM, read, write and multi-write passwords can also be placed on individual minidisks as well the user's account.

Finally, please note that on VMS and Unix systems, not even the System Manager can tell you what your password is. It is encrypted and cannot be decrypted. All he/she can do is set-up a new password. On UQVM, we can tell you what your password is.

File Protection

On any multiuser system, every file must have an owner. On UQVM, the owner of a minidisk implicitly owns all files on that disk. On VMS and Unix systems, however, every individual file is owned by a user.

The owner of a file (whether that file is a normal text file, a program or even a directory) controls who has access to that file through the use of file protection flags.

VMS File Access

From an individual file owner's perspective, there are four types of users, viz. the owner (ie yourself), the group, the world and the system.

In the same way that the university consists of departments, which in turn are made up of individuals, the "world" (ie all users on the computer system) is made up of "groups" of users, all of whom individually own files.

"System" is a special user, and usually has a large range of privileges. However, file owners can still control access to the system account, although this may be overridden through the use of privileges possessed by the system account.

The types of access that a file owner can control are read access (ie read the file), write access (modify the file), execution (execute the file if it is executable) and delete access (erase the file).

This is best illustrated through example.

Consider a computer system with many groups of users, for example the "Lecturers" group, the "Support" group and the "Students" group. The "Support" group consists of many users; among them are Wilber and Danny.

In a particular disk's directory, there may be the following files:

File Name	Owner	Permissions	System Access	Owner Access	Group Access	World Access
File1.c	[Wilber]	(RW,RWD,R,)	rw—	rw-d	r—	—
File1.exe	[Wilber]	(RWE,RWED,RE,)	rwe-	rwed	r-e-	—
File2.pas	[Danny]	(R,RWD,RWD,R)	r—	rw-d	rw-d	r—
File2.exe	[Danny]	(RED,RWED,RWED,RE)	r-ed	rwed	rwed	r-e-

Wilber owns "File1.c" and "File1.exe". Danny owns "File2.pas" and "File2.exe". (Note: In the access matrices in the above table, "r" means read access, "w" means write access, "e" means execute (program) access, and "d" means delete (file) access).

Consider "File1.c". As the owner of the file, Wilber can read it, modify it and delete it. Since Danny is in Wilber's group, he can read the file, but do nothing else with it. (He can copy it though, since he has read access). The system account may read the file and modify it, but not delete it. Users in different groups from Danny and Wilber may not access the file at all.

Now consider "File1.exe". The same access restrictions apply, with the exception that Wilber, Danny and the system account may execute the file (assuming that it is an executable program or command procedure).

Finally, Danny and Wilber may both read, modify (e.g. through compilation of "File2.pas"), execute and delete "File2.exe". The system account may read the file, execute it and delete it. Users in other groups may read the file and execute it, but do nothing else with it.

Note that read access to files is not required to provide execution access.

In VMS, file access (or protection) is set using the SET PROTECTION command. For instance, assume Danny wanted to set the access control protection for the file "File2.exe". This would be done thus:

```
$ SET PROTECTION FILE2.EXE /PROTECTION=(S:RED,O:RWED,G:RWED,W:RE)
```

Similarly, access to a directory and it's contents may be controlled by setting file protection accordingly.

Default file protection may be controlled using the SET PROTECTION/DEFAULT command. This is similar to the normal SET PROTECTION command, except that it controls the default access for all files generated in that login session.

For example:

```
$ SHOW PROTECTION
```

The SHOW PROTECTION command displays what your present defaults are.

```
$ SET PROTECTION=(GROUP:RWED,WORLD:R)/DEFAULT
```

The SET PROTECTION/DEFAULT command in this example sets the default protection to grant unlimited access to other users in the same group and read (R) access to all users. The default protections for system and owner are not changed.

At some stage you may be required to reduce the protection on a directory or directories belonging to you. If this is necessary, then be sure that you never allow write access to either Group or World.

Unix File Access

File ownership and access in Unix is similar to that described above. The same four types of users exist. However, the system (superuser) account on a Unix machine ("root") implicitly has complete access to all files. This means that you can therefore control access to your files by the file owner (ie yourself), your group and the world (known as "others").

The types of access that a file owner can control are read access (ie read the file), write access (modify the file), and execution (execute the file if it is executable). Write access on a parent directory implies delete access for a file. Note that execution access on a directory (rather than a file) is meaningless.

Again, this is best illustrated through example.

In a particular disk's directory, there may be the following files:

File Name	Owner	Permissions	Owner Access	Group Access	World Access
File1.c	Wilber	-rw-r——	rw-	r—	—-
File1.exe	Wilber	-rwxr-x—	rwx	r-x	—-
File2.pas	Danny	-rw-rw-r—	rw-	rw-	r—
File2.exe	Danny	-rwxrwxr-x	rwx	rwx	r-x

Note that in the access matrices above, "r" means read access, "w" means write access, and "x" means execute (program) access.

To delete a file, the user (ie either the file owner, or a person in the same group, or some other user) must have the appropriate write permission on the parent directory. The permissions on the file itself are not relevant.

A user must have execute access for a binary file in order to execute it. For shell scripts, both read and execute access are required.

Read access is not required on the parent directory to execute a file. However, without read access on the directory, you must know that the file is there in order to execute it, as a lack of read access means that you cannot generate a directory listing.

Access control in Unix is set using the "chmod" command.

For instance, to set access permissions for the file "File1.exe", the following commands could be used:

```
$ chmod ugo-rwx File1.exe   {removes all access, for demo only }
$ chmod ug+rwx File1.exe    {give rwx access to owner and group }
$ chmod g-w File1.exe       {remove write access for group }
```

An alternative method allows this in one line:

```
$ chmod 750 File1.exe
```

"750" implies the following bit pattern:

Digit	Bit pattern	Access control
7	111	rwx
5	101	r-x
0	000	—

Default file access control is set using the "umask" command. This command requires a pattern similar to the chmod, except that the pattern describes the access bits to be masked out (ie the reverse of access control value).

For instance, to create files with similar access control to" File1.exe", the umask pattern would be the complement of 750, i.e. 027.

027: 000 010 111
750: 111 101 000

Electronic Mail

All electronic mail boxes are protected by passwords which are chosen by the owners of the accounts. The Prentice Centre cannot accept responsibility for data corruption, loss, or interception through breach of password. It is encumbent on the individual user to choose passwords which are difficult to guess and to guard them carefully. Any breach should be reported immediately to the Prentice Centre.

Electronic mail messages may pass through a number of computers which are not under the control of The University of Queensland. While every care is taken to protect messages from loss, corruption or interception, no responsibility is taken. Clients should treat electronic mail as an insecure medium; however, in practice it is probably no worse than any other form of communications such as postal mail, facsimile or telephone. The service is improving rapidly and users can expect a high degree of reliability and privacy.

It is technically possible for a user to spoof electronic mail messages, that is, to falsify the "From" or "Signature" on the message. In this regard it is similar to postal mail. Cases of this are very rare, but nevertheless clients should bear this in mind.

General Principles

When you login, take notice of what occurs. If the machine you are using issues messages stating when you last logged in, and if there have been any failed login attempts since the last successful login, verify that this information is correct. Ensure that the time of last login was when you were logged in, (if it is different, then clearly somebody else has used your account).

On a VMS system, this message will look like:

```
Last interactive login on Monday, 25-MAY-1992 14:41
Last non-interactive login on Monday, 25-MAY-1992 11:00
```

On a unix system, it will be like:

```
Last login: Mon May 25 15:07:16 from brolga.cc.uq.oz.
```

Ensure that these dates and times are the last times that you did log in.

Also, if the machine issues a message advising you that there have been 6 failed login attempts since you were last logged in, and you have not had a failed attempt since that time, then somebody else has tried (and failed) to login to your account. For example, you could see the following message as you log into a VMS system:

```
Last interactive login on Monday, 25-MAY-1992 15:52
Last non-interactive login on Monday, 25-MAY-1992 11:00
10 failures since last successful login
```

If you have not had 10 unsuccessful attempts to log into your account since 15:52 on May 25, then clearly somebody else has unsuccessfully attempted to log into your account.

Take notice of any changes to your data. This may be in the form or modified or deleted files, or even inserted files. Take note of the last modification date of files, and verify that this is the time that you did modify the files. If not, this may indicate that another user has seen and/or modified your files.

As you login make sure that nobody observes your password as it is being typed. Finally, do not leave your terminal unattended, where it is possible for others to illegitimately use your account. If you are going to be away from your terminal for some period of time (for instance at the end of the day, or when you go to lunch), remember to log off.

What can System Managers do to protect Machines?

Back-ups

There is no substitute for a reliable back-up methodology. Reliable back-up procedures will give you confidence that should disaster strike, an operational system can be restored.

Develop a back-up methodology, and stick to it. For instance, carry out full back-ups on Friday nights, with incremental back-ups each weekday.

Use several sets of media for Back-ups. For example, use perhaps 6 separate sets of media, and rotate them from week to week.

In addition to weekly back-up sets, have long term sets. Examples include an annual back-up set, and perhaps a half-yearly back-up set.

It is particularly important to ensure that important information is recoverable. Furthermore, store the back- up media well away from the computer systems, and preferably not on-site. The Prentice Centre offers secure, air-conditioned storage facilities.

File Protection

Ensure that appropriate access control is implemented on all sensitive and critical system files.

Remember that if a directory is world readable (it should never be world writeable), then at some point, it may well be read by people outside of this University. Ensure that general guest accounts can only access such public areas. Do not allow general guest accounts to have access to critical system files or confidential information.

Periodically ensure that access controls are as you expect. Evaluate the implications of opening or closing access to particular files.

User privileges

When you allocate privileges to users, ensure that:

- they have a need for those privileges;
- they understand what those privileges allow them to do, and the responsibilities that accompany the privileges.

If the privileges are required for a specific time, do not forget to revoke those privileges when that period has expired.

Be careful to whom you allocate privileges. DO NOT allocate privileges to general guest accounts. If a particular person will be a regular user, or a user for a specific purpose, then give him/her a separate account. If users should not be logging on via general guest accounts, then disable such accounts.

Allocate only the minimum required privileges to general user accounts.

For more privileged accounts, allocate the minimum required privileges as the default privileges at login. This forces the user to physically enable required privileges (and thus think about what is required) before carrying out any work that requires greater power.

Accounts and Logging In

Periodically audit accounts. This means that over regular intervals, carry out the following:

- Ensure that all accounts are legitimate.
- If a bogus account has been created, find out when, how, why and by whom. Try to find what files are owned by, or have been accessed by this account.
- If an account is no longer required, back up any files belonging to that account and remove it. One case of attack this year involved the use of an account that was used by a staff member who had resigned. This account had not been removed, and became the point of attack by a hacker.
- Ensure that each account (or group of accounts) has appropriate privileges and operating parameters (eg disk quota).
- If certain accounts should only be used on particular days, or at particular times of the day, then enable these accounts only for these periods. Furthermore, if an account should be used via a particular mode of access (eg local and network, but not dial-up access) then only enable access for that particular mode.

Do not wait for an odd situation to arise. Take the initiative, and keep watch. If an unexpected situation does arise however, investigate the circumstances and implications.

Become familiar with the work patterns of regular users. This will help you spot unusual situations. For instance, if you notice that a particular user logs in at 3:00 am every day, then you may not take action. However, if another user, who is normally logged in only between 9:00 am and 5:00 pm each day unexpectedly logs in at 3:00 am one morning from overseas, then this incident would require some attention.

One method to achieve this is through tracking the last login times of users.

On VMS systems, the security auditor can be used to trace a variety of login and failed login situations. The output from this auditor can be reviewed every day.

On Unix systems, such auditing is a little more diverse.

The following files are used for keeping track of logins:

/etc/utmp	— current users logged in
/usr/adm/wtmp	— login and logout information
/usr/adm/lastlog	— most recent login time for users

The commands "who" and "last" are the easiest way to access the records.

"who" will produce a list of the current users on the system giving the tty name, user name, host where user is logged in from, and time of login.

"last" gives a list of login/logout times for all users and as with "who", tty name, and host for that session. "last" will also accept a username or tty to restrict it's listing.

A useful practice is to monitor logins with "last" and check for "unusual" login times or remote login hosts.

Last login times for users are kept in /usr/adm/lastlog. These times are printed on login, typically in the form:

Last login: Mon May 25 15:07:16 from brolga.cc.uq.oz.

A good idea is to make your users aware of this message and for them to take note of it when they login. An intruder can easily give themselves away to a watchful user this way.

Accounting

System accounting is a useful tool for tracing the activity on a system. Data is typically kept in a central accounting data file, and reports based on the contents of this file can be generated as desired.

On VMS systems, the central repository of data is the system accounting file (ACCOUNTING.DAT). This file is created when the system is initialised, and is updated for each of the following event types:

- process or image termination
- system initialisation
- login failure
- print job events

See HELP ACCOUNTING for more information.

On unix systems, process accounting is kept in /usr/adm/pacct. This file records for each process when it completes: command name (no arguments are kept), user who executed it, time it started, tty associated with it and so on. For complete details, man acct(5). "lastcomm" gives a summary of all the information kept.

Note: You may need to enable process accounting as a kernel option (on SunOS this is SYSACCT) to have it available, plus possibly starting system accounting from an initialisation file (/etc/rc*).

Passwords

On Unix systems, consider the regular use of password cracking tools, packages (such as COPS) which test other aspects of systems integrity (e.g. code running under superuser privileges) and suspicious access controls. Consider also the use of password shadowing, which when used in conjunction with password cracking tools, should provide a good degree of password file robustness. Information on such packages is available from Prentice.

More recent versions of VMS include password oriented features such as history files (to prevent re-use of passwords) and enforced password choosing methods (to prevent easily guessable passwords being used).

All systems should regularly expire passwords (for instance, every 3 months, or every 6 months). Passwords for privileged accounts should be changed very regularly, and be shared only among the smallest group of people possible. Individual passwords with no sharing is the ideal.

Firewalls and Network Protection

It may also be possible to construct a "firewall" for your local network. That is, to use a router to filter traffic so that only specific machines can gain access to your particular local network. If your local network requires a high level of security, and only allows incoming connections from a small number of machines from outside of the network, this could be a solution worthy of investigation.

Other precautions

- Proxy accounts are default accounts which remote machines can use to carry out operations such as file transfer without specifying a username-password combination for your machine. The idea behind this is to allow transfers without broadcasting passwords over the network. There is a tradeoff though, in that it can open you machine up to attack by allowing non-authorised people to transfer potentially harmful files to your machine, without authentication. Be very careful if you allow such accounts on your machines.
- Ensure that objects and daemons have been started under the correct account name.
- Use a security auditor. For instance, on a VMS system enter the command SHOW AUDIT/ALL to see the level of auditing that takes place on your system. This feature audits not only failed access attempts, but a range of events. Auditing can be used to track unsuccessful logins, breakin attempts, the use of the SET AUDIT command and changes to the authorization, rights list and network proxy databases. Inspect audit files daily to determine whether your machine has been attacked, and if so, to what extent.
- If you have a closed VMS system (i.e. a small number of users, all of whom are known to each other), you may consider establishing a system-wide password. This is where users must enter a general password before gaining access to the login prompt. Once at the login prompt, login procedures are the same as any other standard VMS system.

Acknowledgements

1 "The Cuckoo's Egg", Clifford Stoll, The Bodley Head, 1989. ISBN 0 370 31433 6

2 "Almost Everything You Ever Wanted To Know About Security*, *(but were afraid to ask!)", USEnet news Alt.security Frequently Asked Questions, Edited by Alec Muffet.