Security: Bolting the door...

OPEN ALL HOURS

No computer is ever completely secure – there's always risk. The type and extent of that risk can vary. Colin Murphy investigates the dangers

The view from the inside

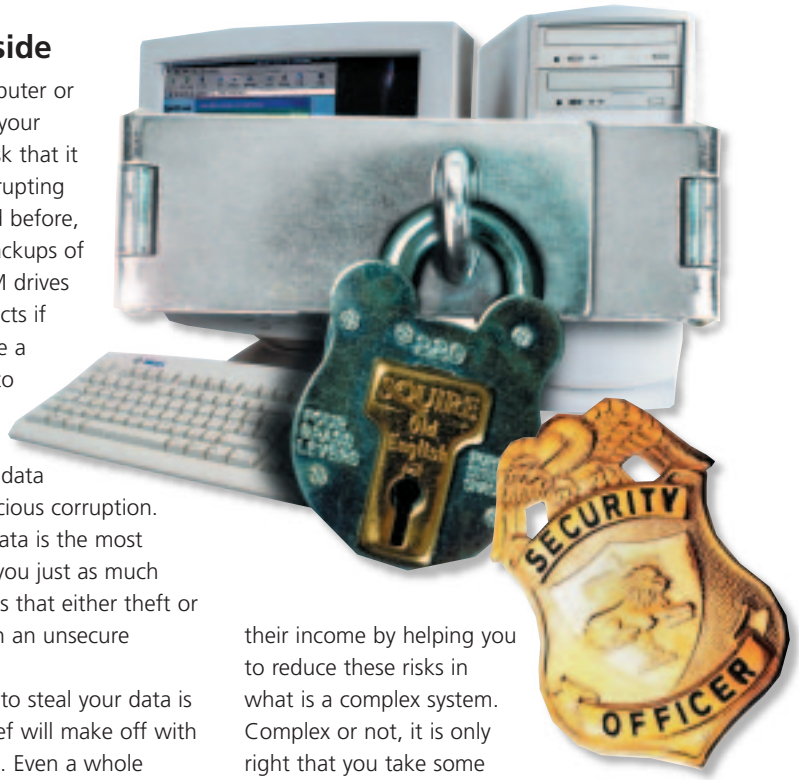
What is more valuable, your computer or your data? No matter how good your hardware is there is always the risk that it will stop working, failing and corrupting your data as it goes. It's been said before, and it will be said again, make backups of your data: Tape drives and WORM drives and buy hardware support contracts if need be. It's not too hard to make a backup of your /home/ directory to a CD writer either.

Along with the household insurance of a data backup, your data is also at risk from theft and malicious corruption. You may think that the theft of data is the most obvious but corruption could do you just as much damage. What is more worrying is that either theft or corruption could easily happen on an unsecure machine.

The simplest way for someone to steal your data is to physically take it. A chance thief will make off with a backup tape or removable drive. Even a whole machine – especially if it's a nice shiny notebook – is very attractive to a passing light-fingered Fred. We can all remember the MI5 worker who lost his laptop in a Tapas bar this summer. The only answer is lock and key and physical security. Following all the best practices and procedures, such as restricting access and bolting the casing down, we are still left with the potential of cyber crime over the network as so loved by fiction writers and **hacker/cracker** wannabes.

With the growth just starting to appear of broadband access in the home, be it ISDN, ASDL or Cable modems, it is no longer just a corporate network administrator's worry. More Small Office/Home Office users will also be tempted to take on these new forms of Internet connection. We all have to be wary of the potential threats.

If your computer is connected to the outside world then there are always risks. Lots of companies make their income by helping you to reduce these risks in what is a complex system. Complex or not, it is only right that you take some precautions yourself, which will also give you the chance to discover more about your system.



Keeping up with the Jones'

When you install a new distribution on a computer it is reasonable to assume it is almost up to date. There is always a delay between the final collection of packages, QA testing, manufacture, distribution and finally sale before you get hold of it. So it's wise to check the Web site for security updates as you install it. This is where the major distributions gain an advantage. They have invested in their upgrade networks fix, and some can make this seem almost automatic. Red Hat uses its Red Hat Network, Mandrake use its local client MandrakeUpdate to look for updates and bugfixes, and SuSE have its security announcements in its support database. All distributions worth their salt, or your money, will also

HACKER/CRACKER The term hacker is always used out of context. Anything good can be a hack: fixing a piece of code to make your company more profitable or producing a meal from leftovers are all good hacks, and you could be proud to be called a hacker. Unfortunately the term is abused by the press and is usually taken to mean Cracker – someone who breaks into computers. Even the term cracker can be subdivided into someone who breaks in for the mental challenge and those whose intent is malicious.

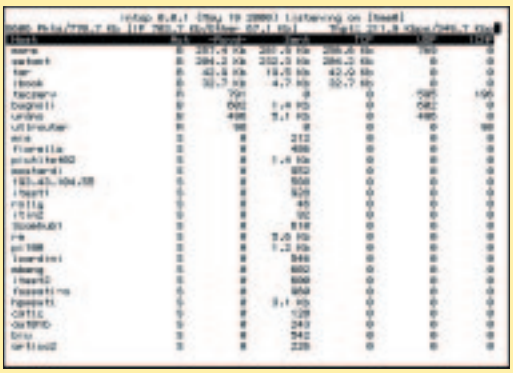
ntop

The first tool to look at is ntop. This program shows the network usage through a simple Web interface. This is a Unix tool that shows the network usage, similar to what the popular top Unix command does for processes.

ntop contains a powerful and flexible interface to the ntop packet sniffer. Since ntop has grown so much in functionality and it cannot be simply considered a network browser, the problem of capturing and showing network usage has been split. The ntop engine captures packets, performs traffic analysis and information storage.

ntop must be run as root, or at least with root permissions. This is best achieved by a user with normal permissions logging into a superuser mode with the su command. This will mean the user will need to have access to the root account, but this shouldn't be a problem because only system administrators should be playing with this anyway.

ntop viewed through a browser



ntop running in a terminal

This is a far more secure method of access than setting the SUID bit on the executable, which would enable anyone to run the program – systems administrator or not.

If ntop is started in a terminal you will be shown a display of network processes, almost like with top. Here you will see some basic information about what data is being sent where and how much.

ntop can also present you with much more information via a Web browser. Start ntop with

```
ntop -i eth0 -w 888
```

and point your browser to *http:///localhost:888*.

With ntop you will be able to sort through the network traffic by protocol and various criteria, look at network statistics, show the IP traffic and sort that by source or destination and much more. This will enable you to get a feel for the movement of data through your network.

run mailing lists to advise you of any security issues. As new development is done, patches are released for your packages and some of these will have security implications.

There is still the risk that a weakness has been identified and that the developers for your distribution have still to find out about it, which means you are your first line of defence.

Ring of fire

Our first port of call is with the **firewall**, a fundamental protection from network intruders. Like a condom, it gives you a sense of security, unlike a condom it will deny someone the penetration. It enables you to control what types of **services** you are happy for your machine to handle – not all are as secure as you would want.

Most distributions will let you set up a personal firewall and configure the type of access you require, usually from their graphical configuration tools. Often a firewall will have been set up during the initial installation, a couple of questions about whether or not you are running Apache or a Web server. If you are not running those services then the **ports** to them will be closed off.

This can range from allowing anything to connect – useful maybe, if you are running a self contained network that never has any access to the outside world and is only connected to another computer in the spare room, which you want to pass files around with ease – all the way to blocking all incoming and outgoing access without direct intervention.

It is at this point we must consider just what is running on our systems. As Linux boxes are used to acting as both server and client at the same time we must be careful about access rights. We also tend to have helpful system services (daemons) running in the background. The daemons wait and when required enable connections via ports but unfortunately this gives huge access holes in the system. You can use

Firewall A firewall is nothing more than a piece of software that enables information to pass through it according to simple rules. The rules can be changed and so care must be taken to ensure everything is double-checked. Firewalls can form part of a workstation machine, but if your network is any more complicated it is often useful to have a dedicated firewall machine.

Services Data coming into your system needs to be handled by the correct type of software. It would be pointless for your email client to be looking at the data coming in from a time server – it would be meaningless. These types of data coming into your machine are broken up into services.

Ports Ports are the means by which your computer knows how to handle services. There is a defined list of ports together with their associated services in the file /etc/services.

the process status utility:

```
ps -aux |less
```

to list which are running, but they will only be running if something has tried to open the port associated with that service. Another utility, netstat, will show you what is listening in your box:

```
netstat -l
```

some of these may not even be wanted, others may be an outright security risk, telenet and finger are just two that spring to mind.

If you are running Red Hat then you can use the chkconfig -list while SuSE users can use YaST and as root System Administration/ Change Configuration/ Services started at boot. If you want a more hands-on approach you can configure access manually. The configuration takes place in one of two places depending on which distribution you are using, inetd.conf or xinetd.conf:

- /etc/inetd.config will contain a list of services and their ports. You can switch off these services by commenting out – putting a '#' in front of the line – lines that concern you.
- /etc/xinetd.conf has a more complex configuration file structure. You still just have to comment out the service lines that you don't need.

The above services are not running until some call has been made to the associated port. inetd or its eXtended daemon cousin will spot this call and then start the required service if it can see it in its configuration file. Commenting out lines like this makes them invisible to the daemon, but they are much easier to reinstate than if we had just deleted the line completely. If you have configured by hand you will need to restart inetd by using the command:

```
killall -HUP inetd
```

Good daemons to remove are the r* services such as rshd or rlogind as well as those just not used like daytime. Fingerd is also worth considering removing as it gives out a lot of information to potential intruders.

From the outside looking in.

Now we have removed some daemons we can add useful software. Not only should you make sure that your distribution is up to date with security packages, but that any other third party is also up to date. Browse the Web and make sure you have the latest servers that you want to run such as Apache.

The system is now more secure but all is still not well. Very often, passwords are still sent in a plain text format. When you set up a connection across a network it is possible for someone to use a password-sniffing tool to listen in to whatever you type. To overcome this we can use ssh. The sshd is the daemon that replaces the r* services we removed earlier and adds encryption to all the communications.

For ssh you will need the following packages:

- Openssh-.rpm
- Openssh-server-.rpm
- Openssh-clients-.rpm
- Openssl-.rpm

To connect with ssh use the following command:

```
ssh -1 usename target.computer.com
```

Copy files by using:

```
scp /where/the/file/is.txt
/where/you/want/it/to/go
```

Nmap

The next program is Nmap, which enables you to run a port scan yourself. This is most useful to highlight what parts of your system are insecure.

A portscan is what a cracker will use to find weaknesses in your system, by sending a stream of data to a range of ports and waiting to see if any of them reply. If they do, then they are prone to attack. Portscanning is a sign of attack, so you should not use this tool against networks that are not under your control. Should you, you are likely to find yourself barred from accessing wherever you scanned and a complaint being sent to your ISP, which could mean them withdrawing their service from you.

NMap shows the list of open ports on a local machine

Nmap (Network Mapper) is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts, which is where you'll probably use it. Nmap uses raw IP packets in novel ways to determine which hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap runs on most types of computers and both console and graphical versions are available.

With Nmap you also get XNmap, enabling you all of the probing, but from the comfort of a graphical user interface.

The primary goals of the Nmap project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. The NMap project also boasts a wealth of tutorials and help files to make sure you get the best out of its powerful features.

Tripwire

Prevention of a crack attack is important and you must treat it with utmost priority. But how will you know that you have succeeded, or more importantly, failed? Should someone sneak past your best efforts and manage to make themselves at home amongst your precious data and computer resources, it's important to know that your defences have been breached.

One of the biggest concerns of a breach is knowing that your data is still accurate and hasn't been tampered with. This is where Tripwire can help. Tripwire is a tool that checks to see what has changed on your system. The program monitors key attributes of files that should not change, including binary signature, size, expected change of size, etc. The hardest part of doing this is balancing security, maintenance, and functionality.

Tripwire maintains database details of all of the files that you have configured for and compares these details against what is really in your directories. Should anything be different then warning bells will sound, or, at least a log file will be written. The important thing is that you know there has been a breach, and you know that you cannot fully rely on all of your data.

The home of Tripwire

Info

Ntop
www.ntop.org
Nmap
www.insecure.org/nmap/index.html
TripWire
www.tripwire.org
PortSentry
www.psionic.com/abacus/portsentry

Portsentry

Portsentry is the most powerful tool we will mention here, running on TCP and UDP sockets to detect port scans against your system. PortSentry is configurable to run on multiple sockets at the same time so you only need to start one copy to cover dozens of tripwired services.

PortSentry will react to a port scan attempt by blocking the host in real time. This is done through a range of configured options. The most useful is when PortSentry drops an illegal packet. Because the packet is dropped and forgotten about, no acknowledgement is received by the cracker sending the packet, who therefore doesn't know they have hit anything and so remain none the wiser regarding you and your machine. PortSentry also takes full advantage of either dropping the local route back to the attacker, using the Linux ipfwadm/ipchains command, and/or dropping the attacker host IP into a TCP Wrappers hosts.deny file automatically, which will further strengthen your system.

PortSentry will detect SYN/half-open, FIN, NULL, X-MAS and oddball packet stealth scans. These are much more obscure form of port scanning and are not usually used by the average script-kiddie. Once a scan is detected your system will turn into a blackhole and disappear from the attacker. This feature stops most attacks cold.

PortSentry has an internal state engine to remember hosts that connected previously. This allows the setting of a trigger value to prevent false alarms and detect "random" port probing, which can happen as part of regular Internet life.

PortSentry in action

PortSentry will report all violations to the local or remote syslog daemons indicating the system name, time of attack, attacking host IP and the TCP or UDP port a connection attempt was made to. When used in conjunction with Logcheck it will provide an alert to administrators via email. Here is your last line of defence, you must regularly check for discrepancies. If you have yet to set up something like Logcheck then you must go through the system log files on a regular basis to make sure that everything is still in order. Without this final effort, the whole exercise is worthless.