Linux Security

— a brief and partial list of issues and tools —

Athens Linux Festival, February 23, 2001
Ed L. Cashin `<ecashin@users.sourceforge.net>`

issue	**description**	**plan**	**solutions**
shell access	Traditional means of accessing a Linux host over the network allow third parties to "sniff" your password as you log in. New technologies use encryption to prevent sniffers from evesdropping on you when you need shell access to a remote host.	encrypt networking	`ssh`, `secure telnet`
inetd	A daemon called inetd launches servers that respond to many popular network requests, e.g., telnet, ftp, finger. Unfortunately, many of these services are easy to attack, especially the older ones that don't use encryption. A new replacement for inetd, *xinetd*, is gaining popularity and gives you more control over who accesses what services.	control access	`TCP_wrappers`, `edit inetd.conf`, `xinetd`
mystery services	A common mistake of a new Linux user is to leave many unecessary and unfamiliar services running. Attackers can often crack such machine, since there are many ways in. By turning off unused services, you eliminate openings in your system's security.	turn off unused services	`chkconfig`
logs	Things happen all the time on a running machine. Knowing what's going on is often critical to making sure a machine doesn't get cracked. UN*X systems feature a way for computer processes to talk to humans: system log files. Often there's so much information in the logs that its cumbersome for a human to browse them. Tools like *logcheck* find important information in your logs.	read logs selectively	`logcheck`
root's mail	As the UGA Workstation Support Group suggests in their "RedHat Linux Security Checklist" (http://www.uga.edu/ucns/wsg-/security/linuxchecklist.html) points out, a human being should receive all the mail that goes to *root*, the system superuser. By creating an alias like "root:ecashin" in the system's mail aliases database, */etc/aliases*, you can make sure you get all of root's mail.	root alias	`edit /etc/aliases`
time	In order for your logs to be useful, it helps to have the correct time on your machine. It's so important, in fact, that there's a special	synchronize time	`xntpd`

	Network Time Protocol, NTP, for synchronizing your machine to the correct time. Modern deamons that speak NTP can keep track of how far the machine's clock drifts from the real time, lowering the number of times they must poll (ask) the time servers for the current time.		
messed-with files	If an attacker does break into your system, odds are they can keep you in the dark about the compromise by replacing the tools you trust, like *ls* and *find*, so that you can't see the porn server or sniffer that they've installed on your machine. File integrity verification systems can help. *Tripwire* is perhaps the most famous, but I prefer *integrit* (disclaimer: I wrote integrit.)	verify system files	`integrit`, `tripwire`
buggy software	Software often contains bugs, and attackers will be quick to exploit them to gain unauthorized access to computer resources. If you are quick to install software upgrades, especially for security-related issues, you are less likely to have to re-install after losing all your data and making your peers mad at you after the intruders use your machine to break into your peers' machines.	update software regularly	`updateme`, `up2date`
crackable passwords	An old way of gaining unauthorized access to a UN*X machine is to "crack" the passwords stored in the system's */etc/passwd* file. An attacker could copy the /etc/passwd file to a different machine and run special software that tries to guess the passwords, encrypting the guesses and comparing the encrypted passwords to those in the stolen copy of the /etc/passwd file. Choosing good passwords like, "BM,sagm!" ("Bob Marley, such a good man!"), will help. A new scheme for system password storage, *shadow passwords*, also keeps would-be attackers from reading the encrypted passwords.	hide good passwords	`shadow passwords`
sendmail	The *sendmail* program is ubiquitous in the UN*X world. It's a very old, complex program that has a long history of security problems. Maybe the problems are all solved. More likely, sendmail is so big and complex that many security holes are waiting to be discovered. Not running sendmail as a daemon is a great idea for anyone who doesn't need to receive incoming email over the network on the host in question. Alternative mail software may be worth looking into. The excellent programmer and cryptographer, Dan Bernstein, has offered $500 – $1000 to anyone who can find a security hole in his *qmail* program, but no one has found a hole, and it's been years.	sendmail only if necessary	`qmail`
vulnerability to the internet	Being on the open internet is rough. You can often protect a small group of machines (or even just one machine) by connecting them	control network traffic	`firewalling`, `ipchains`

together with a small network and putting a *firewall* between the private network and the big, bad internet.
A firewall is conceptually a router that connects one network to another *selectively*. That is, you can control what networking information passes between the internet and your private network.

Resources:

1. UGA WSG RedHat Linux Checklist

   http://www.uga.edu/ucns/wsg/security/linuxchecklist.html

2. Linux Security HOWTO

   http://www.linuxdoc.org/HOWTO/Security-HOWTO.html

3. Google Search Engine

   http://www.google.com/
   http://groups.google.com/

A few things to look into that I haven't covered: Kerberos; SSL; IPSEC; BIOS and hardware-based security.