

FOCUS on Linux: Security Tools

Jonathan Day 2000-02-07

1. Introduction
2. Scanners
3. Firewalls
4. Portscan Detectors
5. Honeypots
6. Authentication
7. Access Control
8. Encryption
9. Virtual Private Networks
10. Developer Libraries

1. Introduction - Know Thine Enemy

There are many security tools for Linux, each good at a specific task or class of tasks. However, there is no universal panacea, which will relieve a system administrator of all worries. The first task, then, is for an administrator to sit down and decide what kind of attacks it is that they wish to be secure against. From there, it is possible to select a tool, or set of tools, which are appropriate for dealing with that specific threat.

Threats can broadly be divided into two categories - electronic and physical. Electronic attacks are those which try to breach security by operating within the environment. This includes computer crackers, viruses, trojans, worms and malicious users. These, in turn, can be crudely divided into two other categories - those that try to attack from the outside and those which attack from the inside.

Physical attacks will usually involve physical removal of the hard disk(s) and/or backup tapes. It's not so common, because of the higher risks involved, but it does happen, and there is security software for Linux that is designed specifically for this kind of threat.

By qualifying the risks, it is possible to take appropriate measures to minimize those risks. For example, if a computer has no connection to the outside world, but is in a high-risk area of being stolen, it makes very little sense to install a firewall on it. However, encrypting the files and/or partitions would at least prevent anyone who wasn't supposed to from accessing the information on the computer. On the other hand, if a server is

locked in a sealed vault, deep underground, where the only access is via an Ethernet cable, there probably isn't much point in using an encrypted file system, or blocking access to the floppy drive. IPSec and a firewall become much more attractive.

2. Scanners - The First Line of Defense

There are a number of security scanners out there, which will probe your computer for open ports and known security holes. These form the first line of defense for a secure system. (Firewalls often get the title, but unless they have been scanned, themselves, the administrator should not trust that they are secure.)

The more common security scanners for Linux are SATAN, Saint, SARA, VeteScan, Nessus, Bass, NMap, IPPS, NSAT, Portscanner, Network Superscanner, CGI Port Scanner, CGI Sonar, strobe-classb and PortZilla.

<http://www.porcupine.org/satan/>

SATAN is very long in the tooth, and no longer actively maintained. However, it is still a good scanner to have around, as many of the more recent scanners are tuned to more recent holes, leaving older (and better-known) flaws undetected. It will run on older Linux distributions without difficulty, but may have problems on newer Linux distributions. It uses a web interface, which potentially allows you to control the program from any machine on the network.

<http://www.wwdsi.com/saint/>

Saint is derived from SATAN, and serves much the same function. It's extremely good at notifying the administrator of what security holes exist, and gives different levels of warning, according to the nature of the problem. It's development seems slow, though, and newer flaws are not included quickly, so whilst it's an excellent scanner to use, it should not be the only one. As with SATAN, it also uses a web interface, and can be used from anywhere on the network.

<http://home.arc.com/sara/index.html>

SARA is a scanner derived from both SATAN and Saint. It interfaces with 3rd-party products, such as NMAP, to provide commonly needed resources. SARA Pro extends this capability, and adds a report writer for producing configurable reports.

<http://www.self-evident.com/splotts.html>

VeteScan is a bulk scanner. It's very good at scanning a large network, very quickly, but

isn't so great at finding all the small holes. This is an inevitable trade-off. For the same period of time, you can either check a large number of systems not so thoroughly, or a small number of systems in great detail. There's a time and a place for each type of scan. For large corporate networks behind a firewall, it's probably pointless conducting detailed scans of every machine on the network. If the firewall does its job, crackers shouldn't get that far, and if they did, most machines shouldn't have any services they can exploit. Bulk scanners of this kind are very useful for this kind of scan, because it's more important to check every machine than to find every minute potential flaw.

<http://www.nessus.org/>

Nessus is a very powerful scanner, of the same ilk as SATAN and Saint. However, it's much easier to add checks to it, for potential vulnerabilities. It's also very actively maintained, and has a sizeable collection of both old and recent security holes. It works with NMap, for identifying active ports. This is optional, but recommended by the authors. There are clients for X11, Windows and Java, though the main server is specific to Unix. Connection is via a public/private key, and password protected.

<http://www.securityfocus.com/tools/394>

Bass is the scanner used in the Internet Auditing Project. It's very fast, but also very limited for that reason, being only able to detect 18 common vulnerabilities. It's good for light scans, or routine checks, of very large networks, but not really much beyond. It's not actively maintained, and contains no recent security checks.

<http://www.insecure.org/nmap/index.html>

NMap is a very powerful tool, designed for scanning which ports are active on a given machine. It's capable of either aggressive or stealthy scans, depending on need, and can launch "decoys" to try and fool Intruder Detection Systems (IDS). It's sometimes used as the port scanner of choice by other scanners (such as Nessus). It's also very useful for testing the responsiveness of IDS software. NMap isn't the latest port scanner around, but it's one of the most widely used. If intrusion detection software fails to identify an NMap scan, or its source, there could be a serious vulnerability that a cracker could exploit.

<http://www.lmn.pub.ro/~bruno/ipps/>

IPPS is a very basic port scanner. Its main attribute is that it can scan within a user-defined range of ports, rather than trying everything it can think of. This makes it very fast, when you know you only want to check a narrow range of ports. It's not actively maintained and performs no security checks.

<http://mixter.void.ru/progs.html>

NSAT is a powerful security scanner. As with Nessus, custom security checks can be written for it. It can only be run on a Linux or Unix platform. There is no ability to control this scanner from a remote computer. Having said that, it does have a very comprehensive list of security checks it can make.

<http://www.ameth.org/~veilleux/portscan.html>

Portscanner is an extremely basic, minimal portscanner. The user can control the degree of information returned and it's designed to interoperate with other packages. It's designed with finding ports in mind, rather than doing so in a way that cannot be readily detected. As such, it's useful for seeing what services are up and running on the network, but isn't that valuable a tool for network testing.

http://members.tripod.de/linux_progz/

Network Superscanner is a very grandiose name for what is essentially a very basic portscanner. It's not actively maintained, but it does have a nice X11 interface.

<http://www.novia.net/~muesu/PORTSCAN/>

CGI Port Scanner is a nice web-based portscanner, which can additionally do traceroutes, pings and whois lookups. It's not actively maintained, however, and therefore might not be suitable for all uses.

<http://www.securityfocus.com/tools/1211>

CGI Sonar is another web-based portscanner, which can detect a large number of common vulnerabilities. It's actively maintained, which is useful, but there is minimal documentation and it's a very basic system.

<http://www.luyer.net/software/strobe-classb/>

strobe-classb is a compact, high-speed port-scanner for large networks. As a result, it's fairly basic and was originally designed just to check for open mail relays. It's not maintained, and was designed for the 2.0 series of Linux kernels.

<http://www.altern.org/vih/pzilla/>

PortZilla is a very basic portscanner with optional GUI interface for Windows or X11. It's temporarily unmaintained, due to a hard disk crash, according to the author.

<http://www.securityfocus.com/tools/1228>

Messala is a fairly advanced vulnerability scanner, looking for a wide range of known

vulnerabilities. It is maintained, which gives it an edge over some of the other scanners.

<http://www.securityfocus.com/tools/200>

Mns is a powerful port scanner, with limited vulnerability checking and stealth capability. It is very nice for scanning large networks. Links to its homepage are broken, so its status is not determinable. It can be obtained from a wide range of security sites, though.

Host-Based Scanners

<http://www.fish.com/cops/>

<http://home.arc.com/tara/index.html>

COPS, TARA and tiger are host scanners. They scan the computer for known security holes and probable weaknesses. Scans include null passwords, world-writable files and directories, misconfigured servers, etc.

<http://ciac.llnl.gov/cstc/spi/spinet.html>

SPI-NET is a host scanner, in much the same way as COPS, above. It also detects file system changes and strong password testing. It is only available to Government employees or contractors.

Windows-Specific Scanners

It may be possible to run Windows security scanners under WINE. Research into this avenue is ongoing. Any results, though, are going to be specific to versions of WINE, owing to the high degree of change between each version.

3. Firewalls - the Outer Moat

Firewalls offer the outermost layer of protection for a network, providing a basic barrier and restricting points of access, much as a moat around a castle did in medieval days. As with moats, their usefulness is highly dependent on how carefully they are configured.

Software for configuring Linux Firewalls includes Instant Firewall, ipchains-firewall, Juniper Firewall Toolkit, KFirewall, PHP Firewall Generator, PMFirewall, TIS Internet Firewall Toolkit, gfcc, gShield

<http://www.jasmine.org.uk/~simon/bookshelf/papers/instant-firewall/instant-firewall.html>

Instant Firewall is a basic firewall script that configures a 3-way firewall, using Linux 2.2's ipchains system. Its author describes it as a 'quick hack'. It does not appear to be

maintained.

<http://ipchains.nerdherd.org/>

ipchains-firewall is a very sophisticated firewall system, allowing the administrator to set up masquerading, logging, external port blocking, control of individual services, and firewalling LAN-LAN or dial-up networks.

<http://www.obtuse.com/juniper/>

Juniper Firewall Toolkit is a fairly sophisticated firewall package. It has no interface, but uses a configuration file for everything. There is no editor for the configuration file, but the format is fairly basic. It is up to the administrator to ensure the configuration file is sound, however.

<http://megaman.ypsilonia.net/kfirewall/>

KFirewall is a very basic X11 interface (using KDE) for editing ipchains rules, toggling masquerading and displaying the status of the firewall. It is not under active development, at present.

<http://www.babel.com.au/phpfwgen/>

PHP Firewall Generator is a web-based IPChains editor, built around a PHP script. It's very primitive, requiring the administrator to enter the static router table, as well as the rules, but not supporting any editing of the protocols, services or interfaces available. The administrator can also display the open ports available, but has no means of adding or removing any, by means of the open ports table.

<http://www.pointman.org/PMFirewall/>

PMFirewall is a simple script for setting up an IPChains-based firewall, and is actively maintained. Its chief limitation is that masquerading is decided at install-time. Otherwise, it supports the same things as ipchains-firewall.

<http://www.tis.com/research/software/>

TIS Internet Firewall Toolkit is a very comprehensive set of utilities for setting up a firewall under a Unix platform. Unlike many other firewall toolkits that run under Linux, this is not specific to the Linux system. There are a number of proxies provided, for common services, such as FTP, HTTP, NNTP, etc. This program does have the ability to test the configuration. There isn't any capability for detecting attacks on the firewall, however. This software is "Source Visible" rather than Open Source, and needs a license for commercial use.

<http://icarus.autostock.co.kr/>

gfcc is a graphical IPChains editor, using X11 and the Gtk toolkit. It's fairly basic, in that there are no special capabilities, but it does provide a simple means of constructing a firewall. This program is not under active development.

<http://linuxmafia.org/~godot/>

gShield is a very flexible firewall package, using IPChains. Its default is to block everything except 'auth', and has very flexible masquerading support. It's configured via a BSD-style configuration file, rather than a curses-based or graphical console. This program is under active development.

<http://www.jukie.net/~bart/gfirewall/>

Fire Gnome is a potentially powerful firewall package that has a graphical X11/Gnome console. Its main strength is that it allows for very fine control of the rules. Fire Gnome is an early release and is under active development.

Windows Firewall Software

Whilst it may be possible to run Windows Firewall software under WINE, there would not be any means for the program to take advantage of any of the Linux IPChains or masquerading drivers. As a result, the usefulness of such software under Linux is questionable. However, this avenue should not be ignored, in the event that there are Windows applications that support required features that do not exist in other packages. Sometimes capability is more important than efficiency and, under those circumstances, a Windows package may prove to be exactly what is required.

4. **Portscan Detectors - Or how to find them before they find you**

There are a number of packages which are specifically designed to look for portscans, and either log the attack and/or retaliate in some way, such as blocking all access from the host identified as the culprit. Back to the medieval analogy, this is the same as having lookouts watching for scouts.

Some packages for defending Linux against portscans include Dragon IDS, Psionic PortSentry, Secure Net Pro, NID, ASAX, Snort, Shadow and AAFID2.

<http://www.network-defense.com/>

Dragon IDS is a powerful distributed system for monitoring networks, detecting intruders

and taking counter-measures. Its main drawback is the company's stringent secrecy, with regards its IDS package, which may deter people who might otherwise be interested.

<http://www.psonic.com/abacus/port Sentry/>

PortSentry monitors for port scans, and logs all suspicious packets in an internal database. Once the number of suspicious packets exceeds the safety threshold, the program will automatically block the site. PortSentry will also detect "stealth" portscans. PortSentry is susceptible to Denial of Service attacks, under certain conditions.

<http://www.mimestar.com/html/products.htm>

Secure Net Pro is a fairly heavyweight distributed package, capable of monitoring any subnet for attacks. Attacks are determined by context and/or content, and are responded to by notifying the administrator and/or terminating the connection.

<http://ciac.llnl.gov/cstc/nid/nid.html>

NID is a very basic network intrusion package. It detects activity from untrusted or unknown hosts and also detects known attacks, such as SYN floods. These are logged for later viewing or replaying. The administrator is optionally informed. NID is free to any Government employee or contractor, but not to the general public.

<http://www.info.fundp.ac.be/~cri/DOCS/asax.html>

ASAX is a very generic system, which handles non-specific, distributed data streams. This system can be used for network intrusion, but is not restricted to this role, and can be used in any situation in which an abnormal data stream can be defined.

<http://www.clark.net/~roesch/security.html>

Snort is a simple, lightweight package which monitors for specific, known types of attack and/or for specific keywords entered by the administrator. It does not take any action against the attacker, preferring to alert the administrator and file a report in both the snort and system logs.

<http://www.nswc.navy.mil/ISSEC/CID/>

Shadow is a distributed intrusion detection system, designed to maximize detection of intruders and minimize the effectiveness of any counter-measures intruders might make.

<http://www.cerias.purdue.edu/coast/projects/autonomous-agents.html>

AAFID2 is a distributed, agent-based intrusion and misuse detection system, written in Perl. It detects attempts to breach a computer's security, as well as attempts by authorized users to misuse the computer's resources.

5. **Butterfly Eyes - Giving the Enemy what they think they want to see...**

Several species of butterfly have developed "eyes" on their wings. This fools predators into thinking it's looking in a direction it isn't, and gives them a "target" that isn't really there. (Big eyes mean a big body behind it.)

Some forms of protection for computers follow the same principle - giving the illusion of common vulnerabilities, appearing to have a port active when it isn't, or even pretending to be an entire network, just waiting to be portscanned - none of it real. Since any activity on these "non-existent" ports or networks has to be from an intruder, it becomes trivial to identify when an attack is taking place, and much easier to identify which packets are from the intruder and which are innocent.

Some packages for protecting Linux systems this way include the Deception Toolkit, FakeBO and Netbusd.

<http://www.all.net/dtk/dtk.html>

Deception Toolkit mimics a number of well-known security holes, in the hope of ensnaring an unwary cracker. The cracker is given plausible responses to their activity, which is then logged, allowing the administrator to take appropriate action.

<http://cvs.linux.hr/fakebo/>

FakeBO mimics trojan servers, such as BackOrifice and NetBus. Any attempt to access or use these "services" is logged, and plausible replies are given back to the client.

<http://services.afternet.org/netbusd/index.html>

Netbusd is a simple tool for mimicking NetBus servers. All NetBus requests are logged.

6. **Authentication - Getting onto the system**

The best firewall in the world can be bypassed, given enough time. However, as with the medieval example above, the first line of defense doesn't have to be the last.

Authentication of connections and/or users is also very important, and many packages exist with this in mind. In terms of the symbolism being used, this is similar to castle walls, and a gatekeeper at the front entrance. Authentication falls into two basic categories - that of authenticating the user or the machine at the other end. Depending

on how comprehensive a security system is required, one or both of these can be deployed.

Software for Machine Authentication includes FreeS/WAN, NIST IPsec and ENSkip. These programs guarantee, through the use of encryption keys, that the computers at each end are who they say they are, and that any packets claiming to be from a given computer is, indeed, from that machine. They also prevent interception and/or modification to packets, in transit.

<http://www.freeswan.org/>

FreeS/WAN is a powerful IPsec implementation. It supports configurations for two subnets (denoted as "left" and "right"), and strong encryption, including 3DES. Machines are identified by means of shared secrets. Patches to support X.509 certificates are also available. FreeS/WAN uses tunnels between nodes, which restricts it to ferrying between machines. FreeS/WAN is under highly active development.

<http://csrc.nist.gov/ipsec/>

NIST IPsec is also a powerful IPsec implementation. It differs from FreeS/WAN in that it's not constrained to only operate using tunnels. However, it has no support for X.509, which means that machines may require multiple identification tags, depending on service. That increases vulnerability, in and of itself. NIST IPsec is under semi-active development, and is rarely up to the latest kernel.

<http://www.tik.ee.ethz.ch/~skip/>

<http://www.kerneli.org/>

ENSkip implements SUN's SKIP encryption/authentication protocol, and identifies hosts by means of certificates at either end. It's faster than IPsec, for many things, but its encryption is not as sophisticated. ENSkip is not under active development, but is maintained to work with recent kernels by the developers of the International Patch.

Software for User Authentication includes SSh, OpenSSh, LSh, Kerberos, OPIE, S/KEY, PAM, HostSentry, the Shadow Password suite, NIS and NIS+. These programs guarantee that the user is who they say they are. In the cases of SSh, OpenSSh, LSh and Kerberos, the connection remains secure, guaranteeing that further packets are from whom they claim to be. They also prevent interception and tampering of packets in transit, provided the packets go through the encryption mechanism. Unlike IPsec, any stream not explicitly using the encrypted connection will be transmitted in the clear.

<http://www.ssh.org/>

SSh provides a secure connection for clones of the "Remote Shell" suite, combined with enforced username/password authentication. Authentication is done through the standard password system, which will usually be either the standard Unix password system or the shadow password system. Because the connection is secure, the password, in theory, cannot be "sniffed" and subsequently used by another individual. This program is under active development. For commercial users, though, the license may prove cumbersome.

<http://www.openssh.com/>

OpenSSH is derived from an earlier version of SSh, and has no licensing issues. It also provides a secure connection, and uses username/password authentication. This program is under active development.

<http://www.net.lut.ac.uk/psst/>

LSh is derived from version 2 of the SSh protocol, but is released under the GNU Public License and contains no patented algorithms. This is an early work, still, and it is not altogether clear how far it can be trusted in providing a secure, non-spoofable connection.

<http://web.mit.edu/kerberos/www/>

Kerberos provides secure connections for derivatives of telnet and ftp. Authentication is done using Kerberos' own password system, rather than the one provided by Unix. This means that a user can have a different password for Kerberos services than for regular ones, so knowledge of one password will not necessarily compromise the other.

<http://www.inner.net/opie>

OPIE is an implementation of the one-time password system, and is compatible with S/KEY. Users logging in need either a modified Telnet client or a response calculator. The user will be given a challenge, which they will need to feed into the calculator, along with their password. They then give the results of that as their "password". This result is different for each connection. If a "password" is intercepted, it will not provide sufficient information to determine what the next "password" should be. OPIE is under very slow development by the NRL.

S/KEY is an alternative to OPIE. It does pretty much the same thing. It has been dropped by it's developers, though, and is no longer supported or developed.

<http://www.us.kernel.org/pub/linux/libs/pam/>

PAM is a pluggable authentication module, and enables software to use any supported authentication system available on the system, transparently to the user or the application. As such, it is a very powerful authentication system. This software is under active, albeit sluggish, development.

<http://www.psionic.com/abacus/hostsentry/>

HostSentry uses authentication by pattern. Instead of authenticating and validating users according to a password, it records their behavior (such as connection time and location) and records significant variations from that behavior. This behavior can be used to flag an alarm with a system administrator or be used to deny access. This software is not under active development.

<ftp://ftp.ists.pwr.wroc.pl/pub/linux/shadow>

The Shadow Password suite consists of a library and a set of utilities that replace the traditional Unix password system. The library allows application developers to write routines that use shadow passwords, rather than "classic" Unix passwords. The main strength of the suite, over the "classic" Unix system is that the password file is not accessible to the general user, making it impossible for non-privileged accounts to use a password cracker to obtain the passwords for other accounts.

<http://www.suse.de/~kukuk/nis/index.html>

<http://www.suse.de/~kukuk/nisplus/index.html>

NIS and NIS+ are password systems based around Sun's NIS/"Yellow Pages" distributed password system. It offers security through consistency, enabling users to have one password for all related systems, rather than maintain many different passwords. Whilst this creates security issues of its own, it does reduce the risk of users picking "obvious" passwords or writing them down.

Software for Strong Password Validation includes John the Ripper and Crack

<http://www.openwall.com/john/>

John the Ripper tests passwords against a dictionary of commonly used words and phrases, and common variants. This is exactly the sort of file that password crackers will be using to attempt to capture passwords from unsecure password files. Any password that can be broken by this can be obtained by anyone.

<http://www.users.dircon.co.uk/~crypto/>

Crack tests the quality of passwords against a dictionary file, and is capable of manipulating wordlists based on an extensive and powerful word alteration scripting

language. It is one of the oldest and most widely used password testers for Unix.

7. Using the System - Keeping software where it belongs

Even if someone manages to break into a computer, all is not lost. If there are constraints on what software people can run, and what files they can access, merely being able to pass instructions to the machine really isn't as useful as it might first appear. Again, there are many different approaches to managing the use of the system. Which approach is most useful depends, as always, on what is being defended against. Again, going back to the symbolism, this is the same as locking all the doors, and handing out keys to those who strictly need them. Fred, the royal thief, probably wouldn't be given the keys to the treasury, for example.

Software packages for Access Control include POSIX ACL and Trustees.

<http://acl.bestbits.at/>

POSIX ACL is an implementation of the POSIX access control lists. It requires patching a number of utilities, including ext2fs, quota and the GNU file utilities. You also need to install programs for setting and displaying the access control lists. This system is not transparent and only works with the ext2 filing system.

<http://www.braysystems.com/linux/trustees.html>

Trustees is another implementation of access control lists, using objects bound to files. It's more transparent, to the user and the system, than POSIX ACL, and will operate with all filesystems Linux supports.

Software packages for Directory Access Control include TCFS and the Encrypted Home Directory patch.

<http://tcfs.dia.unisa.it/>

TCFS is a filesystem for encrypting a home directory of a user. They are the only ones able to read that directory. To all other users, the directory is encrypted and unusable. When mounted across a network, the encryption remains in place over the network, so only the authorized user(s) can see it. This prevents anyone from accessing any software or data within that directory.

<http://members.home.net/id-est/ehd.html>

The Encrypted Home Directory patch uses the Blowfish algorithm to encrypt a user's home directory, preventing unauthorized access. Files are encrypted/decrypted on the fly.

Software packages for Intrusion Detection and Countermeasures include LOMAC, LCAP, LIDS, the Secure-Linux Patch, Tripwire (free), Tripwire (commercial), and Aide.

<ftp://ftp.tislabs.com/pub/lomac/>

LOMAC is a kernel module that protects processes and data from attack by viruses, trojans and malicious users. It does not appear to be under active development.

<http://home.netcom.com/~spoon/lcap/>

LCAP is a package for removing specific capabilities from the system. Any process, including one run under root, cannot use capabilities removed with this package. This is useful in limiting what an intruder can do.

<http://www.lids.org/>

LIDS, the Linux Intrusion Detection System, is a kernel enhancement which prevents key files from being altered, even by the system administrator, and which hides processes from normal users, locks devices to only certain users or applications, and prevents modules or filesystems from being tampered with. It's under active development and provides a very comprehensive set of tools for locking down the system.

<http://www.openwall.com/linux/>

Secure-Linux Patch is part of the Open Wall project, and provides a fairly comprehensive lock against more common buffer overflow exploits. It'll interoperate with LIDS, or operate stand-alone, to provide a much more secure environment.

<http://www.tripwiresecurity.com/>

Tripwire (free) is a program for monitoring files on the system for tampering. It's unsupported and unmaintained, but it will compile and run on all versions of Linux. Unfortunately, critical files for detection are stored as plain text, which leaves them vulnerable to a skilled attacker.

<http://www.tripwiresecurity.com/>

Tripwire (commercial) is essentially the same as it's free counterpart, except that it is supported and maintained. Linux binaries are available for Red Hat 5.2 and Red Hat 6.0, which suggests there may be problems with more recent versions of glibc. There is no support at all for any other version of Linux, though it will work, with a bit of effort.

<http://www.cs.tut.fi/~rammer/aide.html>

Aide is a free "tripwire" clone, for monitoring files for unauthorized tampering. It

supports a substantial number of digest formats, and a fairly extensible configuration file for adding files for monitoring. It is maintained, and is Open Source, so does not suffer from the limitations that the commercial Tripwire has.

Software for Strong Password Validation includes John the Ripper and Crack

<http://www.openwall.com/john/>

John the Ripper tests passwords against a dictionary of commonly used words and phrases, and common variants. This is exactly the sort of file that password crackers will be using to attempt to capture passwords from unsecure password files. Any password that can be broken by this can be obtained by anyone.

<http://www.users.dircon.co.uk/~crypto/>

Crack tests the quality of passwords against a dictionary file, and is capable of manipulating wordlists based on an extensive and powerful word alteration scripting language. It is one of the oldest and most widely used password testers for Unix.

8. **Lock, Stock and Barrel - How to keep your data safe, when your computer isn't.**

It is always possible for someone to wander into a room and then wander out with a computer. If that computer has commercially sensitive data, private documents, or confidential information, this is not a good thing. However, not everything is lost. A good backup plan can ensure that you retain your information, despite the loss. That, however, isn't going to be enough, for the sorts of data listed above. It's important to ensure that the data isn't available to unauthorized people. To complicate things further, it's got to be a method of protection that will work even when nothing else is. It must be possible for the thieves to be able to take the hard drive out, install it into a machine of their own, run any program of their choosing, and still come up empty-handed.

Is this too much to ask? Not really. There are a number of programs that are ideally suited to this kind of work, which work by encrypting the data at different levels. Depending on the sort of data, and where it's located on the disk, a number of schemes can be used.

Software packages for data encryption include TCFS, the International Patch, PGP 2.x, PGP 5/6 and GPG.

TCFS has been covered in the section for Directory Access Control, in part 5.

<http://www.kernel.org/>

The International Patch provides support for encrypted partitions, using a number of encryption schemes. The partition is then password-locked and encrypted. All data within that partition is totally inaccessible until the person mounting the partition unlocks it.

<http://www.pgp.com/>

<http://www.gnupg.org/>

PGP and GPG use public key encryption to encrypt individual files, making the contents of those files totally inaccessible. Even if a skilled cracker were to obtain the encryption key from the disk, by some means, it would not be useful in deciphering the file. That would require the private key, which the owner would be well advised to keep on a separate disk, preferably removable.

PGP 2.x is the last Open Source branch of PGP. As such, it tends to be trusted rather more than later versions. Files aren't 100% interchangeable between the different versions of PGP, though, which caused some problems. A variant, PGP 2.xi, was developed to bypass the ITAR regulations, limiting the export of encryption technology to Europe. This used a European clone of the RSA encryption library. This would work with the "authorized" version written by the RSA, so that files encrypted by one could be decrypted by the other.

PGP 5/6 were commercial offshoots of PGP 2.x. They included a GUI, and various new encryption algorithms. However, because the source was closed, these are generally not as trusted, as it is impossible for security experts to examine the code for flaws or deliberate weaknesses.

GPG is a clone of PGP 2.6, which was the last official Open Source version of PGP. It does not use any of the code from PGP, but can handle any files generated by it.

9. **Secure Virtual Private Networks - An Extranet in an Internet World**

Sometimes, you don't want anyone outside of a known set of local area networks to be able to access anything on that set of networks. Yet, the networks may be in many different buildings, or even many different countries. It would be impossible to wire them up with a physically isolated network. This is where Virtual Private Networks come into play. With these, it is possible to construct the electronic equivalent of a physically isolated network, without the expense or effort involved.

However, building such a network and building one that's secure against intrusion or surveillance are two very different problems. Fortunately, there are packages that can be used to construct such networks, for Linux. These use encryption to protect the contents, and most use IP tunnels to ensure only authorized machines at either end of a connection can send anything.

Software packages for creating secure virtual private networks include FreeS/WAN, NIST IPSec, ENSkip, VPNd, VPNStarter, VTun, PPTP-Linux, PoPToP and Tinc. (This list does not include VPN software that uses external encryption software or does not support encryption at all.)

FreeS/WAN, NIST IPSec and ENSkip are discussed earlier in this document, as a means of authenticating machines.

<http://sunsite.auc.dk/vpnd/>

VPNd is a package for generating secure virtual tunnels. It uses a proprietary protocol and will not interoperate with other packages. It supports moderate-strength encryption. This package is actively maintained.

<http://vtun.netpedia.net/>

VTun is a package for generating secure virtual tunnels. It uses a proprietary protocol and will not interoperate with other packages. It does use strong encryption, so that the contents of packets cannot be easily obtained. This package is actively maintained.

<http://www.pdos.lcs.mit.edu/~cananian/Projects/PPTP/>

PPTP-Linux is a Linux client for Microsoft's PPTP virtual private network system, and supports Microsoft's encryption and authentication systems. It is completely interoperable with Microsoft PPTP systems.

<http://www.moretonbay.com/vpn/pptp.html>

PoPToP is a Linux server for Microsoft's PPTP virtual private network system. As with PPTP-Linux, it is completely interoperable with Microsoft PPTP systems and supports Microsoft's encryption and authentication schemes.

<http://ftp.nl.linux.org/linux/tinc/>

Tinc is a package for generating secure virtual tunnels. It uses a proprietary protocol and will not interoperate with other packages. It does use strong encryption and key-refresh. (Keys expire after an hour.) It also supports strong authentication, so that uninvited

machines cannot join the VPN.

10. Security Libraries - the Developer's Toolkit

Occasionally, it is necessary to either write, or modify, a program to have security. There exist a number of toolkits that make this task considerably simpler, by providing essential functions for this purpose. Some of these toolkits are cryptographic. Others check the authenticity and validity of the connection. A few provide a means to monitor the network to ensure there is no foul play.

Secure Libraries for Linux include Cyrus SASL, Cryptix SASL, Libpcap, tcpdump, libnids, the Shadow Password suite, crypt, mcrypt, mhash, libdes, libresrsa, Kerberos, SSh, OpenSSh and crack.

<http://asg.web.cmu.edu/sasl/>

Cyrus SASL is a library for providing an authentication mechanism for packages. It is used by IMAP, LDAP, ACAP and other standard protocols.

<http://www.cryptix.org/products/sasl/>

Cryptix SASL is a SASL package for Java. It's very alpha in quality, at present, but does provide some basic authentication mechanisms.

Libpcap is a fairly basic library for dumping all TCP packets on the network. It allows the application developer to write a simplistic TCP packet sniffer, to monitor for suspicious packets. Libpcap is not actively maintained, but is sufficiently well written that it is still widely used.

Tcpdump is a simplistic TCP packet sniffer, based on Pcap, which dumps the headers of TCP packets that pass through a named interface.

Libnids emulates a Linux 2.0.x IP stack and includes support for IP packet defragmentation, TCP stream reassembly, and TCP portscan detection. It can be used as a building block for constructing advanced TCP intruder detection systems.

The Shadow Password Suite is described in the section on user authentication.

Crypt offers basic one-way encryption, primarily for use in passwords. The encryption can't be readily broken, but it's only of very limited interest outside of the traditional Unix password environment.

<http://hq.hellug.gr/~mccrypt/>

Mccrypt is a drop-in replacement for Crypt, except that it also supports many modern encryption systems, including "secret key" algorithms which can be decrypted later. This library is actively maintained.

<http://schumann.cx/mhash/>

Mhash is a library that nicely compliments Mccrypt, in that it provides many modern hashing functions, useful in verification and digital signatures, as well as for other circumstances in which one-way encryption functions are useful.

Libdes is a library based on the "classic" DES algorithm. It's pretty useless, now that DES can be readily broken, but there are still plenty of situations in which DES is mandatory.

Libresrsa is the reference library that uses the public-key algorithm developed by RSA. The patent covering this algorithm has now expired, allowing this algorithm to be widely used. It is already used in a great many products, such as PGP, and is widely regarded as one of the most secure public key algorithms in general use.

Kerberos, Ssh and OpenSsh are described in the section on user authentication.

Crack is a library used to develop Unix password crackers. It's a simple library, providing the tools needed to crack passwords. Used for various nefarious purposes, and legitimate evaluation of users' passwords.

This covers some of the better-known security packages for Linux, and how they fit in the jigsaw that is called computer security for Linux.

[Privacy Statement](#)

Copyright 2006, SecurityFocus