Linux Security Basics

The Basics of Securing Your Linux Box

Presented to the 2006
Nebraska CERT Conference

by Adam Haeder
Vice President of Information Technology
adamh@aiminstitute.org

Isn't Linux already secure?

- Why do I need to worry about this? I thought Linux was already secure? Haven't all those eyeballs in the bazaar squashed every possible security bug there was to squash?

Short answer: No

The system is only as secure
as the person managing it.

Some general guidelines

- Be paranoid! Just because you're not paranoid, doesn't mean they're not out to get you
- Don't think you're not a target
- Trust no one but yourself, and still audit yourself
- Assume the worst will happen, and be prepared when it does

Knowledge is key!

What is Linux?

- The proverbial LEGO operating system
  - Made up of thousands of different pieces, most of them following their own rules
  - Advantage: you can make it do whatever you want; you have complete control
  - Disadvantage: you can make it do whatever you want; you have complete control
- Open source nature theoretically makes all bugs shallow. But they're still there!

Identify, Authenticate, Authorize

- Identify – who you are as a user. Usually accomplished with a username.
- Authenticate – validating the identity. Usually done with a password.
- Authorize – what services do you have access to. Implementation is usually application dependent.

Passwords

- The bane of our existence
- Standard good password policy must be followed
- Use MD5 only for backwards compatibility
- Use Blowfish if at all possible
- Password changing options

Shadow Passwords

- Originally, password hashes (one-way) were stored in /etc/passwd
  - World-readable because authentication systems ran with the security level of the user
- Computers got too fast, became too easy to crack a password hash
- Solution: put them somewhere only root can see them
  - /etc/shadow
  - Authentication programs now must be setuid to root to work

Alternatives to passwords

- Some options attempt to replace both the identification and the authentication pieces, some only the authentication piece.
- Identification and Authentication
  - Smart cards
  - Biometric systems
- Authentication only
  - LDAP
  - Kerberos
  - RADIUS
  - TACACS++

The root user, su and sudo

- Standard Unix security is either all or nothing
- root user can do anything with impunity
- Difficult to successfully audit in a multi-admin environment
- Solution: sudo
  - finer grained permission
  - complete logging

Example sudo configuration

- /etc/sudoers
- All users in group 'wheel' can run any command:

```
%wheel ALL=(ALL)   ALL
```

- All users in group 'users' can mount and unmount the cdrom

```
%users ALL=/sbin/mount
  /cdrom,/sbin/umount /cdrom
```

Sudo logs

- Example log entry from sudo:

```
Aug  7 15:21:20 adam sudo: adamh :
TTY=pts/9 ; PWD=/home/adamh ;
USER=root ; COMMAND=/bin/ls /root
```

Process Accounting

- Probably the single most important thing you can do on your linux box – Know it!
- Understand what each and every process on your system does
  - What security context does it use
  - Where are it's binaries
  - Where does it log
  - Where are it's configuration files
  - What ports does it listen on
  - What files or services does it depend on

Process Accounting

- If you don't know what something is, find out or turn it off
- If it's internal only, configure it to listen on the internal port, or localhost
- Use the chkconfig command to maintain runlevels
- sudo ps -aux
- sudo netstat -anp | grep LISTEN
- sudo lsof | grep “IPv”

iptables

- Firewalling control in the 2.4+ kernels
- Iptables is the userspace program, which allows the root user to manipulate the kernel packet filtering rules
- Stateful and stateless packet filtering
- Network address and port translation
- www.netfilter.org
- Complicated, but not complex

Simple host iptables example

- www.adamhaeder.com/sample_firewall.html
- Load the correct kernel modules
- Zero out old rules and set default policies
- Identify some variables
- Set some kernel parameters
- Accept established connections
- Allow accepted tcp ports
- Allow accepted udp ports
- Handle identd requests
- Accept icmp packets
- Block and log all inbound tcp requests
- Block and log everything not already handled

Package management and verification

- The -V option to rpm is used to validate packages

```
# rpm -V sendmail
S.5....T  c /etc/mail/access
```

- S = file size differs
- M = Mode differs
- 5 = MD5 sum differs
- D = device major/minor number mismatch
- L = symlink path mismatch
- U = user ownership mismatch
- G = group ownership mismatch
- T = time mismatch

Package management and verification

- Verify all rpm packages on a system

```
# for package in `rpm -qa`
  {
   echo “Checking package $package...”
   rpm -V $package
   echo
  }
```

- Find all files in a directory not belonging to a package

```
# CHECKDIR=/usr/bin
# for file in `find $CHECKDIR`;
  {
   rpm -q --whatprovides $file;
  } | grep -i "not owned by any package"
```

Logging

- /var/log/messages
- /var/log/secure
- .bash_history
- Application specific logging (for example, apache logs errors to /var/log/apache/error_log)
- dmesg
- last -a

Filesystem integrity check

- Create a filesystem baseline (locations, permissions, size, existence, etc)
- Maintain that baseline on a read-only medium
- Check the filesystem against that baseline as often as you can
- Tools: tripwire, AIDE, Osec, Osiris

Has my system been compromised?

- Filesystem integrity check?
- Package validation check?
- Something weird in the logs?
- New port open?
- New process running?

What can you trust? What should you trust?

The /proc filesystem

- Ultimate source of knowledge for your linux system
- Understanding /proc is an important key for securing and auditing your linux system
- /proc/[0-9].+/
- **ps, top, w, who** all get their data from the /proc filesystem

Policies and Procedures

- Assume the worst will happen (it's a matter of when, not if)
- Plan, plan, plan. Don't keep it all in your head! Don't say 'We'll cross that bridge when we come it it'
- Decide whose job it is

There is knowledge in knowing, but wisdom in doing

Sites

- securityfocus.com
- slashdot.com
- lwn.net
- linuxtoday.com
- packetstormsecurity.com
- cert.org

Contact Info

Adam Haeder
adamh@aiminstitute.org
(402) 345-5025 x115
PGP Key: http://haederfamily.org/pgp.html