



# Hardening Linux

By Michael Shinn  
Prometheus Global

# Contact Information

Prometheus Global  
14121 Parke Long Court  
Suite 220  
Chantilly, VA 20151  
703-266-6006 Voice  
703-266-6007 Fax

Michael Shinn  
mike@proglc.com

# What You Will Learn

- Basic security principles
- Preventing Compromises
- Detecting Compromises
- Recovering from Compromises

# Hardening Security Principles

- Think like the enemy
- Defense in Depth
- Prevent
- Detect
- Recover

# The Golden Security Rule

**Deny all**  
**except that which is**  
**specifically permitted**

# Achieving a State of Security

- Identify the assets you want to protect
- Identify the risks to those assets
- Identify who & how assets are accessed
- Establish checks and balances
- Develop an enforceable security policies
- Use a layered defense in depth approach
- Plan for disasters

# Why Security Policies Fail

- They impair user productivity
- No or Insufficient user education
- No policies for handling the unexpected
- No support from management
- Security policies are not enforced
- Laxed monitoring & auditing practices
- Users having too many privileges

# Threats

- Malicious
  - Unstructured
  - Structured
  - Highly Structured
- Non-Malicious
  - Accidents
  - Weaknesses
  - Exposures

# The Real Threat

- Non malicious damage resulting from:
  - Human error
  - Denial of service
  - Inappropriate disclosure
- Policy Breakdown
  - Key under the doormat
  - Checks and balances bypassed
  - Rogues on your network

# Defense in Depth

- The failure of a single security control should not result in the compromise of the system
- Assume that controls will fail (because they will)
- Always run with the least privileges you can prove you need

# Prevention

- Defense in Depth
- Software/Patch/configuration Management
- Services Management
- Network Protections
- Kernel and Application Hardening
- Authentication/User Management
- Auditing
- RBAC/MAC

# Detection

- Two types of detection
  - Attacks
  - Vulnerability, Exposures and Weaknesses

# Attack Detection

- Attack Detection
  - NIDS
  - HIDS
  - KIDS
  - etc.
- Anomaly Detection
  - Auditing
  - Logging
  - ADS

# Vulnerability Detection

- Are things still in place?
- Are they still effective?
- Vulnerability Testing (Blue team)
  - Openvas, nessus, tiger, etc.
- Penetration Testing (Red team)
  - Metasploit, etc.

# Recovery

Planning

Backups

Testing

# Linux Hardening

## Prevention

# Installation Considerations

- Ensure the hardware clock is accurately set to the current date, time, and time zone.
- Install the latest system BIOS and firmware
- Ensure that all hardware interfaces or devices that are not required are disabled in BIOS
- Password protect BIOS and boot menus
- Consider using a remote management solution and losing the keyboard, and mouse
- Carefully consider disk controllers / spindles

# Installation Considerations (2)

- Verify your installation source is authentic
- Build and harden the system before plugging it into your network
- Test for vulnerabilities after you finish the install
- You can also build from a package distribution server if you and it are on a trusted internal network
- Do a minimum installation
- Do not install a GUI on a server

# Usage Considerations

- Don't use plain text to connect to your systems
- Desktops get owned, use strong authentication
- Servers should do one thing (no single point of failure)

# Software/Patch Management

- Keeping up to date – signed trusted packages
- Removing unnecessary software
- Testing patches
- Just in Time/Virtual Patching
- Know whats on your system
- Control who can change it
- Detect changes
- Prevent changes

# Disable Services

- If you cant prove you need it – turn it off
- `chkconfig --list | grep '3:on'`
- Example services to disable
  - nfs
  - portmap
  - ftp
  - ACPI modules
  - RPC services
- The system runlevel as specified in the 'initdefault' entry in */etc/inittab* MUST BE '3'
- Put banners on services

# Partition Services

- Never run a service as root
- Run everything as a different user
- Run everything in its own chroot
- Trust NOTHING

# Kernel Hardening

- Stack protections
  - PAX
  - ExecShield
- LKM rootkit attack protections
- Chroot hardening
- Trusted Path Execution
- Exec Logging

# Kernel Hardening

- `kernel.exec-shield=1`
- `kernel.randomize_va_space=1`
- `net.ipv4.conf.all.rp_filter=1`
- `net.ipv4.conf.all.accept_source_route=0`
- `net.ipv4.icmp_echo_ignore_broadcasts=1`
- `net.ipv4.icmp_ignore_bogus_error_messages=1`
- `net.ipv4.conf.all.log_martians = 1`
- `echo 1 > /proc/sys/net/ipv4/tcp_cookies`

# Objects Reuse

- the kernel automatically ensures that new objects (disk files, memory, IPC) do not contain any traces of previous contents

# Network Protections

- firewalling/iptables
- TCP Wrappers
- Turn off protocols you don't use

```
vi /etc/modprobe.conf
```

```
install ipv6 /bin/true
```

```
vi /etc/sysconfig/network
```

```
NETWORKING_IPV6=no
```

```
IPV6INIT=no
```

```
service network restart
```

```
rmmod ipv6
```

- Verify:

```
lsmod | grep ipv6
```

```
/sbin/ifconfig
```

# Network Protections

- Popular web languages
  - PHP
    - php.ini
    - Suhosin
  - Perl
  - Python

# Network Protections

- php.ini changes
  - disable\_functions = , dl , exec , passthru , psocketopen , popen , posix\_kill , posix\_mkfifo , posix\_setuid , proc\_close , proc\_open , proc\_terminate , shell\_exec , system , leak , posix\_setpgid , posix\_setsid , proc\_get\_status , proc\_nice , show\_source , escapeshellcmd , phpinfo
  - allow\_url\_fopen = off
  - allow\_url\_include = off
  - safe\_mode = on
  - register\_globals = off

# Network Protections

- Php suhosin
  - PHP module
  - Patch
  - Protects against known and unknown flaws
  - Huge list of features
  - Default configuration works great out of the box

# Disable root Login Over Network

- Login from the network with user ID 0 ('root') MUST NOT be permitted over the network.
- Administrators MUST use an ordinary user ID to log in, and then use the `/bin/su` - command to switch identities.
- The restriction for direct root logins is enforced through two separate mechanisms.
- For network logins using ssh, the `PermitRootLogin` no entry in `/etc/ssh/sshd config` MUST be set.
- logins use the `pam securetty.so` PAM module in the `/etc/pam.d/login` file that verifies that the terminal character device used is listed in the file `/etc/securetty`.

# Reminder Alias for su

- It is RECOMMENDED that you remind administrators of this by adding the following alias to the bash configuration file */etc/bash.bashrc.local* that disables the pathless 'su' command:

```
alias su="echo \"Always use '/bin/su -'  
(see Configuration Guide)\""
```

- This alias can be disabled for the root user in */root/.bashrc*:

```
unalias su
```

# Update permissions for su

- The 'su' binary MUST be restricted to members of the 'trusted' group. This will be enforced both with PAM configuration (configured later) and the binary's permissions.
  - **chgrp trusted /bin/su**
  - **chmod 4750 /bin/su**
- You MUST have at least one user account other than 'root' configured to be a member of the 'trusted' group, otherwise system administration will ONLY be possible from the system console.

# Reminder Alias for su

- It is RECOMMENDED that you remind administrators of this by adding the following alias to the bash configuration file */etc/bash.bashrc.local* that disables the pathless 'su' command:  
**alias su="echo \"Always use '/bin/su -'\""**
- This alias can be disabled for the root user in */root/.bashrc*:  
**unalias su**

# Update permissions for su

- The 'su' binary MUST be restricted to members of the 'trusted' group. This will be enforced both with PAM configuration (configured later) and the binary's permissions.
  - **chgrp trusted /bin/su**
  - **chmod 4750 /bin/su**
- You MUST have at least one user account other than 'root' configured to be a member of the 'trusted' group, otherwise system administration will ONLY be possible from the system console.

# Setting up ssh

- SSH protocol version 1 MUST be disabled.
- The ssh client MUST NOT be set up SUID root
- The SSH Server MUST be configured to reject attempts to log in as root.
- Good authentication mechanisms are per-user (nonempty) passwords
- Better authentication uses per-user RSA/DSA public key authentication.
- All other authentication methods MUST be disabled.
- The setting `PAMAuthenticationViaKbdInt` MUST be disabled, since this would otherwise circumvent the disabled root logins over the network.

# /etc/ssh/sshd.conf

```
# Cryptographic settings. Disallow obsolete insecure protocol version 1, and hardcode a strong cipher.
Protocol 2
Ciphers aes256-cbc
# Configure password-based login. This MUST use the PAM library
# exclusively, and turn off the builtin password authentication code.
UsePAM yes
ChallengeResponseAuthentication yes
PasswordAuthentication no
PermitRootLogin no
PermitEmptyPasswords no
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PubkeyAuthentication yes
RSAAuthentication yes
KerberosAuthentication no
GSSAPIAuthentication no
X11Forwarding no
Subsystem sftp /usr/lib/ssh/sftp-server -f LOCAL7 -l INFO
# Restrict users
AllowUsers user1 user2 user3
DenyUsers baduser1 baduser2
#Send a banner to the user
Banner /etc/issue
```

# Sshd additional

Two factor authentication - keys

Three factor authentication – Ask more questions

<https://calomel.org/openssh.html>

# Identification and Authentication

- Remove all accounts you cant prove you need
- Login with least privileges
- IA tools
  - Pluggable Authentication Module (PAM)
  - OpenSSH
  - su
  - sudo

# Identification and Authentication

- Configure password aging
  - `chage -M 60 -m 7 -W 7 userName`
  - `-M maxdays`
  - `-m mindays`
  - `-W warndays`
- Configure to prevent dictionary attacks
  - `vi /etc/pam.d/system-auth`
  - `password required /lib/security/pam_cracklib.so  
retry=2 minlen=10 difok=6`

# Configure PAM

- Add the *pam\_wheel.so* module to the 'auth' configuration for the 'su' service
- The 'remember=XX' option must be added to the */etc/security/system-auth* file to force users to create new passwords and not re-use:

```
password    sufficient    pam_unix.so sha512 shadow  
remember=12 try_first_pass use_authtok
```

# Setup Login Controls

- The default umask for logged-in users is set in the */etc/profile* file.
- Umask to 077 which is used by useradd and newusers for creating new home directories.

# sudoers

- Use sudoers to restrict users to only the commands they need to run
- Example of a backup user
  - /etc/sudoers:  
backup2 ALL = NOPASSWD: /usr/bin/rdiff-backup
  - ~backup2/.ssh/authorized\_keys  
command="/usr/bin/sudo -u root rdiff-backup  
--server --restrict-read-only /",no-port-  
forwarding,no-X11-forwarding,no-agent-  
forwarding,no-pty ssh-rsa AAAA.....

# Access Control

- DAC
- MAC
- RBAC

# Discretionary Access Control

- Linux is a multi-user operating system. You can control which other users will be able to read or modify your files by setting the Unix permission bits and user/group IDs
- You can achieve more precise control using POSIX-style access control lists (ACLs).
- The administrators ('root') are able to override these permissions and access all files on the system.
- Use of encryption is **RECOMMENDED** for additional protection of sensitive data. (LUKS for example)

# Mandatory Access Control

- SELinux
- Very technical
- Bottom Up
- Application Specific

# Role Based Access Control

- RBAC
- Easy to use
- Top down/Least Priv
- Grsecurity's RBAC is self learning

# Application Hardening

- Input Protections
- Restricted Environments
- Least Privs

# Input Protections

- Web applications - modsecurity
- Database applications -

# Restricted Execution Environment

- Setup a chroot directory structure
- Enable chroot support for those services that can be chroot'd
- Some services can be installed into your chroot'd environment

# Defanging the Environment

- Locking down shells
- Finding SUIDS/GUIDS
- Finding world writable dirs
- Locking down “dangerous” tools
- Upload scanning
- Realtime scanning
- Filesystem security options

# Restricted Execution Environment

- Setup a chroot directory structure
- Enable chroot support for those services that can be chroot'd
- Some services can be installed into your chroot'd environment

# Find dangerous bins and dirs

- Find SUID binaries

```
find / -xdev -type f -perm +u=s -print
```

- Find SGID binaries

```
find / -xdev -type f -perm +g=s -print
```

- Find World Writable Directories

```
find / -xdev -perm +o=w ! \( -type d -perm +o=t \) !  
-type l -print
```

- Find un-owned files

```
find / -xdev \( -nouser -o -nogroup \) -print
```

# Preventing access to Tools

Some programs are used to get around protections and download root kits

Change the permissions on these programs so that only users in the “trusted” group can use them

# Preventing access to Tools

Examples:

Wget – used to download tools

Perl – used to run tools

# Preventing access to Tools

```
chmod o-rwx /usr/bin/wget  
chgrp trusted /usr/bin/wget
```

# malware upload detection

Proftp with mod\_clamav

apache with mod\_security

# Realtime malware detection

modprobe redirfs

modprobe dazuko

ClamukoScanOnAccess yes

ClamukoScanOnOpen yes

ClamukoScanOnClose yes

ClamukoScanOnExec yes

ClamukoIncludePath /var/www

ClamukoIncludePath /tmp

ClamukoExcludePath /some/path

# Configure the Boot Loader

- Ensure the system boots exclusively from the disk partition containing Linux
- Make sure you use BIOS password to protect access to this configuration.
- Use the password command in */boot/grub/menu.lst* to prevent unauthorized use of the boot loader interface.
- Use md5 encoded passwords, run the command *grub-md5-crypt* to generate the encoded version of a password.

# Filesystem protections

- Nodev
- Noexec
- Nosuid
- Umask settings

# Quotas and limits

- /etc/security/limits.conf
- Each line describes a limit for a user in the form:  
    <domain> <type> <item> <value>
- Example to prevent a fork bomb:
  - you hard nproc 300
  - @users hard nproc 50
  - @powerusers hard nproc 200

# Locked down shells

- Upload only (scponly)
- Restricted shell (bash -r)
  - The cd built-in is disabled.
  - Setting or unsetting SHELL, PATH, ENV or BASH\_ENV is not possible.
  - Command names can no longer contain slashes.
  - Filenames containing a slash are not allowed with the . (source) built-in command.
  - The hash built-in does not accept slashes with the -p option.
  - Import of functions at startup is disabled.
  - SHELLOPTS is ignored at startup.
  - Output redirection using >, >|, ><, >&, &> and >> is disabled.
  - The exec built-in, -f and -d are disabled.
  - A default PATH cannot be specified with the command built-in.
  - Turning off restricted mode is not possible.

# Hardening tools

- Sussen
- Bastille (old)
- Titan (older)
- Tiger (oldest)
- Cops (really old)

# Configure the Boot Loader

- Protect all menu entries
- Add a line containing just the keyword lock after the title entry in the */boot/grub/menu.lst* file
- Remove group and world read permissions from the grub configuration file if it contains a password  
**chmod 600 /boot/grub/menu.lst**
- All changes to the configuration take effect automatically on the next boot

# Considerations for Desktops

- `ssh -x -a`
- Run multiple copies of firefox as different users
- Use noscript, requestpolicy, betterprivacy and WOT
- Use OpenDNS
- Set to update automatically
- Use non-routable Ips
- Use a separate hardware firewall and iptables
  - Outgoing only if you can

# Considerations for Desktops (2)

- Lock down boot media
- Cold Memory attacks
- Trivial root login protection (single user mode)

# Considerations for DNS Servers

- Enable bind chroot support.
- Apply port restrictions in firewall.
- Customize logging as desired.
- Authoritative DNS servers should not be used as resolving or caching DNS servers.
- Disable recursive queries on authoritative servers.
- Enable numerous security settings in `/etc/named.conf` to suit your environment.

# Considerations for Email Servers

- Chroot mailserver
- Ensure unauthorized parties can't relay
- Establish port restrictions and access control with iptables.
- Configure smtp restrictions in postfix.
- Use ldap or access file to restrict inbound mail to valid users
- Anti-virus / Anti-Spam

# Linux Hardening

## Detection

# Detecting Attacks

- Attack Detection
  - HIDS
  - NIDS
  - KIDS
  - WIDS
- Anomaly Detection
  - Auditing
  - Logging
  - ADS

# Security monitoring & management

- Setup HIDS, such as OSSEC
- Setup tripwire/AIDE/OSSEC to monitor system file integrity and to audit changes.
- Setup and implement log file rotation policies.
- Setup a central syslog server (syslog-ng, OSSEC, etc.)
- Use a log analyzer, such as **OSSEC**.
- Setup a monitoring system like Nagios or Argus on your network.

# Sec Monitoring & Management (2)

- Created `/var/log/btmp` to log bad login attempts.

```
# touch /var/log/btmp
```

```
# lastb
```

```
btmp begins Sun Dec 14 08:13:22 2008
```

# Detect Attacks

- *Setup snort for NIDS*
- Use a hardened kernel such as [grsecurity.net](http://grsecurity.net) for KIDS
- Use OSSEC for HIDS
- Use `mod_security` for a WIDS

# Anomaly detection

- *Auditing*
- Smart HIDS and a smart operator

# Simple auditing

- **Enable the psacct service**  
`/etc/init.d/psacct start`
- **Lastcomm**
  - **Displays previously executed commands**

# auditd

- **Enable the auditd service**

**`/etc/init.d/auditd start`**

- **chkconfig auditd on**

- **Let us say you would like to audit a `/etc/passwd` file. You need to type command as follows:**

**`auditctl -w /etc/passwd -p war -k password-file`**

# auditd

- `-w /etc/passwd` : Insert a watch for the file system object at given path i.e. watch file called `/etc/passwd`
- `-p war` : Set permissions filter for a file system watch. It can be `r` for read, `w` for write, `x` for execute, `a` for append.
- `-k password-file` : Set a filter key on a `/etc/passwd` file (watch). The password-file is a filterkey (string of text that can be up to 31 bytes long). It can uniquely identify the audit records produced by the watch. You need to use password-file string or phrase while searching audit logs.

# auditd

- `-w /etc/passwd` : Insert a watch for the file system object at given path i.e. watch file called `/etc/passwd`
- `-p war` : Set permissions filter for a file system watch. It can be `r` for read, `w` for write, `x` for execute, `a` for append.
- `-k password-file` : Set a filter key on a `/etc/passwd` file (watch). The password-file is a filterkey (string of text that can be up to 31 bytes long). It can uniquely identify the audit records produced by the watch. You need to use password-file string or phrase while searching audit logs.

# auditd

- `ausearch -f /etc/passwd`
- Another example:
  - To see all syscalls made by a program:
  - `auditctl -a entry,always -S all -F pid=1234`

# Detecting Issues

- **Exposures**
- Weaknesses
- Vulnerabilities
- Compromises
- Are controls still in place?
- Are they still effective against the threat?

# Tools to detect Issues

- Robust free tools
  - Nessus
  - Openvas
- Special free Tools
  - nmap
  - Rkhunter
  - OSSEC
- Regularly and Randomly Test

# Top Down

- Don't use vulnerability analysis to find holes to plug
- Use vulnerability analysis to find what you missed

# Linux Hardening

## Recovery

# Planning for Disasters

- Securely install your operating system
- Accurate time source
- Know every file on your system
- Validate system integrity
- Centralize logging
- Monitor and audit your system regularly
- Documentation and procedures
- Emergency response team
- Backup, backup, backup (Make sure you test your restore procedures periodically)

# Cheap mirror/incremental backups

- Local example
  - `rdiff-backup --include-filelist exclude.txt / /backups`
  - `rdiff-backup --remove-older-than 45B /backups`
- Remote example
  - `Rdiff-backup --include-filelist exclude.txt backup@remote ::/ /backups`
  - `rdiff-backup --remove-older-than 45B /backups`
- fusecompress
  - `fusecompress /storagedir /mountdir`

# Contact Information

Prometheus Global  
14121 Parke Long Court  
Suite 220  
Chantilly, VA 20151  
703-266-6006 Voice  
703-266-6007 Fax

Michael Shinn  
mike@proglc.com