Preparation 1

- A physical access to the suspicious system should be offered to the forensic investigator.
- A physical copy of the hard-disk might be necessary for forensic and evidence purposes. If needed, a physical access could be necessary to disconnect the suspected machine from any network.
- A good knowledge of the usual network activity of the machine/server is needed. You should have a file on a secure place describing the usual port activity, to compare efficiently to the current state.
- A good knowledge of the usual services is needed. Don't hesitate to ask a Unix/Linux Expert for his assistance, when applicable.
- You should have a regularly updated list of all critical files, (especially SUID and GUID files) stored in a secure place out of the network or even on paper. With this list, you can easily separate usual SUID files and detect unusual ones.
- Have a map of your usual port activity/traffic rules.

Identification 2

Unusual Accounts

Look for any suspicious entry in /etc/passwd, especially with UID 0. Also check /etc/group and /etc/shadow.

Look for orphaned files, which could have been left by a deleted account used in the attack:
 # find / \(-nouser –o –nogroup \) -print

Unusual Files

- Look for all SUID and GUID files:

 # find / -uid 0 \(–perm -4000 –o –perm 2000 \) –print

- Look for weird file names, starting with ". " or ".. " or " " :

*# find / -name " *" –print*
*# find / -name ". *" –print*
*# find / -name ".. *" –print*

- Look for large files (here: larger than 10MB)

 # find / -size +10MB –print

- Look for processes running from or to files which have been unlinked :

 # lsof +L1

Identification 2

- Look for unusual files in /proc and /tmp. This last directory is a place of choice for hackers to store data or malicious binaries.

Unusual Services

(Linux only) Run chkconfig (if installed) to check for all enabled services:
 # chkconfig --list

Look at the running processes (remember: a rootkit might change your results for everything in this paper, especially here!).
 # ps -aux
Use *lsof –p [pid]* on unknown processes

You should know your usual running processes, and be able to figure out which processes could have been added by a hacker. Pay a special attention to the processes running under UID 0.

Unusual Network Activity
Try to detect sniffers on the network using several ways:
Look at your kernel log files for interfaces entering promiscuous mode such as :"*kernel: device eth0 entered promiscuous mode*"

Use *# ip link* to detect the "PROMISC" flag. Prefer this method to *ifconfig*, since *ifconfig* does not work on all kernels.

- Look for unusual port activity: *# netstat –nap* and *# lsof –i* to get more information about processes listening to ports.

- Look for unusual MAC entries in your LAN:

 # arp -a

- Look for any unexpected IP address on the network.

Unusual Automated Tasks

- Look for unusual jobs scheduled by users mentioned in /etc/cron.allow. Pay a special attention to the cron jobs scheduled by UID 0 accounts (root):

 # crontab –u root -l

- Look for unusual system-wide cron jobs: *# cat /etc/crontab* and *# ls –la /etc/cron.**

Unusual Log Entries

Look through the log files on the system for suspicious events, including the following:

Identification 2

- Huge number of authentication/login failures from local or remote access tools (sshd,ftpd,etc.)
- Remote Procedure Call (RPC) programs with a log entry that includes a large number of strange characters …)
- A huge number of Apache logs mentioning "error"
- Reboots (Hardware reboot)
- Restart of applications (Software reboot)

Almost all log files are located under /var/log directory in most Linux distributions. Here are the main ones:

/var/log/message: General message and system related stuff
/var/log/auth.log: Authenication logs
/var/log/kern.log: Kernel logs
/var/log/cron.log: Crond logs (cron job)
/var/log/maillog: Mail server logs
/var/log/httpd/: Apache access and error logs directory
/var/log/boot.log: System boot log
/var/log/mysqld.log: MySQL database server log file
/var/log/secure: Authentication log
/var/log/utmp or **/var/log/wtmp:** Login records file

To look through the log files, tools like cat and grep may be useful:
cat /var/log/httpd/access.log | grep "GET /signup.jsp"

Unusual Kernel log Entries

- Look through the kernel log files on the system for suspicious events.

Use :
dmesg
List all important kernel and system information :
lsmod
lspci

- Look for known rootkit (use rkhunter and such tools)

File hashes

Verify all MD5 hashes of your binaries in /bin, /sbin, /usr/bin, /usr/sbin or any other related binary storing place. (use AIDE or such tool)
WARNING: this operation will probably change all file timestamps. This should only be done after all other investigations are done and you feel like you can alter these data.

On systems with RPM installed, use:
 # rpm –Va | sort
On some Linux, a script named *check-packages* can be used.
On Solaris: # pkg_chk –vn
On Debian: debsums –ac
On Openbsd (not really this but a way): pkg_delete -vnx

Containment 3

Backup all important data from the compromised machine, if possible using a bit-by-bit physical copy of the whole hard disk on an external support. Also make a copy of the memory (RAM) of the system, which will be investigated if necessary.

If the machine is not considered critical for the company and can be disconnected, shut the machine down the hard way, removing its power plug. If it is a laptop with a battery on, just push the "off" button for some seconds until the computer switches off.

Offline investigations should be started right away if the identification step didn't give any result, but the system is still suspected of being compromised.

Try to find evidences of every action of the hacker: (using forensic tools like Sleuth Kit/Autopsy for example)

- **Find all files used by the attacker**, including deleted files and see what has been done with them or at least their functionality to evaluate the threat.
- **Check all files accessed recently**.
- **Check log files**.
- More generally, try to **find how the attacker got into the system**. All leads should be considered. If no computer proof of the intrusion is found, never forget it could come from an insider.
- Apply fixes when applicable, to prevent the same kind of intrusion, in case the attacker used a known fixed vulnerability.

Remediation 4

Temporary remove all accesses to the accounts involved in the incident, and remove all fraudulent files.

Recovery 5

No matter how far the hacker has gone into the system and the knowledge you might have about the compromission, as long as the system has been penetrated, the best practice is **to reinstall the system completely and apply all security fixes**.
In case this solution can't be applied, you should:

- Change all the system's accounts passwords, and make your users do so in a secure way: they should use passwords with upper/lower case, special characters, numbers, and at least be 7 characters long.
- Check the integrity of the whole data stored on the system, using MD5 hashes.
- Restore all binaries which could have been changed (Example: /bin/su)

Aftermath 6

Report

A crisis report should be written and made available to all of the actors of the crisis management cell. The following themes should be described:

- Initial detection
- Actions and timelines
- What went right
- What went wrong
- Incident cost

Capitalize

Actions to improve the Unix/Linux intrusion detection management processes should be defined to capitalize on this experience.

IRM #3

Unix/Linux Intrusion Detection

Live Analysis on a suspected system

Abstract

This Incident Response Methodology is a cheat sheet dedicated to incident handlers investigating a precise security issue.
Who should use IRM sheets?

- Administrators
- Security Operation Center
- CISOs and deputies
- CERTs (Computer Emergency Response Team)

Remember: If you face an incident, follow IRM, take notes and do not panic. Contact your CERT immediately if needed.

Incident handling steps

6 steps are defined to handle security Incidents

Preparation: get ready to handle the incident
Identification: detect the incident
Containment: limit the impact of the incident
Remediation: remove the threat
Recovery: recover to a normal stage
Aftermath: draw up and improve the process

IRM provides detailed information for each step.