Rootkits

* A package of malware used to take over a m/c or mobile.
* Collection of attacker tools installed after an intruder has gained access.
  - → log cleaners → turn off logging process.
  - → File / process / user hiding tools
  - → N/w sniffers
  - → Backdoor programs

Rootkit Goals

1. Remove evidence of original attack and activity that led to rootkit installation.
2. Hide future attacker activity (files, n/w connections, processes) and prevent it from being logged.
3. Enable future access to system by attacker.
4. Install tools to widen scope of penetration.
5. Secure system so other attackers cannot take control of system from original attacker.

Concealment Techniques used by Rootkits.

* Remove log and audit file entries.
* Modify system programs to hide attacker files, network connections and processes.
* Modify logging system to not log attacker activities.
* Modify OS kernel system calls to hide attacker activities.

Installation Concealment

* Use a subdirectory of a busy system directory like /dev/, /etc, /lib, or /usr/lib
* Use dot files, which are not in ls output.
* Use spaces to make filenames look like expected dot files: eg. "." and ".."
* Use filenames that system might use
  → /dev/tdd (if no 4th IDE disk exists)
  → /usr/lib/libx.a (libX11 is real Sun X-windows)
* Delete rootkit install directory once installation is complete.

Attack Tools Installed by Rootkits

* N/w Sniffers → including password grabber utility.
* Password cracker
* Vulnerability scanners
* Autorooter → automatically applies exploits to host ranges
* DDOS tools → Distributed Denial of service

History of Rootkits

1989: Phrack 25 - Black Tie Affair ; wtmp wiping.
1994: Advisory CA-1994-01 about SunOS rootkits
1996: Linux Rootkits (Irk3 released.)
1997: Phrack 51 Halflife article: LKM-based Rootkits
1998: Silvio Cesare's kernel patching via kmem.
1999: Greg Hoglund's NT kernel rootkit paper.
2005: Sony ships CDs with rootkits that hide DRM and spyware that auto-installs when CD played.
2006: SubVirt rootkit moves real Os to a VM.

Types of Rootkits

* **User-mode Rootkits** → (a) (Binary Rootkits) they replace normal user programs like ls, netstat, ps with their versions.
  → Create Trojan backdoors : login sshd.

  (b) **Library Rootkits** → replace system libraries
  → Intercept library calls to hide activities & add backdoors

* **Kernel Rootkits**
  → they modify system calls / structures that all user-mode progsams rely on to list users, processes and sockets.
  → Add backdoors to kernel itself. which is more dangerous.

Binary Rootkits : They do the following

* Install trojan-horse versions of common system commands, such as ls, netstat and ps to hide attacker activities.
* Install programs to edit attacker activity from log and accounting files.
* Install trojan-horse ~~variation~~ variants of common programs like login, passwd, and sshd to allow attacker continued access to system.
* Install network sniffers.

Linux Root kit (LRK) V4 Trojans :-

Following linux programs are replaced by Trojans (because of linux Rootkit)
ifconfig, login, ls, Passwd, PS, rshd, syslogd etc.

Binary Rootkit Detection → involves follo

* Use non-trojaned programs such as following
  - → Ptree is generally uncompromised
  - → tar will archive hidden files. the list with -t
  - → lsof is also generally safe.
  - → use known good tools from CD-ROM.

* File Integrity checks
  - → tripwire, AIDE, Osiris
  - → rpm -V -a
  - → Must have known valid version of database offline or attacker may modify file structures to match trojans.

Library Rootkits → also known as torn rootkit
- → uses special system library libproc.a to intercept process information requested by user utilities.
- → Modify libc → they intercept system call data returning from kernel, stripping out evidence of attacker activities.

→ Library rootkits, alternately, ensure that rootkit library providing system calls is called instead of libc by placing it in /etc/ld.so.preload

Kernel Rootkits :- Since kernel runs in supervisor mode, kernel rootkit have complete control over machine.

* Rootkits modify kernel system calls
  ex → execve modified to run Trojan horse binary for some programs, while other system calls used by integrity checkers read original binary file.
  → setuid modified to give root to a certain user.
* Stealth - Provided by kernel Rootkits
  Runtime integrity checkers cannot see rootkit changes.
  → All programs impacted by kernel Trojan horse.
  → Open backdoors / sniff n/w without running process.

Types of Kernel Rootkits → come in many forms

① Loadable Kernel Modules. (LKMs)

→ Device Drivers are LKMs.

→ LKMs can be defeated by disabling them.

ex – Adore, knark

② Alter running kernel in memory

→ they can modify /dev/kmem directly (this file gives current state of memory)

ex. Suckit.

③ Alter Kernel on Disk → when system is booting, it can be infected.

Kernel Rootkit Detection → For detection following can be done –

* List kernel modules [lsmod / cat/proc/modules

* Examine kernel symbols (/proc/kallsyms)

→ Module name listed in [] after symbol name.

→ Check system call addresses (Compare running kernel syscall addresses with those listed in System.map generated at kernel compile.

→ But sometimes all of these signatures can be hidden/forged.

Knark – Linux based LKM rootkit

* Features of knark are following:
  → Hide/unhide files or directories
  → Hide TCP or UDP connections
  → Execute redirection
  → Unauthenticated, privilege escalation
  → Utility to change UID/GID of a running process.
  → Unauthenticated, privileged remote execution daemon.
  → Kill -31 to hide a running process.

* Modhide :- an assistant LKM that hides knark from module listing attempts.

Rootkit Detection :- Following can clone for dete

* **Offline System Examination**
  - → Mount and examine disk using another OS kernel+image.
  - → Knoppix : live CD linux distribution

* **Computer Forensics**
  - → Examine disk below filesystem level.
  - → Helix ; live CD linux forensics tool.

Rootkit Detection Utilities.

The other utilites for detection are following.

* **Chkrootkit**
  - → detects > 50 rootkits on multiple UNIX types.
  - → Checks commonly trojaned binaries.
  - → Examines log files for modifications.
  - → Checks for LKM rootkits.
  - → Use -p option to use known safe binaries from CDROM.

* **Carbonite**
  - → LKM that searches for rootkits in kernel.
  - → Generates and searches frozen image kernel process structures.

Detection Countermeasures by Hackers.

- Hide rootkit in unused sectors or in unused fragments of used sectors.
- Install rootkit into flash memory like PC BIOS ensuring that rootkit persists even after disk formatting and OS re-installation.

Rootkit Recovery (Recovery from Rootkit)

- Restore compromised programs from backup
  - → Lose evidence of intrusion
  - → Did you find all the trojans?
- Backup System, then restore from tape
  - → Save image of hard disk for investigation.
  - → Restore known safe image to be sure that all trojans have been eliminated.
  - → Patch system to repair exploited vulnerability.

Key points about Rootkits

* Backdoors allow intruder into system without using exploit again.
* Rootkits automatically deeply compromise a system once root access is attained.
* Rootkits are easy to use, difficult to detect.
* Don't trust anything on a compromised system and access disk from a known safe system like a knoppix CD.
* Recovery requires a full re-installation of the OS and restoration of files from a known good backup.