Homework #5 Solution

Contact TAs: vegetable@csie.ntu.edu.tw

System Administration 1

Debian Package Manager

1. `update`: Update the list of the latest packages available in remote repository.
   `upgrade`: Upgrade all installed packages without any removals.
   `dist-upgrade`: Upgrade all installed packages. Remove some packages if needed.
2. `remove`: Remove the specified package.
   `autoremove`: Remove pacakges that are no longer required (installed as dependencies).
   `purge`: Remove the specified package and its configuration files.
3. `apt-cache search --names-only perl`
   (`dpkg -l` only saearh installed packages)
4. `apt-file search /usr/bin/ncat`
   (`dpkg -S` only search installed packages, `apt-file search` can be used to search what package to install to get the executable)
5. `apt-mark showmanual gcc`
   (if output contains gcc, it's installed as explicit, otherwise as dependencies).
6. `apt-mark manual gcc`
7. ```
   gpg --gen-key
   dpkg-sign --sign origin -k $keyid nasa-meta.deb
   ```
8. `gpg --armor --export $keyid | apt-key add -`
9. ```
   mkdir -p /srv/repo
   cp nasa-meta.deb /srv/repo
   cd /srv/repo
   apt-ftparchive packages . > Packages
   apt-ftparchive release . > Release
   gpg --default-key \$keyid --clearsign -o InRelease Release
   gpg --default-key \$keyid -abs -o Release.gpg Release
   echo "deb file:///srv/repo /" >> /etc/apt/sources.list
   ```

System Administration 2

System Log

The standard system logging facility

Note

There is no "standard" or "most correct" answer in this problem. Your answer may be completely different from examples shown here because it is highly dependent on the system you use.

Processes and packages

Process: Get the list of listening sockets or all sockets, and filter the output with `grep`.

Table 1: Get the list of all sockets

OS	Process	How to find
Systemd-based GNU/Linux	`systemd`, `systemd-journald`	`ss -ap`, `netstat -ap`
Other GNU/Linux	`syslogd`, `syslog-ng`, `rsyslogd`	`ss -ap`, `netstat -ap`
FreeBSD	`syslogd`	`sockstat`
NetBSD	`syslogd`	`sockstat`
DragonFlyBSD	`syslogd`	`sockstat`
OpenBSD	`syslogd`	Run `netstat -a` to get the address of socket PCB and use `fstat` to get the process associated with the address.

Package: Get the path to the executable file of the process, and search for the path in the package manager database.

Table 2: Get the path to the executable

OS	How to find
GNU/Linux	`readlink /proc/<pid>/exe`
FreeBSD	`procstat -e <pid>`
NetBSD	`readlink /proc/<pid>/exe`
DragonFlyBSD	`readlink /proc/<pid>/file`
OpenBSD	Run `fstat -p <pid>` to get the mountpoint and the inode number of the file and use `find -x <mountpoint> -inum <inum>` to find the path.

Table 3: Search for the path in the package manager database

Package Manager	How to find
RPM	`rpm -qf <path>`
Debian (dpkg)	`dpkg -S <path>`
Pacman (alpm)	`pacman -Qo <path>`
Portage (emerge)	`qfile <path>`

Table 4: Possible search results

OS	Package
Systemd-based GNU/Linux	`systemd`
Other GNU/Linux	`sysklogd`, `syslog-ng`, `rsyslog`
FreeBSD	none, included in the base system
NetBSD	none, included in the base system
DragonFlyBSD	none, included in the base system
OpenBSD	none, included in the base system

Log Files

We use command "`logger SAHW5`" as the example here.

Table 5: Possible locations and file types

OS	Location	Type	Command
Systemd-based GNU/Linux	`/var/log/journal`	binary	`journalctl`
Other GNU/Linux (Debian)	`/var/log/syslog`	text	none
Other GNU/Linux (Gentoo)	`/var/log/messages`	text	none
FreeBSD	`/var/log/messages`	text	none
NetBSD	`/var/log/messages`	text	none
DragonFlyBSD	`/var/log/messages`	text	none
OpenBSD	`/var/log/messages`	text	none

Table 6: Possible lines and entries

OS	Line or entry
Systemd-based GNU/Linux	`<date> <time> <hostname> <username>[<pid>]: <msg>`
Other GNU/Linux	`<date> <time> <hostname> <username>: <msg>`
FreeBSD	`<date> <time> <hostname> <username>: <msg>`
NetBSD	`<date> <time> <hostname> <username>: <msg>`
DragonFlyBSD	`<date> <time> <hostname> <username>: <msg>`
OpenBSD	`<date> <time> <hostname> <username>: <msg>`

Forged Messages

On systemd-based systems, messages sent by users are usually stored in the same file as long as `logger` commands are executed by the same user in the same session. The file that new messages are written to is usually `/var/log/journal/<machine-id>/user-<uid>.journal`. Systemd journal stores messages with trusted metadata. Fields prefixed with an underscore can only be set by systemd journal service and cannot be set or modified by users, so it can be used to determine whether an entry is really produced by a system service. These additional data fields can be shown by running `journalctl -o verbose`, `journalctl -o json`, or other formats that can display structured data. A list of available fields can be found in `systemd.journal-fields(7)`. Difference in `_UID`, `_GID`, `_EXE`, `_SYSTEM_CGROUP`, `_SYSTEMD_UNIT` fields can be easily found.

On other systems, messages may be written to the same file or different files, depending on the configuration used. Most implementations don't store trusted metadata by default, and not all implementations provide access to trusted information. `rsyslog` can be configured to ignore PIDs provided by clients and use trusted values obtained

from UNIX domain sockets with `SysSock.UsePIDFromSystem` parameter. `syslog-ng` allow access to trusted UNIX credentials with `${.unix.uid}`, `${.unix.gid}`, `${.unix.exe}` and other macros. `sysklogd` cannot retrieve trusted information from sockets, but third-party patches are available to support it.

Systemd journal service

Version

Run any systemd-provided commands with `--version` option. Example:

```
$ systemctl --version
systemd 229
+PAM +AUDIT +SELINUX +IMA -APPARMOR +SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP +GCRYPT +GNUTLS +ACL +XZ +LZ4 +SECCOM
```

Persistence

Yes if files are stored in `/var/log/journal`. No if files are stored in `/run/log/journal`. It can be made persistent by creating the directory `/var/log/journal` and restarting `systemd-journald.service`.

Dmesg of previous boot

```
journalctl -b -1 _TRANSPORT=kernel
```

Messages generated by the SSH server

```
journalctl _SYSTEMD_UNIT=sshd.service
```

Messages produced by both dbus user and your own user account

```
journalctl _UID=$(id -ru dbus) + _UID=$(id -ru)
```

Messages generated by /usr/bin/sudo

```
journalctl _EXE=/usr/bin/sudo
```

Network Log

Use Linux netfilter to log packets

Enable logging in nftables or iptables

Set up `nftables` to send packets to `NFLOG`, an interface which allows packets to be logged by userspace applications.

Use `nft` command to add the rule.

```
# nft add table inet filter
# nft add chain inet filter output { type filter hook output priority 0 ';' }
# nft add rule inet filter output oif '!=' lo log group 10425
```

Or load rules written in a file with `nft -f`.

```
# cat /etc/nftables.conf
table inet filter {
    chain output {
        type filter hook output priority 0;
        oif != lo log group 10425
    }
}

# nft -f /etc/nftables.conf
```

Alternatively, you can use `iptables` if you prefer it or your distribution doesn't support `nftables`.

Use `iptables` command to add the rule.

```
# iptables -t filter -A OUTPUT '!' -o lo -j NFLOG --nflog-group 10425
# ip6tables -t filter -A OUTPUT '!' -o lo -j NFLOG --nflog-group 10425
```

Run a program to receive packets from kernel

We use `ulogd` as an example here. If you don't like it, you can write your own program with a library called `libnetfilter_log`.

Create a configuration file and start `ulogd`.

```
# cat ulogd.conf
[global]
logfile="syslog"
plugin="/path/to/ulogd_inppkt_NFLOG.so"
plugin="/path/to/ulogd_raw2packet_BASE.so"
plugin="/path/to/ulogd_filter_IFINDEX.so"
plugin="/path/to/ulogd_filter_IP2STR.so"
plugin="/path/to/ulogd_filter_PRINTPKT.so"
plugin="/path/to/ulogd_output_LOGEMU.so"
stack=sal:NFLOG,sab:BASE,saif:IFINDEX,saip:IP2STR,sap:PRINTPKT,sao:LOGEMU

[sal]
group=10425

[sao]
file="/var/log/ulogd/workstation_packets.log"
sync=1

# ulogd -c ulogd.conf
```