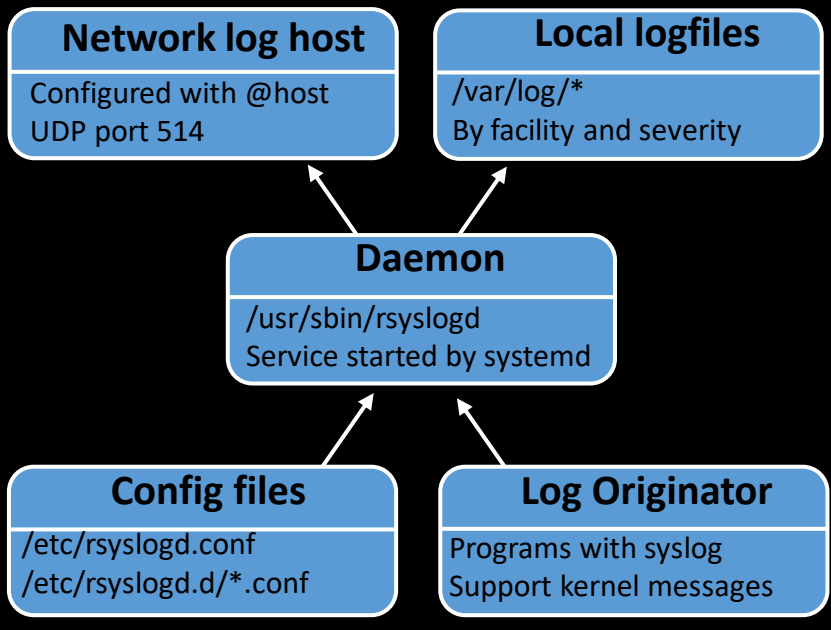
Linux Investigation through System Logs

Syslog: Syslog listens for log messages from multiple sources, such as packets arriving over network sockets (UDP port 514), local named pipes, or syslog library calls.

Syslog Architecture:



Eight severity levels with the short or alternate names and description:

0 emergency (emerg or panic):
system is unusable
1 alert (alert):
action must be taken immediately
2 critical (crit):
critical conditions
3 error (err):
error conditions
4 warning (warn):
warning conditions
5 notice (notice):
normal but significant condition
6 informational (info):
informational messages
7 debug (debug):
debug-level messages

Common syslog daemon configuration file locations:

- /etc/syslog.conf
- /etc/rsyslog.conf
- /etc/rsyslog.d/*.conf
- /etc/syslogng.conf
- /etc/syslogng/*

****All are plaintext files, can be read by any text editor.**

Example of a syslog configuration file having two field selector and action:

Sector	Action
#*.debug	/var/log/debug
kern.*	/var/log/kern.log
mail.err	/var/log/mail.err
*.info	@loghost

****The selector field is composed of the facility and severity (separated by a dot).**
The action field defines the destination or other action taken when logs match the selector

Use of logger tool for generating syslog messages :

$ logger -p auth.emerg “We have been hacked!“

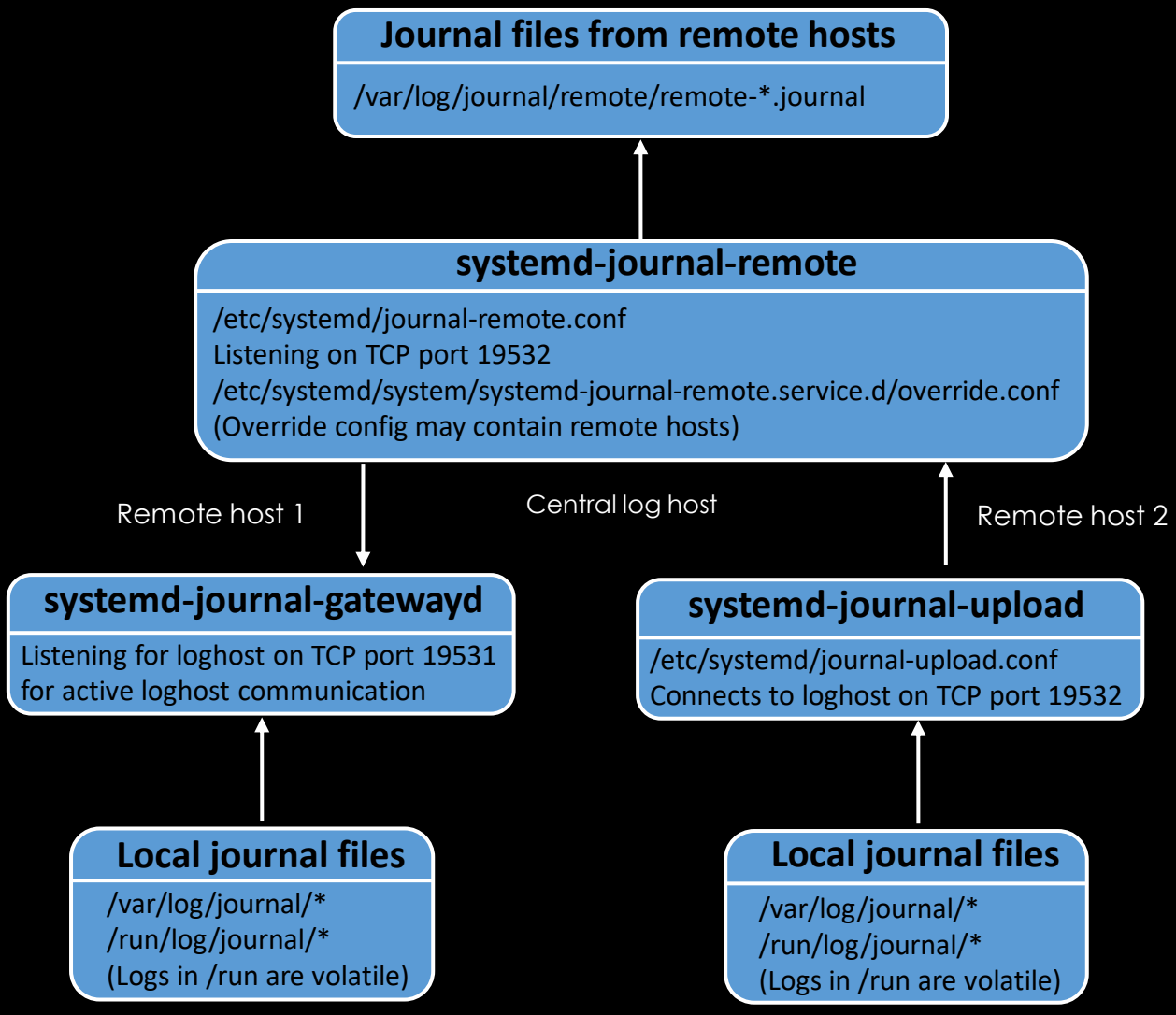
Message looks like:

```
21:56:32.635903 IP (tos 0x0, ttl 64, id 12483, offset 0, flags [DF],
proto UDP (17), length 80)
pc1.42661 > loghost.syslog: SYSLOG, length: 52
Facility auth (4), Severity emergency (0)
Msg: Sep 2 21:56:32 pc1 sam: We have been hacked!
```

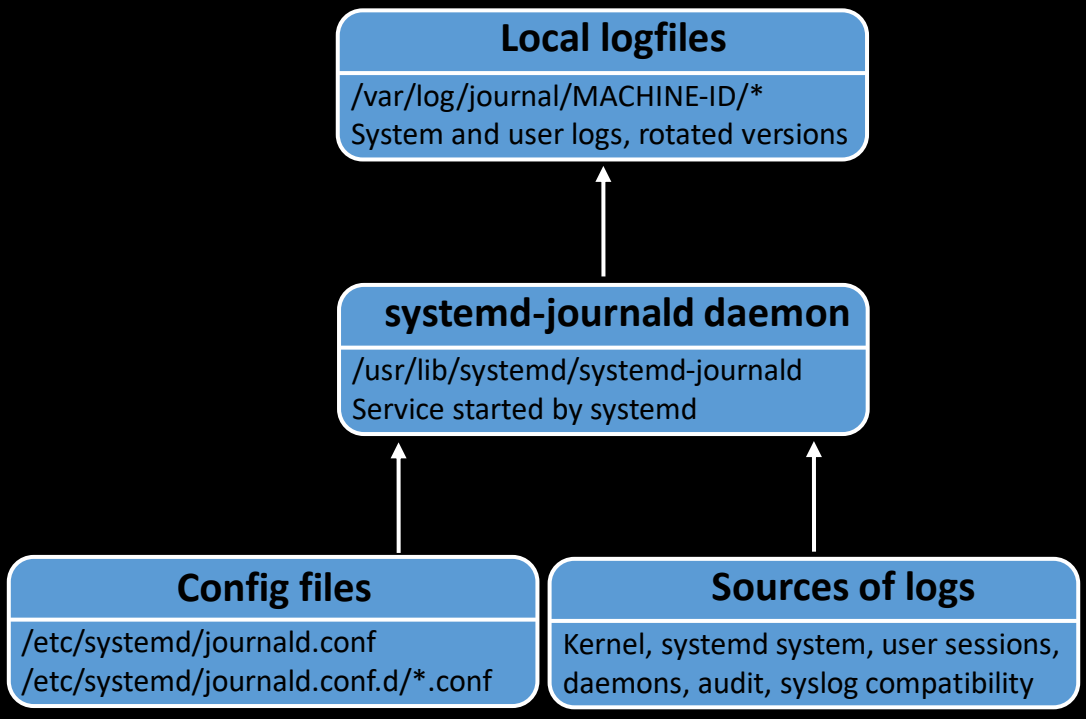
Caution when analyzing syslog messages:

- Programs can generate messages with any facility and severity they want.
- Syslog messages sent over a network are stateless, unencrypted, and based on UDP, which means they can be spoofed or modified in transit.
- Syslog does not detect or manage dropped packets. If too many messages are sent or the network is unstable, some messages may go missing, and logs can be incomplete.
- Textbased logfiles can be maliciously manipulated or deleted.

Systemd Journal Networking Architecture:



Systemd Journal Daemon:



Command for Analysis of Journal File Contents:

$ journalctl --file system.journal –header
$ journalctl --file system.journal
$ journalctl --file system.journal -o verbose
$ journalctl --file system.journal -o json > system.journal.json
$ journalctl --file system.journal -o export > system.journal.export
$ journalctl --file system.journal _SYSTEMD_UNIT=sshd.service
$ journalctl --file user-1000.journal _TRANSPORT=stdout
$ journalctl --file system.journal --verify
$ journalctl --file user-1002.journal –verify
$ journalctl --directory ./evidence -S 2022-12-01 -U 2022-12-31
$ journalctl --file ./evidence/system.journal -S "2022-11-05 08:00:00" -U "2022-11-05 09:00:00“

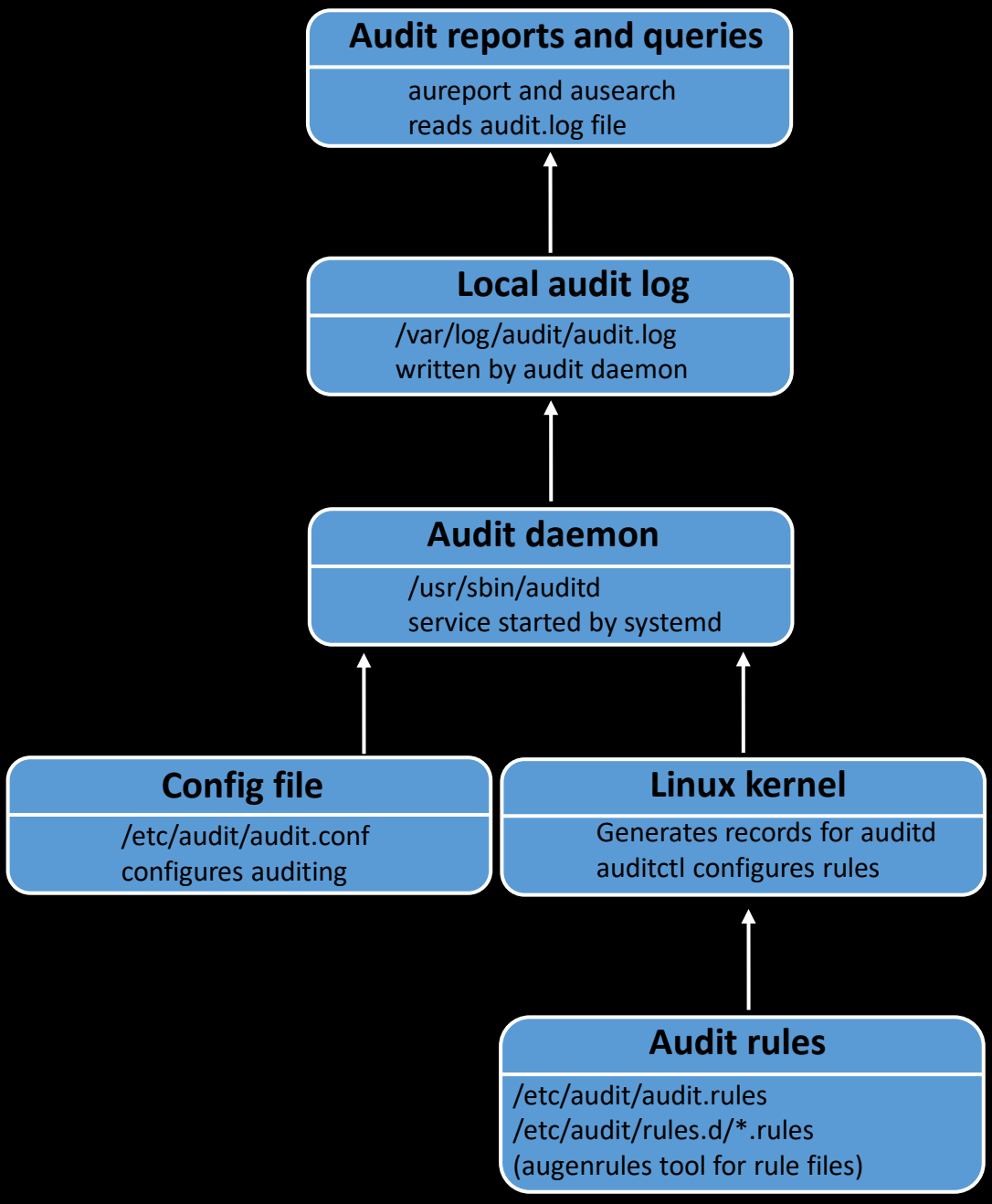
Command for Independent Server Application Logs:

$ ls -gfo /usr/bin/vi /etc/alternatives/vi /usr/bin/vim.basic
$ cat /var/log/alternatives.log

Command for Independent User Application Logs:

$ cat ~/.config/Jitsi\ Meet/logs/main.log
$ cat ~/.zoom/logs/zoom_stdout_stderr.log
$ cat ~/.cache/libvirt/qemu/log/pc1.log

Linux Auditing System:



Audit report using Ausearch Tool:

$ ausearch --input audit.log
$ ausearch --input audit.log --format text
$ aureport --input audit.log –login
$ aureport --input audit.log --start 2021-11-08 09:00:00 --end 2021-11-08 09:59:59