Logging

Logging

- Why do we care?
- What possible information can we get from a logfile?
- Where are logfiles stored?
- How can we view logfiles?

Why do we care?

A logfile can contain useful information about a process, service, or general state of the machine. A system administrator needs to be able to interpret the messages within a logfile, especially when things are not working correctly.

As we continue throughout the semester, we will utilize several log files.

What possible information can we get from a logfile

Let's look at some examples:

```
    Jun 18 06:26:11 ssh sshd[29658]: pam_sss(sshd:auth): authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=222.85.188.84 user=minecraft
    Jun 18 06:26:11 ssh sshd[29658]: pam_sss(sshd:auth): received for user minecraft: 10 (User not known
to the underlying authentication module)
    Jun 18 06:26:11 ssh sshd[29662]: pam_unix(sshd:auth): check pass; user unknown
    Jun 18 06:26:11 ssh sshd[29662]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=222.85.188.84
```

Taken from `/var/log/auth.log`.

More examples

```
    Jun 20 11:30:49 ns1 named[6603]: client @0x7f2ce47531c0 144.38.192.80#56573
(gitlab2450.cs.dixie.edu.cs.dixie.edu): query failed (SERVFAIL) for gitlab2450.cs.dixie.edu.cs.dixie.edu
/IN/A at ../../../bin/named/query.c:6885
```

From the syslog.

More examples

```
    2023-06-20 09:02:20 status half-configured google-chrome-stable:amd64 114.0.5735.133-1
    2023-06-20 09:02:20 status installed google-chrome-stable:amd64 114.0.5735.133-1
    2023-06-20 09:02:20 configure libgjs0g:amd64 1.72.2-0ubuntu2 <none>
```

From `/var/log/dpkg.log`.

More examples

Here are two lines from another log. Any guesses on what it is logging?

```
    144.38.192.200 - - [20/Jun/2023:06:34:49 -0600] "GET /style/plain.css HTTP/1.1" 200 1176
"https://computing.utahtech.edu/it/4200/sources/xenial_network_how_to.html" "Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/602.1 (KHTML, like Gecko) wkhtmltopdf Version/9.0 Safari/602.1"
    66.249.66.70 - - [20/Jun/2023:06:34:48 -0600] "GET /it/4200/sources/xenial_network_how_to.pdf
HTTP/1.1" 200 28291 "-" "Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/114.0.5735.90 Mobile Safari/537.36 (compatible; Googlebot/2.1;
```

```
+http://www.google.com/bot.html)"
```

Some explanation of logfiles

As you can see, each log file might have a unique or different format, you can even customize or modify a lot of them, however one comminality that most logfiles have is a timestamp.

- Why would that be important?
- What could a system administrator do with date/time information?

Where are logfiles stored?

For the most part, on an Ubuntu system they are stored in:

- `/var/log` directory

Some applications like to create their own directory like:

- `/var/log/apache2`

How to view logfiles?

Hmmmm... They are plaintext files, so this may seem rather obvious, BUT since logfiles can sometimes be HUGE, being able to search through a logfile is a critical skill.

Commands that I find most useful:

- cat, awk, head, tail, grep
  - For example: `cat syslog | grep -i cron`
- Some logfiles are automatically compressed to save space after a while (this is also configurable). They will usually end with a gz extension. You could ungzip it to view, but you could also view with `zcat`.
- If you have loaded the logfile with `less`, you can search for strings using the `/` forward slash.

The syslog

- Pretty much the most important logfile
- We will just say that a lot of services and processes write their log information into this file (NOT ALL services, but a lot of them), like:
  - CRON
  - Network Manager
  - systemd
  - kernel
- Let's say that pretty much any process or service that is built-in to Ubuntu will log messages there.
- Send your own messages there with the `logger` command.

The troubleshooting process

As we install and configure various services, the logfile can provide valuable information to you, but you must be patient and persist. Here are some general steps for debugging:

- restart the service
- if the service has problems
- check the appropriate logfile
  - Filtering your way through the information in the logfile will likely be necessary, you may do this by grepping for a timestamp, or an error code
  - Many times the logfile will tell you what the problem is, but they won't tell you how to solve it.
- solve the problem
- restart the service