Hands On Lab Unit 4

MICS-252, Fall 2024

Prepared by: Karl-Johan Westhoff

email: kjwesthoff@berkeley.edu

UC Berkleley School of Information

MICS Course 252 Fall 2024 (Kristy Westphal)

1 Log Analysis

I used parts of the "Six Step Analysis" we discussed in class, which I have personalized a follows[1] (when has overkill ever stopped us, the assignment instruction said "just look at the logs and explain"):

1. *Build knowledge, ask around for help, invite peoples inputs and opinions
    - Gather Facts
    - mobilize help, informally and formally
2. *Systematize facts, and communicate
    - This is where post-it sessions go
    - categorize validity inputs based on facts, including what you don't know
3. *Determine biases
    - Including your own!
4. *Translate Jargon (reference to Richard Feynman[2])
5. *Ensure that test platform works
    - Calibrate models and remove biases[3]
6. *Assure you get the most direct answer
    - Simplify your reasoning/models as much as possible in your reporting

[1] Where each is applied, it is denoted as 'No'*

[2] Richard Feynman was an american theoretical physicist often cited for: "If you can't explain something in simple terms, you don't understand it."[1] [2]

[3] Here goes another Feynman Quote: "You must not fool yourself — and you are the easiest person to fool."[3]

2 Observations (1*,2*)

I mean 'Facts' but that may hold some stakeholders back from speaking out or providing their input. In an investigation want to end up as close to factual objectivity as possible but in the beginning, I want everything on the table also the strongly opinionated inputs, so everything can undergo factual analysis. It is the 'nothing is wrong phase' of the investigation

There are 3 files, two named 'log' and one named messages, there may be more but this is what we have now.

- `log file 1.txt`
- `log file 2.txt`
- `messages`

2.1 log file 1.txt:

ASCII text, with CRLF line terminators, 68 lines, each is a log entry (example in Figure 1 below):

```
Aug  1 18:27:45 knight sshd[20325]: Illegal user test from 218.49.183.17
Aug  1 18:27:46 knight sshd[20325]: Failed password for illegal user test from 218.49.183.17 port 48849 ssh2
Aug  1 18:27:46 knight sshd[20325]: error: Could not get shadow information for NOUSER
Aug  1 18:27:48 knight sshd[20327]: Illegal user guest from 218.49.183.17
Aug  1 18:27:49 knight sshd[20327]: Failed password for illegal user guest from 218.49.183.17 port 49090 ssh2
Aug  1 18:27:49 knight sshd[20327]: error: Could not get shadow information for NOUSER
Aug  1 18:27:52 knight sshd[20329]: Failed password for admin from 218.49.183.17 port 49266 ssh2
Aug  1 18:27:56 knight sshd[20331]: Failed password for admin from 218.49.183.17 port 49468 ssh2
Aug  1 18:27:58 knight sshd[20334]: Illegal user user from 218.49.183.17
Aug  1 18:27:59 knight sshd[20334]: Failed password for illegal user user from 218.49.183.17 port 49680 ssh2
Aug  1 18:27:59 knight sshd[20334]: error: Could not get shadow information for NOUSER
```

Figure 1: *Snippet from "log file 1.txt"*

Log columns/data: The log entries/data columns are:

`'Month' 'day' 'HH:MM:SS' 'knight' 'sshd[nnnnn]': 'message string'`

- The dates are all August 1st, time 18:27:45 to 18:36:28
- The `knight` sting is present in all log entries
- The `sshd[number]` is related to an OpenSSH server process with the number being the process number on an OpenSSH server
- The `message` string varies, sometimes containing a username and and ip address/port number.
  - The message string can be further subdivided into:`'Action', 'IP', 'Port'`
  - The only IP address occurring is `"218.49.183.17"`

- `"Failed Password"` string occurs in 36 entries
- The failed password attempts occur with varying usernames: `'test'`, `'guest'`, `'admin'`, `'root'` on varying ephemeral port numbers (ranging 39604 to 54423)
- `"error: Could not get shadow information for NOUSER"` occurs in 16 entries

2.2 log file 2.txt

ASCII text, with CRLF line terminators 26 lines, each is a log entry, see snippet in Figure 2 below:

```
Jul 14 04:44:46 opala proftpd[30812] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): mod_delay/0.5: delaying for 14871 usecs
Jul 14 04:44:46 opala proftpd[30813] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): no such user 'guest'
Jul 14 04:44:46 opala proftpd[30813] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): USER guest: no such user found from sieapp.ufpel.edu.br [200.17.161.73] to 192.168.2.5:21
Jul 14 04:44:46 opala proftpd[30813] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): mod_delay/0.5: delaying for 86 usecs
Jul 14 04:44:46 opala proftpd[30815] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): FTP session opened.
Jul 14 04:44:46 opala proftpd[30814] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): no such user 'guest'
Jul 14 04:44:46 opala proftpd[30814] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): USER guest: no such user found from sieapp.ufpel.edu.br [200.17.161.73] to 192.168.2.5:21
```

Figure 2: *Snippet from "log file 2.txt"*

Log columns/data: The log entries/data columns are:

`Month day H:MM:SS opala proftpd[nnnnn] opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73]): message string`

- The dates are all July 14th, time 04:44:46 to 04:44:47
- The `opala` sting is present in all log entries.
- The `proftpd[number]` indicates a process on a FTP server from the open source ProFTP server
- The `opala.xxxxxx.edu.br (sieapp.ufpel.edu.br[200.17.161.73])` is the same for all entries, noting the .edu.br and consistent ip address[200.17.161.73]
- The `message` string varies, sometimes containing a username and and ip address/port number.
  - FTP sessions are opened and closed,
  - The username `'guest'` is attempted repeatedly for login
  - The logging time interval is 1s, multiple attempts are done within this short time
  - mod delay 0.5: delaying for XXX usecs occurs 8 times

2.3 messages

Non-ISO extended-ASCII text, with very long lines (1025), with LF, NEL line terminators. 948 lines, each representing an entry

Logs entries (example in Figure 3 below):

```
Sep  8 13:43:03 brjdev syslogd 1.3-3: restart.
Sep  8 13:43:03 brjdev syslogd 1.3-3: restart.
Sep  8 13:43:03 brjdev syslogd 1.3-3: restart.
Sep  8 13:44:04 brjdev anacron[803]: Job `cron.daily' terminated
Sep  8 13:48:02 brjdev anacron[803]: Job `cron.weekly' started
Sep  8 13:48:02 brjdev anacron[1070]: Updated timestamp for job `cron.weekly' to 2003-09-08
Sep  8 13:48:35 brjdev anacron[803]: Job `cron.weekly' terminated
Sep  8 13:48:35 brjdev anacron[803]: Normal exit (2 jobs run)
Sep  8 13:49:10 brjdev PAM_unix[4171]: authentication failure; (uid=0) -> matt for system-auth service
Sep  8 13:49:10 brjdev login[4171]: FAILED LOGIN 1 FROM corp FOR matt, Authentication failure
Sep  8 13:49:11 brjdev PAM_unix[4171]: bad username []
Sep  8 13:49:11 brjdev login[4171]: FAILED LOGIN 2 FROM corp FOR , Authentication failure
Sep  8 13:49:12 brjdev PAM_unix[4171]: bad username []
```

Figure 3: *Snippet from "messages"*

Log columns/data: The log entries/data columns are:

`'Month' 'day' 'HH:MM:SS' 'brjdev' 'varying string': 'message string'`

- The dates are all Sep.8th, time 13:43:03 to 17:20:00
- The `brjdev` string is present in all log entries.
- The `message` string varies, sometimes containing a username and and ip address/port number.
  - there are log entries from various sources:
    - syslogd, process doing logging in endpoints (Windows, Mac, Linux)[4] in the first log entries this process is restarted 3 times (see Figure3)
    - anachron, performs periodic command scheduling of processes [5]
    - CHRON[NNNN], presumably something with chron jobs in Linux/Unix
    - pam_rhosts_auth, module provides rhost authentication services[6]
    - etc.
  - There are a majority of very long log entries, starting with: `Sep 8 HH:MM:SS brjdev SERVER[NNNN]: Dispatch_input: bad request line` and then some non ASCII characters and numbers

1

3 Biases (3*,5*)

This is a solo assignment so the biases are my own, I have identified the following, which I will need to consider in my analyses:

- I suspect there is a some spectacular breach to be identified from these logs, and i want to find it
- This is a course assignment, I think there is a lesson to be learned somewhere

4 Conclusions (6*)

I have concluded that the files are not related, the logs are from different dates (separated by months) and from different systems (SSH, FTP and various in 'messages'), and the IP addresses involved are different in each case. Below are conclusions for each log.

4.1 log file 1

- OpenSSH is attacked
- Log entries over 8 minutes and 43 seconds, with 68 log entries
- Log-in attempts with failed passwords and attempts to retrieve 'shadow information'
- different, but generic user names attempted, hereunder 'root' and 'admin'
- Attempts from only one ip addess: 218.49.183.17, but from varying remote/ephemeral ports

Hypothesis I think this is a brute force attack on an OpenSSH server, SSH gives direct terminal access on the remote host, and high privilege usernames are attempted. The number of attempts and irregular choice of ports and number of attempts on each username suggests a semi manual approach using a script but not a fully automated attack.

4.2 log file 2

- Many login attempts in short time from IP 200.17.161.73 to FTP server: 192.168.2.5 on port 21, TCP port 21 is the 'control' channel used to negotiate access etc.
- The username ued is 'guest' which sounds generic
- FTP sessions are opened and closed many times over 1 second
- The server attempts to slow down communication using 'mod_delay'

Hypothesis The log-in attempts occur over a short period indicating that it was done automatically, the use of a generic username suggests it may be part of a brute force attack. However, as the log snippet is short, it may also be a part of a DDoS attack and the server does go into mod_delay 'slowdown mode', what speaks against this is that the attack only comes from one IP. Given the size of the log snippet both brute-force and DDoS are feasible explanations.

4.3 messages

- Log entries are from different sources, relating both to internal errors (syslogd is restarted)

- The long SERVER[NNNN] entries, with 'bad request line' could be a log of malformed http requests to a server

The messages log seems to be from a SIEM system collecting logs from various places. The long malformed 'SERVER' messages could indicate and attack on a web server, possibly DDoS or something more fancy on another protocol. I would need some more information (IP addresses, protocols ports etc.) to come up with any conclusions, these can probably be retrieved somehow as the 'messages' file is a log aggregation.

To better understand the messages, they should be analyzed and sorted by type time etc. I threw the messages (disregarding the long 'SERVER' entries) into ChatGPT, see Appendix A Figure 6 and Figure 7. Of security interest, the chat GPT 'analysis' revealed:

ChatGPT: Authentication Failures 13:49:10 to 13:49:12: (see Figure 4) There are repeated login attempts for matt from a source labeled corp. Authentication fails, including attempts with a blank username. This could indicate a brute-force or automated script trying to log in to the system, as multiple failed login attempts are recorded in a very short span.

```
Sep  8 13:49:10 brjdev PAM_unix[4171]: authentication failure; (uid=0) -> matt for system-auth service
Sep  8 13:49:10 brjdev login[4171]: FAILED LOGIN 1 FROM corp FOR matt, Authentication failure
Sep  8 13:49:11 brjdev PAM_unix[4171]: bad username []
Sep  8 13:49:11 brjdev login[4171]: FAILED LOGIN 2 FROM corp FOR , Authentication failure
Sep  8 13:49:12 brjdev PAM_unix[4171]: bad username []
Sep  8 13:49:12 brjdev login[4171]: FAILED LOGIN 3 FROM corp FOR , Authentication failure
Sep  8 13:49:12 brjdev PAM_unix[4171]: bad username []
Sep  8 13:49:12 brjdev login[4171]: FAILED LOGIN SESSION FROM corp FOR , Authentication failure
```

Figure 4: *13:49:10 to 13:49:12*

Hypothesis Looking at the 'raw' logs, I am not so sure about this, it is 'only' 3 attempts and could be someone misspelling username etc. 3 attempts in 2 seconds I assume that to be humanly possible.

ChatGPT: Failed Remote Shell (RSH) Attempts 14:55:41 to 14:59:22:(see Figure 5) Several attempts to use the rsh service are made from root@94.90.84.93 as user lpd. All these are denied due to "access not allowed." These multiple attempts indicate an attempt to exploit weak authentication methods like rsh, which is less secure than SSH.

Hypothesis There are 5 attempts over 2 minutes, this could be someone poking around with rsh, or an administrator forgetting a password, would need to check up on who 94.90.84.93 is.

This log analysis has identified items for further investigation.

```
Sep  8 14:55:41 brjdev pam_rhosts_auth[5266]: denied to root@94.90.84.93 as lpd: access not allowed
Sep  8 14:55:41 brjdev in.rshd[5266]: rsh denied to root@94.90.84.93 as lpd: Permission denied.
Sep  8 14:55:41 brjdev in.rshd[5266]: rsh command was '/bin/sh -i'
Sep  8 14:56:04 brjdev pam_rhosts_auth[5267]: denied to root@94.90.84.93 as lpd: access not allowed
Sep  8 14:56:04 brjdev in.rshd[5267]: rsh denied to root@94.90.84.93 as lpd: Permission denied.
Sep  8 14:56:04 brjdev in.rshd[5267]: rsh command was 'ls'
Sep  8 14:57:29 brjdev pam_rhosts_auth[5270]: denied to root@94.90.84.93 as lpd: access not allowed
Sep  8 14:57:29 brjdev in.rshd[5270]: rsh denied to root@94.90.84.93 as lpd: Permission denied.
Sep  8 14:57:29 brjdev in.rshd[5270]: rsh command was 'ls'
Sep  8 14:57:51 brjdev sshd[5271]: Failed password for ROOT from 94.90.84.93 port 2093 ssh2
Sep  8 14:57:57 brjdev last message repeated 2 times
Sep  8 14:57:57 brjdev sshd[5271]: Connection closed by 94.90.84.93
Sep  8 14:57:57 brjdev PAM_unix[5271]: (system-auth) session closed for user lpd
Sep  8 14:59:22 brjdev pam_rhosts_auth[5273]: denied to root@94.90.84.93 as lpd: access not allowed
Sep  8 14:59:22 brjdev in.rshd[5273]: rsh denied to root@94.90.84.93 as lpd: Permission denied.
Sep  8 14:59:22 brjdev in.rshd[5273]: rsh command was 'ls'
```

Figure 5: *14:55:41 to 14:59:22*

In "retrospect" (after being revealed in class) the "messages" log file is obviously a buffer overflow/heap groomin attack on a webserver with all the long log messages in the logfile. The attacker also manages to log into the system.

References

[1] *Richard Feynman, Medium on explainin things*. https://medium.com/@markolson_25124/for-richard-feynman-if-you-cant-explain-something-in-simple-terms-you-don-t-understand-it-408e9e743668. Accessed: 2024-20-13.

[2] *Richard Feynman, Wikipedia*. https://www.snort.org/https://en.wikipedia.org/wiki/Richard_Feynman. Accessed: 2024-20-13.

[3] *Richard Feynman, Medium, Ilsoe*. https://ilsoe.medium.com/you-must-not-fool-yourself-and-you-are-the-easiest-person-to-fool-82da1ddeaa6b. Accessed: 2024-20-13.

[4] *syslog*. https://en.wikipedia.org/wiki/Syslog. Accessed: 2024-20-13.

[5] *anacron*. https://en.wikipedia.org/wiki/Anacron. Accessed: 2024-20-13.

[6] *pam rhosts auth*. https://www.ibm.com/docs/en/aix/7.2?topic=files-pam-rhosts-auth-module. Accessed: 2024-20-13.

Appendices

A ChatGPT analysis of the messages file

1. System Restart and CRON Jobs

- **13:43:03**: `syslogd` is restarted three times. This might be part of regular system maintenance or could indicate an issue causing repeated restarts.
- **CRON Jobs**: Regular jobs are being executed (e.g., `/sbin/rmmod -as` for removing unused modules). This happens frequently every 10 minutes, which could be a system optimization or clean-up task.

2. Anacron Jobs

- **13:44:04 to 13:48:35**: `anacron` completes running the daily and weekly scheduled jobs. These jobs are part of system maintenance routines.

3. Authentication Failures

- **13:49:10 to 13:49:12**: There are repeated login attempts for `matt` from a source labeled `corp`. Authentication fails, including attempts with a blank username. This could indicate a brute-force or automated script trying to log in to the system, as multiple failed login attempts are recorded in a very short span.

4. Network Connection Issues

- **14:02:40 to 14:02:42**: There are errors related to `rlogind`, `rshd`, and `identd`, where connections to remote hosts are not working properly. The error "Can't get peer name of remote host" and "Connection reset by peer" suggests connectivity issues or possibly network disruptions.
- **14:02:42**: `sshd` logs an error indicating that the ident string wasn't received from `94.90.84.93`, a suspicious IP address. This might indicate an attempt to connect without proper identification, which could be part of a malicious action.

5. User Logins

- **14:09:11**: `richard` logs in from IP `102.60.21.97` on terminal `pts/0`.
- **14:12:05**: `curtis` logs in on `tty2`.
- **14:14:24**: `matt` successfully logs in on terminal `pts/1` from `corp`, despite earlier failed attempts. This could be a legitimate user or a compromised account.

6. Further Connection Issues

- **14:20:40 to 14:20:41**: Similar to the earlier errors, `rshd`, `rlogind`, `identd`, and `sshd` again report connection resets and issues with peer name identification. This could suggest network instability, improper remote access attempts, or a malicious actor attempting to connect improperly.

7. Potential Server Issue

- **14:35:55**: The log shows `SERVER[4303]: Dispatch_input: bad request line`. This might indicate a malformed request sent to the server, possibly from an application or a misconfigured client.

Figure 6: *ChatGPT analysis of logs before the large chunk of 'SERVER' logs*

1. SSH Activity

- **14:37:52 & 15:37:52**: New 768-bit RSA key generated. These are part of SSH key generation, likely for encryption or a system update.
- **14:57:51**: A failed password attempt from IP `94.90.84.93` for the `root` user via SSH. The connection is eventually closed.
- **15:00:01**: A successful password-based SSH login for the `root` user from the same IP (`94.90.84.93`). This could indicate a brute force attempt followed by a successful login.

2. Failed Remote Shell (RSH) Attempts

- **14:55:41 to 14:59:22**: Several attempts to use the `rsh` service are made from `root@94.90.84.93` as user `lpd`. All these are denied due to "access not allowed." These multiple attempts indicate an attempt to exploit weak authentication methods like `rsh`, which is less secure than SSH.

3. CRON Jobs

- Regular CRON jobs running the command `/sbin/rmmod -as`. This is likely a script or task removing unused modules at regular intervals.
- **16:01:00**: The system runs scheduled tasks from `/etc/cron.hourly`.

4. PAM (Pluggable Authentication Module) Logs

- **Multiple logins and logouts** for various users (e.g., `richard`, `matt`, `curtis`). These sessions open and close at various times.
- The `pam_rhosts_auth` module logs frequent denials for the `root` user from `94.90.84.93`, which indicates failed attempts to authenticate via rhosts authentication.

5. System Errors

- **Rhnsd (Red Hat Network)**: There are repeated errors from the Red Hat Network daemon (`rhnsd`), indicating that the system is unable to read the system ID. This might be due to configuration or subscription issues.
- **Kernel Errors**: **16:46:33 to 16:56:27**: Continuous `lost interrupt` messages related to device `hdc`, a CD-ROM drive. It seems there was an issue with the hardware, but the system eventually resets the IDE and reports success.

6. Potential Security Concerns

- **IP** `94.90.84.93`: Multiple unauthorized attempts to log in as `root` through `rsh` and SSH. Eventually, a successful login occurs via SSH. This could suggest a compromised system or an active attack, particularly given the sensitive nature of `root` access.

Figure 7: *ChatGPT analysis of logs after the large chunk of 'SERVER' logs*