



Artifact Genome Project



This material is based upon work supported by the National Science Foundation under Grant No. 1900210. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1. Assignment Type

Learn By Doing

2. Assignment Name

Web Forensics: Using Logs to investigate SQL Injection Attack

3. Assignment Overview

Despite the advantage that web applications have, they frequently become targets for attackers because of poor coding or inadequate security monitoring. Attackers attempt to take advantage of coding flaws and access database contents in order to obtain sensitive data. When such web application attacks take place, forensic investigation enters the picture. Web application forensics entails forensic investigation of web applications and their data, such as logs, to track down the attack, identify its source, ascertain how it spread, as well as the tools used and the individuals responsible for carrying it out. A server's log file is one of the most important pieces of data it offers. A log file keeps track of all the activities that happen while a service or application is running. Log files give us a clear picture of a server's activities as well as vital details like when, how, and by whom a server is being used. This information helps forensic investigators unfold the chain of events that led to a malicious activity. This module discusses the steps of analyzing log files to find digital evidence of SQL Injection on a compromised web application running on Apache web server and MySQL 8.0.28 database in an Ubuntu 20.04.03 operating system environment. This assignment will provide a deep understanding of how defenders can detect the use of SQLMAP tool to exploit SQL injection flaws.

4. Learning Objectives

- To find digital evidence of SQL Injection attack
- To detect that a MySQL database records has been extracted
- To understand the commands used in Linux filesystem

5. Tools Needed

- Epoch Converter
- Oracle VirtualBox 6.1
- Ubuntu 20.04.3

6. Artifact Tags

Demo Website running on Ubuntu Operating System - Open Virtual Application file (.ova)

7. Task 1: Artifact Investigation

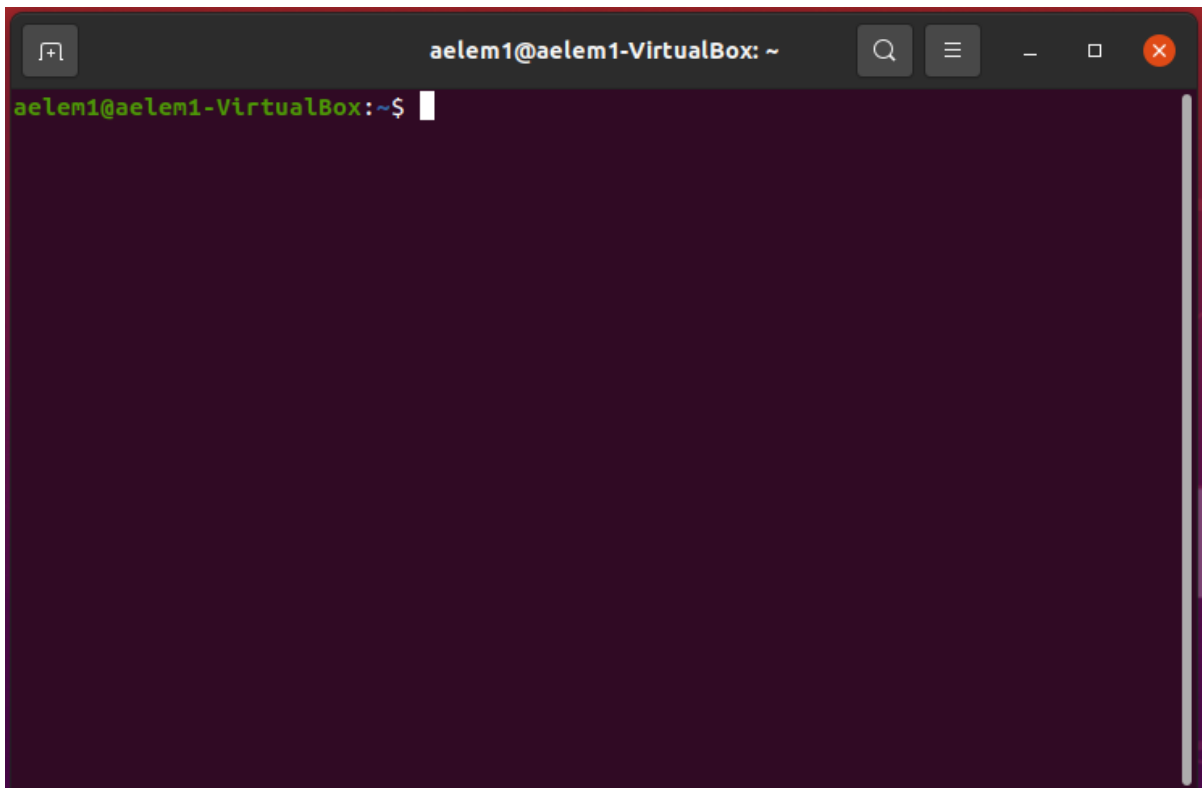
Click on the Artifact Tag above or search the artifact names within AGP. Read through each artifact and gather an understanding of all the presented information. You may view the download files through the artifact view, or if a different view is desired, you may use the download files. Review the information. Once you have a basic understanding of the information these artifacts are presenting, continue with Task 2.

8. Task 2: Load the demo website in Oracle VirtualBox

- If you do not have VirtualBox already installed please use the link above to install it.
 - Once VirtualBox is installed click on the artifact and download it.
 - After installing VirtualBox, open it and import the artifact which contains the compromised demo website.
 - Load the demo website by entering the URL - "http://localhost/beunique" in the browser.
 - The Ubuntu operating system password is sqlinjection\$12&
-]

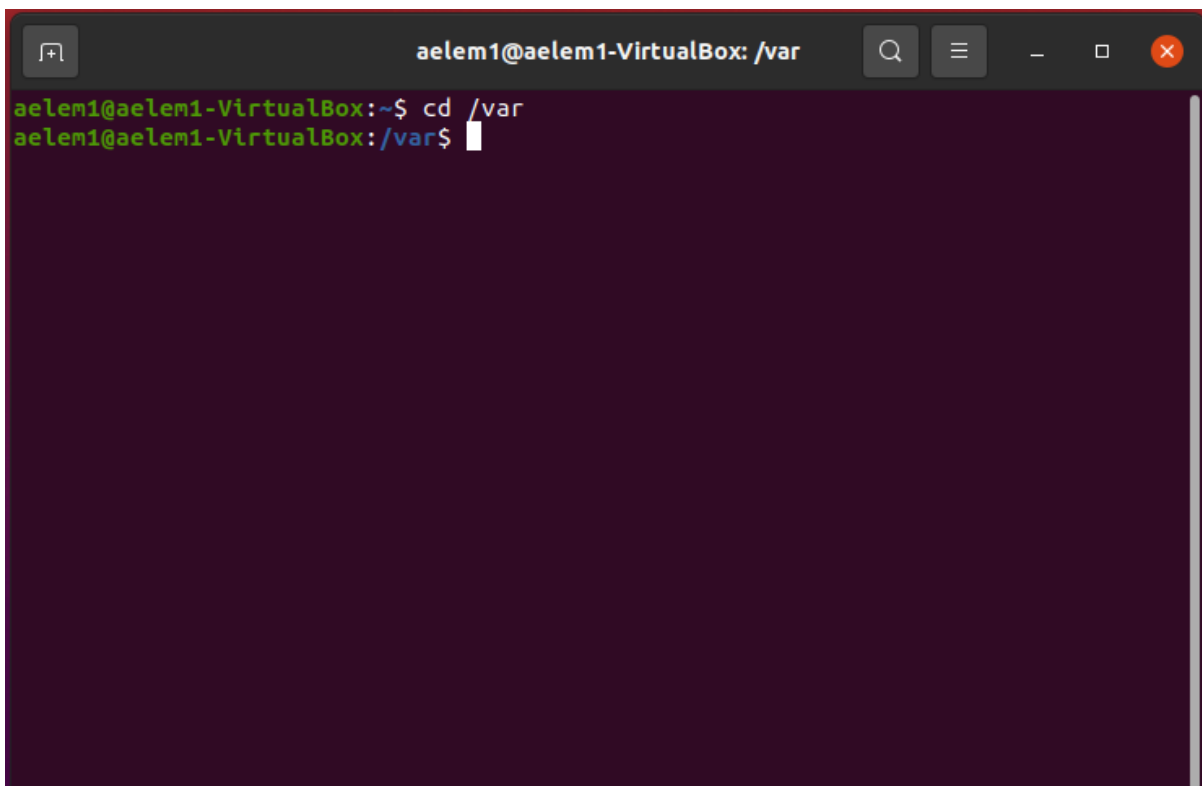
9. Task 3: Examine Apache access log files

- The Apache access logs are text files that include information about all the requests processed by the Apache server. You can expect to find information like the time of the request, the requested resource, the response code, time it took to respond, and the IP address used to request the data.
- The Apache access logs are located in the directory: var/log/apache2
- Open the terminal in the Ubuntu VM.
- Here is what the output might look like:



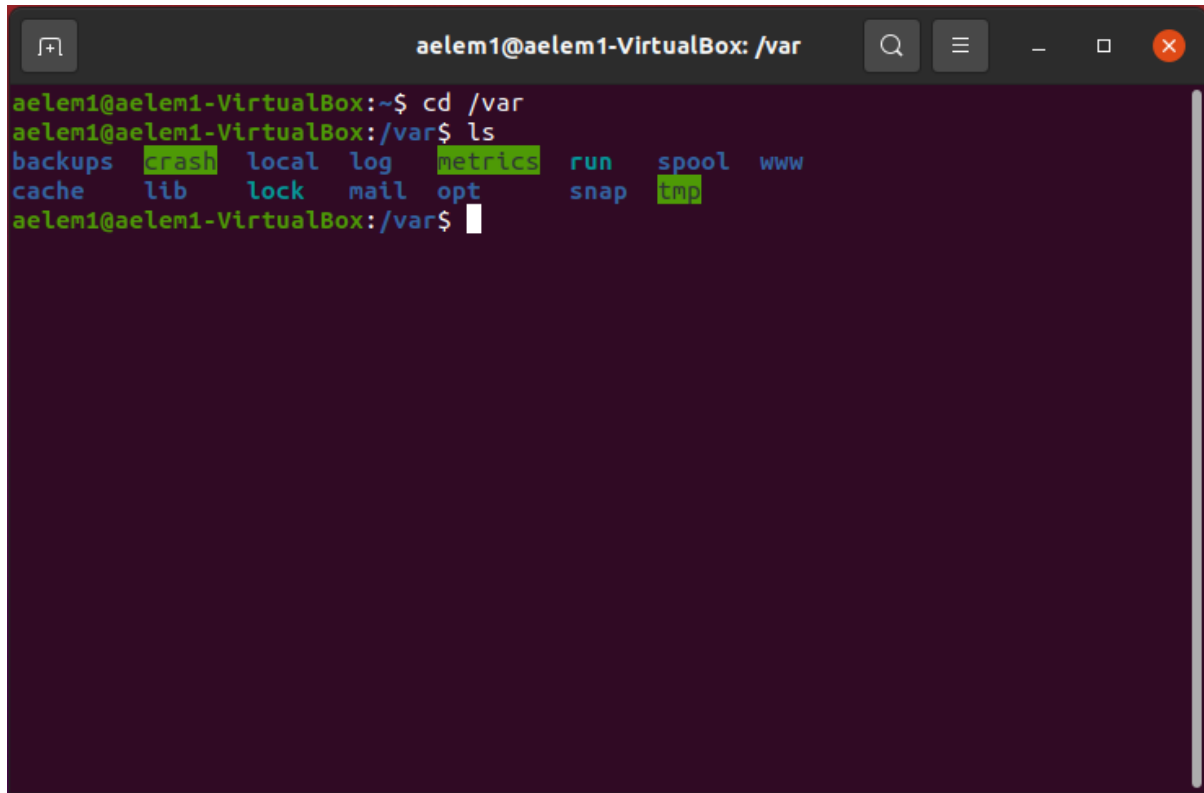
Ubuntu Terminal

- Run the command: `cd /var`
- Here is what the output might look like:



changing directory to var

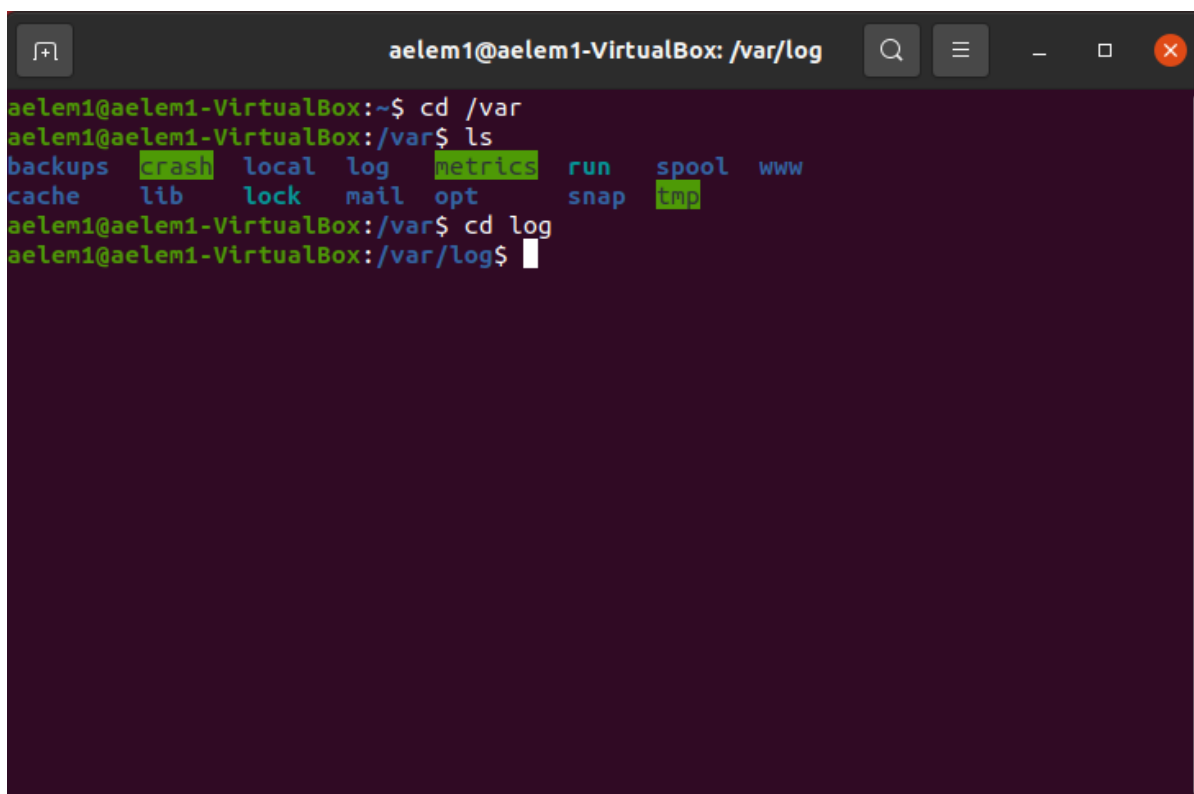
- Run the command: `ls` to lists files and directories within the `var` directory
- Here is what the output might look like:



```
aelem1@aelem1-VirtualBox: /var
aelem1@aelem1-VirtualBox:~$ cd /var
aelem1@aelem1-VirtualBox:/var$ ls
backups  crash  local  log  metrics  run  spool  www
cache    lib    lock   mail  opt      snap  tmp
```

Listing files and directories within `var`

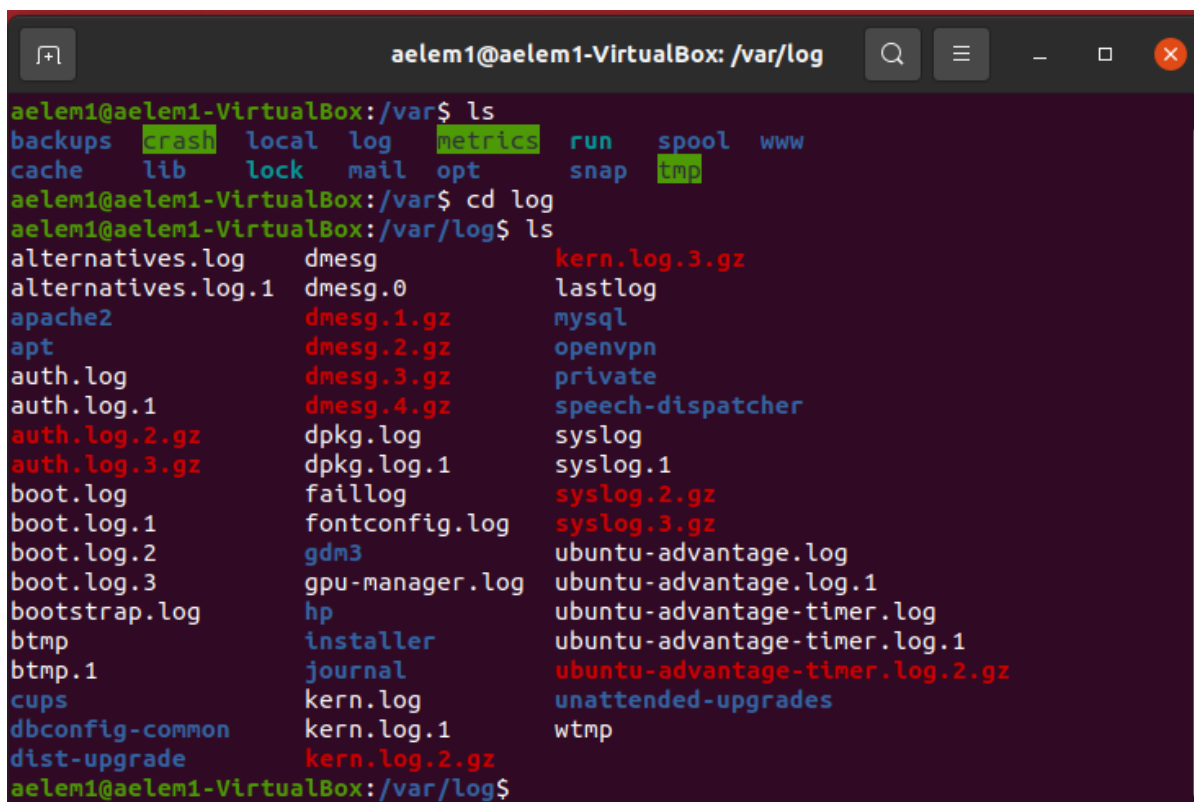
- Run the command: `cd log`
- Here is what the output might look like:



```
aelem1@aelem1-VirtualBox: /var/log
aelem1@aelem1-VirtualBox:~$ cd /var
aelem1@aelem1-VirtualBox:/var$ ls
backups  crash  local  log  metrics  run  spool  www
cache    lib    lock   mail  opt      snap  tmp
aelem1@aelem1-VirtualBox:/var$ cd log
aelem1@aelem1-VirtualBox:/var/log$
```

changing directory to log

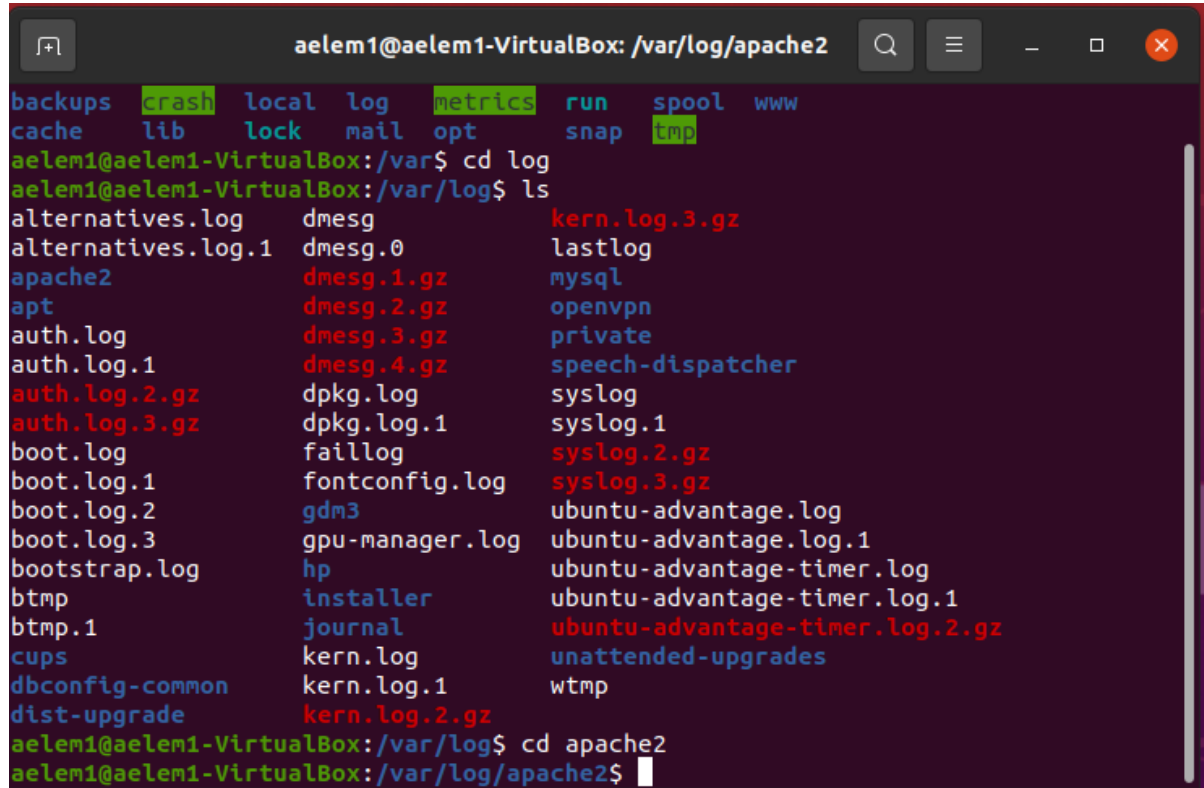
- Run the command: `ls` to lists files and directories within the log directory
- Here is what the output might look like:



```
aelem1@aelem1-VirtualBox: /var/log
aelem1@aelem1-VirtualBox:/var$ ls
backups  crash  local  log  metrics  run  spool  www
cache    lib    lock   mail  opt      snap  tmp
aelem1@aelem1-VirtualBox:/var$ cd log
aelem1@aelem1-VirtualBox:/var/log$ ls
alternatives.log      dmesg                kern.log.3.gz
alternatives.log.1    dmesg.0              lastlog
apache2               dmesg.1.gz           mysql
apt                   dmesg.2.gz           openvpn
auth.log               dmesg.3.gz           private
auth.log.1            dmesg.4.gz           speech-dispatcher
auth.log.2.gz          dpkg.log             syslog
auth.log.3.gz          dpkg.log.1           syslog.1
boot.log               faillog              syslog.2.gz
boot.log.1             fontconfig.log        syslog.3.gz
boot.log.2             gdm3                 ubuntu-advantage.log
boot.log.3             gpu-manager.log       ubuntu-advantage.log.1
bootstrap.log          hp                   ubuntu-advantage-timer.log
btmtp                  installer            ubuntu-advantage-timer.log.1
btmtp.1                journal              ubuntu-advantage-timer.log.2.gz
cups                   kern.log             unattended-upgrades
dbconfig-common        kern.log.1           wtmp
dist-upgrade           kern.log.2.gz
aelem1@aelem1-VirtualBox:/var/log$
```

Listing files and directories within log

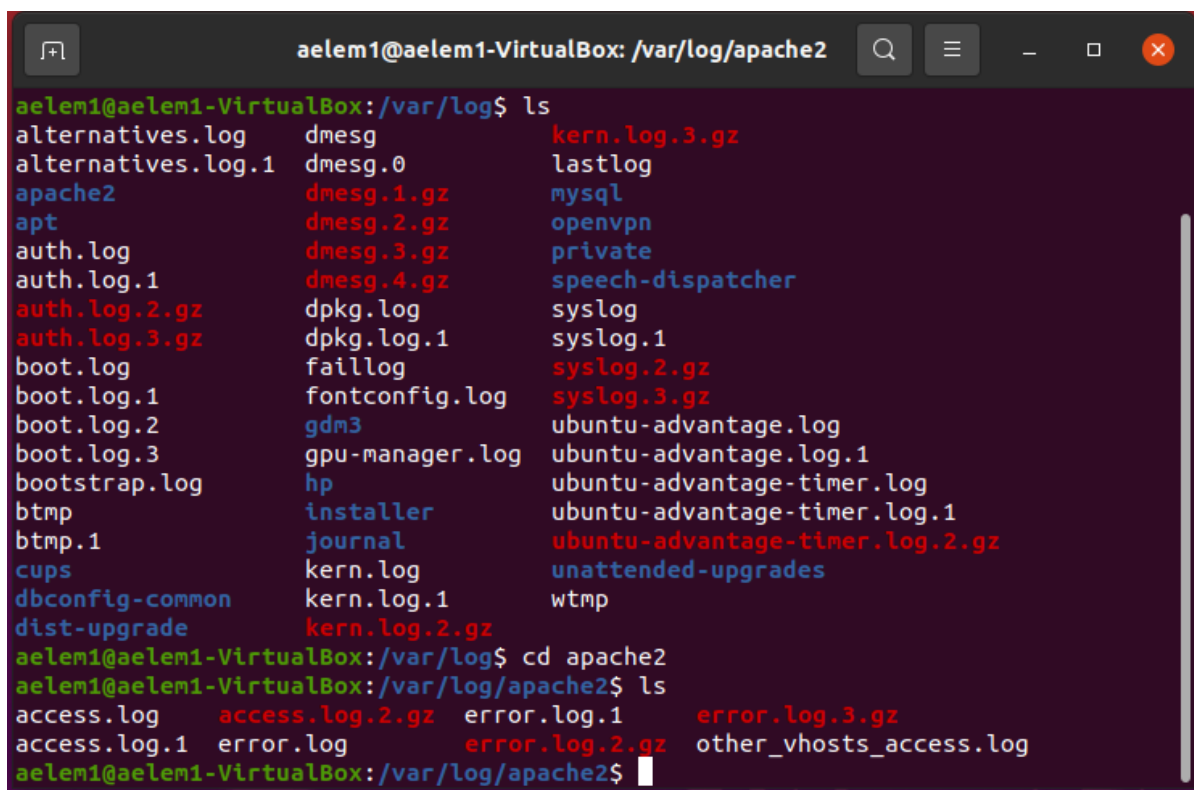
- Run the command: `cd apache2`
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/apache2' showing a directory listing of /var/log. The prompt is 'aelem1@aelem1-VirtualBox:/var\$' and the command 'cd log' has been executed. The prompt is now 'aelem1@aelem1-VirtualBox:/var/log\$' and the command 'ls' has been executed. The output is a long list of files and directories, including 'backups', 'cache', 'crash', 'lib', 'local', 'lock', 'log', 'mail', 'metrics', 'opt', 'run', 'snap', 'spool', 'tmp', 'www', 'alternatives.log', 'dmesg', 'kern.log.3.gz', 'lastlog', 'alternatives.log.1', 'dmesg.0', 'mysql', 'apache2', 'dmesg.1.gz', 'openvpn', 'apt', 'dmesg.2.gz', 'private', 'auth.log', 'dmesg.3.gz', 'speech-dispatcher', 'auth.log.1', 'dmesg.4.gz', 'syslog', 'auth.log.2.gz', 'dpkg.log', 'syslog.1', 'auth.log.3.gz', 'dpkg.log.1', 'syslog.2.gz', 'boot.log', 'faillog', 'syslog.3.gz', 'boot.log.1', 'fontconfig.log', 'ubuntu-advantage.log', 'boot.log.2', 'gdm3', 'ubuntu-advantage.log.1', 'boot.log.3', 'gpu-manager.log', 'ubuntu-advantage-timer.log', 'bootstrap.log', 'hp', 'ubuntu-advantage-timer.log.1', 'btmp', 'installer', 'ubuntu-advantage-timer.log.2.gz', 'btmp.1', 'journal', 'unattended-upgrades', 'cups', 'kern.log', 'wtmp', 'dbconfig-common', 'kern.log.1', 'dist-upgrade', 'kern.log.2.gz'. The prompt is now 'aelem1@aelem1-VirtualBox:/var/log\$' and the command 'cd apache2' has been executed. The prompt is now 'aelem1@aelem1-VirtualBox:/var/log/apache2\$'.

```
aelem1@aelem1-VirtualBox: /var/log/apache2
backups crash local log metrics run spool www
cache lib lock mail opt snap tmp
aelem1@aelem1-VirtualBox:/var$ cd log
aelem1@aelem1-VirtualBox:/var/log$ ls
alternatives.log dmesg kern.log.3.gz
alternatives.log.1 dmesg.0 lastlog
apache2 dmesg.1.gz mysql
apt dmesg.2.gz openvpn
auth.log dmesg.3.gz private
auth.log.1 dmesg.4.gz speech-dispatcher
auth.log.2.gz dpkg.log syslog
auth.log.3.gz dpkg.log.1 syslog.1
boot.log faillog syslog.2.gz
boot.log.1 fontconfig.log syslog.3.gz
boot.log.2 gdm3 ubuntu-advantage.log
boot.log.3 gpu-manager.log ubuntu-advantage.log.1
bootstrap.log hp ubuntu-advantage-timer.log
btmp installer ubuntu-advantage-timer.log.1
btmp.1 journal ubuntu-advantage-timer.log.2.gz
cups kern.log unattended-upgrades
dbconfig-common kern.log.1 wtmp
dist-upgrade kern.log.2.gz
aelem1@aelem1-VirtualBox:/var/log$ cd apache2
aelem1@aelem1-VirtualBox:/var/log/apache2$
```

changing directory to apache2

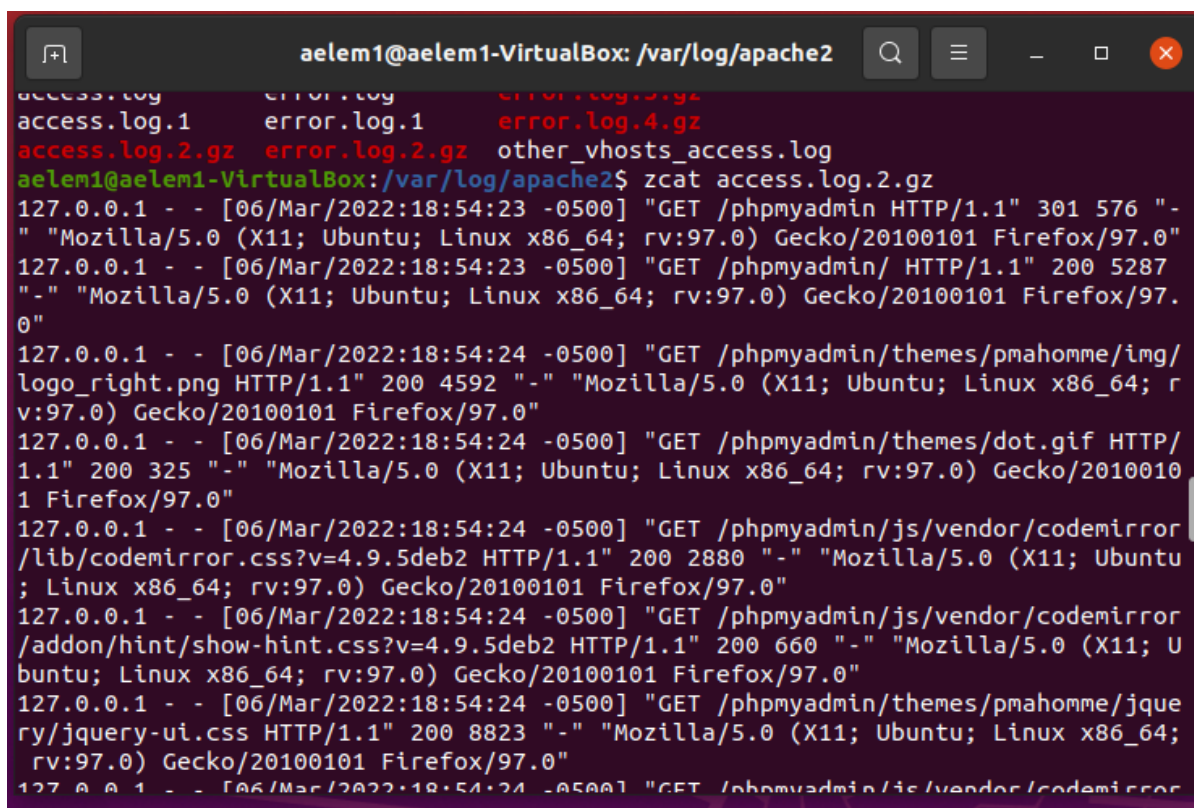
- Run the command: `ls` to lists files and directories within the apache2 directory
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/apache2' with search, menu, and window control icons. The terminal shows the command 'ls' being executed in '/var/log', listing various system logs. Then, the command 'cd apache2' is executed, and 'ls' is run again in the '/var/log/apache2' directory, listing Apache-related log files. The prompt returns to '/var/log/apache2\$'.

```
aelem1@aelem1-VirtualBox: /var/log/apache2
aelem1@aelem1-VirtualBox:/var/log$ ls
alternatives.log      dmesg                kern.log.3.gz
alternatives.log.1    dmesg.0              lastlog
apache2               dmesg.1.gz           mysql
apt                  dmesg.2.gz           openvpn
auth.log              dmesg.3.gz           private
auth.log.1            dmesg.4.gz           speech-dispatcher
auth.log.2.gz          dpkg.log             syslog
auth.log.3.gz          dpkg.log.1           syslog.1
boot.log              faillog              syslog.2.gz
boot.log.1            fontconfig.log        syslog.3.gz
boot.log.2            gdm3                 ubuntu-advantage.log
boot.log.3            gpu-manager.log       ubuntu-advantage.log.1
bootstrap.log         hp                   ubuntu-advantage-timer.log
btm                    installer            ubuntu-advantage-timer.log.1
btm.1                 journal              ubuntu-advantage-timer.log.2.gz
cups                  kern.log              unattended-upgrades
dbconfig-common       kern.log.1            wtmp
dist-upgrade          kern.log.2.gz
aelem1@aelem1-VirtualBox:/var/log$ cd apache2
aelem1@aelem1-VirtualBox:/var/log/apache2$ ls
access.log            access.log.2.gz      error.log.1          error.log.3.gz
access.log.1          error.log             error.log.2.gz       other_vhosts_access.log
aelem1@aelem1-VirtualBox:/var/log/apache2$
```

Listing files and directories within apache2

- The command "cat" is used to output the contents of the log file to the screen.
- The command "zcat" is used to output the contents of the compressed log files(.gz) to the screen.
- The log files are numbered in a reverse order i.e the highest number is the oldest log file.
- From the above, we can see that access.log.2.gz is the oldest and first access log file.
- View the contents of access.log.2.gz by running the command: zcat access.log.2.gz
- Here is what the output might look like:



```
aelem1@aelem1-VirtualBox: /var/log/apache2
access.log      error.log      error.log.2.gz
access.log.1    error.log.1    error.log.4.gz
access.log.2.gz error.log.2.gz other_vhosts_access.log
aelem1@aelem1-VirtualBox:/var/log/apache2$ zcat access.log.2.gz
127.0.0.1 - - [06/Mar/2022:18:54:23 -0500] "GET /phpmyadmin HTTP/1.1" 301 576 "-"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:97.0) Gecko/20100101 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:23 -0500] "GET /phpmyadmin/ HTTP/1.1" 200 5287
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:97.0) Gecko/20100101 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:24 -0500] "GET /phpmyadmin/themes/pmahomme/img/
logo_right.png HTTP/1.1" 200 4592 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; r
v:97.0) Gecko/20100101 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:24 -0500] "GET /phpmyadmin/themes/dot.gif HTTP/
1.1" 200 325 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:97.0) Gecko/2010010
1 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:24 -0500] "GET /phpmyadmin/js/vendor/codemirror
/lib/codemirror.css?v=4.9.5deb2 HTTP/1.1" 200 2880 "-" "Mozilla/5.0 (X11; Ubuntu
; Linux x86_64; rv:97.0) Gecko/20100101 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:24 -0500] "GET /phpmyadmin/js/vendor/codemirror
/addon/hint/show-hint.css?v=4.9.5deb2 HTTP/1.1" 200 660 "-" "Mozilla/5.0 (X11; U
buntu; Linux x86_64; rv:97.0) Gecko/20100101 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:24 -0500] "GET /phpmyadmin/themes/pmahomme/jque
ry/jquery-ui.css HTTP/1.1" 200 8823 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:97.0) Gecko/20100101 Firefox/97.0"
127.0.0.1 - - [06/Mar/2022:18:54:24 -0500] "GET /phpmyadmin/js/vendor/codemirror
```

Contents of access.log.2.gz

- Scrolling through the contents of the access.log.2.gz file shows that the attacker's system had an IP Address of 192.168.100.4
- We need to see all the requests that came from the IP Address 192.168.100.4 using the "grep" command thereby filtering out unneeded resources like images, css and javascript files.
- Run the command: `zcat access.log.2.gz | grep 192.168.100.4`
- Here is what the output might look like:

```

aelem1@aelem1-VirtualBox: /var/log/apache2
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:97.0) Gecko/20100101 Firefox/97.0
aelem1@aelem1-VirtualBox: /var/log/apache2$ zcat access.log.2.gz | grep 192.168.1
00.4
192.168.100.4 - - [07/Apr/2022:20:57:41 -0400] "GET /beunique/view_student.php?id=001 HTTP/1.1" 200 2959 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:20:57:41 -0400] "GET /beunique/view_student.php?id=001&FRFD=4217%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1" 200 2959 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:20:57:42 -0400] "GET /beunique/view_student.php?id=001 HTTP/1.1" 200 2959 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:20:57:42 -0400] "GET /beunique/view_student.php?id=3943 HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:20:57:43 -0400] "GET /beunique/view_student.php?id=001%22%28%28%27%2C.%2C%28%2C%2C HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:20:57:43 -0400] "GET /beunique/view_student.php?id=001%27gameAU%3C%27%22%3ExxVzoE HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:20:57:44 -0400] "GET /beunique/view_student.php?id=001%29%20AND%202600%3D4589%20AND%20%282142%3D2142 HTTP/1.1" 200 2959 "-" "sqlmap/1.6#stable (https://sqlmap.org)"

```

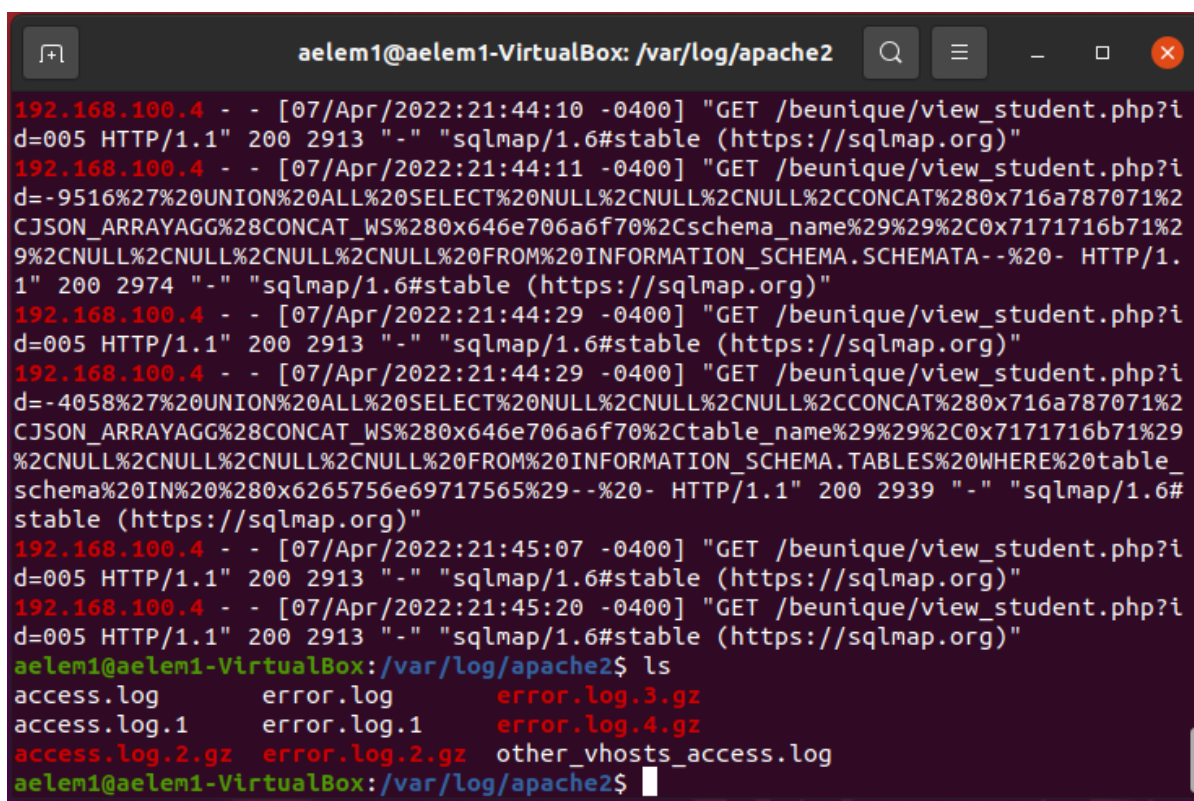
Requests of IP ADDRESS 192.168.100.4 in access.log.2.gz

- The first entry in the log shows that the attack took place on the 7th of April 2022 at 20:57:41 UTC.
- The SQL Injection attack utilized GET method requests which means parameter values was transmitted to the web server within the URL to query the backend database.
- The vulnerable webpage is `/beunique/view_student.php?id=001`
- The HTTP 200 status response code indicates that the request was successful.
- The quantity of entries in the `access.log` and the pattern indicate that a SQL injection vulnerability was exploited by the attacker using a SQL injection exploitation tool. The exploitation tool is SQLMAP as shown in the first entry that SQLMAP version 1.6 was used to scan the website. Some log entries of the attack may look like gibberish , but they are SQL queries that are frequently used to extract data via a SQL injection vulnerability. As part of the enumeration process, SQLMAP employs a variety of SQL injection techniques to discover the database name, table name, and columns.
- You can repeat the same process for the `access.log.1` file and the `access.log` file.

10. Task 4: Examine Apache error log files

- The Apache error files are in the same `apache2` directory.
- Run the `ls` command to list the files within the `apache2` directory

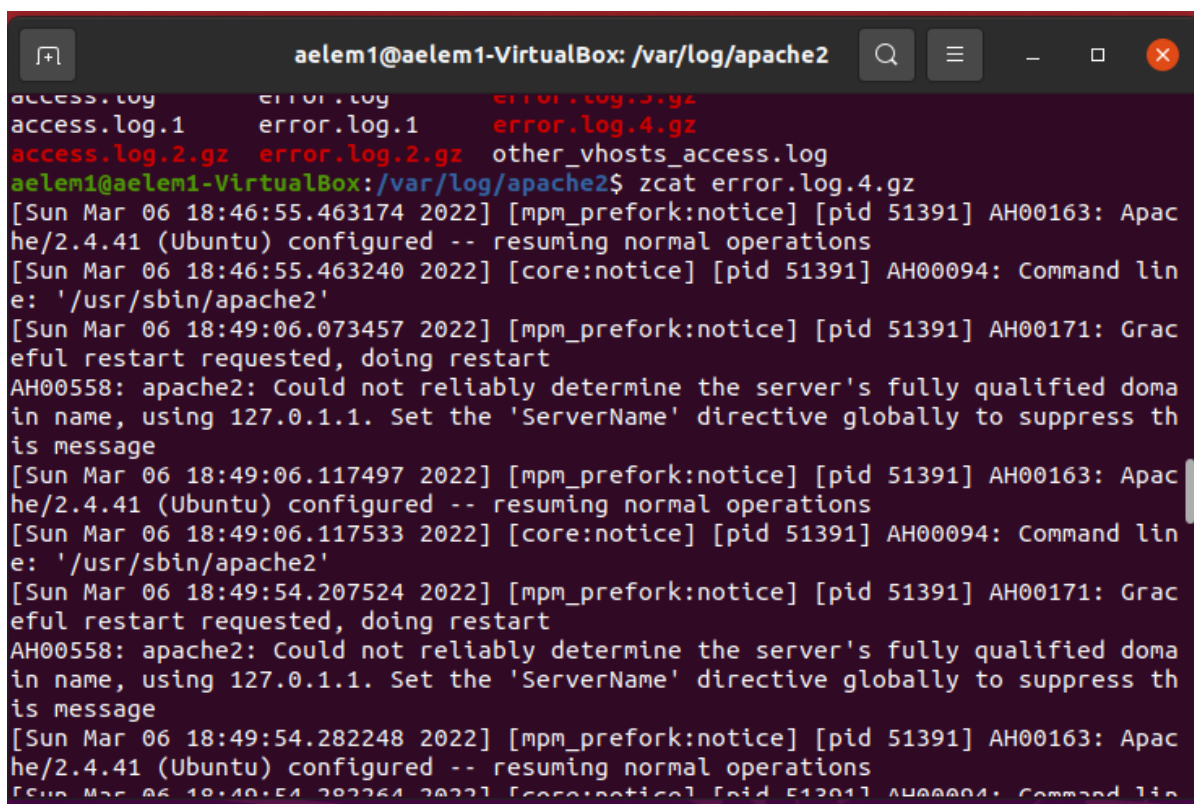
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/apache2' showing Apache2 logs and a directory listing. The logs show several GET requests to /beunique/view_student.php?id=005 from 192.168.100.4, with the last one containing a large SQL injection payload. The directory listing shows files: access.log, error.log, error.log.3.gz, error.log.4.gz, error.log.1, error.log.2.gz, other_vhosts_access.log, and error.log.2.gz.

```
aelem1@aelem1-VirtualBox: /var/log/apache2
192.168.100.4 - - [07/Apr/2022:21:44:10 -0400] "GET /beunique/view_student.php?id=005 HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:21:44:11 -0400] "GET /beunique/view_student.php?id=-9516%27%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CCONCAT%280x716a787071%2CJSON_ARRAYAGG%28CONCAT_WS%280x646e706a6f70%2Cschema_name%29%29%2C0x7171716b71%29%2CNULL%2CNULL%2CNULL%2CNULL%20FROM%20INFORMATION_SCHEMA.SCHEMATA--%20- HTTP/1.1" 200 2974 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:21:44:29 -0400] "GET /beunique/view_student.php?id=005 HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:21:44:29 -0400] "GET /beunique/view_student.php?id=-4058%27%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CCONCAT%280x716a787071%2CJSON_ARRAYAGG%28CONCAT_WS%280x646e706a6f70%2Ctable_name%29%29%2C0x7171716b71%29%2CNULL%2CNULL%2CNULL%2CNULL%20FROM%20INFORMATION_SCHEMA.TABLES%20WHERE%20table_schema%20IN%20%280x6265756e69717565%29--%20- HTTP/1.1" 200 2939 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:21:45:07 -0400] "GET /beunique/view_student.php?id=005 HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
192.168.100.4 - - [07/Apr/2022:21:45:20 -0400] "GET /beunique/view_student.php?id=005 HTTP/1.1" 200 2913 "-" "sqlmap/1.6#stable (https://sqlmap.org)"
aelem1@aelem1-VirtualBox:/var/log/apache2$ ls
access.log      error.log      error.log.3.gz
access.log.1    error.log.1    error.log.4.gz
access.log.2.gz error.log.2.gz other_vhosts_access.log
aelem1@aelem1-VirtualBox:/var/log/apache2$
```

Files in the apache2 directory

- The error log files are also listed in a reverse order so error.log.4.gz is the first and oldest log file
- View the contents of error.log.4.gz by running the command: `zcat error.log.4.gz`
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/apache2' with search, menu, and window control icons. It shows the command 'zcat error.log.4.gz' being executed. The output displays Apache error logs for March 6, 2022, at 18:46:55 and 18:49:06. The logs show that Apache 2.4.41 on Ubuntu was configured and resumed normal operations. It also shows graceful restarts requested at 18:49:06 and 18:49:54. A recurring error message (AH00558) states: 'Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message'. The command line is shown as '/usr/sbin/apache2'.

Contents of error.log.4.gz

- Scrolling through the contents shows the same IP Address of the attacker - 192.168.100.4
- View the error generated by the attacker's IP Address by running the command: `zcat error.log.4.gz | grep 192.168.100.4`
- Here is what the output might look like:

```

aelem1@aelem1-VirtualBox: /var/log/apache2
aelem1@aelem1-VirtualBox: /var/log/apache2$ zcat error.log.4.gz | grep 192.168.100.4
[Thu Apr 07 20:57:42.868410 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 40
[Thu Apr 07 20:57:42.868441 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 41
[Thu Apr 07 20:57:42.868450 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 42
[Thu Apr 07 20:57:42.868459 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 43
[Thu Apr 07 20:57:42.868468 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 46
[Thu Apr 07 20:57:42.868478 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 60
[Thu Apr 07 20:57:42.868489 2022] [php7:notice] [pid 859] [client 192.168.100.4:37400] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 71
[Thu Apr 07 20:57:43.445906 2022] [php7:warn] [pid 861] [client 192.168.100.4:37400] PHP Warning: Invalid argument supplied for foreach() in /var/www/html/beunique/view_student.php on line 71

```

Errors generated by 192.168.100.4 in error.log.4.gz

- Examining the errors shows the PHP Notice on several lines of the code: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php
- This proves the vulnerability exists in the view_student.php file.
- The first entry shows that the error was generated the same day and time the attack took place in access.log.2.gz - 7th of April 2022 at 20:57 UTC.
- The attacker used an automated tool(SQLMAP) to query the database using an id that does not exist in the table.
- This led to the PHP Notice: Trying to access array offset on value of type null.

11. Task 5: Examine MySQL query log files

- The MySQL log files are located in the /var/log/mysql directory
- Run the command: cd ../ to go back to the log directory
- Here is what the output might look like:

```

aelem1@aelem1-VirtualBox: /var/log
www/html/beunique/view_student.php on line 71
[Thu Apr 07 21:45:20.896350 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 40
[Thu Apr 07 21:45:20.896383 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 41
[Thu Apr 07 21:45:20.896393 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 42
[Thu Apr 07 21:45:20.896402 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 43
[Thu Apr 07 21:45:20.896411 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 46
[Thu Apr 07 21:45:20.896421 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 60
[Thu Apr 07 21:45:20.896432 2022] [php7:notice] [pid 920] [client 192.168.100.4:57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 71
aelem1@aelem1-VirtualBox:/var/log/apache2$ cd ../
aelem1@aelem1-VirtualBox:/var/log$

```

Back to the log Directory

- Run the ls command to list the folders in the log directory
- Here is what the output might look like:

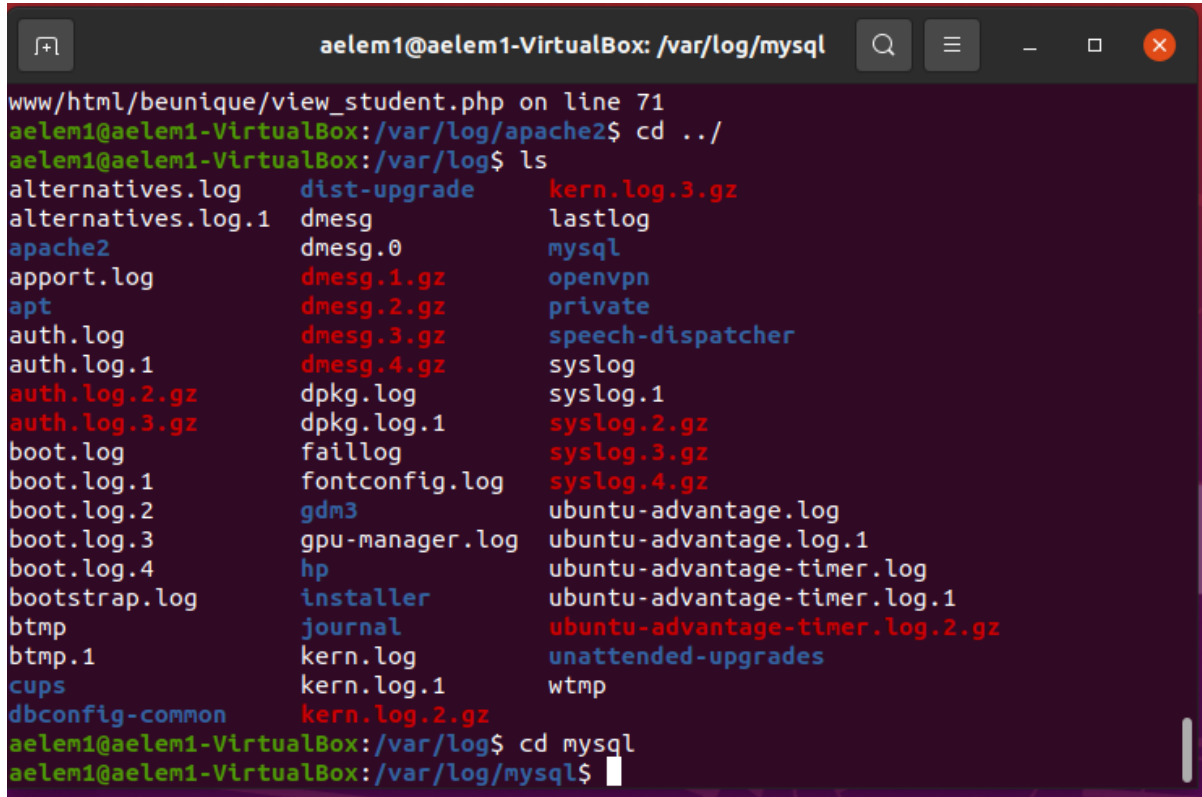
```

aelem1@aelem1-VirtualBox: /var/log
57330] PHP Notice: Trying to access array offset on value of type null in /var/www/html/beunique/view_student.php on line 71
aelem1@aelem1-VirtualBox:/var/log/apache2$ cd ../
aelem1@aelem1-VirtualBox:/var/log$ ls
alternatives.log      dist-upgrade          kern.log.3.gz
alternatives.log.1    dmesg                 lastlog
apache2               dmesg.0              mysql
appport.log           dmesg.1.gz           openvpn
apt                   dmesg.2.gz           private
auth.log              dmesg.3.gz           speech-dispatcher
auth.log.1            dmesg.4.gz           syslog
auth.log.2.gz         dpkg.log              syslog.1
auth.log.3.gz         dpkg.log.1            syslog.2.gz
boot.log              faillog               syslog.3.gz
boot.log.1            fontconfig.log        syslog.4.gz
boot.log.2            gdm3                  ubuntu-advantage.log
boot.log.3            gpu-manager.log       ubuntu-advantage.log.1
boot.log.4            hp                     ubuntu-advantage-timer.log
bootstrap.log         installer             ubuntu-advantage-timer.log.1
btmtp                 journal               ubuntu-advantage-timer.log.2.gz
btmtp.1               kern.log              unattended-upgrades
cups                  kern.log.1            wtmp
dbconfig-common        kern.log.2.gz
aelem1@aelem1-VirtualBox:/var/log$

```

Folders in the log Directory

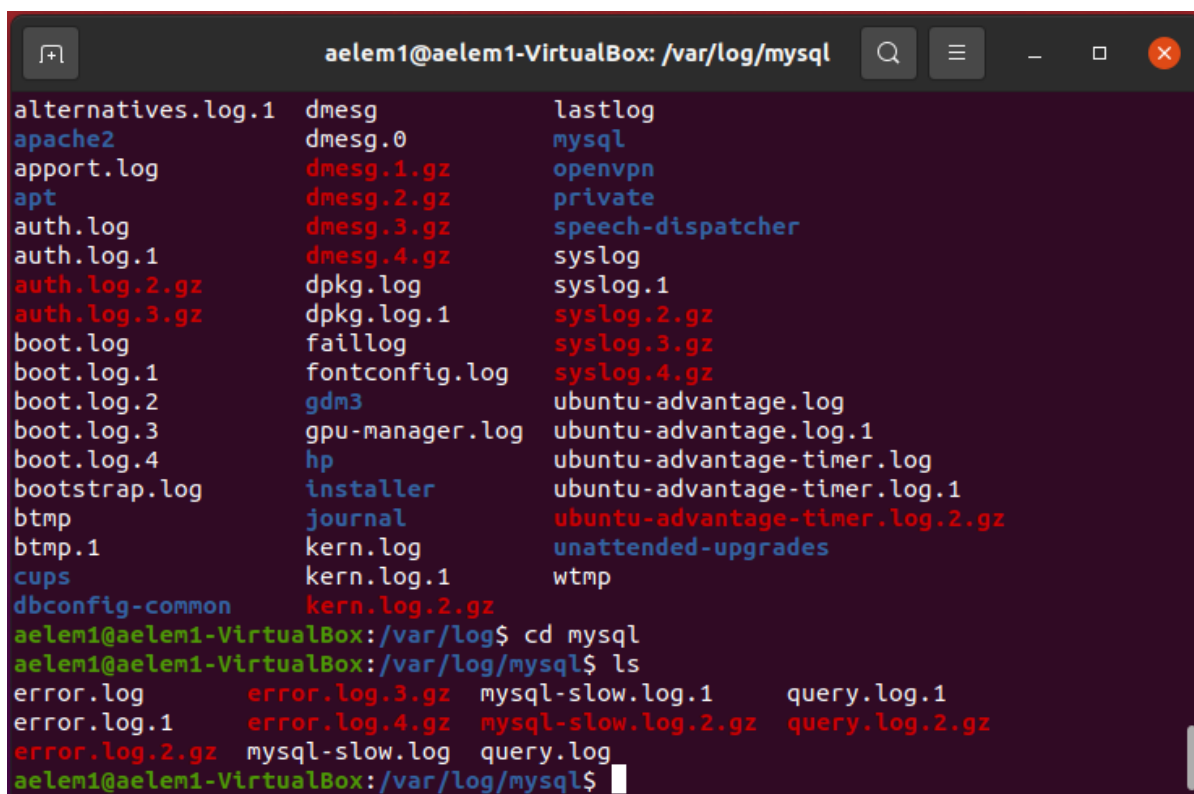
- Change the directory to Mysql by running the command: `cd mysql`
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/mysql' showing a directory listing of /var/log. The output of the 'ls' command is displayed in three columns. The user then navigates to the 'mysql' directory.

```
www/html/beunique/view_student.php on line 71
aelem1@aelem1-VirtualBox:/var/log/apache2$ cd ../
aelem1@aelem1-VirtualBox:/var/log$ ls
alternatives.log  dist-upgrade      kern.log.3.gz
alternatives.log.1  dmesg             lastlog
apache2           dmesg.0           mysql
appport.log       dmesg.1.gz        openvpn
apt              dmesg.2.gz        private
auth.log          dmesg.3.gz        speech-dispatcher
auth.log.1        dmesg.4.gz        syslog
auth.log.2.gz     dpkg.log          syslog.1
auth.log.3.gz     dpkg.log.1        syslog.2.gz
boot.log          faillog           syslog.3.gz
boot.log.1        fontconfig.log    syslog.4.gz
boot.log.2        gdm3              ubuntu-advantage.log
boot.log.3        gpu-manager.log   ubuntu-advantage.log.1
boot.log.4        hp                ubuntu-advantage-timer.log
bootstrap.log     installer         ubuntu-advantage-timer.log.1
btmtp             journal           ubuntu-advantage-timer.log.2.gz
btmtp.1           kern.log          unattended-upgrades
cups              kern.log.1        wtmp
dbconfig-common   kern.log.2.gz
aelem1@aelem1-VirtualBox:/var/log$ cd mysql
aelem1@aelem1-VirtualBox:/var/log/mysql$
```

Changing directory to mysql

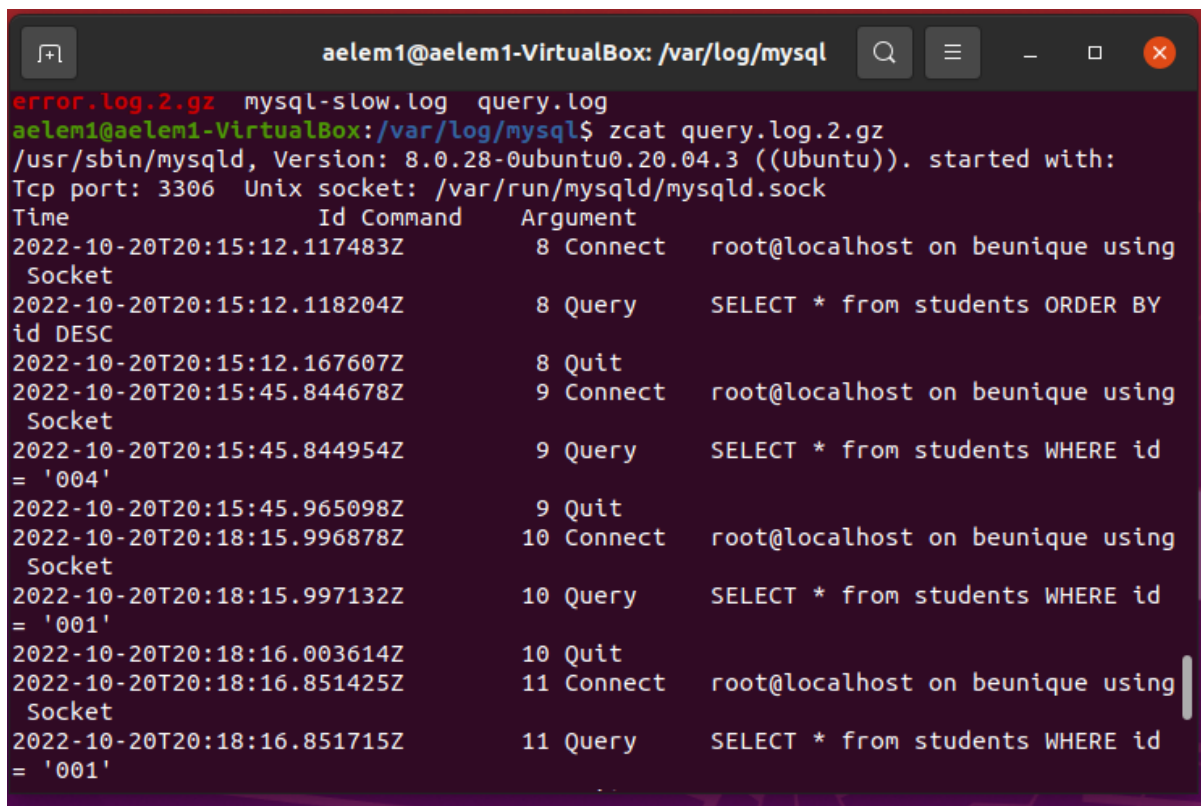
- Run the `ls` command
- Here is what the output might look like:



```
aelem1@aelem1-VirtualBox: /var/log/mysql
alternatives.log.1 dmesg lastlog
apache2 dmesg.0 mysql
apport.log dmesg.1.gz openvpn
apt dmesg.2.gz private
auth.log dmesg.3.gz speech-dispatcher
auth.log.1 dmesg.4.gz syslog
auth.log.2.gz dpkg.log syslog.1
auth.log.3.gz dpkg.log.1 syslog.2.gz
boot.log faillog syslog.3.gz
boot.log.1 fontconfig.log syslog.4.gz
boot.log.2 gdm3 ubuntu-advantage.log
boot.log.3 gpu-manager.log ubuntu-advantage.log.1
boot.log.4 hp ubuntu-advantage-timer.log
bootstrap.log installer ubuntu-advantage-timer.log.1
btmpt journal ubuntu-advantage-timer.log.2.gz
btmpt.1 kern.log unattended-upgrades
cups kern.log.1 wtmp
dbconfig-common kern.log.2.gz
aelem1@aelem1-VirtualBox:/var/log$ cd mysql
aelem1@aelem1-VirtualBox:/var/log/mysql$ ls
error.log error.log.3.gz mysql-slow.log.1 query.log.1
error.log.1 error.log.4.gz mysql-slow.log.2.gz query.log.2.gz
error.log.2.gz mysql-slow.log query.log
aelem1@aelem1-VirtualBox:/var/log/mysql$
```

Listing contents of mysql directory

- The query log files follow the same reverse naming system therefore query.log.2.gz is the first and oldest log file.
- View the contents of query.log.2.gz by running the command: `zcat query.log.2.gz`
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/mysql' with search, menu, and window control icons. The terminal shows the command 'zcat query.log.2.gz' being executed. The output displays MySQL log entries for connections and queries. The log entries are as follows:

```
error.log.2.gz  mysql-slow.log  query.log
aelem1@aelem1-VirtualBox:/var/log/mysql$ zcat query.log.2.gz
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time                Id Command  Argument
2022-10-20T20:15:12.117483Z      8 Connect  root@localhost on beunique using
Socket
2022-10-20T20:15:12.118204Z      8 Query    SELECT * from students ORDER BY
id DESC
2022-10-20T20:15:12.167607Z      8 Quit
2022-10-20T20:15:45.844678Z      9 Connect  root@localhost on beunique using
Socket
2022-10-20T20:15:45.844954Z      9 Query    SELECT * from students WHERE id
= '004'
2022-10-20T20:15:45.965098Z      9 Quit
2022-10-20T20:18:15.996878Z     10 Connect  root@localhost on beunique using
Socket
2022-10-20T20:18:15.997132Z     10 Query    SELECT * from students WHERE id
= '001'
2022-10-20T20:18:16.003614Z     10 Quit
2022-10-20T20:18:16.851425Z     11 Connect  root@localhost on beunique using
Socket
2022-10-20T20:18:16.851715Z     11 Query    SELECT * from students WHERE id
= '001'
```

Viewing contents of query.log.2.gz

- Scrolling through the contents shows the various queries that were executed.
- Scroll down to the point where the entire database is extracted.
- Here is what the output might look like:

```

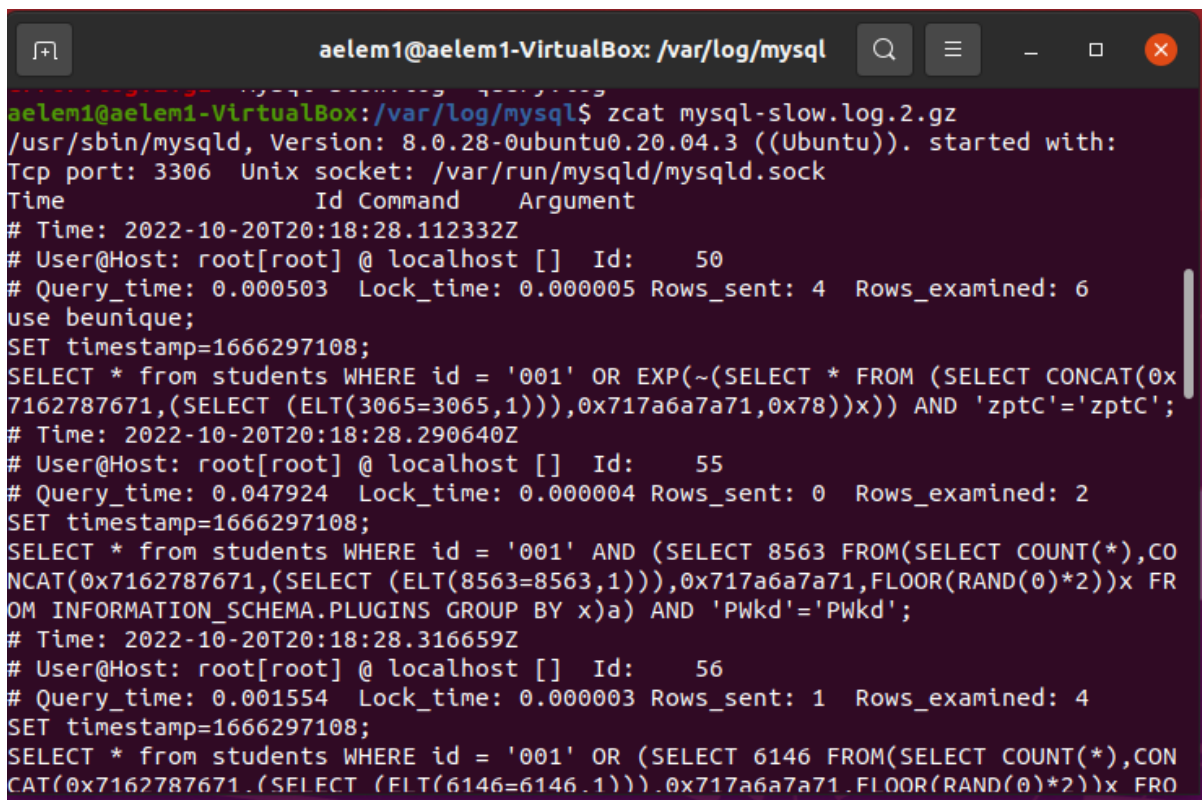
aelem1@aelem1-VirtualBox: /var/log/mysql
2022-10-20T20:20:27.445061Z      116 Quit
2022-10-20T20:20:28.238756Z      117 Connect  root@localhost on beunique using
Socket
2022-10-20T20:20:28.239054Z      117 Query    SELECT * from students WHERE id
= '-8864' UNION ALL SELECT NULL,CONCAT(0x7162787671,JSON_ARRAYAGG(CONCAT_WS(0x6a
6572777872,`session`,`timestamp`,email,first_name,id,last_name,lesson,phone_numb
er)),0x717a6a7a71),NULL,NULL,NULL,NULL,NULL,NULL FROM beunique.students-- -'
2022-10-20T20:20:28.239926Z      117 Quit
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:

```

Query showing all the fields in the students table were extracted

12. Task 6: Examine MySQL mysql-slow log files

- The mysql-slow.log files follow the same reverse naming system therefore mysql-slow.log.2.gz is the first and oldest log file.
- View the contents of mysql-slow.log.2.gz by running the command: `zcat mysql-slow.log.2.gz`
- Here is what the output might look like:

A terminal window titled 'aelem1@aelem1-VirtualBox: /var/log/mysql' with standard window controls. The terminal shows the command 'zcat mysql-slow.log.2.gz' being executed. The output displays MySQL log entries for three queries. Each entry includes a timestamp, user information, query statistics (Query_time, Lock_time, Rows_sent, Rows_examined), and the SQL command. The first query is a complex SELECT statement with a WHERE clause involving a timestamp and a user. The second query is a SELECT statement with a WHERE clause involving a timestamp and a user. The third query is a SELECT statement with a WHERE clause involving a timestamp and a user. The output is truncated on the right side, indicated by a vertical scrollbar.

```
aelem1@aelem1-VirtualBox: /var/log/mysql$ zcat mysql-slow.log.2.gz
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument
# Time: 2022-10-20T20:18:28.112332Z
# User@Host: root[root] @ localhost [] Id: 50
# Query_time: 0.000503 Lock_time: 0.000005 Rows_sent: 4 Rows_examined: 6
use beunique;
SET timestamp=1666297108;
SELECT * from students WHERE id = '001' OR EXP(~(SELECT * FROM (SELECT CONCAT(0x
7162787671,(SELECT (ELT(3065=3065,1))),0x717a6a7a71,0x78))x)) AND 'zptC'='zptC';
# Time: 2022-10-20T20:18:28.290640Z
# User@Host: root[root] @ localhost [] Id: 55
# Query_time: 0.047924 Lock_time: 0.000004 Rows_sent: 0 Rows_examined: 2
SET timestamp=1666297108;
SELECT * from students WHERE id = '001' AND (SELECT 8563 FROM(SELECT COUNT(*),CO
NCAT(0x7162787671,(SELECT (ELT(8563=8563,1))),0x717a6a7a71,FLOOR(RAND(0)*2))x FR
OM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) AND 'PWkd'='PWkd';
# Time: 2022-10-20T20:18:28.316659Z
# User@Host: root[root] @ localhost [] Id: 56
# Query_time: 0.001554 Lock_time: 0.000003 Rows_sent: 1 Rows_examined: 4
SET timestamp=1666297108;
SELECT * from students WHERE id = '001' OR (SELECT 6146 FROM(SELECT COUNT(*),CON
CAT(0x7162787671,(SELECT (ELT(6146=6146,1))),0x717a6a7a71,FLOOR(RAND(0)*2))x FRO
```

Viewing contents of mysql-slow.log.2.gz

- Scrolling through the contents shows the various queries that were executed.
- Scroll down to the point where the entire database is extracted.
- Here is what the output might look like:

```

aelem1@aelem1-VirtualBox: /var/log/mysql
, NULL, NULL, NULL, NULL FROM INFORMATION_SCHEMA.SCHEMATA-- -';
# Time: 2022-10-20T20:20:28.239469Z
# User@Host: root[root] @ localhost [] Id: 117
# Query_time: 0.000501 Lock_time: 0.000003 Rows_sent: 1 Rows_examined: 4
SET timestamp=1666297228;
SELECT * from students WHERE id = '-8864' UNION ALL SELECT NULL, CONCAT(0x7162787
671, JSON_ARRAYAGG(CONCAT_WS(0x6a6572777872, `session`, `timestamp`, `email`, `first_nam
e`, `id`, `last_name`, `lesson`, `phone_number`)), 0x717a6a7a71), NULL, NULL, NULL, NULL, NULL
FROM beunique.students-- -';
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument
/usr/sbin/mysqld, Version: 8.0.28-0ubuntu0.20.04.3 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument

```

Query showing all the fields in the students table were extracted

13. Analysis

- Who was responsible for the attack is one of the unanswered mysteries. The attacker's IP address is the only information that is currently accessible. If the attacker did not leave behind tangible proof that could be linked to a specific person's identity, it would be difficult to try and credit most attacks. Keep in mind that most assaults are conducted by attackers using proxies and anonymity networks like Tor to conceal their true location.
- The bottom line is that `view_student.php` included vulnerable code that enabled a SQL injection attack. The attacker would not have been able to exploit the security flaw that led to the extraction of the database if the website had undergone security vulnerability testing before going live in a production environment.
- In the fictitious example above, the attacker was very careless and left behind a lot of evidence, which made the investigation relatively simple. However, keep in mind that this is not always the case, particularly when dealing with more advanced attacks.

14. Task 7: Test Your Understanding

Complete the questions listed below. At the end of this exercise, attempt to analyze the whole case and answer the *Thought Question(s)* on your own before displaying the answer(s). Note, the latter are not graded.

15. Task 8: Additional Resources

Continuing SQL Injection with SQLMap - Detection via logs

15.1 Flag Questions:

1. What exact time did the attacker first access the website?

- (a) 05/Apr/2022:20:57:41
- (b) 13/Nov/2022:21:47:52
- (c) 07/Apr/2022:20:57:41
- (d) 15/Nov/2022:21:47:53

- **Correct Answer:** 15/Nov/2022:21:47:53
- **Answer Description:** To find this, Go to Task 1: Examine Apache access log files
- **Difficulty:** Hard
- **Total Points:** 8
- **Retries:** 2

2. What is the linux command to view a compressed file without uncompressing that file

- (a) cat
- (b) zcat
- (c) head
- (d) tail

- **Correct Answer:** zcat
- **Answer Description:** To find this, check Task 1:Examine Apache access log files
- **Difficulty:** easy
- **Total Points:** 2
- **Retries:** 2

3. What linux command would you use to filter log lines of a suspicious IP address?

- (a) ls
- (b) cd
- (c) grep
- (d) var

- **Correct Answer:** grep
- **Answer Description:** To find this, check Task 1:Examine Apache access log files
- **Difficulty:** easy

- **Total Points:** 2
- **Retries:** 2

4. What exact time was data extracted from the students table in the query log?

- (a) 2022-21-16T02:45:15
- (b) 2022-10-20T20:20:24
- (c) 2022-11-16T02:50:15
- (d) 2022-10-20T20:20:28

- **Correct Answer:** 2022-11-16T02:50:15
- **Answer Description:** To find this, Go to Task 3:Examine MySQL query log files
- **Difficulty:** Hard
- **Total Points:** 8
- **Retries:** 2

5. What is the first process ID of the second Apache error log generated by the attacker?

- (a) 878
- (b) 51391
- (c) 859
- (d) 3319

- **Correct Answer:** 878
- **Answer Description:** To find this, Go to Task 2:Examine Apache error log files
- **Difficulty:** Hard
- **Total Points:** 8
- **Retries:** 2

6. What is the IP Address of the attacker?

- (a) 192.168.100.5
- (b) 192.168.100.3
- (c) 192.168.100.2
- (d) 192.168.100.4

- **Correct Answer:** 192.168.100.5
- **Answer Description:** To find this, Go to Task 1:Examine Apache access log files
- **Difficulty:** Intermediate
- **Total Points:** 4
- **Retries:** 2

7. What is the time in UTC when the extraction of the data took place in mysql-slow.log(Hint: Use Epoch Converter)

- (a) Friday, October 21, 2022 4:20:28 PM
- (b) Tuesday, November 15, 2022 9:50:15 PM
- (c) Thursday, October 20, 2022 4:20:28 PM
- (d) Wednesday, November 16, 2022 9:50:15 PM

- **Correct Answer:** Tuesday, November 15, 2022 9:50:15 PM
- **Answer Description:** To find this, Go to Task 4:Examine MySQL mysql-slow log files
- **Difficulty:** Hard
- **Total Points:** 8
- **Retries:** 2

8. What is the ID of the query that extracted all the records from the students table in the query log?

- (a) 117
- (b) 114
- (c) 116
- (d) 118

- **Correct Answer:** 116
- **Answer Description:** To find this, Go to Task 3:Examine MySQL query log files
- **Difficulty:** Intermediate
- **Total Points:** 4
- **Retries:** 2

9. What is the linux command to determine the type of a file?

- (a) filename
- (b) file
- (c) filetype
- (d) filekind

- **Correct Answer:** file
- **Answer Description:** To find this, Go to Task 6: Additional Resources
- **Difficulty:** Easy
- **Total Points:** 2
- **Retries:** 2

10. What linux command is used to view a text file without modifying it?

- (a) cat
- (b) type
- (c) zcat
- (d) head

- **Correct Answer:** type
- **Answer Description:** To find this, Go to Task 6: Additional Resources
- **Difficulty:** Easy
- **Total Points:** 2
- **Retries:** 2

15.2 Thought Questions:

1. Create the chain of events that led to this attack?

- **Correct Answer:**



Chain of events that led to the attack

2. Why is log analysis important in cyber security?

- **Correct Answer:** Organizations that want to improve their security posture must develop capabilities in log analysis that will enable them to actively recognize and counteract cyber attacks. Businesses that use log analysis to successfully monitor their systems can increase the defenses around their network assets. Splunk and Datadog, as well as other log monitoring technologies, can assist firms achieve compliance standards for cyber security, such as ISO/IEC 27002:2013, PCI DSS V3.1, and NIST 800-137.

They can also lessen the frequency and severity of cyberattacks. The following logs need to be monitored by organizations: System activity logs, Endpoint logs, Application logs, Authentication logs, Physical security logs, Email logs, Firewall logs, VPN logs, Netflow logs, HTTP proxy logs, DNS, DHCP and FTP logs, AppFlow logs, Web and SQL server logs, Malware protection software logs, Network intrusion detection system (NIDS) logs, Network intrusion prevention system (NIPS) logs and Data Loss Prevention (DLP) logs. To save time and resources, cyber security specialists should identify the most crucial logs for ongoing monitoring and use automated or software-based log analysis techniques.

16. References

<https://www.acunetix.com/blog/articles/using-logs-to-investigate-a-web-application-attack/>