

**ATTACK**

**DEFENSE**

by PentesterAcademy

<b>Name</b>	LNAV : Apache Log Analysis
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=1184">https://attackdefense.com/challengedetails?cid=1184</a>
<b>Type</b>	Log Analysis : Other Tools

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Open the apache logs using Inav for analysis.

**Command:** Inav access.log

```

Mon Aug 05 17:43:47 UTC /root/access.log access_log LOG
192.131.71.7 - - [04/Aug/2019:18:30:21 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable ("
192.131.71.5 - - [04/Aug/2019:18:29:37 +0000] "GET /btslab/vulnerability/ForumPosts.php?id=20sleep%288000%2F1000%29%3B HTTP/1.1" 200 2084 "-" "Arachni/v1
192.131.71.5 - - [04/Aug/2019:18:29:45 +0000] "GET /btslab/vulnerability/ForumPosts.php?id=1 HTTP/1.1" 200 2166 "-" "Arachni/v1.5.1"
127.0.0.1 - - [04/Aug/2019:18:33:22 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.2.22 (Ubuntu) (internal dummy connection)"
Last message: 23 hours ago; Files: 1; Error rate: 0.00/min; Time span: 8719h18m45s
L82506 100% 0 hits ?View Help
Press e/E to move forward/backward through error messages

```

**Q1. SQLmap was used to perform SQL injection attack against a particular webpage. Find the relative path of the vulnerable webpage.**

**Answer:** /index.php

**Solution:**

**Step 1:** Apply a filter searching for sqlmap in the logs.

**Filter:** :filter-in sqlmap

```
Tue Aug 06 00:33:50 UTC
192.131.71.7 - - [04/Aug/2019:18:30:22 +0000] "POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php"
```

After applying the filter, the scroller moves to the last record. Press "g" to move to the first line of the returned logs.

```
/root/access.log: access_log
"POST /index.php?page=login.php HTTP/1.1" 200 314 "-" "sqlmap/1.2.11#stable (http://sqlmap.org)"
"POST /index.php?page=login.php&wdQu=8985%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
"POST /index.php?page=login.php HTTP/1.1" 200 314 "http://192.131.71.3:80/index.php" "sqlmap/1.2.11#stable
```

The returned output indicates that /index.php page was attacked using sqlmap.

**Note:** To exit out of the filter, press Ctrl + R. It will reset the session.

Alternative approach using SQL query:

**Step 1:** Execute the following SQL query to get the distinct requested files for which the user-agent string contains "sqlmap".

**Query:** select DISTINCT cs\_uri\_stem from access\_log where cs\_user\_agent LIKE "%sqlmap%";



**Note:** Type ";" before entering the SQL query.

```
Tue Aug 06 00:41:07 UTC
cs_uri_stem
/index.php
```

**Note:** If the above window doesn't show up, then the output is shown in the bottom panel. To switch to SQL result view, press "v". To exit the SQL result view, press "q" or press "v" again.

The vulnerable web page was located at "/index.php".

**Q2. The most active user of the web application was suspected to be attacking it. Verify if that was actually the case. Also find out the IP address of that user.**

**Answer:** 192.131.71.7

**Solution:**

**Step 1:** Execute the following SQL query to get the IP address of the most active client.

**Query:** select c\_ip , COUNT(\*) as count from access\_log group by c\_ip order by count desc;

c_ip	count
192.131.71.7	43399
192.131.71.6	17551
192.131.71.2	14658
192.131.71.5	6521
127.0.0.1	368
172.17.0.1	5

The machine at IP address "192.131.71.7" was the most active client.

**Step 2:** Execute the following SQL query to get the list of request paths and the request queries issued by the most active client.

**Query:** select DISTINCT cs\_uri\_stem, cs\_uri\_query from access\_log where c\_ip="192.131.71.7";

cs_uri_stem	
/index.php	page=login.php
/index.php	page=login.php&wdQu=8985%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%20
<NULL>	<NULL>
/index.php	page=3715
/index.php	page=login.php%29%2C%29%29%29.%28%22%27%29
/index.php	page=login.php%27cSXCKm%3C%27%22%3EaCFppk
/index.php	page=login.php%29%20AND%208882%3D2297--%20XbYX
/index.php	page=login.php%29%20AND%201960%3D1960--%20HHFQ
/index.php	page=login.php%27%29%20AND%205015%3D9711--%20lDqI
/index.php	page=login.php%27%29%20AND%201960%3D1960--%20vQCq
/index.php	page=login.php%27%20AND%204215%3D9605--%20HZiF
/index.php	page=login.php%27%20AND%201960%3D1960--%20nNRy
/index.php	page=login.php%22%20AND%208573%3D6611--%20lLgz
/index.php	page=login.php%22%20AND%201960%3D1960--%20kTqA
/index.php	page=login.php%29%20AND%201914%3D4192%20AND%20%284950%3D4950
/index.php	page=login.php%29%20AND%201960%3D1960%20AND%20%284199%3D4199
/index.php	page=login.php%29%29%20AND%203852%3D4687%20AND%20%28%282163%3D2163
/index.php	page=login.php%29%29%20AND%201960%3D1960%20AND%20%28%288482%3D8482
/index.php	page=login.php%29%29%29%20AND%206433%3D6774%20AND%20%28%28%287318%3D7318
/index.php	page=login.php%29%29%29%20AND%201960%3D1960%20AND%20%28%28%285272%3D5272

The above results show that the most active client, having IP address "192.131.71.7" was attacking the web application.

**Q3. Which attacker machine was able to find out the existence of directory '/btslab/tmp/'?**

**Answer:** 192.131.71.5

**Solution:**

**Step 1:** Execute the following SQL query to retrieve the IP address, the request path and the status code of the client that was able to locate `"/bt slab/tmp/"` directory.

**Query:** select c\_ip, cs\_uri\_stem, sc\_status from access\_log where cs\_uri\_stem LIKE "/btslab/tmp/";

Tue Aug 06 01:06:52 UTC		
c_ip	cs_uri_stem	sc_status
192.131.71.5	/btslab/tmp/	200

The client at IP address "192.131.71.5" was able to find out the existence of the "/btslab/tmp/" directory.

**Note:** If the above window doesn't show up, then the output is shown in the bottom panel. To switch to SQL result view, press "v". To exit the SQL result view, press "q" or press "v" again.

**Step 2:** Execute the following SQL query to retrieve the client IP addresses and count of the requests that resulted in the status code of 404.

**Query:** select c\_ip, count(\*) as count from access\_log where sc\_status="404" group by c\_ip order by count desc;

c_ip	count
192.131.71.6	17152
192.131.71.5	1546
192.131.71.2	703

The client at IP address "192.131.71.5" received 404 Not Found response 1546 times, which indicates that it could be a possible attacker machine.

**Step 3:** Execute the following SQL query to retrieve the timestamp and the request path from the logs where the client IP address was "192.131.71.5".

**Query:** select log\_time, cs\_uri\_stem from access\_log where c\_ip="192.131.71.5";



log_time	cs_uri_stem
2019-08-04 18:19:54.000	/
2019-08-04 18:19:54.000	/
2019-08-04 18:19:54.000	/%3E%22'%3E%3Cmy_tag_ddc5dc63e2371d6df9695175659fb79b/%3E
2019-08-04 18:19:54.000	/%3Cmy_tag_ddc5dc63e2371d6df9695175659fb79b/%3E
2019-08-04 18:19:54.000	/
2019-08-04 18:19:54.000	/
2019-08-04 18:19:54.000	/
2019-08-04 18:19:54.000	/
2019-08-04 18:19:54.000	/ddc5dc63e2371d6df9695175659fb79b
2019-08-04 18:19:54.000	/clientaccesspolicy.xml
2019-08-04 18:19:54.000	/robots.txt
2019-08-04 18:19:54.000	/Arachni-ddc5dc63e2371d6df9695175659fb79b
2019-08-04 18:19:54.000	/_mmServerScripts/MMHTTPDB.asp
2019-08-04 18:19:54.000	/_mmDBScripts/MMHTTPDB.php
2019-08-04 18:19:54.000	/_mmDBScripts/MMHTTPDB.asp
2019-08-04 18:19:54.000	/config/database.yml
2019-08-04 18:19:54.000	/install.php
2019-08-04 18:19:54.000	/config.php
2019-08-04 18:19:54.000	/php.ini

```

2019-08-04 18:20:35.000|/btslab/index.php%20script:;arachni_xss_in_element_event=ddc5dc63e2371d6df969
2019-08-04 18:20:35.000|/btslab/index.php%20;arachni_xss_in_element_event=ddc5dc63e2371d6df9695175659
2019-08-04 18:20:35.000|/btslab/index.php%20script:\";arachni_xss_in_element_event=ddc5dc63e2371d6df96
2019-08-04 18:20:35.000|/btslab/vulnerability/url/forward.php
2019-08-04 18:20:35.000|/btslab/vulnerability/url/forward.php
2019-08-04 18:20:35.000|/btslab/vulnerability/url/forward.php
2019-08-04 18:20:35.000|/btslab/index.php%20script:';arachni_xss_in_element_event=ddc5dc63e2371d6df96
2019-08-04 18:20:35.000|/btslab/index.php%20\";arachni_xss_in_element_event=ddc5dc63e2371d6df96951756
2019-08-04 18:20:35.000|/btslab/index.php%20';arachni_xss_in_element_event=ddc5dc63e2371d6df969517565
2019-08-04 18:20:35.000|/btslab/
2019-08-04 18:20:35.000|/btslab/
2019-08-04 18:20:35.000|/btslab/
2019-08-04 18:20:35.000|/btslab/
2019-08-04 18:20:35.000|/btslab/%3Cmy_tag_ddc5dc63e2371d6df9695175659fb79b/%3E
2019-08-04 18:20:35.000|/btslab/
2019-08-04 18:20:35.000|/btslab/
2019-08-04 18:20:35.000|/btslab/%3E%22'%3E%3Cmy_tag_ddc5dc63e2371d6df9695175659fb79b/%3E
2019-08-04 18:20:38.000|/btslab/vulnerability/rfi/RFI.php

```

```

2019-08-04 18:23:19.000 |/btslab/index.php<xss_ddc5dc63e2371d6df9695175659fb79b/>
2019-08-04 18:23:19.000 |/btslab/vulnerability/url/forward.php
2019-08-04 18:23:19.000 |/btslab/index.php%3Cxss_ddc5dc63e2371d6df9695175659fb79b%2F%3E
2019-08-04 18:23:19.000 |/btslab/vulnerability/url/forward.php
2019-08-04 18:23:19.000 |/btslab/index.php()\"&%1'-;<xss_ddc5dc63e2371d6df9695175659fb79b/>'
2019-08-04 18:23:19.000 |/btslab/vulnerability/url/forward.php
2019-08-04 18:23:19.000 |/btslab/index.php%28%29%22%26%25%27-%3B%3Cxss_ddc5dc63e2371d6df9695175659fb79b%2F%
2019-08-04 18:23:19.000 |/btslab/vulnerability/url/forward.php
2019-08-04 18:23:19.000 |/btslab/index.php</textarea>--><xss_ddc5dc63e2371d6df9695175659fb79b/><!--<textare
2019-08-04 18:23:19.000 |/btslab/vulnerability/url/forward.php
2019-08-04 18:23:19.000 |/btslab/index.php%3C%2Ftextarea%3E--%3E%3Cxss_ddc5dc63e2371d6df9695175659fb79b%2F%

```

The client at IP address "192.131.71.5" issued a huge number of suspicious requests in a short span of time.

This indicates that the client at "192.131.71.5" was trying to attack the web application and it was also able to find the existence of "/btslab/tmp/" directory.

#### Q4. How many union based SQL injection attacks were attempted on the web application?

**Answer:** 2138

**Solution:**

**Step 1:** Execute the following SQL query to retrieve the count of all the log instances in which the request query contains "union" keyword.

**Query:** select COUNT(\*) from access\_log where cs\_uri\_query LIKE "%union%";

COUNT(*)
2138

There were 2138 log instances where union based SQL injection attack was attempted.

**Note:** If the above window doesn't show up, then the output is shown in the bottom panel. To switch to SQL result view, press "v". To exit the SQL result view, press "q" or press "v" again.



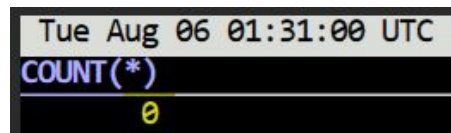
**Q5. How many of the requests resulted in an internal server error?**

**Answer:** 0

**Solution:**

**Step 1:** Execute the following SQL query to retrieve the count of all the log instances where the request status was 500, i.e, internal server error.

**Query:** select COUNT(\*) from access\_log where sc\_status=500;



COUNT(*)
0

There were no requests that resulted in an internal server error.

**Note:** If the above window doesn't show up, then the output is shown in the bottom panel. To switch to SQL result view, press "v". To exit the SQL result view, press "q" or press "v" again.

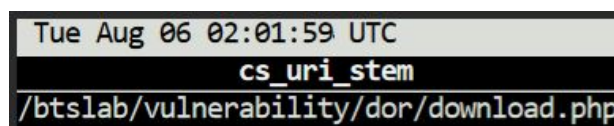
**Q6. An HTTP GET request was made to a web page to download a PDF file. The same web page was vulnerable to PATH traversal attack and was exploited to download /etc/passwd file. Find the approximate size of /etc/passwd file in KB.**

**Answer:** 1

**Solution:**

**Step 1:** Execute the following SQL query to retrieve the distinct request paths which were queried to download PDF files.

**Query:** select DISTINCT cs\_uri\_stem from access\_log where cs\_uri\_query LIKE "%.pdf";



cs_uri_stem
/btsslabs/vulnerability/dor/download.php

The PDF files were downloaded from the page located at  
"/btslab/vulnerability/dor/download.php".

**Note:** If the above window doesn't show up, then the output is shown in the bottom panel. To switch to SQL result view, press "v". To exit the SQL result view, press "q" or press "v" again.

**Step 2:** Execute the following SQL query to retrieve the response size and the request query for the logs having the request path as "/btslab/vulnerability/dor/download.php".

**Query:** select sc\_bytes, cs\_uri\_query from access\_log where cs\_uri\_stem =  
"/btslab/vulnerability/dor/download.php" order by sc\_bytes asc;

```
sc_bytes
176|file=hTtP%3A%2F%2Ftests.arachni-scanner.com%2Frfi.md5.txt%00
176|file=doc1.pdfhTtP%3A%2F%2Ftests.arachni-scanner.com%2Frfi.md5.txt%00
176|file=http%3A%2F%2Ftests.arachni-scanner.com%2Frfi.md5.txt%00
176|file=doc1.pdfhttp%3A%2F%2Ftests.arachni-scanner.com%2Frfi.md5.txt%00
176|file=tests.arachni-scanner.com%2Frfi.md5.txt%00
176|file=doc1.pdftests.arachni-scanner.com%2Frfi.md5.txt%00
176|file=any%0D%0ASet-cookie%3A+Tamper%3Df9253962-cb46-444d-899a-2142f4eaebe7
176|file=any%3F%0D%0ASet-cookie%3A+Tamper%3Df9253962-cb46-444d-899a-2142f4eaebe7
176|file=any%0ASet-cookie%3A+Tamper%3Df9253962-cb46-444d-899a-2142f4eaebe7
176|file=any%3F%0ASet-cookie%3A+Tamper%3Df9253962-cb46-444d-899a-2142f4eaebe7
176|file=any%0D%0ASet-cookie%3A+Tamper%3Df9253962-cb46-444d-899a-2142f4eaebe7%0D%0A
176|file=any%3F%0D%0ASet-cookie%3A+Tamper%3Df9253962-cb46-444d-899a-2142f4eaebe7%0D%0A
176|file=%00
176|file=_arachni_trainer_ddc5dc63e2371d6df9695175659fb79b%00
176|file=doc1.pdf_arachni_trainer_ddc5dc63e2371d6df9695175659fb79b%00
176|file=doc1.pdf%0D%0AX-CRLF-Safe-ddc5dc63e2371d6df9695175659fb79b%3A%20no
176|file=%0D%0AX-CRLF-Safe-ddc5dc63e2371d6df9695175659fb79b%3A%20no
176|file=%0D%0AX-CRLF-Safe-ddc5dc63e2371d6df9695175659fb79b%3A%20no%00
176|file=doc1.pdf%0D%0AX-CRLF-Safe-ddc5dc63e2371d6df9695175659fb79b%3A%20no%00
176|file=php%3A%2F%2Finput%00.pdf
176|file=%2Fproc%2Fself%2Fenviron%00.pdf
176|file=file%3A%2F%2F%2Fproc%2Fself%2Fenviron%00.pdf
176|file=file%3A%2F%2F%2Fetc%2Fpasswd%00.pdf
176|file=%2Fetc%2Fpasswd%00.pdf
176|file=%2F..%2F%2Fproc%2Fself%2Fenviron%00.pdf
176|file=file%3A%2F%2F%2F..%2F%2Fproc%2Fself%2Fenviron%00.pdf
176|file=%2F..%2F..%2F%2Fproc%2Fself%2Fenviron%00.pdf
176|file=file%3A%2F%2F%2F..%2F..%2F%2Fproc%2Fself%2Fenviron%00.pdf
176|file=%2F..%2F..%2F..%2F%2Fproc%2Fself%2Fenviron%00.pdf
176|file=file%3A%2F%2F%2F..%2F..%2F..%2F%2Fproc%2Fself%2Fenviron%00.pdf
```

The results indicate that the normal size of the page `"/btslab/vulnerability/dor/download.php"` was around 176 bytes.

**Step 3:** Execute the following SQL query to retrieve the number of bytes in the response for the logs having the request path as `"/btslab/vulnerability/dor/download.php"` and the query containing `"/etc/passwd"`.

**Query:** `select sc_bytes, cs_uri_query from access_log where cs_uri_stem = "/btslab/vulnerability/dor/download.php" AND cs_uri_query LIKE "%etc%passwd" AND sc_status=200 order by sc_bytes desc;`

sc_bytes	cs_uri_query
1067	file=file%3A%2F%2F%2F..%2F..%2F..%2F%2Fetc%2Fpasswd
1064	file=file%3A%2F%2F%2F..%2F..%2F%2Fetc%2Fpasswd
1063	file=%2F..%2F..%2F..%2F..%2F%2Fetc%2Fpasswd
1061	file=file%3A%2F%2F%2F..%2F%2Fetc%2Fpasswd
1060	file=%2F..%2F..%2F..%2F%2Fetc%2Fpasswd
1057	file=file%3A%2F%2F%2Fetc%2Fpasswd
1057	file=%2F..%2F..%2F%2Fetc%2Fpasswd
1054	file=%2F..%2F%2Fetc%2Fpasswd
1050	file=%2Fetc%2Fpasswd
1050	file=%2Fetc%2Fpasswd
249	file=%2Fbin%2Fcat%20%2Fetc%2Fsecurity%2Fpasswd
247	file=%2Fbin%2Fcat%20%2Fetc%2Fmaster.passwd
240	file=%2Fbin%2Fcat%20%2Fetc%2Fpasswd

The above results show that the response size varies between 1050 to 1067 bytes when `/etc/passwd` file was downloaded. Also, as it was observed before that the `download.php` page had the size around 176 bytes or slightly more (depending upon the number of characters passed in the GET request).

So, the approximate size of `"/etc/passwd"` was 1 KB.

**Q7. Which attacker machine had tried to perform the most command injection attempts using the 'exec' payload?**

**Answer:** 192.131.71.2

**Solution:**



**Step 1:** Execute the following SQL query to retrieve client IPs and the request query from the logs which contain "exec" in the request query.

**Query:** select c\_ip, cs\_uri\_query from access\_log where cs\_uri\_query LIKE "%exec%";

Tue Aug 06 01:42:41 UTC	
c_ip	
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E
192.131.71.2	query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C
192.131.71.2	query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C

**Step 2:** Execute the following SQL query to get the list of client IP addresses and the count of the log instances in which the request query contains "exec" keyword.

**Query:** select c\_ip, count(\*) from access\_log where cs\_uri\_query LIKE "%exec%" group by c\_ip;

Tue Aug 06 01:46:47 UTC	
c_ip	count(*)
192.131.71.2	404
192.131.71.6	5
192.131.71.7	1

The client at IP address "192.131.71.2" tried to perform the most command injection attempts using the "exec" payload.

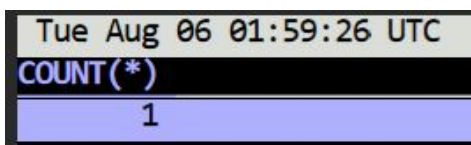
**Q8. How many onerror based XSS attacks were attempted on the web application?**

**Answer:** 1

**Solution:**

**Step 1:** Execute the following SQL query to get the number of log instances where the request query contains "onerror" keyword.

**Query:** select COUNT(\*) from access\_log where cs\_uri\_query LIKE "%onerror%";



Tue Aug 06 01:59:26 UTC
COUNT(*)
1

There was only one log instance where onerror based XSS attack was attempted.

**Note:** If the above window doesn't show up, then the output is shown in the bottom panel. To switch to SQL result view, press "v". To exit the SQL result view, press "q" or press "v" again.

**References:**

1. LNAV (<https://github.com/tstack/lnav>)