Mason Competitive Cyber

DawgCTF and UMDCTF Recap

Upcoming Events

- PatriotCTF - April 25-26
    - #patriotctf-online in Slack for team making
- NCL Recap
    - Our NCL teams placed 69th (M0nkeyBra1nZ), 91st (MasonCC), and 399th (Surviving Powell) out of 931 teams :)
- RIP VTSummit - not happening until next Fall
    - Play PatriotCTF instead (we promise it will have more easy challenges than last time)

DawgCTF

Misc: Qwerky Qwerty

Challenge text:

Oh no... whays.. ,dats hall.bing yr me... nr br brw ,df M>vv ,df BR<vvvvv Xgy ,day-o ydcovv yd.p. co a bry. cb mf dabeS U.ap bry e.ap jdcnew ydco co rbnf a ep.amvv A ep.am yday dao x..b jago.e xf JRKCE[19v Mabf 'g.oycrbo frg dak.w ,dcn. frg-k. x..b aon..lv D.p.cb ydco bry. nc.o yd. abo,.p frg o..tS Ea,iJYU?L4ydu1be3p+

Solution:

Dvorak keyboard layout to Qwerty keyboard layout

DawgCTF{P4thf1nd3r}

Misc: Qwerky Qwerty

Source Text:

Oh no... whays.. ,dats hall.bing yr me... nr br brw ,df M>vv ,df BR<vvvvv Xgv ,day-o ydcovv yd.p. co a bry. cb mf dabeS U.ap bry e.ap idcnew ydco co rbnf a ep.amvv A ep.am yday dao x..b jago.e xf JRKCE[19v Mabf 'g.oycrbo frg dak.w ,dcn. frg-k. x..b aon..lv D.p.cb ydco bry. nc.o yd. abo,.p frg o..tS Ea.iJYU?L4ydu1be3p+

Output Text:

Sj lseee ,jat:ee whak; iappenglu to mdeee lo no no, why ME.. why NOW..... But what's this.. there is a note in my hand: Fear not dear child, this is only a dream.. A dream that has been caused by COVID-19. Many questions you have, while you've been asleep. Herein this note lies the answer you seek: DawgCTF{P4thf1nd3r}

To QWERTY | To DVORAK

Forensics: Impossible Pen Test

froogle

Please choose one of the following

- SyncedIn (someone's name)
- facespace (someone's name)
- Data Breaches (not someone's name)

Search for page:

Search

OR

Corporate Website

Burke Defense Solutions & Management

Here at Burke Defense Solutions & Management, we have all the defense and solutions you could ever want or need. Our products are listed below!

- Defense
- Solutions
- Management

A message from our CEO

Special thanks to Todd Turtle from Combined Dumping & Co for babysitting our kids!

Special thanks to Mohamed Crane from Babysitting, LLC for helping us take out the trash!

Special thanks to Sonny Bridges from Oconnell Holdings for freeing up our finances!

Special thanks to Emery Rollins from Combined Finance, Engineering, Scooping, Polluting, and Dumping, Incorporated for helping make the world a better place!

- Truman Gritzwald, CEO

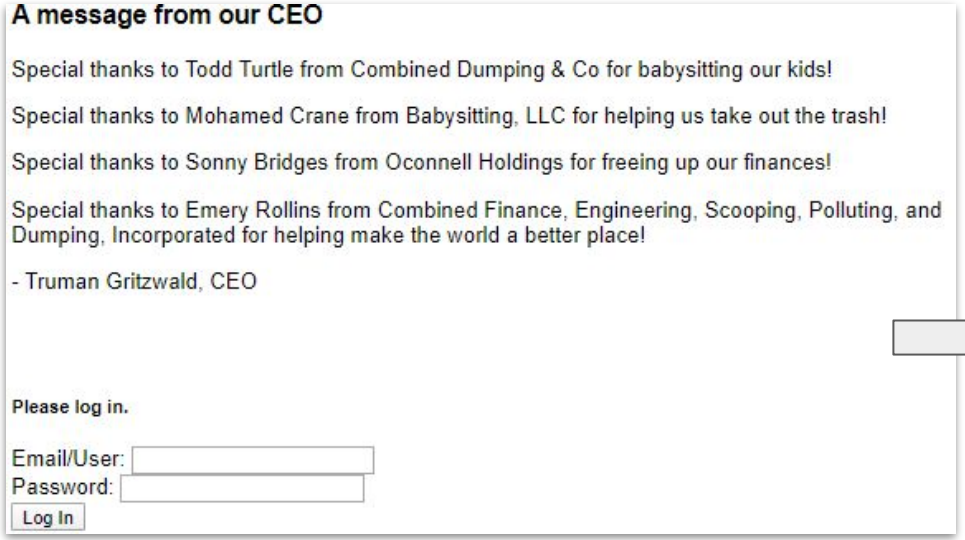
Please log in.

Email/User:
Password:
Log In

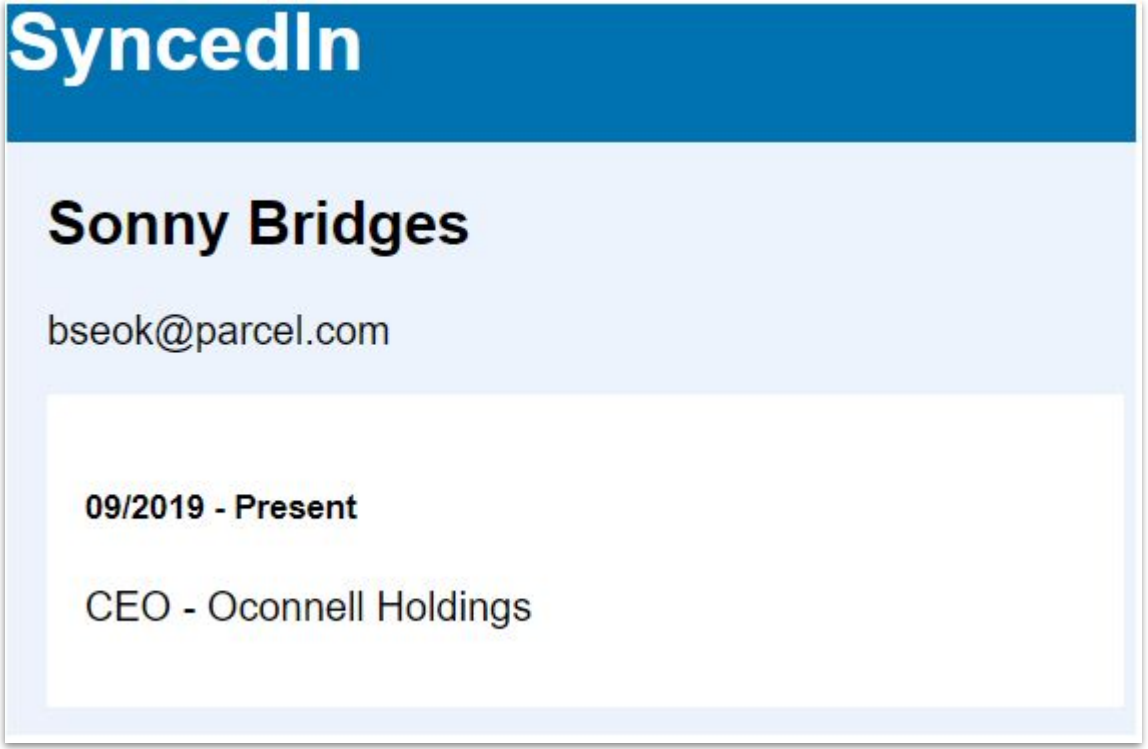
Forensics: Pen Test 1

Challenge Text:

Welcome! We're trying to hack into Burke Defense Solutions & Management, and we need your help. Can you help us find the password of an affiliate's CEO somewhere on the internet and use it to log in to the corporate site?



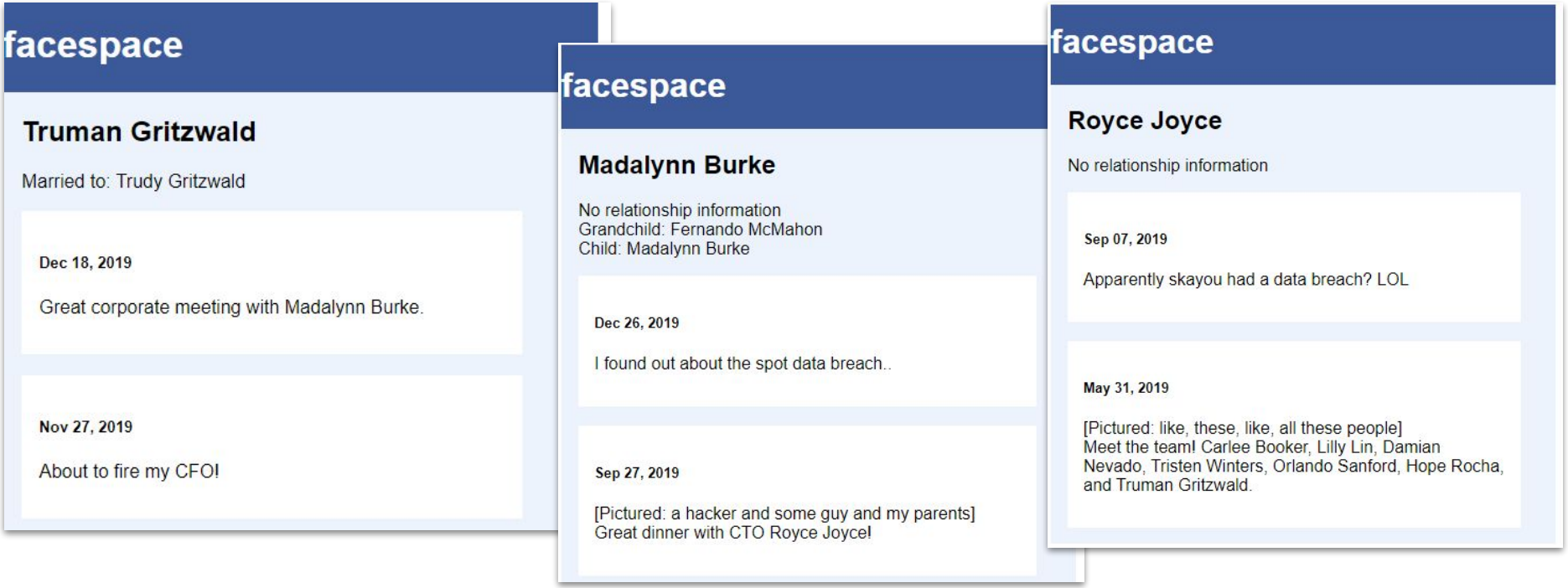
Forensics: Pen Test 1



Forensics: Pen Test 2

Challenge Text:

Welcome! We're trying to hack into Burke Defense Solutions & Management, and we need your help. Can you help us find a disgruntled former employee somewhere on the internet (their URL will be the flag)?



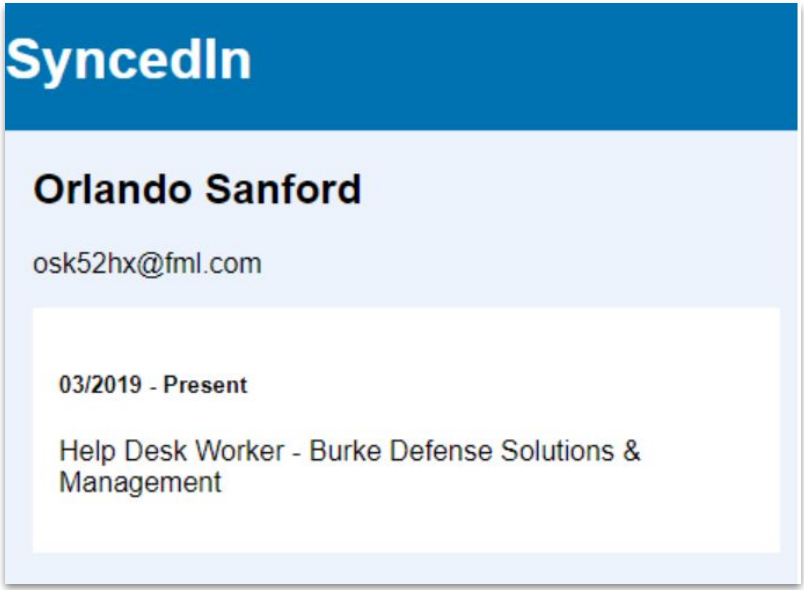
Forensics: Pen Test 2

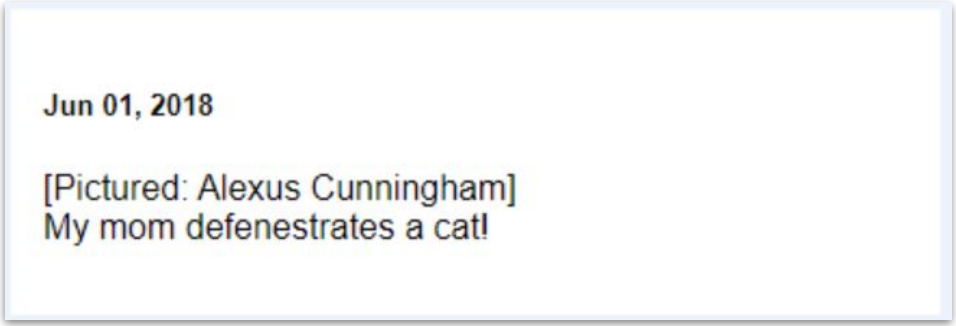
https://theinternet.ctf.umbccd.io/SyncedIn/DawgCTF{RudyGrizwald}.html

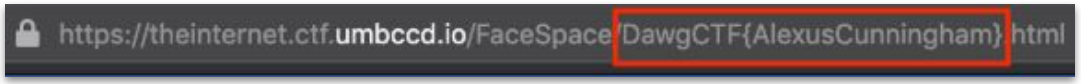
Forensics: Pen Test 3

Challenge Text:

Welcome! We're trying to hack into Burke Defense Solutions & Management, and we need your help. Can you help us find the mother of the help desk employee's name with their maiden name somewhere on the internet (the mother's URL will be the flag)?



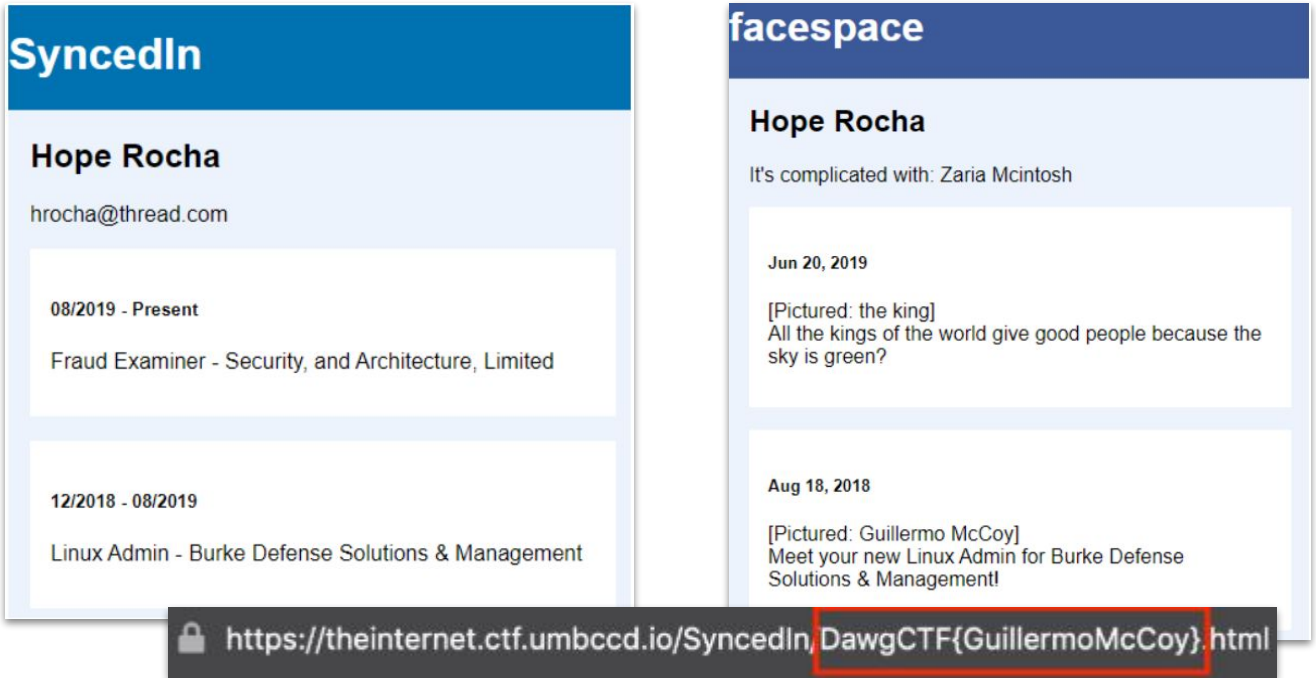




Forensics: Pen Test 4

Challenge Text:

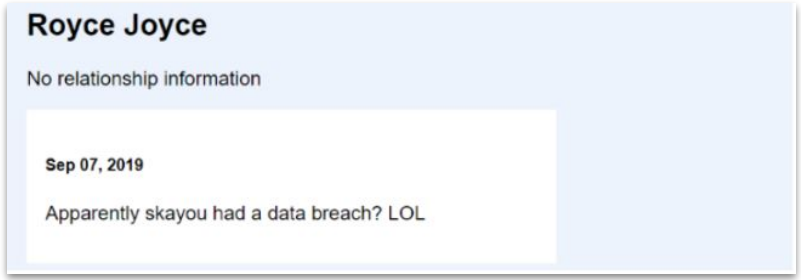
Welcome! We're trying to hack into Burke Defense Solutions & Management, and we need your help. Can you help us find the syncedin page of the linux admin somewhere on the internet (their URL will be the flag)?

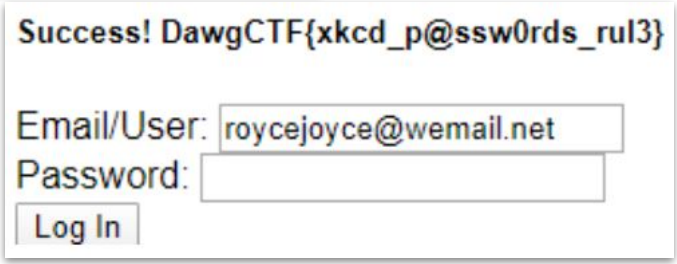


Forensics: Pen Test 5

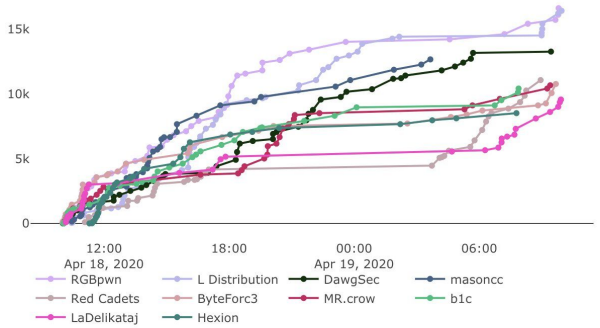
Challenge Text:

Welcome! We're trying to hack into Burke Defense Solutions & Management, and we need your help. Can you help us find the CTO's password somewhere on the internet and use it to log in to the corporate site?





UMDCTF



Place	Team	Score
1	RGBpwn	16615
2	L Distribution	16415
3	DawgSec	13265
4	masoncc	12665
5	Red Cadets	11065

Challenges are still up

https://umdctf.io/

Forensics: Sensitive

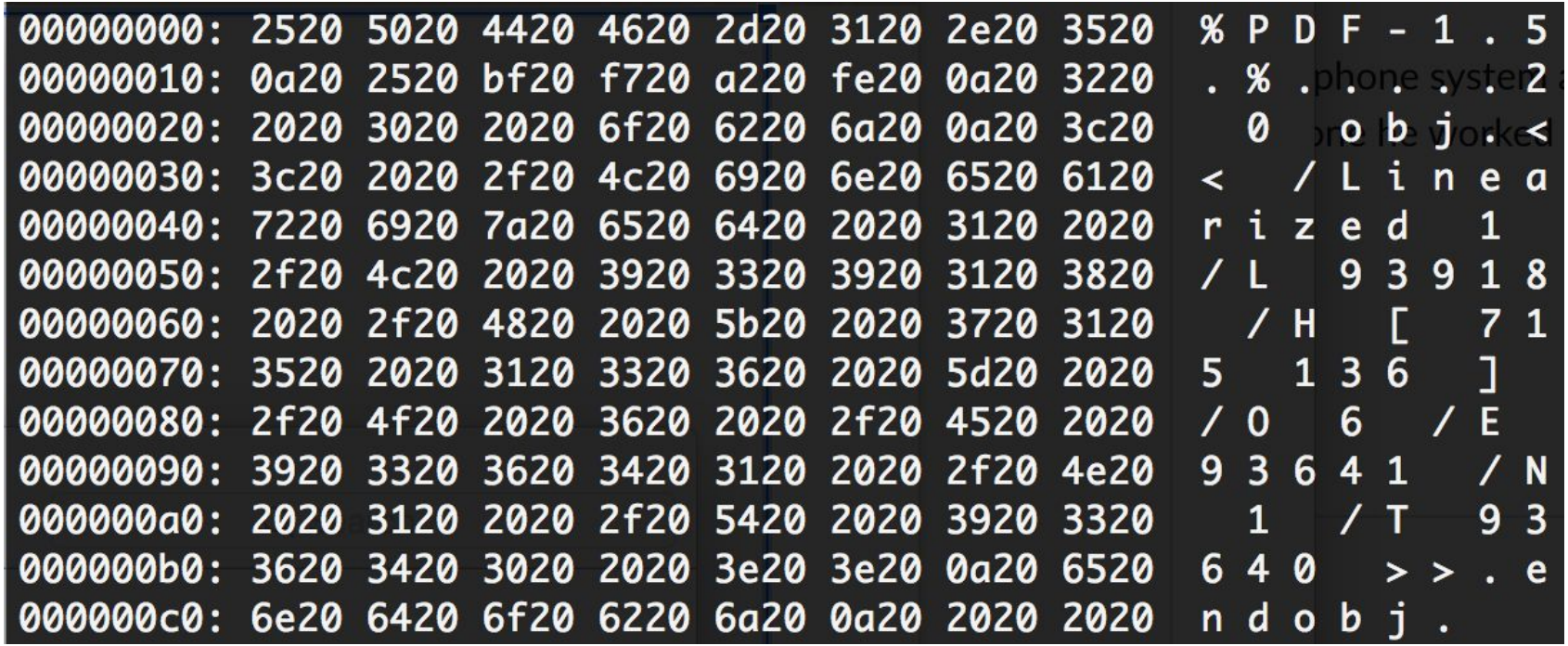
There is an ASCII text file with this thing that seems like a PDF file with a lot of extra spaces.

```
Calebs-MacBook-Pro:UMD20 calebyu$ cat sensitive
% P D F - 1 . 5
 % ? ? ? ?
 2   0   o b j
 < <   / L i n e a r i z e d   1   / L   9 3 9 1 8   / H   [   7 1 5   1 3 6   ]
   / O   6   / E   9 3 6 4 1   / N   1   / T   9 3 6 4 0   > >
 e n d o b j



 3   0   o b j
 < <   / T y p e   / X R e f   / L e n g t h   6 8   / F i l t e r   / F l a t e
 D e c o d e   / D e c o d e P a r m s   < <   / C o l u m n s   5   / P r e d i
 c t o r   1 2   > >   / W   [   1   3   1   ]   / I n d e x   [   2   1 9   ]
 / I n f o   1 5   0   R   / R o o t   4   0   R   / S i z e   2 1   / P r e v
 9 3 6 4 1                                   / I D   [ < 0 f 5 b e 0 8 c c f c c
 9 9 4 7 3 b b 1 3 5 7 b 3 4 e 8 3 1 1 a > < 0 f 5 b e 0 8 c c f c c 9 9 4 7 3 b
 b 1 3 5 7 b 3 4 e 8 3 1 1 a > ]   > >
 s t r e a m
 x ? c b d ` ? g ` b ` ` 8       " ?  ? ? F   ? ?  L 2  I F ? ?   ? ? ? ? d ?
  ?  V ?
     " % ? ? ? v  & ? ? S ? ? & 0 0 ? "  = m          <
 e n d s t r e a m
 e n d o b j
```

Forensics: Sensitive

There is a 0x20 byte in between every valid byte, so this will just require some scripting.

Pitfall: Don’t delete every 0x20 byte, some are valid!



Forensics: Sensitive

Once the PDF was fixed, I got this:

UMDCTF 2020

There's a very faint QR code, so I did some image manipulation in Mac Keynote

UMDCTF 2020

Forensics: Jarred 1

Given: .vmem, module.dwarf, and system.map

Prompt: “Jarred was working on a challenge when I took a snapshot of his VM. Can you find the flag he was working on?”

Solve: Zip up both the module.dwarf and system.map, name it something that reflects the system the dump is coming from (in this case lubuntux64), and save in ‘volatility/plugins/overlays/linux/’ directory. Use the linux plugin that dumps the most recent bash commands

```
root ~ umdctf jarred1 volatility -f lubuntu-Snapshot2.vmem --profile Linuxlubuntux64 linux_bash
Volatility Foundation Volatility Framework 2.6
Pid      Name                 Command Time                   Command
-------- -------------------- ------------------------------ -------
    1825 bash                 2019-11-13 03:59:04 UTC+0000   how do I linux?
    1825 bash                 2019-11-13 03:59:21 UTC+0000   UMDCTF-{falskdfklashdkjfhaskljfhakljsdhflkjasdhflkashdk}
    1825 bash                 2019-11-13 04:00:22 UTC+0000   echo -n "VU1EQ1RGLXtKYXJyZWRfU2gwdWxEX0hhVjNfTDBjazNkX0gxc19DT21wdTdlcn0=" | base64 -d | sha256sum
    1825 bash                 2019-11-13 04:01:32 UTC+0000   UMDCTF-{STRINgz_W0n't_Get_Th3_FLVG}
;
'
```

Forensics: Jarred 2

Given: .vmem and password protected zip file

Prompt:

Solve: Look into the memory dump and see it has been infected with the reptile module.

```
root ~ umdctf jarred2 volatility -f jarred2vm.vmem --profile Linuxlubuntux64 linux_check_modules
Volatility Foundation Volatility Framework 2.6
    Module Address       Core Address       Init Address Module Name
------------------ ------------------ ------------------ ------------------------
0xffffffffc0574400 0xffffffffc0571000                0x0 reptile_module
```

Unzip the zip file with the password `reptile`. The will give you the flag format “UMDCTF-{0xmodule_addr:uptime_when_loaded:pid_of_evil_process:attacker_ip}”

So we need to find all those components to construct our flag. Already found the module address above: 0xffffffffc0574400

Forensics: Jarred 2

Solve: To get the “uptime_when_loaded” I clarified with the author what he meant and he said that uptime = kernel time. I know dmesg has some kernel info/logs so I used the linux_dmesg plugin to dump the contents and grep’d for reptile. It asked for “loaded” so I used the 2nd time

```
root ~ umdctf jarred2 volatility -f jarred2vm.vmem --profile Linuxlubuntux64 linux_dmesg | grep reptile
Volatility Foundation Volatility Framework 2.6
[235932388794.235] reptile: module verification failed: signature and/or required key missing - tainting kernel
[235967862310.235] reptile_module: loading out-of-tree module taints kernel.
[276484713402.276]  khook_filldir64+0x58/0x70 [reptile_module]
[278871121666.278]  khook_filldir64+0x58/0x70 [reptile_module]
[278876567092.278]  khook_filldir64+0x58/0x70 [reptile_module]
[279095933010.279]  khook_filldir64+0x58/0x70 [reptile_module]
```

To get the pid of evil we need to save which process is infected by the reptile rootkit. We can do that by searching the results of the linux_psscan for anything reptile related. We find it to be at pid 8741

```
root ~ umdctf jarred2 volatility -f jarred2vm.vmem --profile Linuxlubuntux64 linux_psscan | grep reptile
Volatility Foundation Volatility Framework 2.6
0x00000000515b4500 reptile_shell        8741          -              -1             -1     0xf000f859f000e739 -
```

I did not find the evil_ip but I was told it was a local ip address so you could find it using linux_netstat plugin

Forensics: Jarred 3

Given: .vmem (windows)

Prompt: “Jarred is always having issues. He thinks he got malware from doing something dumb, but won’t tell me what he was doing?”

Solve: I first thought of possible things Jarred could've done that would be considered dumb. Click download malicious file, click malicious link, get phished. I check the IE history and see he just downloaded Tor (pid 3668) and Thunderbird (pid 424)

```
Location: Visited: Jarred Mclovin@https://www.thunderbird.net
Location: Visited: Jarred Mclovin@https://www.thunderbird.net/en-US
Location: Visited: Jarred Mclovin@https://download-installer.cdn.mozilla.net/pub/thunderbird/releases/68.7.0/win32/en-US/Thunderbird%20Setup%2068.7.0.exe
Location: Visited: Jarred Mclovin@http://www.bing.com/search?format=rss&q=how+to+buy+bitcoin&src=IE-SearchBox&FORM=IE8SRC
Location: Visited: Jarred Mclovin@http://www.bing.com/search?q=how+to+buy+bitcoin&src=IE-SearchBox&FORM=IE8SRC
Location: Visited: Jarred Mclovin@http://www.bing.com/search?q=how+to+use+tor&src=IE-SearchBox&FORM=IE8SRC
Location: Visited: Jarred Mclovin@https://www.pcworld.com/article/2686467/how-to-use-the-tor-browser-to-surf-the-web-anonymously.html
Location: Visited: Jarred Mclovin@http://www.bing.com/search?format=rss&q=tor&src=IE-SearchBox&FORM=IE8SRC
Location: Visited: Jarred Mclovin@http://www.bing.com/search?q=tor&src=IE-SearchBox&FORM=IE8SRC
Location: Visited: Jarred Mclovin@https://www.torproject.org/download/download
Location: Visited: Jarred Mclovin@https://www.torproject.org
Location: Visited: Jarred Mclovin@https://www.torproject.org/download
Location: Visited: Jarred Mclovin@https://www.torproject.org/dist/torbrowser/9.0.9/torbrowser-install-win64-9.0.9_en-US.exe
Location: Visited: Jarred Mclovin@https://dist.torproject.org/torbrowser/9.0.9/torbrowser-install-win64-9.0.9_en-US.exe
Location: Visited: Jarred Mclovin@https://dist.torproject.org/torbrowser/9.0.9/torbrowser-install-win64-9.0.9_en-US.exe
```

Forensics: Jarred 3

Solve: I first thought of possible things Jarred could've done that would be considered dumb. Click download malicious file, click malicious link, get phished. I check the IE history and see he just downloaded Tor and Thunderbird. I didn't know what Thunderbird was initially so I dumped the process and looked further into it. Find mclovin.zip gets sent and it is saved in …\Inbox directory

```
0x000000005e778300      1      0 R--rw- \Device\HarddiskVolume1\Users\Jarred Mclovin\AppData\Roaming\T
hunderbird\Profiles\rv7o17ni.default-release\Mail\pop.gmail.com\Inbox
0x00000000650aa840      3      0 RW-rw- \Device\HarddiskVolume1\Users\Jarred Mclovin\AppData\Roaming\T
hunderbird\Profiles\rv7o17ni.default-release\Mail\pop.gmail.com\Inbox
0x00000000679974e0      1      1 RW-rw- \Device\HarddiskVolume1\Users\Jarred Mclovin\AppData\Roaming\T
hunderbird\Profiles\rv7o17ni.default-release\Mail\pop.gmail.com\Inbox.msf
```

```
root ~ umdctf jarred3 volatility -f Windows\ 7\ x64\ -\ Jarred-Snapshot1.vmem --profile Win7
SP1x64 dumpfiles -Q 0x00000000650aa840 -D dump/
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x650aa840   None   \Device\HarddiskVolume1\Users\Jarred Mclovin\AppData\Roaming\Th
underbird\Profiles\rv7o17ni.default-release\Mail\pop.gmail.com\Inbox
SharedCacheMap 0x650aa840   None   \Device\HarddiskVolume1\Users\Jarred Mclovin\AppData\Roaming\Thund
erbird\Profiles\rv7o17ni.default-release\Mail\pop.gmail.com\Inbox
```

Forensics: Jarred 3

Solve: Find the zip there with the password of @uVmG4NMus.

This is as far as I got because I actually didn't see the password. I was too focused on finding the Mclovin.zip.

To solve it you need extract the zip with the password. It will give you a docx file and with oletools you find a macro with the flag inside after a reverse and caesar shift.

```
Hi Jarred Mclovin,

We are glad you chose us for your CTF Flag needs, and hope you are happy
with our product!

Password: @uVmG4NMus

Sincerely,

ALEX THROE
alex@throe.com <axel@throe.com>
SGALF LLC


DO NO REPLY TO THIS EMAIL

--000000000000e6890c05a2e08d6d
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr">Hi Jarred Mclovin,<br><br>We are glad you chose us for you=
r CTF Flag needs, and hope you are happy with our product!<div><br></div><d=
iv>Password:=C2=A0@uVmG4NMus<br><br>Sincerely,<br><br>ALEX THROE<br><a href=
=3D"mailto:axel@throe.com" target=3D"_blank">alex@throe.com</a><br>SGALF LL=
C<br><br><br>DO NO REPLY TO THIS EMAIL</div></div>

--000000000000e6890c05a2e08d6d--
--000000000000e6890d05a2e08d6f
Content-Type: application/zip; name="Mclovin.zip"
Content-Disposition: attachment; filename="Mclovin.zip"
Content-Transfer-Encoding: base64
Content-ID: <f_k8t5fbn30>
X-Attachment-Id: f_k8t5fbn30

UEsDBBQACQAIAPSGiFCgeQ8xU4MAAI2bAAARABwATG92aW5JbnZvaWNlLmRvY3hVVAkAA0w6jl63
Q45edXgLAAEE6AMAAAToAwAA5ISKblxOQNsWBnvQmA/Sqm8ph/3ekS3ylL5L55N9fT7tp8yoQrej
ejTqIJNLZernsfzuR4zpwwAWu3SDYKekpG93xkZb6zxPnGY5i/hw7glA0qVL1S2X8lpk6GNy/QqH
aOMd55xJ6mO1uqzD9wN71uilXeKWR8FVJiUDeN1XmsC0N2Hvh/hhWfqq97GLM0+j6ty2I4VmQVIQ
A3F5h4HS/Kp+epagEDHYjIJYqYY8nMM88TgjgapfOkJsdKW6oKuhS1FRrdjrZiwh8yb1zvFiYWUS
eaZ2JHp7/qYMVgigBbdNfd4AvIzKZecxDAGJq+9uPgW2spKHkp6XzXTTTkNL7Fml6GPdMG2UCjzx
bst09JZ0z6TmtikfbzQ0HiY72Gho4h++WemMtS/MxhHPB8zx49lexUEwMLTWalOYkFV7VBg2Kvdi
A+AThEiFhVN/VjojoRxKL4BtHc/INI9YneaYtLimvaEw4z08HNRKxiMuoC5Aj+m9czbxgcY+Q0st
YfOGdpb3gzQjj/Sv4Q+ZPkh6cNhgFuTkcFXBLnlUhQEk6+vwVM16afzTHuiMn1oeghx/hzPsQJg+
W7G3eZKpN6jiu/iNDjV8ihMQAoAw+8txrTnP/0RPbI0zwvu+5XOLh5DaAS6cLnU0BgXDzmGLduE1
RGEFpS+y72cI2pYERXcaiTda4sJpuW3+d4rw7KrOMm9h8uq5C/RAqRtIe/eQ2e/IYZJNJlyX55y+
```

Forensics: UMDBOMB

Solve: Password protected zip.

Prompt: IDK

Solve: Guess password. Get README file with more hints and another password protected zip. Guess password with hint. Guess password with hint. Finally you to the bomb. It is in the structure of years -> [1-12] -> [1-31].

```
root ~ umdctf umdbomb 7z x umdbomb.7z -oyears -y

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,2 CPUs Intel(R) Core(TM) i5-8600K CPU @ 3.60GHz (906EA),ASM,AES-NI)

Scanning the drive for archives:
1 file, 2113618 bytes (2065 KiB)

Extracting archive: umdbomb.7z
--
Path = umdbomb.7z
Type = 7z
Physical Size = 2113618
Headers Size = 726
Method = BZip2
Solid = -
Blocks = 60

Everything is Ok

Files: 60
Size:       2124659
Compressed: 2113618
root ~ umdctf umdbomb ls years
1840  1842  1844  1846  1848  1850  1852  1854  1856  1858  1860  1862  1864  1866  1868  1870  1872  1874  1876  1878  1880  1882  1884  1886  1888  1890  1892  1894  1896  1898
1841  1843  1845  1847  1849  1851  1853  1855  1857  1859  1861  1863  1865  1867  1869  1871  1873  1875  1877  1879  1881  1883  1885  1887  1889  1891  1893  1895  1897  1899
```

Forensics: UMDBOMB

Solve: Tried many scripts but I didn't realize the magnitude of this bomb. When you get to the last layer of the bomb each file is 805306368 bytes long with ~25 byte long potential flag at the head of the file. That would make the bomb about 17.9744 TB.

The final solve was to realize the last hint in the README file was `opening day`. With a little googling we find this sentence on a wiki page.

"On October 5, 1859, the first 34 students entered the Maryland Agricultural College, including four of Charles Calvert's sons, George, Charles, William, and Eugene. The keynote speaker on opening day was Joseph Henry, the first Secretary of the Smithsonian Institution. "

Open up that date in the bomb and get the flag.

```
root ~ umdctf umdbomb cat exrm.sh
#!/usr/bin/env bash

files=$(ls|grep -v sh)

for f in $files
do
        mv $f $f.7z
        7z x $f.7z -y
        rm $f.7z
done
root ~ umdctf umdbomb cat final.sh
#!/usr/bin/env bash

years=$(ls | grep -v sh)

compile='~/umdctf/umdbomb/compile'

format='UMDCTF'

for a in $years
do
        7z x $a -otmp
        cd tmp
        newfiles=$(ls)
        for b in $newfiles #01-12
        do
                7z x $b -onewtmp
                cd newtmp
                asciifiles=$(ls)
                for c in $asciifiles #01-31
                do
                        flag=$(head -c 25 $c)
                        if [[ "$flag" == *"$format" ]]; then
                                echo $flag
                                exit 1
                        fi
                        echo $flag >> ~/umdctf/umdbomb/compile
                done
                cd ..
                rm -rf newtmp
        done
        cd ..
        rm -rf tmp
done
~
```

```
root ~ umdctf umdbomb cat give.sh
#!/usr/bin/env bash

files=$(ls | grep -v 7z | grep -v sh)

for f in $files;
do
        cat $f | head -c 25 >> PLEASE
        echo "" >> PLEASE
done

root ~ umdctf umdbomb cat check.sh
#!/usr/bin/env bash

files=$(ls | grep -v sh | grep -v PLEASE)

for f in $files
do
        tr < $f -d '\000' >> give
        echo "" >> give
done
root ~ umdctf umdbomb
```

Web: CSEC Invasion

Three different challenges:

Still online at https://csec.umd.edu

1. Oh no! It looks like UMD CSEC's website has been invaded by Aliens! Can you help us fend them off?

2. Good job fending off the aliens on UMD CSEC's website, but it looks like some robots have started to invade as well.

3. Those robots were tough, but are you any match for the monsters that are looking to mash their way through UMD CSEC's website

Web: CSEC Invasion

Three different challenges:

1. Oh no! It looks like UMD CSEC's website has been invaded by Aliens! Can you help us fend them off?

```
<!doctype html>
<html lang="en">
▼<head>
    <meta charset="utf-8">
    <link rel="icon" href="/images/favicon.ico">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="theme-color" content="#000000">
    <meta name="description" content="University of Maryland Cybersecurity Club">
    <link rel="apple-touch-icon" href="/images/favicon.ico">
    <link rel="manifest" href="/manifest.json">
    <title>UMD CSEC</title>
    <link href="/static/css/2.cb2a3b3c.chunk.css" rel="stylesheet">
    <link href="/static/css/main.6a7f23be.chunk.css" rel="stylesheet">
  ▶<style type="text/css" data-styled-components="NUEHg kmZVBJ" data-styled-components-is-local="true">…
  </style>
 </head>
▼<body>
  ▼<noscript>
     "You need to enable JavaScript to run this app. UMDCTF-{@l13ns_@r3_b3tt3r_th@n_hum@ns}"
   </noscript>
... ▶<div id="root">…</div> == $0
  ▶<script>…</script>
    <script src="/static/js/2.ab05ba77.chunk.js"></script>
    <script src="/static/js/main.68c06c32.chunk.js"></script>
 </body>
</html>
```

Web: CSEC Invasion

Three different challenges:

2. Good job fending off the aliens on UMD CSEC's website, but it looks like some robots have started to invade as well.

https://csec.umd.edu/robots.txt

https://www.robotstxt.org/robotstxt.html

UMDCTF-{d0m0_@r1g@t0_mr_r0b0t0}

User-agent: *

Disallow:

Web: CSEC Invasion

Three different challenges:

3. Those robots were tough, but are you any match for the monsters that are looking to mash their way through UMD CSEC's website

```
<!doctype html>
<html lang="en">
▼<head>
    <meta charset="utf-8">
    <link rel="icon" href="/images/favicon.ico">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="theme-color" content="#000000">
    <meta name="description" content="University of Maryland Cybersecurity Club">
    <link rel="apple-touch-icon" href="/images/favicon.ico">
    <link rel="manifest" href="/manifest.json">
    <title>UMD CSEC</title>
    <link href="/static/css/2.cb2a3b3c.chunk.css" rel="stylesheet">
    <link href="/static/css/main.6a7f23be.chunk.css" rel="stylesheet">
  ▶<style type="text/css" data-styled-components="NUEHg kmZVBJ" data-styled-components-is-local="true">…
  </style>
 </head>
▼<body>
  ▼<noscript>
     "You need to enable JavaScript to run this app. UMDCTF-{@l13ns_@r3_b3tt3r_th@n_hum@ns}"
   </noscript>
... ▶<div id="root">…</div> == $0
  ▶<script>…</script>
    <script src="/static/js/2.ab05ba77.chunk.js"></script>
    <script src="/static/js/main.68c06c32.chunk.js"></script>
 </body>
</html>
```

Web: CSEC Invasion

Three different challenges:

3. Those robots were tough, but are you any match for the monsters that are looking to mash their way through UMD CSEC's website

https://csec.umd.edu/manifest.json


```
 "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff",
  "calc": "UMDCTF-{w3_d1d_th3_m@th}"
```

Forensics: Nation State Musical

Challenge text:

Oh no! It looks like a nation state is trying to attack one of UMDs routers! Using a pcap generated from the attack, try to determine which nation state the attack is coming from.

Forensics: Nation State Musical

No.	Time	Source	Destination	Protocol	Length	Leftover Capture Data	Info
1	0.000000	77.123.100.186	128.8.0.0	TCP	54		6000 → 5000 [SYN] Seq=0 Win=8192 Len=0
2	0.035959	77.123.100.186	128.8.0.0	TCP	54		6001 → 5000 [SYN] Seq=0 Win=8192 Len=0
3	0.092684	77.123.100.186	128.8.0.0	TCP	54		6002 → 5000 [SYN] Seq=0 Win=8192 Len=0
4	0.136617	77.123.100.186	128.8.0.0	TCP	54		6003 → 5000 [SYN] Seq=0 Win=8192 Len=0
5	0.184696	77.123.100.186	128.8.0.0	TCP	54		6004 → 5000 [SYN] Seq=0 Win=8192 Len=0
6	0.228662	77.123.100.186	128.8.0.0	TCP	54		6005 → 5000 [SYN] Seq=0 Win=8192 Len=0
7	0.288799	77.123.100.186	128.8.0.0	TCP	54		6006 → 5000 [SYN] Seq=0 Win=8192 Len=0
8	0.324771	77.123.100.186	128.8.0.0	TCP	54		6007 → 5000 [SYN] Seq=0 Win=8192 Len=0
9	0.364553	77.123.100.186	128.8.0.0	TCP	54		6008 → 5000 [SYN] Seq=0 Win=8192 Len=0
10	0.420843	77.123.100.186	128.8.0.0	TCP	54		6009 → 5000 [SYN] Seq=0 Win=8192 Len=0
11	0.456541	77.123.100.186	128.8.0.0	TCP	54		6010 → 5000 [SYN] Seq=0 Win=8192 Len=0
12	0.500829	77.123.100.186	128.8.0.0	TCP	54		6011 → 5000 [SYN] Seq=0 Win=8192 Len=0
13	0.543498	77.123.100.186	128.8.0.0	TCP	54		6012 → 5000 [SYN] Seq=0 Win=8192 Len=0
14	0.596612	77.123.100.186	128.8.0.0	TCP	54		6013 → 5000 [SYN] Seq=0 Win=8192 Len=0
15	0.660736	77.123.100.186	128.8.0.0	TCP	54		6014 → 5000 [SYN] Seq=0 Win=8192 Len=0
16	0.700838	77.123.100.186	128.8.0.0	TCP	54		6015 → 5000 [SYN] Seq=0 Win=8192 Len=0
17	0.748789	77.123.100.186	128.8.0.0	TCP	54		6016 → 5000 [SYN] Seq=0 Win=8192 Len=0
18	0.792129	77.123.100.186	128.8.0.0	TCP	54		6017 → 5000 [SYN] Seq=0 Win=8192 Len=0
19	0.863374	77.123.100.186	128.8.0.0	TCP	54		6018 → 5000 [SYN] Seq=0 Win=8192 Len=0

Wireshark · Protocol Hierarchy Statistics · attack.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▼ Frame	100.0	5001	100.0	270292	8531	0	0	0
▼ Ethernet	100.0	5001	25.9	70014	2209	0	0	0
▼ Internet Protocol Version 4	100.0	5001	37.0	100020	3157	0	0	0
▼ Transmission Control Protocol	100.0	5001	37.1	100258	3164	5000	100000	3156
Data	0.0	1	0.1	238	7	1	238	7

Forensics: Nation State Musical

Challenge text:

Oh no! It looks like a nation state is trying to attack one of UMDs routers! Using a pcap generated from the attack, try to determine which nation state the attack is coming from.

Solution:

Looked at Statistics > Protocol Hierarchy and there was only one packet which had actual data in it. I looked at the contents of the data packet and this is what I found.

```
..'.;.
.r.m. .-.f. .b.a.c.k.d.0.0.r.
.m.k.f.i.f.o. .b.a.c.k.d.0.0.r.
.n.c. .-.l.k. .1.3.3.7. .0.<.b.a.c.k.d.0.0.r. .|. ./.b.i.n./.b.a.s.h. .1.>.b.a.c.k.d.0.0.
.e.c.h.o. .'.<.5.=. .:.V.@.5.<.V.=.'. .|. .n.c. .3.7...4.6...9.6...0. .1.3.3.7.
```

Looks like the IP address is 37.46.96.0. When I googled the geolocation of the IP address, it was found to be in Kazakhstan.

Crypto: Baby's First Crypto


```
>>> for i in range(256):
...   tmp=b''
...   for j in data:
...    tmp+=bytes([(j+i)%128])
...   print(tmp)
```

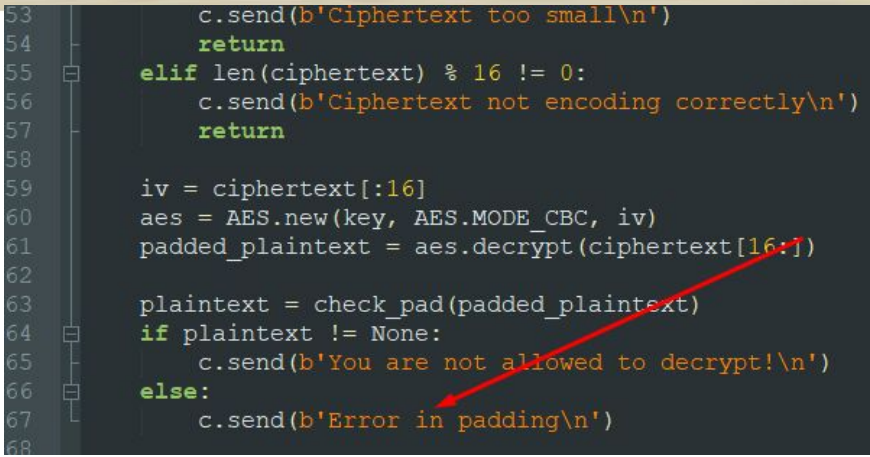
```
b'TLCBSE.z0^0tu^baxos/'
b'UMDCTF-{1_1uv_crypt0'
b'VNEDUG.|2`2vw`dszqu1'
b'WOFEVH/}3a3wxaet{rv2'
```

Crypto: Low Effort Required

Small Ciphertext Size RSA. Threw it against RsaCtfTool and the flag fell out. I don't have screenshots for this one because I had to try like 4 different RsaCtfTool installs on 3 different machines and I legitimately don't remember which one worked and *fuck* going on that goose chase again.

Crypto: Padme Twice

Crypto: Sideways Ciphering



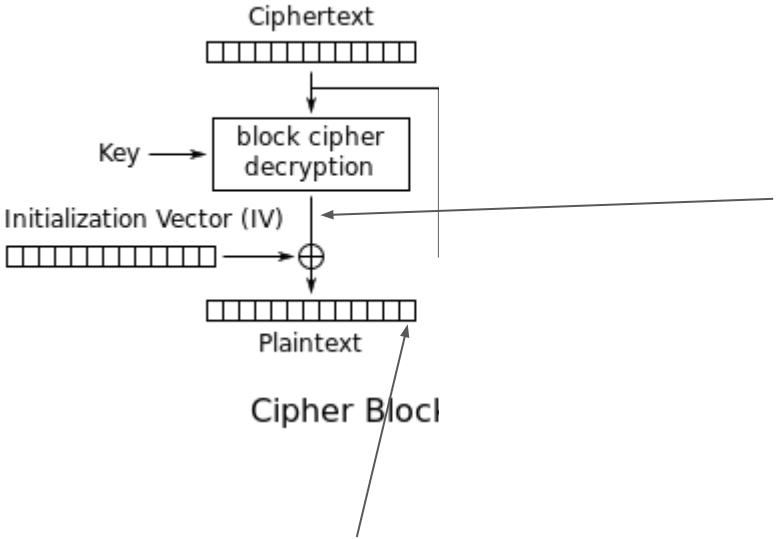
```
53          c.send(b'Ciphertext too small\n')
54          return
55      elif len(ciphertext) % 16 != 0:
56          c.send(b'Ciphertext not encoding correctly\n')
57          return
58
59      iv = ciphertext[:16]
60      aes = AES.new(key, AES.MODE_CBC, iv)
61      padded_plaintext = aes.decrypt(ciphertext[16:])
62
63      plaintext = check_pad(padded_plaintext)
64      if plaintext != None:
65          c.send(b'You are not allowed to decrypt!\n')
66      else:
67          c.send(b'Error in padding\n')
68
```

Padding Oracle attack

Service is down :(

```
zibsec@ZIBAWARE-1:~/CTF/umdctf2020$ python3 ./poracle.py
[-] Opening connection to 192.241.138.174 on port 1337: Failed
Traceback (most recent call last):
  File "./poracle.py", line 21, in <module>
    conn = remote("192.241.138.174",1337)
  File "/home/zibsec/.local/lib/python3.6/site-packages/pwnlib/tubes/remote.py", line 75, in __init__
    self.sock   = self._connect(fam, typ)
  File "/home/zibsec/.local/lib/python3.6/site-packages/pwnlib/tubes/remote.py", line 109, in _connect
    sock.connect(sockaddr)
KeyboardInterrupt
```

Crypto: Sideways Ciphering



Intermediate state xor actual IV = actual plaintext

Padding bytes must be valid

Crypto: Charlotte's Crazy Conundrum

Good ol' "Guess what I'm thinking" crypto

```
00 12 11 13 01 14 52 67 38 09 4a 3b 64 33 09 6b 38 46 6b 33 4c 09 43 38 6b 4d 33 65 38 4c 46 62 //'guessed' hex conversion from heiroglyphs
```

```
1  00 12 11 13 01 14 52 67 38 09 4a 3b 64 33 09 6b 38 46 6b 33 4c 09 43 38 6b 4d 33 65 38 4c 46 62
2
3  55 4d 44 43 54 46 2d 7b                                                                 7d
```

Crypto: Charlotte's Crazy Conundrum

```
00 12 11 13 01 14 52 67 38 09 4a 3b 64 33 09 6b 38 46 6b 33 4c 09 43 38 6b 4d 33 65 38 4c 46 62

55 4d 44 43 54 46 2d 7b                                                                 7d
```

```
//initial cribs from the flag format
0=5
1=4
2=D
3=3
4=6
5=2
6=7
7=q
8=r
9=s
a=t
b=u
c=v
d=w
e=x
f=y
```

```
55 4d 44 43 54 46 2d 7b 3r 5s 6t 3u 76 33 5s 7u 3r 67 7u 33 6v 5s 63 3r 7u 6w 33 72 3r 6v 47 7d //flag hex with placeholder variables
U  M  D  C  T  F  -  {  ?  ?  ?  0  v  3  ?  ?  ?  g  ?  3  ?  ?  c  ?  ?  ?  3  r  ?  ?  G  }  //translation using ascii table
U  M  D  C  T  F  -  {  ?  ?  ?  0  v  3  ?  p  ?  g  p  3  ?  ?  c  ?  z  ?  3  r  ?  ?  G  }     u=0
U  M  D  C  T  F  -  {  ?  _  ?  0  v  3  _  p  ?  g  p  3  ?  _  c  ?  z  ?  3  r  ?  ?  G  }     u=0, s=f
U  M  D  C  T  F  -  {  l  _  ?  0  v  3  _  p  l  g  p  3  ?  _  c  l  z  ?  3  r  l  ?  G  }     u=0, s=f, r=l
U  M  D  C  T  F  -  {  l  _  l  0  v  3  _  p  l  g  p  3  ?  _  c  l  p  ?  3  r  l  ?  G  }     u=0, s=f, r=l, t=c
U  M  D  C  T  F  -  {  l  _  l  0  v  3  _  p  l  g  p  3  ?  _  c  l  p  ?  3  r  l  ?  G  }     u=0, s=f, r=l, t=c, w=8
U  M  D  C  T  F  -  {  l  _  l  0  v  3  _  p  l  g  p  3  ?  _  c  l  p  h  3  r  l  ?  G  }     u=0, s=f, r=l, t=c, w=8
U  M  D  C  T  F  -  {  l  _  l  0  v  3  _  p  l  g  p  3  n  _  c  l  p  h  3  r  l  n  G  }     u=0, s=f, r=l, t=c, w=8, v=e
```

Proud Sponsors

CACI

EVER VIGILANT

BATTELLE

It can be done™

CRYPSIS™