SANS Holiday Hack Challenge 2019

KringleCon 2

- Write-Up –

By James Baldacchino (betaj)

Table of Contents

1. Main Objectives

a. Objective 0 – Talk to Santa in the Quad

OK – not sure whether this even warrants an entry in this write-up, but I Clicked on Santa in the "Quad" area and read what he had to say.

b. Objective 1 – Find the Turtle Doves

After some exploring around the campus, I spotted the Turtle Doves next to the fire place in the Student Union building – just dumb luck I guess.

c. Objective 2 – Unredact Threatening Document

The document is lying on the ground of the Quad (top left corner).

Text is easily un-redacted by following these steps:

- Open the pdf
  - Select all
  - Paste in a text editor

d. Objective 3 – Windows Log Analysis: Evaluate Attack Outcome

We start off by opening the Security.evtx file. Looking through its contents one immediately notices multiple failed login attempts. Furthermore the attempts have usernames advancing in alphabetical order – the hallmark of an automated attack.

Filter by Event ID 4624 brings up the successful logons. There are some domain controller logins, but we quickly get to a successful login by user: "***supatree".***

e. Objective 4 – Windows Log Analysis: Determine Attacker Technique

I opened the sysmon-dat.json file in a text editor and searched for "lsass.exe". This gives a single entry with logon_id 999.

So I ran a search for ""logon_id": 999," and there is only one other entry for process_name "ntdsutil.exe". This must be the tool used to dump the hashes from lsass.exe

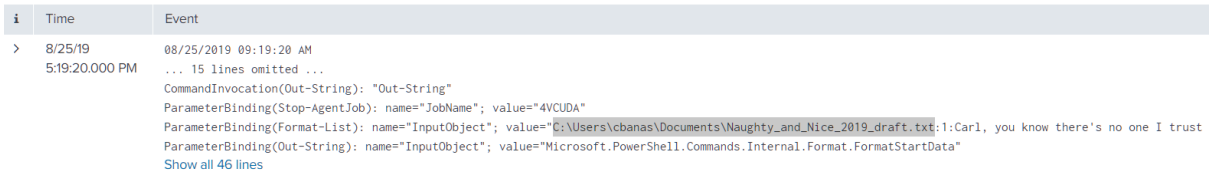
f. Objective 5 – Network Log Analysis: Determine Compromised System

I opened the index.html page in the "ELFU" folder included with the logs which conveniently brings up a RITA GUI.

Looking under "Beacons" I noticed an extraordinarily large amount of connections from **192.168.134.130** – this must be the system that's infected with malware.

g. Objective 6 – Splunk

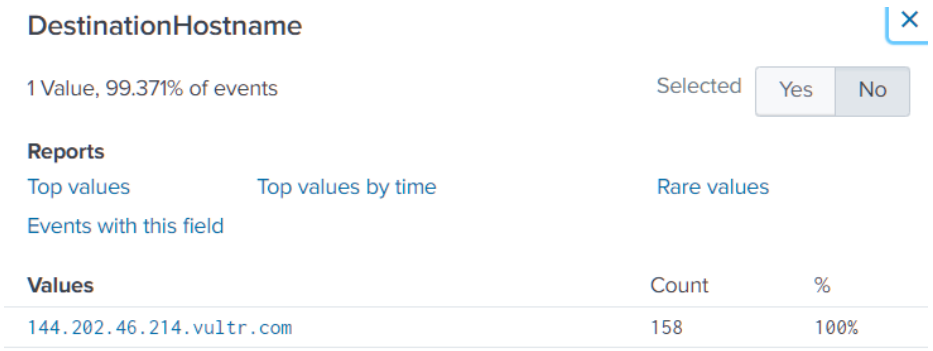
1. What is the short host name of Professor Banas' computer? ***sweetums***
    - Just read through the chat with #ELFU SOC
2. What is the name of the sensitive file that was likely accessed and copied by the attacker? Please provide the fully qualified location of the file.
   C:\Users\cbanas\Documents\Naughty_and_Nice_2019_draft.txt
    - Search for "index=main santa"
    - First entry is the following



3. What is the fully-qualified domain name(FQDN) of the command and control(C2) server? ***144.202.46.214.vultr.com***
    - Search for

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

        - Look at "Destination Hostname" under "INTERESTING FIELDS"



4. What document is involved with launching the malicious PowerShell code? Please provide just the filename. ***19th Century Holiday Cheer Assignment.docm***
    - Search for

index=main sourcetype="WinEventLog:Microsoft-Windows-Powershell/Operational" | reverse

        - Select first entry time and search +/- 5 seconds from the event
        - Run search again for "index=main sysmon"
            - This shows two process_id values: 5864 and 6268
    - Convert 5864 and 6268 to hexadecimal:
        - 5864 = 0x16E8
        - 6268 = 0x187C
    - Search for *"index=main sourcetype=WinEventLog EventCode=4688 0x16E8"*
    - Search for *"index=main sourcetype=WinEventLog EventCode=4688 0x187C"*

- This search returns a WINDOWRD process for "C:\Windows\Temp\Temp1_Buttercups_HOL404_assignment (002).zip\19th Century Holiday Cheer Assignment.docm"

```
m Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE

andatory Label\Medium Mandatory Level

s\explorer.exe
am Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Windows\Temp\Temp1_Buttercups_HOL404_assignment (002).zip\19th Century Holiday Cheer Assignment.docm" /o ""

f token that was assigned to the new process in accordance with User Account Control policy.
```

5. How many unique email addresses were used to send Holiday Cheer essays to Professor Banas? Please provide the numeric value. ***21***
   - Search for

```
index=main sourcetype=stoq | table _time results{}.workers.smtp.to results{}.workers.smtp.from results{}.workers.smtp.subject results{}.workers.smtp.body | sort -_time "Holiday Cheer Assignment Submission"
```

   - Count the number of unique email addresses under results{}.workers.smtp.from

6. What was the password for the zip archive that contained the suspicious file? ***123456789***
   - Add "password" to the search term in (5) – i.e. Search for

```
index=main sourcetype=stoq | table _time results{}.workers.smtp.to results{}.workers.smtp.from results{}.workers.smtp.subject results{}.workers.smtp.body | sort -_time "Holiday Cheer Assignment Submission" password
```

   - Password is shown in plain text:

results{}.workers.smtp.to	results{}.workers.smtp.from	results{}.workers.smtp.subject	results{}.workers.smtp.body
carl.banas@faculty.elfu.org carl.banas@faculty.elfu.org	bradly buttercups <bradly.buttercups@eifu.org> Bradly Buttercups <Bradly.Buttercups@eIfu.org>	holiday cheer assignment submission Holiday Cheer Assignment Submission	professor banas, i have completed my assignment. please open the attached zip file with password 123456789 and then open the word document to view it. you will have to click "enable editing" then "enable content" to see it. this was a fun assignment. i hope you like it! -- bradly buttercups

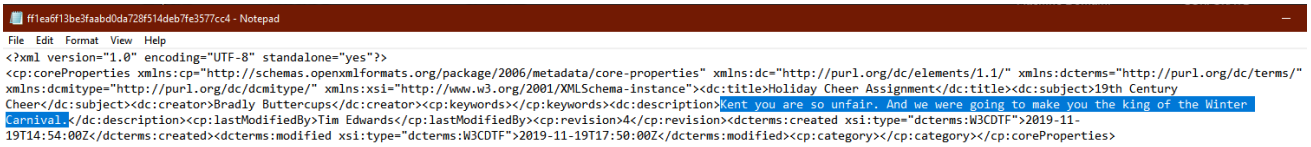
7. What email address did the suspicious file come from? ***bradly.buttercups@eifu.org***
   - The search term in (6) also gives the answer to this question

- Following the hints from Alice Bluebird we finally get to the following search term:

```
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
```

- Follow the archive path for "19th Century Holiday Cheer Assignment.docm" – i.e. "/home/ubuntu/archive/c/6/e/1/7/c6e175f5b8048c771b3a3fac5f3295d2032524af/19th Century Holiday Cheer Assignment.docm"
  - Opening the downloaded file with a text editor gives us a message pointing us towards "core.xml" instead

- So, follow the archive path for "core.xml"- i.e. "/home/ubuntu/archive/f/f/1/e/a/ff1ea6f13be3faabd0da728f514deb7fe3577cc4/core.xml"
    - Opening the file with a text editor reveals the following message:



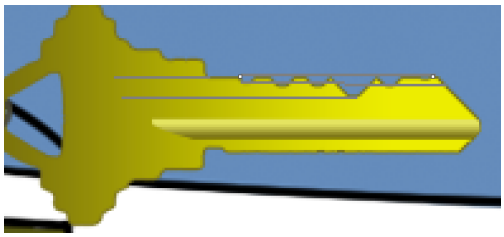
h. Objective 7 – Get Access to the Steam Tunnels

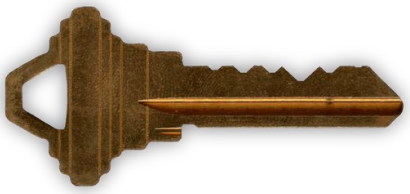
When walking into the dorm, we see a weird guy running away from us. Following him into the next room we see that he's disappeared into another room that needs a key.

Running through the above sequence again I noticed that the guy has a key on his belt and the room is conveniently equipped with a key-cutting machine. If only we could get a closer look at that key!

Bringing up the "Web Developer -> Networks" tab in Firefox we see a number of GET requests for .png files, including "Krampus.png", which leads us to https://2019.kringlecon.com/images/avatars/elves/krampus.png which is a conveniently high-res image of Krampus and his key.

Zooming in on the key (and drawing some straight lines in MS paint) and assuming that the deeper the cut, the higher the number on the cutting machine, we get to the following combination: ***122520*** which, when entered into the key cutter, produces a replica key





The replica key could then be used in the next room to gain access to the steam tunnels.

i. Objective 8 – Bypassing the Sleigh CAPTEHA

- First I sat through the suggested youtube video.
    - The video makes use of a convenient but of code at https://github.com/chrisjd20/img_rec_tf_ml_demo
    - This has a helpful README.md file with step-by-step installation and usage instructions

- Following the instructions in a linux VM terminal:

```
> sudo apt install python3 python3-pip -y
> sudo python3 -m pip install --upgrade pip
> sudo python3 -m pip install --upgrade setuptools
> sudo python3 -m pip install --upgrade tensorflow==1.15
> sudo python3 -m pip install tensorflow_hub
```

- Still following the instructions we retrain on the downloaded CAPTEHA image set

2. Other Achievements:

a. Mongo Pilfer Challenge:

When logging in to the terminal, the prompt tells us that the system is running MongoDB.

Trying to run “mongo” fails and returns a hint:

```
Hmm... what if Mongo isn't running on the default port?
```

So running “> ps –edaf” returns;

```
ID         PID  PPID  C STIME TTY          TIME CMD
elf          1     0  0 15:40 pts/0    00:00:00 /bin/bash
mongo        9     1  0 15:40 ?        00:00:02 /usr/bin/mongod --
quiet --fork --port 12121 --bind
elf         84     1  0 15:45 pts/0    00:00:00 ps -edaf
```

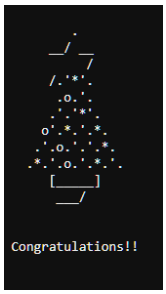
So I just run mongo again with a –port switch:

- *mongo –port 12121*

Now that we’re in mongo we can look around and we get a super helpful hint:

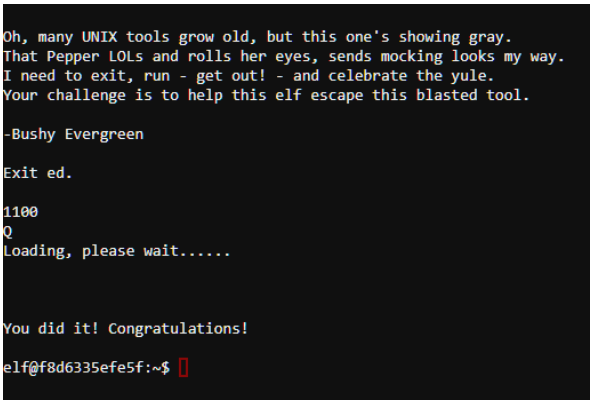
```
> show collections
bait
chum
line
metadata
solution
system.js
tackle
tincan
> db.solution.find()
{ "_id" : "You did good! Just run the command between the stars: ** db.loadServerScripts();displaySolution(); **" }
>
```

Happy to oblige:



b. Escape Ed

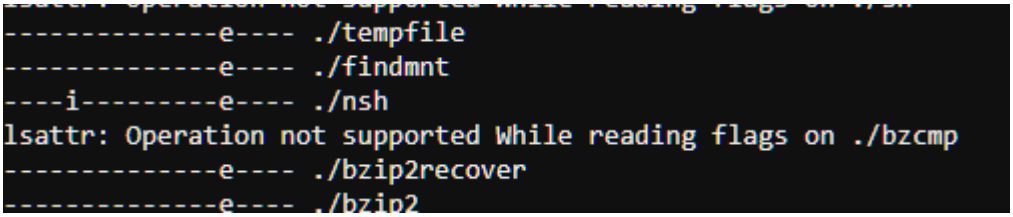
Well this was an easy one – a quick google search to learn some "ed" commands and type "Q" into the terminal – that's it!



c. Nyanshell

Running sudo –l we see that we are only allowed to run chattr as root. A quick Google search shows that this tool is used to change file attributes.

Looking at the /etc/passwd file we see that user alabaster_snowball is booting with the shell "/bin/nsh" which probably explains the Nyan Cat popping up on logon. Running > lsattr –aR in /bin shows us that there is only one immutable file in the directory and unsurprisingly it's /nsh



Chattr comes in handy now – we run > sudo chattr –i /bin/nsh to remove the immutable attribute from nsh.

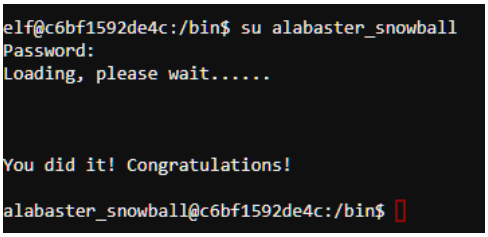
We cannot delete nsh, but we can edit it. So the solution is now quite simple:

```
> vi /bin/nsh
```

Replace the contents with:

```
#!/bin/bash
/bin/bash
```

I can now log in as alabaster snowball…



And we're in! ☺

d. Frosty Keypad:

Looking at the keypad, we can tell that the key-code is composed of the digits 1, 3 and 7. Seeing as we also know that only one of the digits is repeated once, the key-code must be 4 digits long. So we have the following parameters:

- Prime number between 1137 and 7731
- Using only the digits 7,3 and 1
- With only one repeated digit

To solve the above, I wrote the following python script:

```
##############################################################################
## SANS Holiday Hack Challenge 2019
##
## JAMES BALDACCHINO - 26-DEC-2019
##
## Frosty Keypad Challenge
## find 4 digit combinations that are prime using 1,3 and 7
## One of the digits is repeated once
##############################################################################

digits=set('137');  #these are the allowed digits based on the frost on the keypad
num=0;
x=0;

#Check if a given number is prime or not
def check_if_prime(candidate):
test=0
for x in range (2,candidate):
if (candidate % x)==0:
test= 1;
if test==1:
return 0;
if test==0:
return 1;


# Check whether a given number has repeated digits and how many times they are repeated
def repeated_digits(x):
count=0;
checkstring=str(x);
for el in digits:
count=checkstring.count(el);
if count>1:
return count;


for num in range(1137,7731):  #smallest possible number with available digits and constrains is 1137,
largest is 7731
if check_if_prime(num)==1: #If the combination is prime..
if all((d in digits) for d in str(num)) and len(str(num))==4:   #and the combination is only composed of
1,3 and 7 and is 4 digits long...
if repeated_digits(num)<3:  #and has no digits repeated more than twice
x=x+1
print(x,".  ",num, "is a PRIME candidate");

num = num +1;
print("\n\nThere are ",x," possible valid combinations.");
```

Running this script we only get 5 possible valid combinations which are easy enough to try on the keypad:

```
d.py
1 .   1373 is a PRIME candidate
2 .   1733 is a PRIME candidate
3 .   3137 is a PRIME candidate
4 .   3371 is a PRIME candidate
5 .   7331 is a PRIME candidate


There are  5  possible valid combinations.
>>>
```

e. Graylog

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?

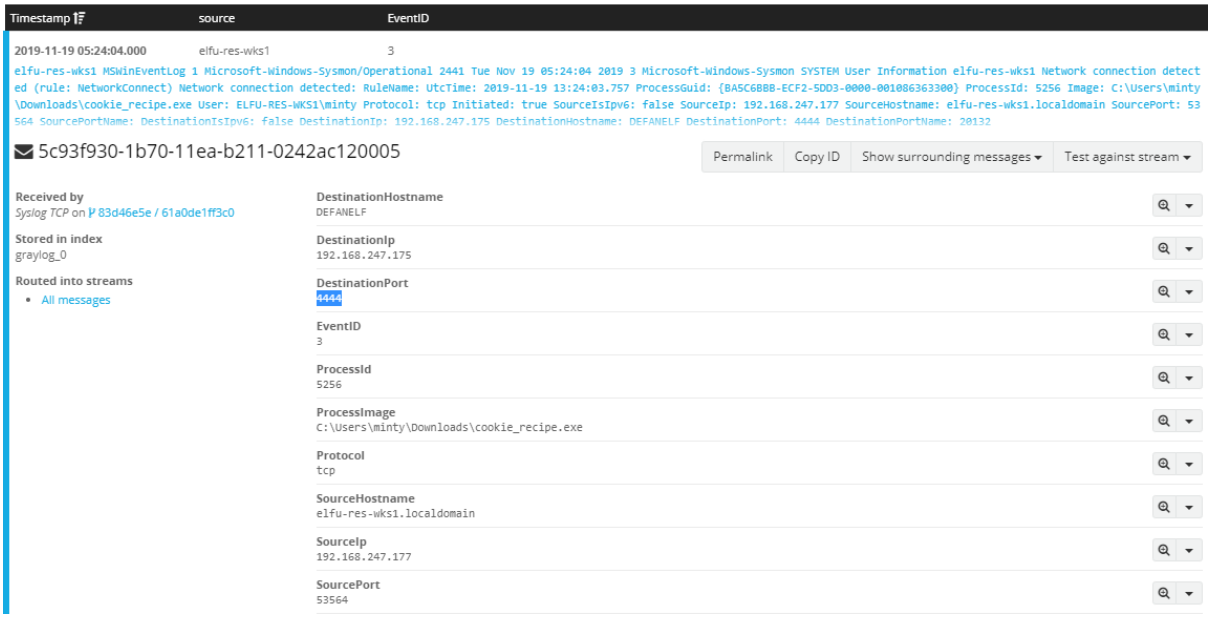
Answer: C:\Users\minty\Downloads\cookie_recipe.exe

- Search for "TargetFilename:/.cookie.+/" to find all file names with "cookie" in them

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?

Answer: 192.168.247.175:4444

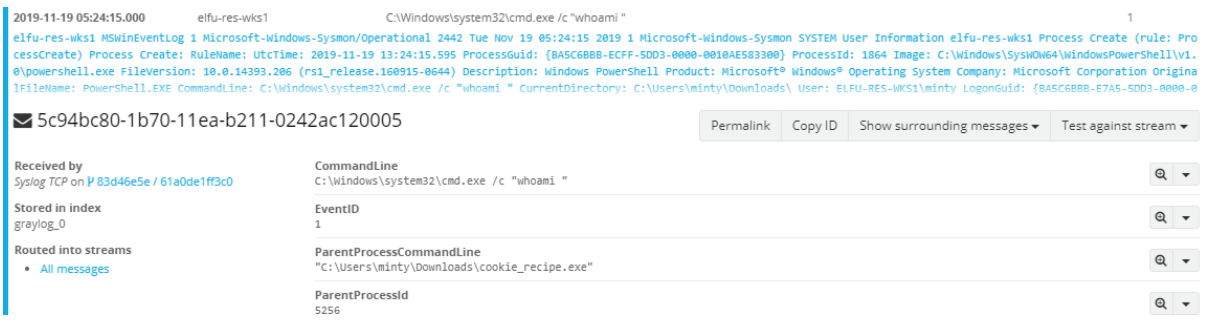
- Search for "ProcessImage:/.+cookie_recipe.exe/ AND EventID:3" to find Network Events related to cookie_recipe.exe
- This returns a single log entry:



Question 3 - What was the first command executed by the attacker?

Answer: whoami

- Search for "ParentProcessImage:/.+cookie_recipe.exe/" to find processes initiated by cookie_recipe.exe.
- Looking at the first logs we find this one:



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

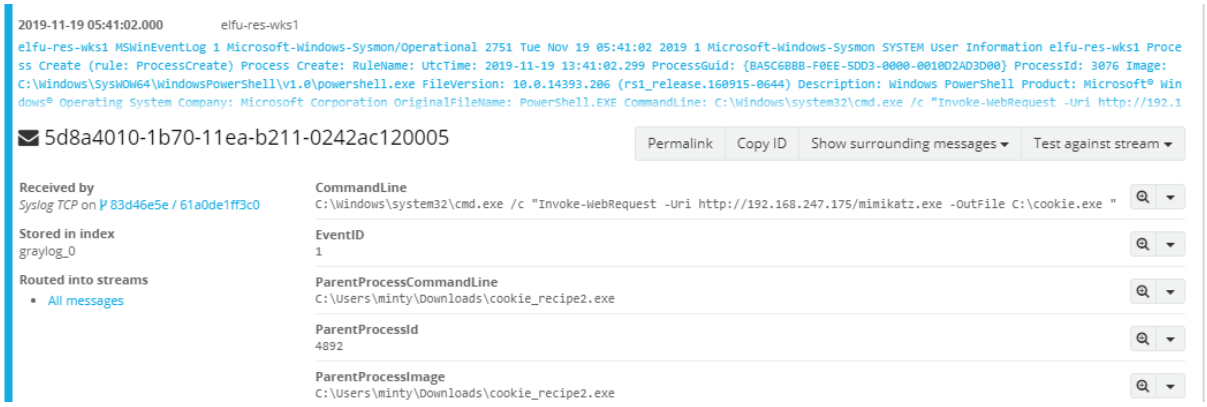
Answer: webexservice

- Looking through the results from Question 3 we see that the user runs this service

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

Answer: C:\cookie.exe

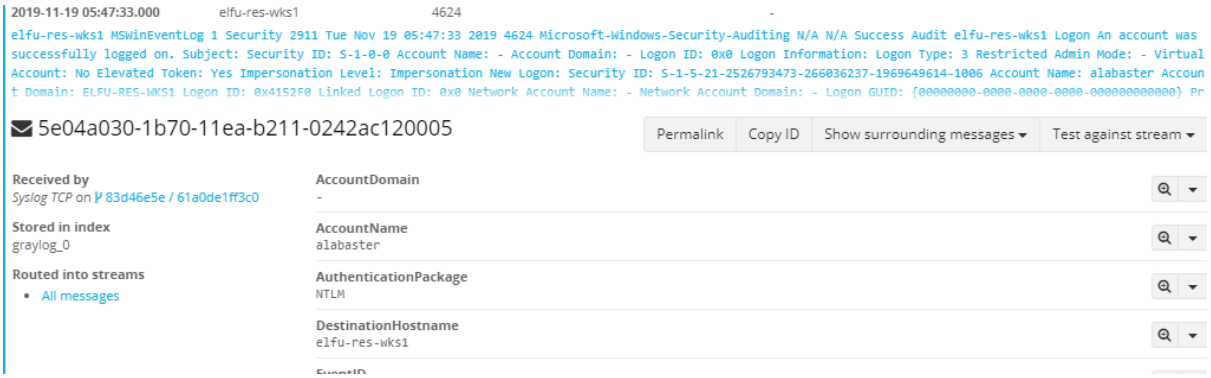
- Searching for "ParentProcessImage:/.+cookie_recipe.+/" and tracking the User over time we see that all of a sudden the user stops being "minty" right after running the webexservice.
- We see that the user then runs mimikatz with the switch "-Outfile C:\cookie.exe"



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?

Answer: alabaster

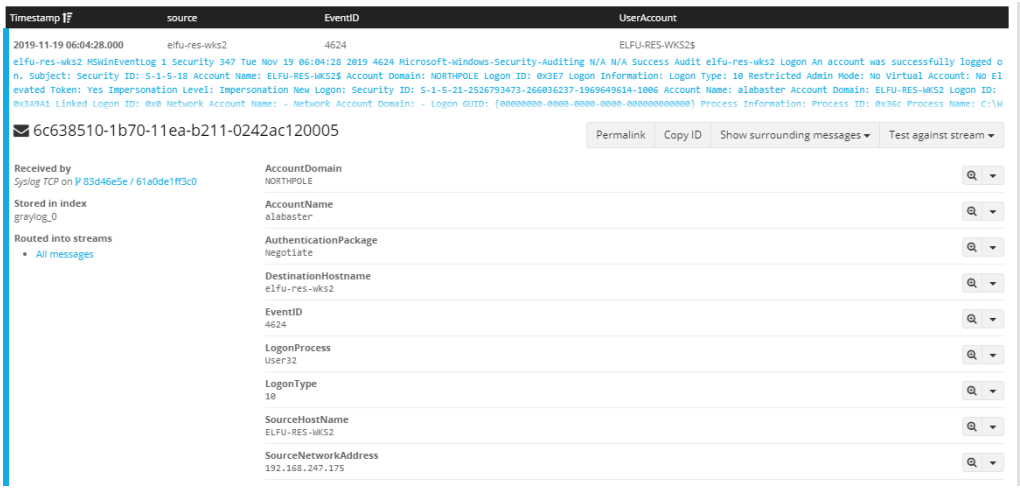
- Searching for connections from the attackers IP address 192.168.247.175 AND
- Searching for successful logon events: EventID 4624
- Search for: SourceNetworkAddress:192.168.247.175 AND EventID:4624



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

Answer:06:04:28

- Searching for connections from the attackers IP address 192.168.247.175 AND
- Searching for successful RDP connection: LogonType:10
- SourceNetworkAddress:192.168.247.175 AND LogonType:10



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?

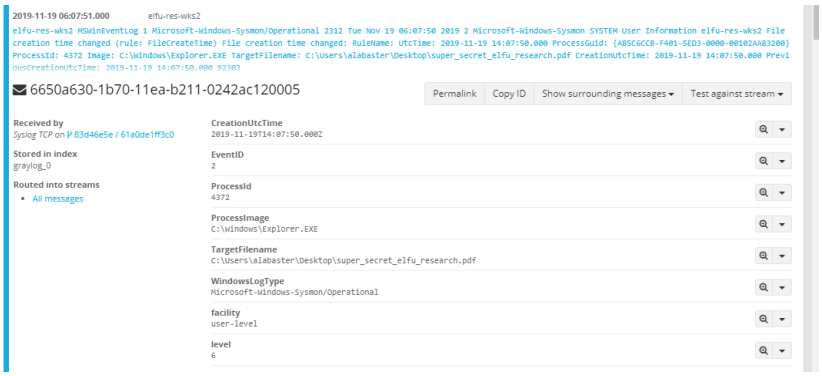
Answer: elfu-res-wks2,elfu-res-wks3,3

- Searching for succesful logon originating from ELFU-RES-WKS2: SourceHostName:"ELFU-RES-WKS2" AND EventID:4624

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

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

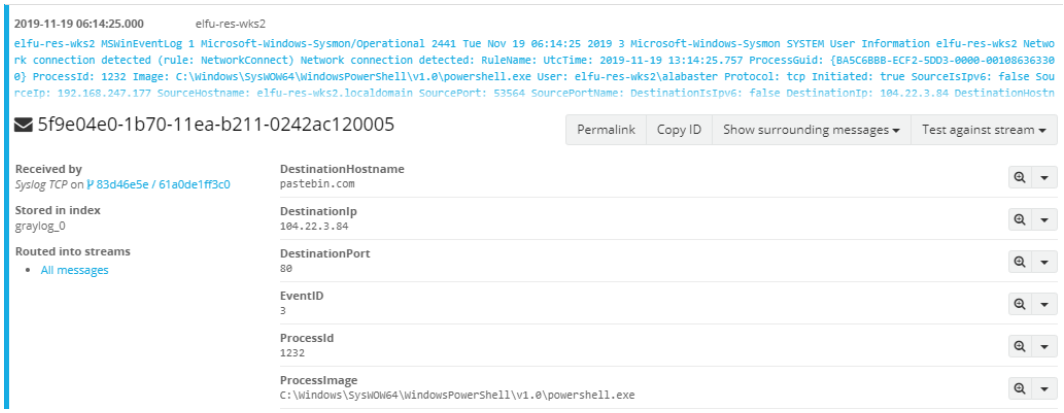
- Search for source:"elfu-res-wks2" AND EventID:2
- Look through the entries for something that is not system generated and that happened after **06:04:28**
- After just a couple of entries we come across this log:



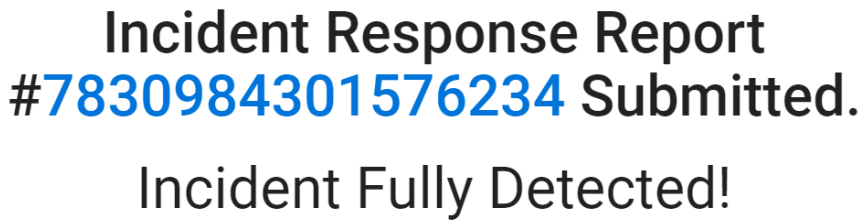
Question 10 - What is the IPv4 address (as found in logs) the secret research document was exfiltrated to?

Answer: 104.22.3.84

- Run a search for "super_secret_elfu_research.pdf"
- The most recent entry shows a powershell Invoke-Webrequest to https://pastebin.com/post.php
- Searching for logs in the surrounding 5 seconds, we find this log:



That's it – task completed!



f. IOT Braces:

Reading the contents of /home/elfuuser/IOTteethBraces.md we have a list of steps to follow:

1. Set the default policies to DROP for the INPUT, FORWARD, and OUTPUT chains.

```
> sudo iptables –P INPUT DROP
> sudo iptables –P FORWARD DROP
> sudo iptables –P OUTPUT DROP
```

2. Create a rule to ACCEPT all connections that are ESTABLISHED,RELATED on the INPUT and the OUTPUT chains.

```
> sudo iptables –A INPUT –m state –state ESTABLISHED,RELATED –j ACCEPT
> sudo iptables –A OUTPUT –m state –state ESTABLISHED,RELATED –j ACCEPT
```

3. Create a rule to ACCEPT only remote source IP address 172.19.0.225 to access the local SSH server (on port 22).

```
> sudo iptables –A INPUT –p tcp –s 172.19.0.225 –dport 22 –j ACCEPT
> sudo iptables –A OUTPUT –p tcp –s 172.19.0.225 –dport 22 –j ACCEPT
```

4. Create a rule to ACCEPT any source IP to the local TCP services on ports 21 and 80.

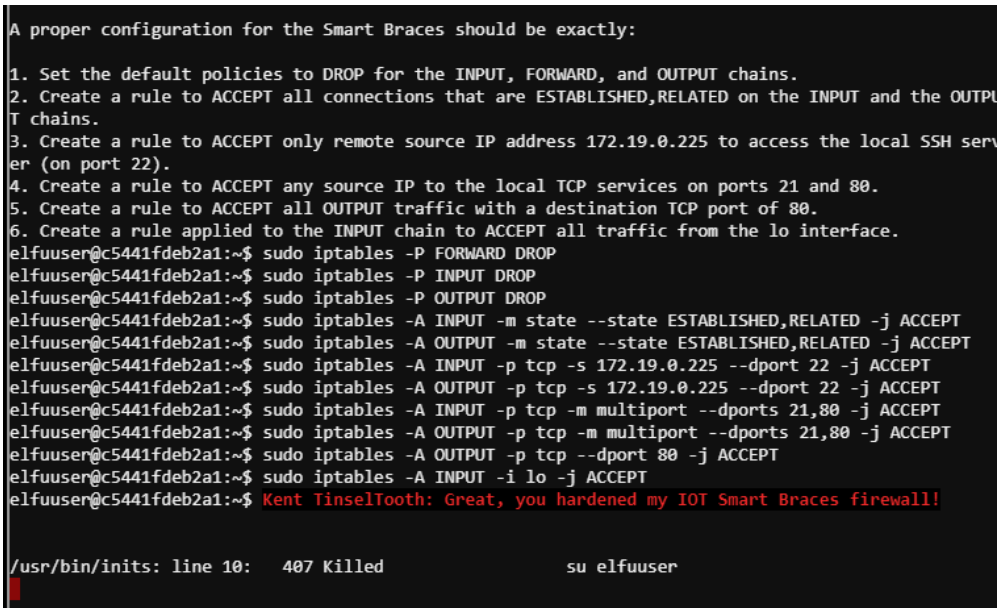
```
> sudo iptables –A INPUT –p tcp –m multiport –dports 21,80 –j ACCEPT
> sudo iptables –A OUTPUT –p tcp –m multiport –dports 21,80 –j ACCEPT
```

5. Create a rule to ACCEPT all OUTPUT traffic with a destination TCP port of 80.

```
> sudo iptables –A OUTPUT –p tcp –dport 80 –j ACCEPT
```

6. Create a rule applied to the INPUT chain to ACCEPT all traffic from the lo interface.

```
> sudo iptables –A INPUT –I lo –j ACCEPT
```



g. Linux Path

I quickly notice that someone has messed with PATH:

Running 'ls' doesn't work.

On the other hand;

> Echo $PATH

Gives: /usr/local/bin/ls.

There is something wrong with this – ls should be run in /bin

So I simply try running /bin/ls and IT WORKS!

```
Get a listing (ls) of your current directory.
elf@001b52688f4e:~$ ls
This isn't the ls you're looking for
elf@001b52688f4e:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
elf@001b52688f4e:~$ which ls
/usr/local/bin/ls
elf@001b52688f4e:~$ /bin/ls
' '   rejected-elfu-logos.txt
Loading, please wait......



You did it! Congratulations!

elf@001b52688f4e:~$
```

h. Xmas Cheer Laser

This one was particularly challenging for me as it uses Windows Powershell commands. I have absolutely no experience with Powershell so I had to do tons of Googling for every command I wanted to run.

> Get-Content /home/callingcard.txt

This gives a hint to check command history, so…

> Get-History

```
PS /home/elf> Get-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 cat /home/callingcard.txt
  11 /home/callingcard.txt
  12 echo /home/callingcard.txt
  13 Get-Content /home/callingcard.txt
  14 Get-EventL
  15 Get-EventLog

PS /home/elf>
```

It is also worth noting the entry: **angle?val=65.5** - is this the angle to use?

Running `> Get-History | Format-List –Property *` makes the output more readable.

This is particularly interesting:

```
Id                : 8
CommandLine       : Get-EventLog -Log "Application"
ExecutionStatus   : Stopped
StartExecutionTime : 11/29/19 4:56:56 PM
EndExecutionTime  : 11/29/19 4:57:14 PM
Duration          : 00:00:18.7496697

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
StartExecutionTime : 11/29/19 4:57:16 PM
EndExecutionTime  : 11/29/19 4:57:16 PM
Duration          : 00:00:00.6090308
```

Let's have a look at the environment variables

```
> Get-ChildItem Env: | Format-List
```

```
Name  : PATH
Value : /opt/microsoft/powershell/6:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
        /usr/games:/usr/local/games

Name  : PSModuleAnalysisCachePath
Value : /var/cache/microsoft/powershell/PSModuleAnalysisCache/ModuleAnalysisCache

Name  : PSModulePath
Value : /home/elf/.local/share/powershell/Modules:/usr/local/share/powershell/Modules:/opt/micros
        oft/powershell/6/Modules

Name  : PWD
Value : /home/elf

Name  : RESOURCE_ID
Value : 0e2a45bd-9d89-4e8e-a579-0ad7d0f4cac4

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.

Name  : SHELL
Value : /home/elf/elf

Name  : SHLVL
```

Looks like we're looking for a compressed file somewhere…let's follow the instructions:

```
> Get-ChildItem –R | LastWriteTime
```

And here is the latest entry:

```
    Directory: /etc/apt

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
--r---          12/27/19 12:46 PM        5662902 archive
```

Now to uncompress the archive

```
> Expand-Archive –Path /etc/apt/archive –DestinationPath /tmp
```

We now have a folder containing two files: riddle and runme.elf

Setting permissions for runme.elf and executing it, we get the following value for refraction: **1.867**

```
PS /tmp/archiveout/refraction> chmod 777 ./runme.elf
PS /tmp/archiveout/refraction> ./r
riddle     runme.elf
PS /tmp/archiveout/refraction> ./runme.elf
refraction?val=1.867
```

Let's have a look at the riddle file now:

```
PS /tmp/archiveout/refraction> Get-Content ./riddle
Very shallow am I in the depths of your elf home. You can find my entity by using my md5 identity:

25520151A320B5B0D21561F92C8F6224
```

It sounds like we need to recursively list the files in the home directory along with their MD5 hashes and compare those to this hash.

To do this we run:

> Get-ChildItem –R –File | Foreach {Get-FileHash –Algorithm MD5 $_.fullname} | where-Object {$_.Hash –eq '25520151A320B5B0D21561F92C8F6224''} | Format-List

```
PS /home/elf/depths> Get-ChildItem -R -File | Foreach {Get-FileHash -Algorithm MD5 $_.fullname} |
Where-Object {$_.Hash -eq '25520151A320B5B0D21561F92C8F6224'} | Format-List

Algorithm : MD5
Hash      : 25520151A320B5B0D21561F92C8F6224
Path      : /home/elf/depths/produce/thhy5hll.txt


PS /home/elf/depths>
```

So let's have a look at thhy5hll.txt

We have a temperature value = **-33.5** and another hint

```
PS /home/elf/depths> Get-Content /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 Len
gth.
PS /home/elf/depths>
```

So we sort the files in /home/elf/depths according to their FullName size:

```
> Get-ChildItem –R –File | Select-Object FullName, @{Name="length";Expression={$_.FullName.Length}} | Sort-Object length | select –last 1 | Format-List
```

```
PS /home/elf/depths> Get-ChildItem -R -File | Select-Object FullName, @{Name="length";Expression={
$_.FullName.Length}} | Sort-Object length | select -last 1 | Format-List

FullName : /home/elf/depths/larger/cloud/behavior/beauty/enemy/produce/age/chair/unknown/escape/v
           ote/long/writer/behind/ahead/thin/occasionally/explore/tape/wherever/practical/therefo
           re/cool/plate/ice/play/truth/potatoes/beauty/fourth/careful/dawn/adult/either/burn/end
           /accurate/rubbed/cake/main/she/threw/eager/trip/to/soon/think/fall/is/greatest/become/
           accident/labor/sail/dropped/fox/0jhj5xz6.txt
length   : 388
```

Let's have a look inside this text file:

```
rew/eager/trip/to/soon/think/fall/is/greatest/become/accident/labor/sail/dropped/fox> Get-Content
./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 .
PS /home/elf/depths/larger/cloud/behavior/beauty/enemy/produce/age/chair/unknown/escape/vote/long/
```

So let's follow the instructions:

```
> Get-Process –IncludeUserName
> Stop-Process 24
> Stop-Process 25
> Stop-Process 27
> Stop-Process 29
```

```
PS /home/elf> Get-Process -IncludeUserName

    WS(M)   CPU(s)      Id UserName                      ProcessName
    -----   ------      -- --------                      -----------
    26.79     0.47       6 root                          CheerLaserServi
   113.82     3.08      31 elf                           elf
     3.63     0.02       1 root                          init
     0.72     0.00      24 bushy                         sleep
     0.86     0.00      25 alabaster                     sleep
     0.72     0.00      27 minty                         sleep
     0.86     0.00      29 holly                         sleep
     3.25     0.00      30 root                          su
```

```
PS /home/elf> Stop-Process 24
PS /home/elf> Stop-Process 25
PS /home/elf> Stop-Process 27
PS /home/elf> Stop-Process 29
PS /home/elf> Get-Process -IncludeUserName

    WS(M)   CPU(s)      Id UserName                      ProcessName
    -----   ------      -- --------                      -----------
    27.14     0.56       6 root                          CheerLaserServi
   113.94     3.20      31 elf                           elf
     3.63     0.02       1 root                          init
     3.25     0.00      30 root                          su
```

There's a reference to "/shall/see" - /shall is a root directory so…

```
> Get-Content /shall/see
```

```
PS /> Get-Content /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.
PS />
```

Ok let's run a recursive search for an xml file in /etc/

```
> Get-ChildItem –R /etc –include *.xml
```

```
PS /etc> Get-ChildItem -R /etc -include *.xml
Get-ChildItem : Access to the path '/etc/ssl/private' is denied.
At line:1 char:1
+ Get-ChildItem -R /etc -include *.xml
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo          : PermissionDenied: (/etc/ssl/private:String) [Get-ChildItem
dAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.G
mand



    Directory: /etc/systemd/system/timers.target.wants

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
--r---         11/18/19  7:53 PM       10006962 EventLog.xml
```

There's the event log

We now need to sort and count the event IDs:

```
> Get-Content EventLog.xml | Select-String –Pattern '<I32 N="id"' | Group-Object | Select-Object –Property Count, Name | Sort-Object –Property Count -Descending
```

```
PS /etc/systemd/system/timers.target.wants> Get-Content EventLog.xml | Select-String -Pattern '<I3
2 N="id"' | Group-Object | Select-Object -Property Count, Name | Sort-Object -Property Count -Desc
ending

Count Name
----- ----
  905       <I32 N="Id">5</I32>
  179       <I32 N="Id">3</I32>
   98       <I32 N="Id">6</I32>
   39       <I32 N="Id">2</I32>
    2       <I32 N="Id">4</I32>
    1       <I32 N="Id">1</I32>

PS /etc/systemd/system/timers.target.wants>
```

There is only a single instance for event id "1" – so we need to output the lines next to this event entry to find its properties.

I used this command:

```
> Get-Content ./EventLog.xml | Select-String -Pattern '<I32 N="id">1' -Context 20,200
```

Reading through the output we find:

```
                <S N="Value">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}`n"</S>
            </Props>
```

Those look like the gas mixtures we need!!

O = 6 H = 7 He = 3 N = 4 Ne = 22 Ar = 11 Xe = 10 F = 20 Kr = 8 Rn =9

Now we're ready to input the values – let's look at the instructions for the laser again

```
> Invoke-WebRequest –Uri http://localhost:12225/).RawContent
```

```
--------------------------------------------------
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=-10

Change the mirror angle value (0 - 359):
GET http://localhost:1225/api/angle?val=45.1

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
--------------------------------------------------
```

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

```
PS /home/elf> (Invoke-WebRequest -Uri http://localhost:1225/api/output).RawContent
HTTP/1.0 200 OK
Server: Werkzeug/0.16.0
Server: Python/3.6.9
Date: Fri, 27 Dec 2019 14:13:03 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 200

Success! - 5.37 Mega-Jollies of Laser Output Reached!
```

FINALLY! - That was a tough one!

i. Holiday Hack Trail

I figure it's best to start with "Easy" and see where that gets us. I'll leave all values set to default for the time being and start the game.

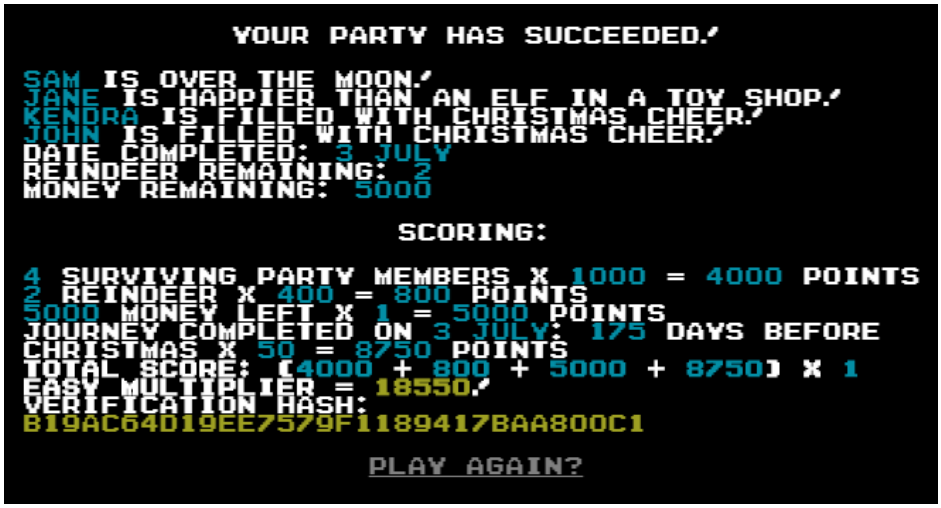
Looks like the objective is for Santa's sleigh to travel a distance "8000". It also looks like the game parameters are being passed in the URI as clear text.

Clicking on "GO", I note that the "distance remaining" drops down to 7973 (i.e it decreases by 27) and the "&distance" parameter in the URI changes from &distance=0 to &disctance=27. Next step seems obvious – I changed the &distance parameter to 8000:

*hhc://trail.hhc/trail/?difficulty=0**&distance=8000**&money=5000&pace=0&curmonth=7&curday=2&reindeer=2&runners=2&ammo=100&meds=20&food=392&name0=Sam&health0=100&cond0=0&causeofdeath0=&deathday0=0&deathmonth0=0&name1=Jane&health1=100&cond1=0&causeofdeath1=&deathday1=0&deathmonth1=0&name2=Kendra&health2=100&cond2=0&causeofdeath2=&deathday2=0&deathmonth2=0&name3=John&health3=100&cond3=0&causeofdeath3=&deathday3=0&deathmonth3=0*

and "Distance Remaining" dropped down to 0 .

I clicked on "GO" one last time and that's it:

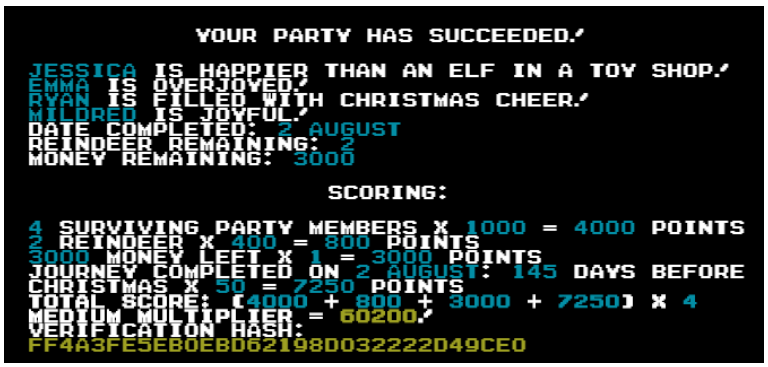


Just for fun I decided to try my hand at the Medium difficulty level next. This time around the parameters are not shown in the URI. However a quick look at the page source whilst playing the game reveals an element “ which is being updated with every run.

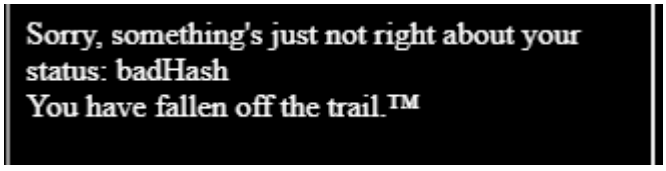
Sure enough the container contains all the game parameters in clear text, so once again, I simply update

*<input type=”hidden” name=”distance” class=”distance” value=”**8000**> == $0*

And that does the trick:



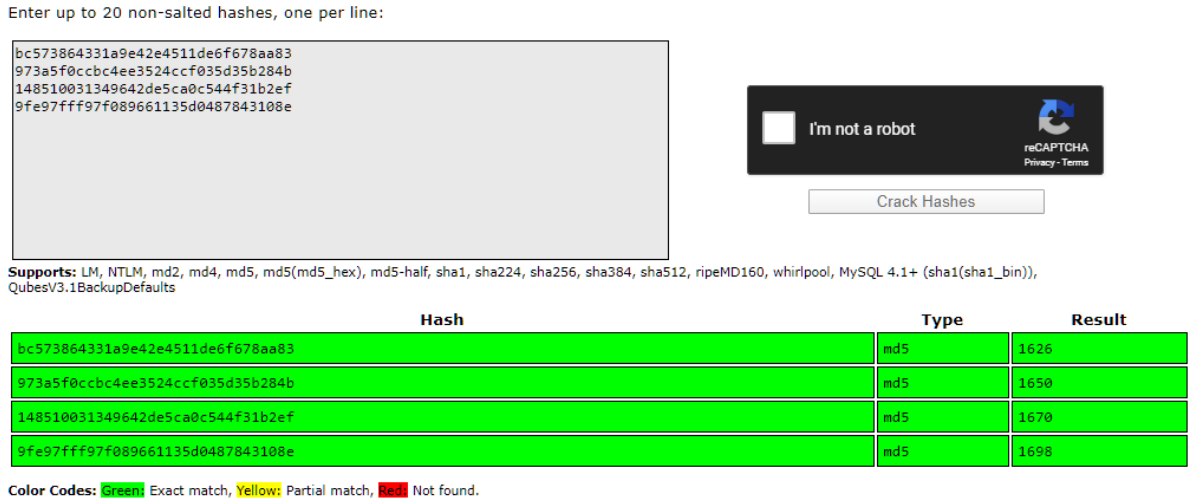
Now to attempt the Hard Mode – at first glance everything looks identical to “Medium” mode, but when editing the source I get an error saying “status: badHash”. Looks like the game is a bit smarter now



On closer inspection of the “statusContainer” element I notice that there is a new variable at the end called “hash”. The hash seems to change with each step progression of the game

I searched for an online hash cracker to see if this hash could give me some information and settled on https://crackstation.net/

With this tool the hashes where identified as MD5 and gave a number which was “1626” at the start of the game and then increased by a seemingly arbitrary amount with every turn.



So the game must be generating a MD5 hash based on all the game parameters at each turn and submitting this to the server along with the parameters. The server will return an error and stop the game if the hash does not match the given parameters.

So, I ran through a few game steps and recorded the parameter values in a table, along with the cleartext value of the generated hash

money	1500	1500	1500	1500
distance	0	31	58	93
curmonth	9	9	9	9
curday	1	2	3	4
reindeer	2	2	2	2
runners	2	2	2	2
ammo	10	10	10	10
meds	2	2	2	2
food	100	92	84	76
Unhashed:	1626	1650	1670	1698

It was quickly apparent that the hash value was being calculated simply by adding the values of all the other parameters (excluding the difficulty level and the runners’ health level).

The next step was quite easy – I changed the parameter values for the following:

Distance = “8000”

Curday = “1”

Food = “100”

Note: I could have just changed the distance, but in for a penny, in for a pound, am I right?

Then worked out the checksum by adding all the parameters:

money	1500
distance	8000
curmonth	9
curday	1
reindeer	2
runners	2
ammo	10
meds	2
food	100
Sum:	**9626**

And then generating a MD5 hash of the checksum using https://www.md5hashgenerator.com/

Your Hash: **649d45bf179296e31731adfd4df25588**
Your String: 9626

I plugged all the parameters (including the new hash) into the browser’s developer console and clicked on “GO”

```
<input type="hidden" name="name3" class="name3" value="Anna">
<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=
"649d45bf179296e31731adfd4df25588"> == $0
</div>
```

That’s it – mission accomplished with a beautiful score of 96000 ☺

```
YOUR PARTY HAS SUCCEEDED.'
LILA IS FILLED WITH CHRISTMAS CHEER.'
CHRIS IS WICKED PSYCHED.'
JEN IS FILLED WITH CHRISTMAS CHEER.'
ANNA IS HAPPIER THAN AN ELF IN A TOY SHOP.'
DATE COMPLETED: 2 SEPTEMBER
REINDEER REMAINING: 2
MONEY REMAINING: 1500

SCORING:

4 SURVIVING PARTY MEMBERS X 1000 = 4000 POINTS
2 REINDEER X 400 = 800 POINTS
1500 MONEY LEFT X 1 = 1500 POINTS
JOURNEY COMPLETED ON 2 SEPTEMBER: 114 DAYS
BEFORE CHRISTMAS X 50 = 5700 POINTS
TOTAL SCORE: (4000 + 800 + 1500 + 5700) X 8
HARD MULTIPLIER = 96000.'
VERIFICATION HASH:
57EC46350CE6DBD9881127DD6D102CFB
```