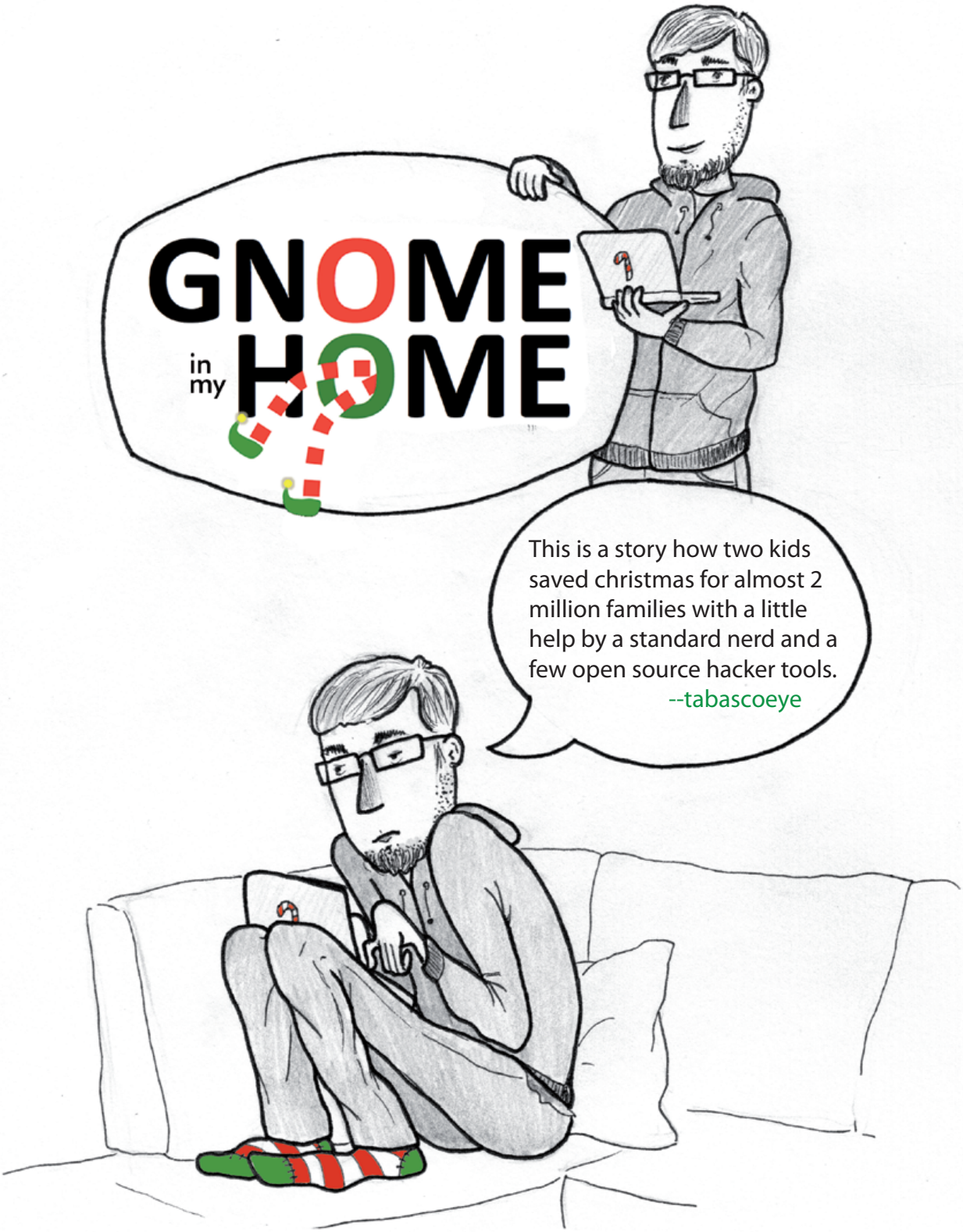


The 2015 SANS Holiday Hack Challenge Diary

Introduction

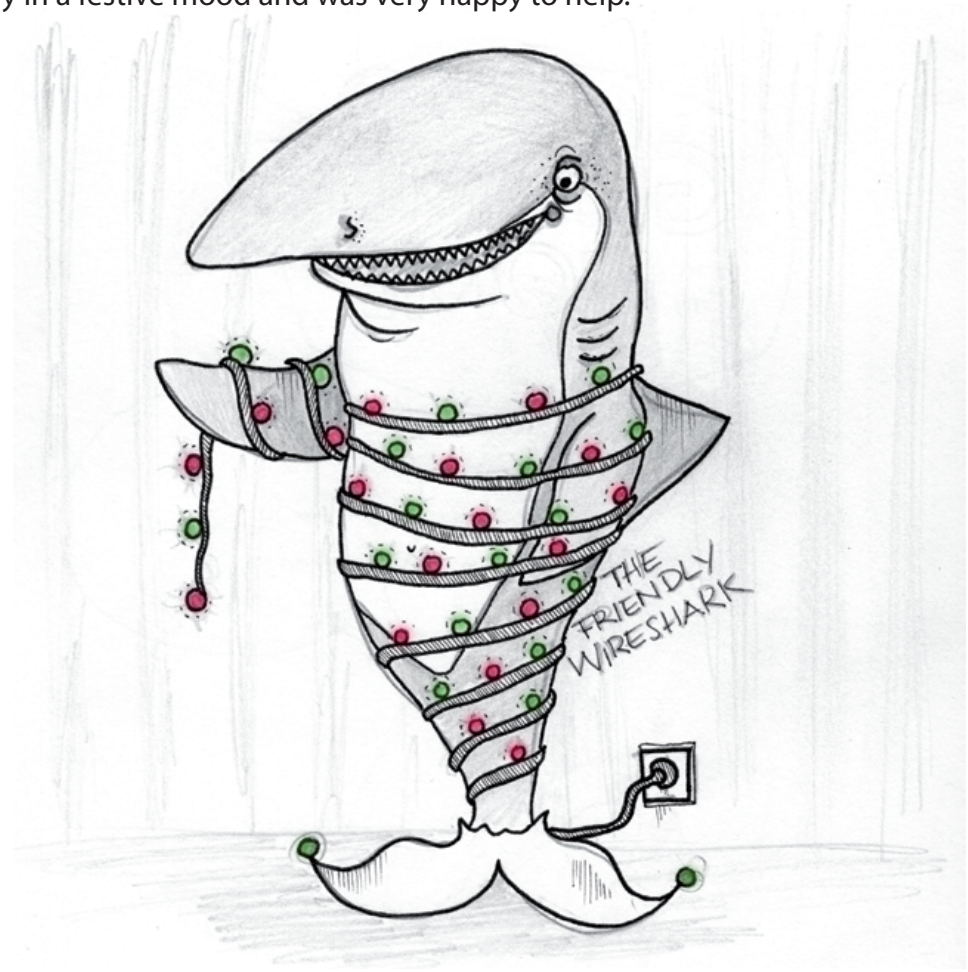
A few weeks before christmas, I was suprised to be contacted by the Dosis family about a suspicious toy they got.
Since my xmas vacation was only a few days away and I was still working on the SANS ICS Security Challenge, I told Josh that I would look at his packet capture when I find the time and almost forgot about it.
A few days later, I stumbled across tweets about gnomes in my feed and remembered that I promised to help Josh and Jess, so I got into my 2015 FireFox and drove to the Dosis neighborhood to find what the fuzz about this toy really was all about.

Part 1: Dance of the Sugar Gnome Fairies:

Since I didn't remember which of the houses was the Dosis residence, I was happy to find Lynn standing around in the neighborhood. She helped me to find Josh and also noted that a Netwars tournament was going on in a nearby hotel, which also meant that many SANS people were around in case I needed more help with this project.

Josh was really excited about a capture file that he had and told me something about the gnomes being very busy on the wireless interface. He noted that the gnomes were a huge hit and completely sold out everywhere. I though about the infamous "Hello Barbie" and was suddenly eager to check out if these kids found something interesing in this toy.

I called up my buddy the friendly wireshark to help me analyze the pcap. He was already in a festive mood and was very happy to help.



"Well look at those weird DNS requests" he said after filtering out the 802.11 beacon frames. "Those are not normal and contain an odd amount of data"
I immediately had a thought after looking at the content "It looks like base64. Maybe it really is a command&control channel".
"Can you extract only those text parts for me, sharky?" I asked the friendly wireshark.
"Well. I'm a very GUI oriented guy and it might not be that easy to extract these specific parts on all those packets. But my brother Tshark might be able to"
After calling his brother with

```
tshark -T fields -e dns.txt -r giyh-capture.pcap > dns_base64
```

He gave us a single file containing all the data, which I could send through the base64 decoding

```
base64 -d dns_base64 > dns_decoded
```

And with that, all doubts were gone. This "toy" was being controlled remotely, and it sniffed around for WIFI and even sent back a file called "snapshot_CURRENT.jpg".
"Wow," I said "they were issuing commands."
"What? Which ones?" Josh and Jess wanted to know. "Well" I said "these:"

```
EXEC:iwconfig
EXEC:cat /tmp/iwlistscan.txt
FILE:/root/Pictures/snapshot_CURRENT.jpg
```

"They scanned our WIFI and got a file?!" Josh was shocked.
"What is inside that image?!"

So I isolated the binary data of the image, removed all the newlines and search&replaced all instances of "FILE:" with nothing. After I opened the file, the kids were even more shocked:

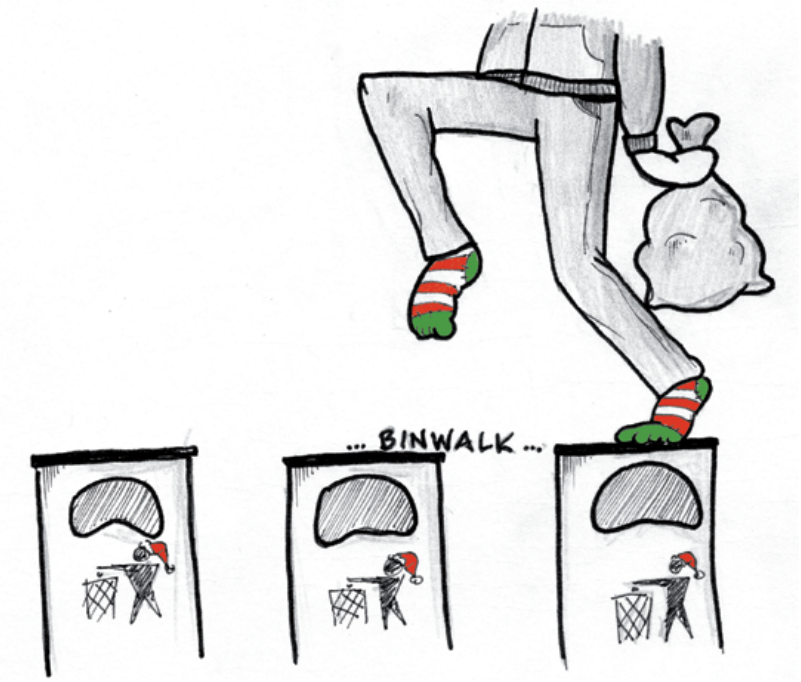
But after the shock changed back to curiosity, Jess whipped out her Xeltek SuperPro 6100 to get the firmware of the annoying gnome in their home.
"While you work on this, I'll check out the neighborhood and talk to some of the SANS people. Maybe I can learn a thing or two that will help us find out whats going on inside this gnome." I said and headed out.

Interlude:

After mapping out the neighborhood, I ran around the block to get all the treats that the SANS people needed. A gift for Dan, blinky lights for Tom, a cookie for Jeff, a mint for Josh, hot chocolate for Tim.
And finally, Ed wanted me to find his intern. I found a PIN code for the most mysterious building in the neighborhood but it only lead into a labyrinth. After going "up, up" intuitively, I found another "up" just leads back to the street, so I tried "up, up, down" which worked and I started to have a gut feeling combined with a smile. As I tried "up, up, down, down, left, right, left, right" I laughed really hard at the nice reference chosen to hide the NOC.
(MUCH later, I found that Jeff would have given an obvious hint on the Konami code. Oh well…)

Part 2: I'll be Gnome for Christmas:

After I returned to the Dosis residence, Jess had already pulled the full firmware of the gnome and gave me the file asking "what can we do with it now?"
"Well, I'll do the binwalk now"
Josh was laughing and told me how he imagined me walking across a couple of bins.



"No binwalk is actually a tool that goes through the whole binary and looks for magic values that are known to be beginnings of files. Basically it's like someone taking a looking glass and finding file types contained in the image."

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

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PEM certificate
1809          0x711           ELF, 32-bit LSB shared object, ARM, version 1 (SYSV)
116773        0x1C825         CRC32 polynomial table, little endian
168803        0x29363         Squashfs filesystem, little endian, version 4.0,
compression:gzip, size: 17376149 bytes, 4866 inodes, blocksize: 131072 bytes, blocksize:
131072 bytes, created: 2015-12-08 18:47:32
```

"Well. This looks like an embedded Linux device" I said. "I'll extract this filesystem and take a look around."
I found a full root filesytem with several components in there. It looked like a custom made OpenWRT Linux variant which runs on an ARM Cortex A9 CPU. There were even large parts of a webinterface in there, using node.js as a webserver and express.js as a web framework with bootstrap and jquery for the front-end.

I even found a MongoDB database in there. "Hmm does that database contain passwords?" Jess asked. "Well it's a big binary file, since mongodb works different from normal SQL, but I can simply run the strings command against it to see if there is any clear text in there" I said and found the admin account with that: admin:SittingOnAShelf

Part 3: Let it Gnome! Let it Gnome! Let it Gnome!

"So if these talk to a command&control sever, can we find it?" Josh asked excitedly.
"Well, if we find something that makes this webserver unique, we can search for that on shodan."
"On what?" Jess wanted to know.
"Well Shodan is a search engine based on scans of the whole internet and you can find devices with open services (ports) all over the IPv4 address space" I explained
"That's cool!" both kids exclaimed.

After I found the Title of the admin webpages to be "GIYH::ADMIN PORT V.01" from /www/routes/index.js, I searched shodan.io for "title:GIYH" and got 5 results back. Connecting to one of the addresses revealed the first "SuperGnome".

"Gosh! Five of them? Where are they?"

- SuperGnome 04, 52.192.152.132, Japan, Tokyo
- SuperGnome 01, 52.2.229.189, United States, Ashburn
- SuperGnome 05, 54.233.105.81, Brazil, Sao Paulo
- SuperGnome 03, 52.64.191.71, Australia, Sydney
- SuperGnome 02, 52.34.3.80, United States Boardman

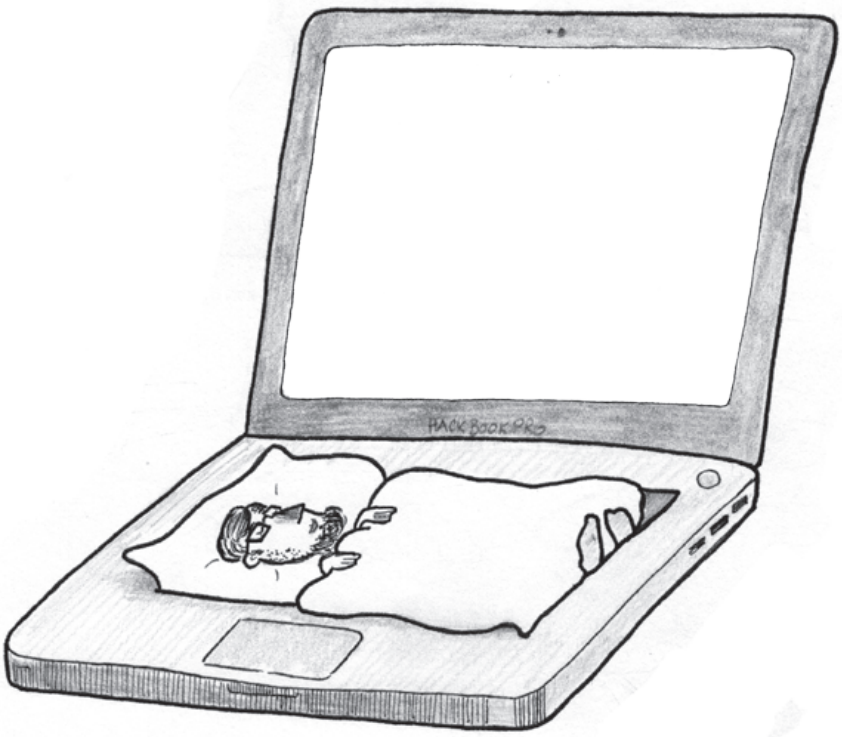
"We need to find out more about this. Can you help us some more?" Josh asked.
"Well I'll need to look for flaws in the firmware to attack those SuperGnomes. We'll start at Number 1." I explained and got going.

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

After digging through the firmware, I realized that a number of people have been working on this "project" so I used grep on the firmware with the names I found to get some pointers about the files which where explicitly coded/modified by them. Stuart, Louise, Auggie and Nedford left several comments in the firmware and it looked like they were stressed and running out of time because they left interesting problems in there:

SuperGnome 1 was fully accessible with the admin:SittingOnAShelf credentials. So I just downloaded all the files from the "file" view. I took a screenshot of the "Gnome-NET" page for later and exclaimed victory "Well that was easy, kids".
"What did you find?" asked Josh. "Well the gnome.conf contains NCC1701 which is the serial number of the USS Enterprise from StarTrek"

SuperGnome 2 did let me in, but file download was disabled. "What now?" asked Josh. "Well I can see that there is an upload mechanism in /settings/ enabled. let me browse the source code we got from the firmware image."
So I found that the settings upload is actually broken (/www/routes/index.js, line 127-151), but before the upload, a new directory is created according to the name submitted. "Hmm looks like a dead end" I sighed.
Then I shifted my focus on the separate camera viewer, which wasn't linked in the admin webinterface, but index.js showed in line 182-198 that /cam had a handler. After seeing those comments and unfinished code, I wondered which version would be running on SG02, so I tried "/cam?camera=.png_foo" and could see in the error message, that the request hadn't been corrected to ".png_foo.png", so SG02 still ran the code that was already corrected in our version.
"I just need to put the .png in the beginning and then I can do directory traversal, Josh!" I cried. "But I would need to have a directory named .png for that or something. That'll never work"
Both paths seemed to lead nowhere and I decided to sleep over it as it was getting really late...

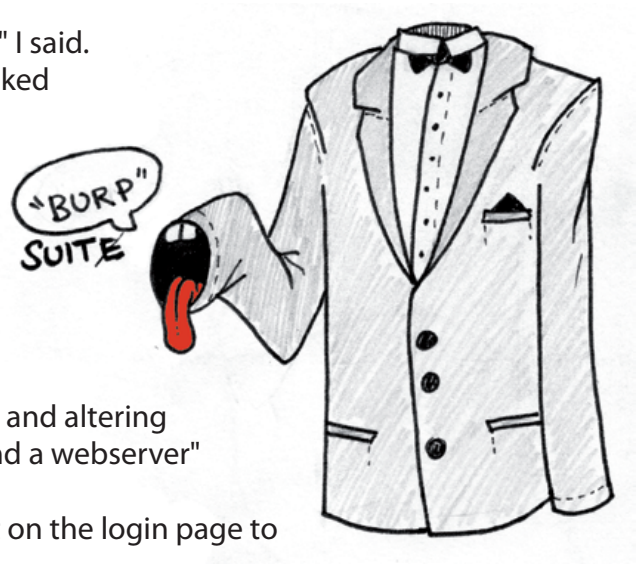


In my dreams I drove down a road which suddenly split and I had to choose which way to go, so I took the left one. But after a few miles, I decided that the other road would maybe lead to where I wanted to go. Since I didn't want to drive all the way back, I just drove cross-country towards the other road.
And then I woke up and had the solution: BOTH roads shall be used. With Josh and Jess watching closely, I first used the settings upload to create a directory named /upload/jNWAYRqi/.png/ and then used this for the directory traversal in /cam with

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

"Wow. There is another serial number in there: XKCD988. What could that mean?" Josh asked. "Well xkcd is a very popular web comic by Randall Monroe. Let's check out comic number 988." I said and we found it to be a nice christmas related panel.

SuperGnome 3 suddenly did not accept our well known credentials anymore.
"Bummer. Now what?" Josh was giving up already. "Let's check out how the login mechanism works and maybe we can bypass it" I said and had the kids' eyes glowing back on me.
"Since the login mechanism actually uses the input fields directly as input to mongodb, it is potentially vulnerable. Look here in index.js line 109" I pointed the kids to the node.js code.
"I need to get my burp suite for this" I said.
"Hahaha. A suit that burps?" Josh asked



"No Josh. It's a _suite_ for analyzing and altering the traffic between your browser and a webserver" I explained.
So after changing the POST request on the login page to

```
POST / HTTP/1.1
Host: 52.64.191.71
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:43.0)
Gecko/20100101 Firefox/43.0
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/?logout=1
Cookie: sessionid=KU7jhlS0KHJe3V4zXCKe
Connection: close
Content-Type: application/json
Content-Length: 69

{
    "username": {"$lt": "user"},
    "password": {"$gt": ""}
}
```

we got logged in as admin and where able to download all the files from the SuperGnome. "Another weird serial number?" Josh asked. "Yes. This time it's THX1138. That is an old movie by George Lucas. You know, the Star Wars guy?"
"Oh yes! Can't wait to see Episode VII!" Josh shouted.

SuperGnome 4 let us back in with the known admin password but again file download was disabled, but file upload was available. "Hmm. That's interesting. I saw that there was some really strage code about image processing in there for the file upload. Let's check that out a bit more"
And so we looked at lines 153-180 of index.js and like a bold red sign, the line using eval() hit me. "You see that line there? That looks bad for them... and good for us!" I told the kids. "This means we can potentially execute any JavaScript we want on the server"
Since the upload only accepted .png files, I created a single pixel one and used that in the upload form. The selected image processing didn't really matter as I changed the actual request in burp suite before it would reach the SuperGnome:

```
res.send(require('fs').readFileSync('/gnome/www/files/gnome.conf'))
```

"This is an API feature of the express.js framework they use. It will just send us the file like a browser download" I explained.
"Thats really cool! Do it for all the other files too!"
So I did:

```
res.send(require('fs').readFileSync('/gnome/www/files/20151203133815.zip'))
res.send(require('fs').readFileSync('/gnome/www/files/factory_cam_4.zip'))
```

"Wait. That last one failed! Maybe the file is too large?!" Josh noticed. So I had to improvise. "This is not nearly as elegant, but we don't have much time left until christmas, so this will have to do:"

```
require('fs').readFileSync('/gnome/www/files/factory_cam_4.zip',
'base64', function(err,data){
    if(!err){
        res.end(data.toString())
    }
})
```

So we got the base64 version of the file as a string inside the resulting webpage.
"Good enough." I said and the kids were happy.
"What is the significance of this serial number then?" Josh wanted to know.
"Well. BU22_1729_2716057 that is supposed to mean: Bending Unit 22, unit number 1729, serial number 2716057, which is the robot name of Bender Bending Rodriguez from the TV show Futurama. You should watch it, it's really funny" I geeked out.

SuperGnome 5 turned out to be tough. "The admin credentials work but file download is disabled and I don't see anything that looks like a flaw" Jess declared.
"I saw some comments in a file called sgstatsd.c, maybe that program is active on the SG. The default TCP port seems to be 4242." I explained
"Like the answer to life, the universe and everything, doubled!" Josh noticed.
"Yeah. Kind of like that..." those kids really were premium nerds. Like me.

The compiled version of this program in the gnome's firmware was an x86 executable instead of ARM like all the rest, which might explain the problems they had to get it to work on the gnomes. But SG05 really was running this service, which I checked with

```
telnet 54.233.105.81 4242
```

The menu appeared and I tried the "secret command" which I deducted from the source code to be a capital X (ascii 88).
"This calls the sgstatd() function which contains a stack buffer overflow. Maybe we could exploit that. The program has an executable stack according to `execstack -q sgstatd`"

I showed the kids.

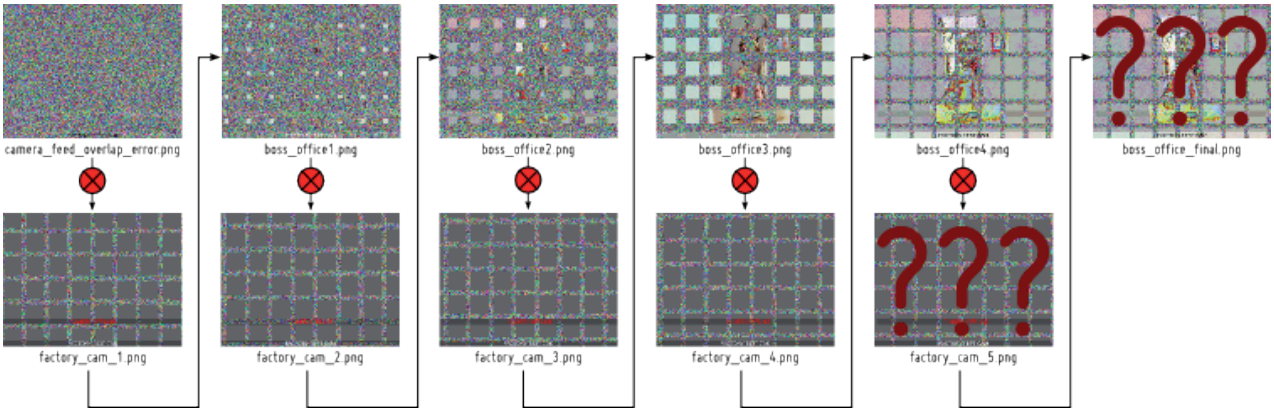
"And how do we exploit that?" Josh asked eagerly.
"Well. The sgnet_readn() function reads up to 200 bytes into the character array 'bin' which is only 100 bytes big. So if we overwrite this array, we will also overwrite the stack information like the return pointer. In this case, we will also overwrite a custom stack canary, so we will have to reconstruct that correctly. Otherwise, our exploit won't work. Also it will most likely use ASLR, so we can't just put some static address into the return pointer and hope that works..." I tried to explain the details.
"So what do we do?" Jess needed answers.
"Well. We would run the program in gdb and use a pattern to find out what offsets the stack variables are at, reconstruct the canary in there, fill the rest with some shellcode like a reverse netcat shell and then have EIP be a jmp esp instruction on return. Maybe throw in a NOP slide to be sure?" I hoped my instructions wouldn't overwhelm them.
"Okay. Let's do that!" they both exclaimed.

"Kids. I would have to set up a debug environment and fiddle around with all the breakpoints and attaching to the child processes. I really don't have time for that right now. I still have to get all the christmas presents!"
The kids were disappointed, but there wasn't much I could do. Being a grown-up is hard. They would learn that soon enough.

"Tell you what. Let's look at all the stuff we found on SG1-4 and maybe we can already tell what this ATNAS corp is really all about. Then we don't need to pwn SG05 and I can still get home in time." I tried to reason with them.
"Okay. Let's find out what's going on"

Part 5: Baby, It's Gnome Outside:

"So we got all the configurations and we found these pcaps and some strange images. What now?" Jess was still trying to make sense of it all.
"Well the GnomeNET page said something about the cameras being named the same and the images getting fused into one. So if we XOR all these static images with the fused one, we should get the camera screenshot from the boss' office." I explained to the kids.
"That sounds so complicated." Josh was upset.
"Well I will use the convert command and it works like this"

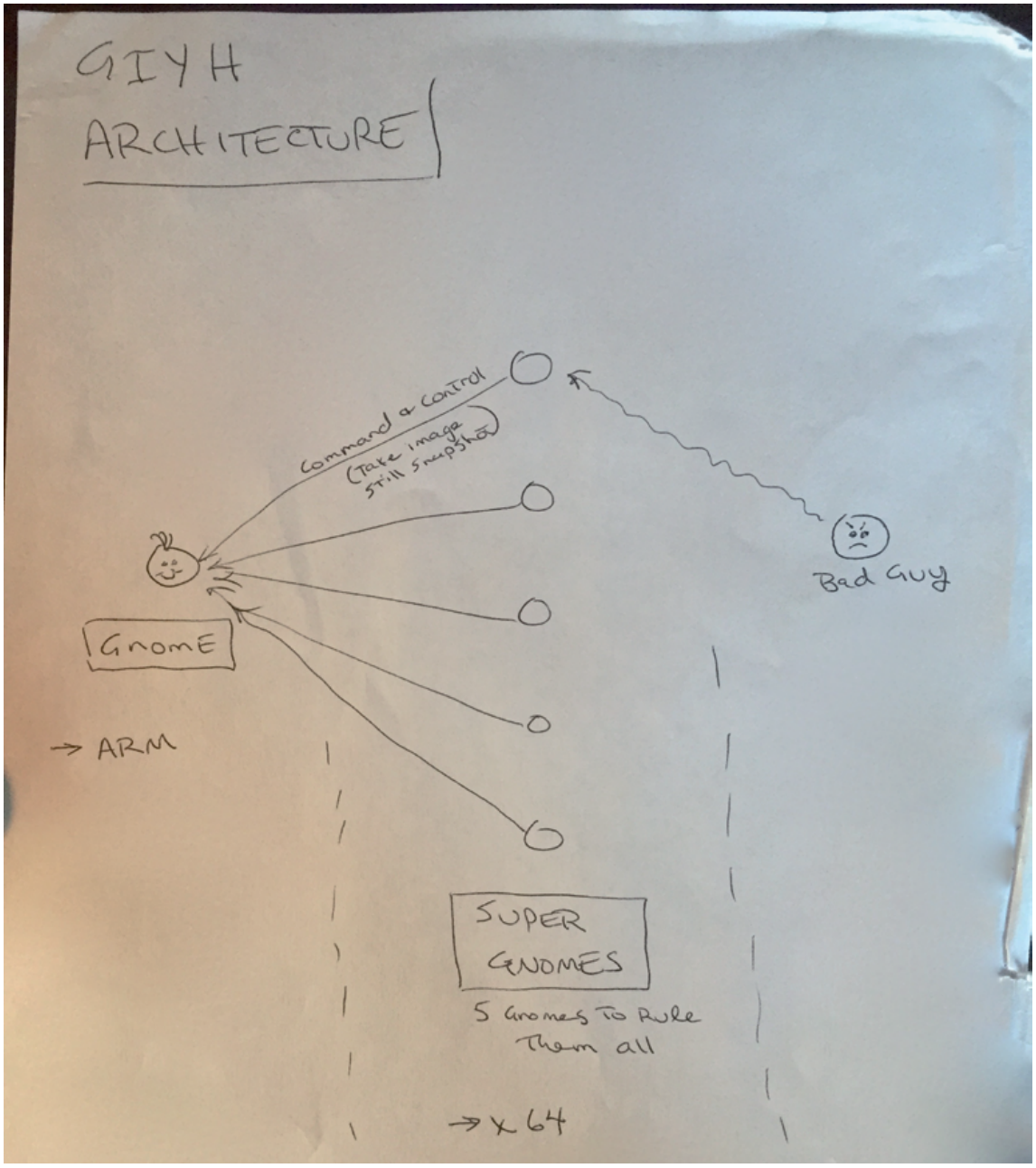



```
convert camera_feed_overlap_error.png factory_cam_1.png -fx
"(((255*u)&(255*(1-v)))|((255*(1-u))&(255*v)))/255" boss_office.png
```

"How did you know that command?" Jess wanted to know.
"Well… erm… Sometimes people already had a problem similar to yours and then you can find answers on stackoverflow."

So in the boss_office image, we found a name tag "Cindy Lou Who". And the pcaps contained several emails which mapped out the nefarious plot that Cindy had in mind: The almost 2 million gnomes where sending pictures back (we even found a crude picture of the architecture) and those images were analyzed for valuable goods in the homes of the unsuspecting families.
Those goods were compiled into lists for burglars to be stolen on 24th of december (after dark) to ruin christmas for all those families and steal back christmas.
Even the company name "ATNAS" was a play on this with it being "SANTA" spelled backwards.

And with Cindy's traumatizing experience with the Grinch as a child, her hatred of christmas led to this whole plot.



"Alright Cindy. I'm sorry for being your personal Grinch once again, but I'll ruin this nefarious plan for you now.
Come on kids, let's inform the authorities about this." I said with a grin on my face.

And as the federal police caught Cindy and her bunch of burglars just in time before christmas, we had a party under the tree at the Dosis home and re-flashed the gnome to be a motion detector camera. It now protects the Dosis kids' room against stealthy guests like green monsters. :o)

The End.


Gnome with it

Credits:

hacking, story: tabascoeye

illustrations, layout: flederrattie

Thank you for reading.