CHECKMARX

Exposing Wireless IP Camera Security Flaws

by Checkmarx Application Security Research Team, August 2017

Introduction

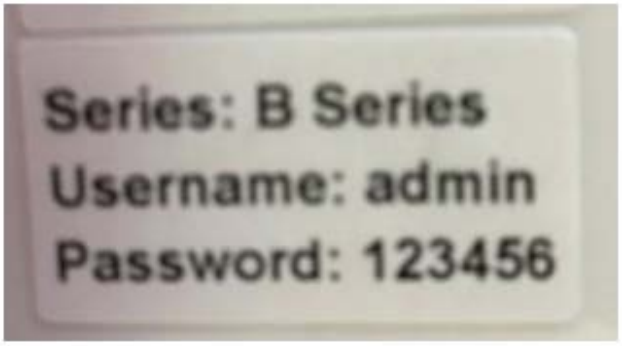
Now more than ever, the exposure of sensitive data is easily one of the top concerns among users, and the thought of cybercriminals exploiting your home security to watch you, steal your data and more, is already a reality that does not seem to be addressed sufficiently. Checkmarx set out with the mission of figuring out just how much risk are we exposed to by some of these connected devices. Our overall objective was to focus on the security states of the featured wireless IP cameras. Additionally, with the help of previous vulnerabilities disclosed by our fellow security researchers, we aimed to uncover additional attack vectors and scenarios.

Methodology

To start off the research, we dove into the wireless IP camera market. We quickly discovered, without surprise, that the most popular devices are also among the least expensive on the market. The two chosen models covered in this research are the Loftek CXS 2200 and VStarcam C7 37WIP.

While concentrating on two specific models, many of the other devices available on the market are based on the same or similar software. Therefore, it is important to state that the findings in this report are applicable to wider range of models.

The general setup process included connecting to a WiFi network, connecting to the power, and ensuring that the device has the same IP range of our network. Afterwards, an authentication window popped up, and we submitted the given admin credentials. Along with purchasing the camera, a set of default credentials were printed on small stickers attached to the bottom of the device



One of the first red flags to pop up in this testing session happened during this stage, as there wasno recommendation or enforcement for a password change. We found the results of the VStarcam particularly interesting. We learned that this device has telnet access, yet there was no information about this connection channel in the device's manual. According to the manufacturer, this is not something we should have access to. So, the question was raised whether or not this may be a backdoor.

Telnet is used frequently by IoT devices, though we still don't understand why manufacturers don't allow the device owner to access it. From that point in the research on, the vulnerabilities just kept on coming.

Analysis

The following section covers a range of vulnerabilities detected and exploited by our team in a lab environment. As mentioned earlier, there have been previous reports on IP Cameras however all findings mentioned in this document are either new or expansions on previous findings by other security researchers.

Cross-Site Request Forgery (Loftek) - Poor Username and Password Policies

This vulnerability, previously brought to light by researcher Craig Young, is a wellknown issue (CVE-2013-3312), that seems to have been ignored by the vendors. Rather than addressing the vulnerability, Loftek, for example, no longer lists this camera on their website

What Can Attackers Do?

Get Requests sent in clear text allows the attacker to send all kinds of commands to the camera. Among them is the ability to create new users. As admin passwords do not need to be changed, the chance is likely that the default password may work. The following request is a simple example we executed:

```
<iframe
src="http://victim/set_users.cgi?next_url=rebootme.htm&user1=admin&pwd1=CxSRT&pri1=2&user2=Attacker&pwd2=123456&pri2=2&user3=&pwd3=&pri3=0"></iframe>
```

1. eep the original admin password to avoid any suspicions
2. Add a new user called 'attacker' with admin privileges (pri2=2)
3. **Checkmarx Finding** An attacker who would want to stay in stealth mode could actually name his user "%20" (a hex representation of a space) and not define a password. This will actually add an admin user without showing up on the camera's admin interface.

Server-Side Request Forgery (Loftek) Checkmarx Finding

Loftek uses both FTP and email settings, which may be accessed via the 'alarm settings' option. This led us to two attack scenarios:

1. **Portscan**

Based on the response when clicking the "test" button on an FTP scan, an attacker could figure out what port is open or closed. The errors returned are indicative of the state of the port. "Server error" means the port is open and "cannot connect to server" means that the port is closed. This may allow malicious users to transform your IP camera into a scanner for internal and external networks.

2. **Deinal of Service**

Using a method similar to Portscan, a malicious user is able to flood a system with requests via a specific port. Should an attacker grab a bunch of vulnerable cameras, which may be an easier than one assumes, they will be able to launch a DDoS attack using the IP cameras as hosts.

Denial of Service Using Requests on Camera Services Checkmarx Finding

Loftek seem to offer a free DDNS service. The format of the DDNS domain is: 002gyfl.nwsvr.com where002gyfl is the name of the device we purchased. This clarifies that device IDs are built into each camera, meaning other camera IDs are publically available.

What Can Attackers Do?

Using the manufacturer's DDNS, an attacker could run a script that would put together four letter combinations and see if they return a 302 code. If it returns, you will be redirected to the IP address of the relevant camera. Another interesting point is that the server itself uses an old Ubuntu version and a vulnerable Apache version. An attacker who gains access to this server has just hit the jackpot.

Stored Cross-Site Scripting in //proc/ Checkmarx Finding

Loftek

During our tests, we noticed that //proc/core renders the page in HTML, which was good from a research point of view (and bad from a camera owner point of view), as this allowed us to inject a XSS payload and execute it on this file.

VStarcam

When it came to experimenting with XSS, we found the VStarcam much more interesting. Adding a user with the name '-alert(1)-' will show the stored XSS on every page. This is due to that every request receives a get_params.cgi, and you are able to see the usernames. We noticed this whenwe checked the HTMl source code: url+='&loginuse='+loginuser +'&loginpas='+encodeURIComponent(loginpass). t 'loginpass' is encoded, while 'loginuser' isn't.

Additionally, on the VStarcam, you can own a camera just by using a rogue SSID. In our lab, we changed our SSID wireless access ApointPpoint to <img src=x onerror=confirm(1)>. When the victim scans their Wi-Fi, the payload is triggered automatically.

In a nutshell, if the user scans for nearby rogue Wi-Fi AP, they will get compromised. As we aimed to get a better PoC, we changed our SSID (following the 32 character max) to a vector which would send us the admin credentials.

HTTP Response Splitting Checkmarx Finding

Loftek

You are able to send a direct XSS payload to an authenticated user. The “next_url” input parameter (which is used in almost every CGI file) isn’t properly sanitized, thus allowing the launch of an attack using the HTTP response splitting vulnerability.

VStarcam

Similar to the Loftek, this attack style is effective here too. An attacker is able to manipulate the HTTP responses, which allows them to conduct cross-site scripting, cross-user defacement, cache poising, and page hijacking.

Open Redirect (VStarcam)

In addition to HTTP response splitting, an attacker can use the “next_url” input parameter for leveraging an Open Redirect vulnerability.

File Disclosure (Loftek)

Again, by using the “next_url” parameter, an attacker may also output File Server files – at the very least, they may output the ones with the right permissions (such as resolv.conf).

External Service Interaction (DNS) (Loftek)

As the “myserver” will receive a DNS lookup for type A by the camera’s IP, this may not be considered a vulnerability in its’ own right and could be considered to be part of the app’s normal behavior. However, this still could be used in a malicious way. In our case, it was possible to use an internal network scan to potentially create a Denial of Service attack.

Forced Factory Reset (VStarcam)

We discovered this vulnerability while testing the ‘backdoor’ access, as mentioned in the CSRF section using the space-hex representation (20%) to be used as a username. VStarcam has some JavaScript code forbidding the use of special characters, however, you may disable it using any browser inspector in order to add other characters. Validating client-side is not the best choice for this. We added ‘20%’ to the admin username. After the automatic reboot due to the modification, the access was blocked. Even the default credentials didn’t work. In order to get it working again, you would need to press the reset button and proceed to reconfigure everything.

Attackers can disable the camera functionality until the reset button is pressed.

Conclusion

A wide range of camera manufacturers use very similar hardware and software in their cameras. The main difference is with the UI and specific firmware updates. We noticed that many wireless IP cameras on the market, especially the cheaper ones available for purchase on popular sites such as eBay or AliExpress, run on a specific server signature called Netwave IP Camera. Using Shodan.io, we were able to check how many cameras are available with the Netwave and GoAhead signatures listed.

We estimate that more than 1 million cameras and IoT devices using these server signatures are vulnerable. In the interest of user's privacy, we did not check for vulnerable cameras in the wild. However, it is safe to say that the cameras we tested are indeed vulnerable in the wild

Searching for other devices based on firmware (which is publically available), we found different camera models using the same server signature and software. The only difference is the web UI itself. They include: Foscam, Advance, Wanscan, Apexis, Visioncam, Eshine and EyeSight. Most of these cameras have old firmware and hardware.

As our initial scans came to an end, we reached the conclusion that if your camera is connected - you're definitely at risk. It's as simple as that. A malicious user can exploit your device to track your day-to-day, know when you're home or out, steal your email information, steal your wireless connection, gain control of other connected devices, use your camera as a bot, listen in to your conversations, record video, and more. Attackers may even track your location. This is done by doing a simple search on WiGLE to find the SSID (assuming it's unique) and can find your location on a map, or use the MAC address. During our testing, we found the location of the test lab with a 200m error range. An attacker can take it one step further by using the wireless settings of the web

application to scan other available networks. By doing so, the attacker can create a kind of triangulation to determine where the camera is physically located. Users and the global web may encounter DDoS attacks on their systems, websites and/or servers, as the devices are fertile ground for a botnet army created along with other IoT devices. The Mirai attack serves as a great example.

There may be a scenario where an attacker could use the camera's settings to send spam emails, flooding the victim's inbox. With a simple script, an attacker could launch such an attack with little-to-no effort.

The cameras are vulnerable by default, and - especially the Loftek 2200 – may be a backdoor to your network. It is clearly worth spending a bit more money on a more secure camera.

For the cameras tested, no firmware updates appear publicly nor have been released on their pages. We learned that some users have successfully uploaded new firmware from other vendors that fixed most issues, though caused some functionalities to stop working. A great alternative is to only access the camera through a private VPN. It may not be a simple process for the average user, but without doubt, this can be a good option for users up for the challenge.

We sent both vendors emails (dated March 24th, 2017) asking for a security contact. We are yet to receive replies.

The information above is a partial representation of the issues discovered during our research. The full list of attacks/exploits we tested and confirmed can be found below split into existing research findings and new finding by Checkmarx's research team:

Found by Checkmarx

Stored Cross-site Scripting in //proc/kcore
HTTP Response Splitting
Bad username and password policies which allow an attacker to obfuscate his account
Server-side Request Forgery
Denial of Service using requests on camera services
Manufacter DDNS security issues
File disclosure using next_url parameter
Stored Cross-site Scripting in user name
Stored Cross-site Scripting in SSID
Open Redirect using next_url parameter
Disable IP Cam and force a reset from factory
Sniffing the network using the wifi driver
Bitcoin mining, password cracking, or any other abuse of distributed computing
Android app clear text authentication
Admin password disclosure to Eye4 API
Android app has malware

Existing Research

Cross-Site Request Forgery
Information disclosure (both cameras)
Netwave IP remote exploit
Remote root access
Memory dump

About Checkmarx

Checkmarx is an Application Security software company whose mission is to provide enterprise organizations with application security testing products and services that empower developers to deliver secure software faster. Amongst the company's 1,400+ customers are five of the world's top ten software vendors and many Fortune 500 and government organizations, including SAP, Samsung, and Salesforce.com.