EZVIZ Security Camera Analysis

Executive Project Summary

In the modern age, Internet of Things (IoT) devices are becoming more and more common. One portion of this growing market includes security cameras. Through research and development, Wi-Fi enabled cameras have become quite affordable. They also have a growing list of features that are attractive to consumers. Some examples of the convenient tools available include remote two-way communications, remote access to live video, integration with Alexa and IFTTT, cloud storage, SD storage, and much more.

Our main goal is to perform a penetration test on an EZVIZ CTQ2C 720p security camera. Our scope will include testing the vulnerabilities of the camera itself, along with utilizing other devices to attack it over the network. These can be the smartphone companion app, web app, and other IoT devices. We will not be directly pen testing the official servers that host the cloud service and allow remote camera access.

A compromised security camera can lead to unwanted spying and the patterning of tenants' activities. If remote access is achieved by a malicious individual, the user could be spied on without ever knowing it. In the modern day of sacrificing security for convenience, it's important for consumers to remain aware of the dangers around them. To help people best defend themselves, we aim to present a report of our findings and an establishment of best practices for securing IoT cameras.

Illegal camera surveillance has adverse effects on society, from the perspective that people feel their privacy is ever invaded via camera systems. Generally, the fact that people are becoming more computer literate cases of hacking cameras are rapidly being common. Industries like the hotels and surveillance units suffer greatly. An instance of the hotel sector, once their customers' privacy is invaded through camera hackings, the hotels credibility is lost, thus they end up losing financially.

Goals and Objectives

- Build experience practicing skills learned in coursework
- Pentest camera with a variety of methodologies
  - Intercept video/remotely control device
  - Test the security of its mobile applications
  - Analyze packets passively and see what can be observed
  - Attempt to leverage the HTTP requests it transmitts
  - Use parent companies software to achieve more access to the camera
  - Attempt to connect to the camera through a TCP connection

Merit of the Project

- With IoT cameras gaining popularity, finding and sharing weaknesses so they can be patched is beneficial to everyone
- It is important for IT companies and individuals to know that how to protect their assets during the current revolution in the IT industry when their cameras are connected to the internet.
- If no vulnerabilities are found, we can add comfort to the idea of consumers trusting these devices

Proposed Project Timeline

Tasks and Expected Completion Time

Task	Start Date	End Date	Time Needed

Investigate Camera Hardware	2/28	3/07	1 week
Build Test Beds (iOS, Android, WebApp)	2/28	3/03	4 days
Test (iOS, Android, WebApp)	3/03	3/17	2 weeks
Setup WiFi Pumpkin	2/28	3/03	4 days
Network attacks	3/03	3/17	2 weeks
Prepare MicroSD for Firmware Attack	3/03	3/10	1 weeks
Implement Firmware Attack	3/10	3/17	1 weeks
Formalize Write up for M2	3/17	3/28	9 days
Project Realization	4/04	4/11	1 week
Review Project Documentation	4/11	4/18	1 week
Produce Final Presentation	4/18	5/03	2 weeks

Gantt Chart

Alt Text

Project-oriented Risk List

Risk name (value)	Impact	Likelihood	Description
Brick security camera (30)	6	5	It's possible that we may brick the security camera while trying to gain access to it via the hardware
Corrupt micro SD card (20)	4	5	It's possible that we may corrupt an SD card while attempting to gain access to the device using the SD card
Team member being unavailable/unwilling to help (32)	8	4	There may be a loss in productivity if one or more team members are unable to cooperate
Cannot attack via network (15)	3	5	There may be no network-based vulnerabilities
Cannot attack via micro SD card (15)	3	5	There may be no way to attack through the SD card

Cannot attack via IOS/Android Apps (15)	3	5	There may be no vulnerabilities allow us to get access to the camera via the phone apps

Diagrams

Use Case: Camera
Use Case: API
Process: Emulator

Project Methodology

Literature Review

1. **Resource:** https://github.com/OWASP/owasp-mstg#android-testing-guide
   a. **Author(s):** OWASP
   b. **Importance:** There are in-depth guides on testing mobile device applications based on the operating system
   c. **Significance to the Group:** This is useful to the group because the device we are testing has an application on both android and iOS

2. **Resource:** https://arxiv.org/ftp/arxiv/papers/1709/1709.05742.pdf
   a. **Author(s):** Mordechai Guri, Dima Bykhovsky, Yuval Elovici
   b. **Importance:** Touches on how attackers use infrared waves to read signals from cameras remotely. The main mean of attacks discussed is exfiltration and infiltration.
   c. **Significance to the Group:** It's useful to the group since the EZVIZ CTQ2C 720p security camera can be penetrated both internally and externally.

3. **Resource:** http://s3.eurecom.fr/docs/trusted16_costin.pdf
   a. **Author(s):** Andrei Costin
   b. **Importance:** There is a detailed explanation of various attacks styles like steganography on surveillance systems and how to mitigate them
   c. **Significance to the Group:** The information is beneficial to the study in the sense that it gives an insight on how to overcome some of the vulnerabilities that might arise from the study of the EZVIZ CTQ2C 720p security camera

4. **Resource:** https://bit.ly/2E6c3qW
   a. **Author(s):** Qasim Mahmood Rajpoot, Christian D. Jensen
   b. **Importance:** Covers how the legal sector is coming in to counter attack the growing vice of breaches through camera systems.
   c. **Significance to the Group:** Material is important as it helps explain how demographic intervention promotes change on technological lawlessness.

5. **Resource:** https://arxiv.org/ftp/arxiv/papers/1802/1802.03110.pdf
   a. **Author(s):** Wei Zhou, Yuqing Zhang, and Peng Liu
   b. **Importance:** In length discussions on IoT, and especially its efficiency and possible breaches.
   c. **Significance to the Group:** Since the EZVIZ CTQ2C 720p security camera is an IoT technology most of the possible vulnerability routes affecting it are mentioned.

6. **Resource:** https://www.irks.at/assets/irks/Publikationen/Forschungsbericht/SurPRISE%20D3.3%20WP3%20Exploring%20the%20Challenges%20-%20Privacy%20Security%20Acceptability%20Alternatives%202013.pdf
   a. **Author(s):** Regina Berglez, Reinhard Kreissl
   b. **Importance:** Explains how to enhance security when using surveillance systems.

c. **Significance to the Group:** It has some possible schemes that can come in handy to explain how security cameras can be used without fear of being hacked.

7. **Resource:** https://s3.amazonaws.com/mfs.ezvizlife.com/Video%20plugin%20for%20Windows.exe
   a. **Author(s):** Video Plugin for Windows
   b. **Importance:** One possible entry
   c. **Significance to the Group:** This is important because we could exploit it from our own computers

8. **Resource:** https://s3.amazonaws.com/mfs.ezvizlife.com/Video%20plugin%20for%20Mac.zip
   a. **Author(s):** Video Plugin for Mac
   b. **Importance:** One possible entry
   c. **Significance to the Group:** This is important because we could exploit it from our own computers

9. **Resource:** http://download2.ezvizlife.com/assets/deps/EzvizStudioSetups.exe
   a. **Author(s):** EZVIZ PC Studio
   b. **Importance:** Remote Access without the need of a browser
   c. **Significance to the Group:** In the case we aren't able to use a supported browser, we can use this software

10. **Resource:** https://s3.amazonaws.com/mfs.ezvizlife.com/UD10510B-B_C2C_QSG_V1.0_180626.pdf
    a. **Author(s):** Mini O User Manual
    b. **Importance:** Instructions on how to operate the security camera
    c. **Significance to the Group:** We must be able to setup the device properly in order to begin testing

11. **Resource:** https://s3.amazonaws.com/mfs.ezvizlife.com/C2C%20(Mini%20O)%20Datasheet.pdf
    a. **Author(s):** Device Data Sheet
    b. **Importance:** Device Specifications
    c. **Significance to the Group:** Could be used to find a vulnerability in the hardware that the device uses

12. **Resource:** https://github.com/P0cL4bs/WiFi-Pumpkin
    a. **Author(s):** Marcos Bomfim
    b. **Importance:** WiFi Pumpkin has a wide breadth of features focused on the viewing and manipulation of network communications
    c. **Significance to the Group:** It will be useful for performing attacks such as Rogue Access Point, Credentials Monitor, DNS Spoofing, and much more

13. **Resource:** https://ipcamtalk.com/threads/custom-initrun-sh-firmware-tools-not-working.28054/
    a. **Author(s):** Reilly Chase
    b. **Importance:** User was able to gain root privileges using the sd card of a similar camera
    c. **Significance to the Group:** If we can translate this over to our device we should be able to have a similar result

14. **Resource:** https://null-byte.wonderhowto.com/how-to/intercept-images-from-security-camera-using-wireshark-0191735/
    a. **Author(s):** Kody
    b. **Importance:** A tutorial on how to capture images from a security camera via Wireshark
    c. **Signigicance to the Group:** The article will guide us to intercept images from a security samera using Wireshark

Technical Plan

Threat Modeling

One threat space that we will attempt to use to gain access to the EZVIZ security camera is the smartphone applications, this includes the android and iOS applications, that can be used to setup the device as well as watching the live feed from the camera. In order to do this, we will be using OWASP's penetration testing standard (from #1 in Literature Review). This standard has instructions how to setup a testing environment for each operating system and suggestions how to test the security of android applications.

In this study the group will attempt to access EZVIZ CTQ2C 720p security camera. To gain access to the camera, we will attempt getting its IP address though a web application called the angry IP scanner. Follow the steps as arraigned in #5 from the Literature Review. If any vulnerability is found, the test results will be recorded as further investigations are conducted.

Since all of us have a camera at our disposal, we plan on accepting the potential loss of one. To appease our curiosity, we will take one apart and see what can be learned about how its hardware functions. May prove to be helpful in compromising the device's security.

Vulnerabilities and Exploitation

If we discover vulnerabilities using the techniques discussed in the threat modeling stage, we will assess their impact on the security camera. We will then attempt to use these vulnerabilities to gain access to the device and take video surveillance.

Will need to start by probing the camera and validate that our scoped exploitation methods are viable. Individual reconnaissance may be necessary as the team has discovered variations in camera models and not all exploits may work across all devices. Device probing and passive exploitation should start in our local area networks, so we do not put external servers at risk. Once we have completed our probing and reconnaissance, we can proceed with the next face of the attack.

The next face of the attack will be performing the exploits. As previously stated in our goal section, the target will be submitted to Remote Access Exploits, Network spoofing attacks, DNS poisoning, Man In The Middle attacks, and other attacks applicable to this scenario. To perform these attacks, we will utilize the WiFi Pumpkin on a Linux VM on a LAN. Then we could use a tool like Wireshark to capture traffic from the camera. We require confidence that the exploits can be replicated on multiple cameras. Once that is accomplished, we can investigate other misuses for the camera and the data we exploited.

If we successfully obtain Root on the device, we will push the limits of what we can do with the new privileges. This could be key creating the botnet. If we manage to reach our goal of accessing the camera controls and obtain video feeds without authorization, we could proceed to connect the cameras and create a botnet of EZVIZ IOT cameras in our controlled environment.

Reporting

Once we have concluded our device analysis, we will be creating a report with all our findings. The report will include the vulnerabilities discovered, or the lack-there-of if none are found. It will also include possible remediations for the vulnerabilities discovered. Finally, it will mention where the project could be taken next in the future.

Resources/Technology needed

Resource	Dr. Hale needed?	Investigating Team member	Description
EZVIZ CTQ2C	no	Everyone	this allows everyone to do independent research
SD card	no	Everyone	Needed for the camera to be able to store recordings, does not come included with device
iPhone	no	Mohammed & Khalid	Needed to investigate the iOS app
Android Device	no	Jose & Thomas	Needed to investigate the android app
Burp suite	no	Christian	Needed to test the webapp
Server Space	no	Christian	Needed to test MITM attacks

Wireshark software	no	Mohammed & Khalid	Software used to monitor network traffic
Workstations and writing materials	no	Everyone	For recording test results during practical
WiFi Pumpkin	no	Everyone	For performing network attacks
Postman	no	Jose Hernandez	Used for sending get requests to the camera on port 8000
Visual Studio Code	no	Jose Hernandez	Used to write python program to establish a TCP Connection with the Camera
iVMS	no	Jose, Christian, Khalid, & Mohammed	Hikvision software that offers management features over EZVIZ Camera

Results / Findings

Below are the final quantitative results of our work on this security camera pentest project. The efforts that ended up making the findings list are just the tip of the iceberg of our work performed. We started out with ambitious goals and with that came probability that some testing would yield nonmaterial results or not even be testable like we initially believed. While there were many roadblocks along the way, we managed to adapt as we progressed by finding new ways to overcome challenges or discovering even more attack vectors to shift our focus onto.

- **Hikvision Software:**
  Hikvision (the parent company of EZVIZ) has software for its branded cameras that are used for commercial/enterprise surveillance. This software is called iVMS. It is meant for system administrators to configure and edit settings in the camera that are hidden from the average consumer. Using credentials, one can access any camera in the network and take ownership of the camera. The software is not intended for the consumer

- **Open Port 554:**
  Utilizing a third party video streaming software (VLC) we were able to stream the video feed from the camera. Since the credentials are weak, a brute force attack allows us to access the video feed with ease. One misuse could be obtaining video feed from an enterprise network; and then watching and recording the video feed using a third party software without trace.

- **Weak Credentials:**
  Brute forcing the credentials used for accessing the camera via port 554 would not be hard. The username is "admin" and the password is 6 capital letters in a random order. The same password is used for the default encryption password and is on the bottom of the camera's base (easy physical access).

- **Unable to Change Credentials Using the Application:**
  The application used to control the camera has the option to change the encryption password, which we believed to be the same as the one used in port 554. The password change in the application had no effect on the password used to connect over port 554; the default password remained the same. This leaves the camera vulnerable to a brute force attack even if remediation is attempted.

- **Connection Over Port 8000:**
  We discovered that port 8000 is used to connect to the Hikvision management software. While viewing the connection between the camera and iVMS, we are able to see some http requests. When attempting to send the same GET requests

using postman, we do not get a response. The connection appeared to be some sort of TCP connection when attempting to connect to it; we were unable to get a response. If we had more knowledge on how the connection worked, this may be a port that could be leveraged to attain additional information.

- **Aireplay-ng Deauthentication Attack:** After setting up Kismet and the wireless adapter on a Linux Kali box, we found that the camera could be disrupted using the Aireplay-ng attack. The camera stays unresponsive for as long as the command is running.
- **Implemented Android Application Security:**
  While pentesting the Android EZVIZ app, we learned that the app is fairly well secured. The analysis performed revealed proper encryption of network traffic and the hashing of passwords when they were present in logs. Also, EZVIZ coded it in such a way that it cannot be installed or run without crashing on emulated Android devices. They covered their bases in secure app design to prevent malicious tinkering.

Moving Forward

While something like a deauthentication attack does not have a realistic defense, issues such as open ports or weak default credentials can feasibly be remediated. Being able to tap into someone else's camera feed through an open port is a big privacy issue for consumers looking to monitor their homes without their own devices being used against them. While using Hikvision software on an EZVIZ camera is not part of the intended use case, the weak admin credentials can have a relatively simple fix. First of all, the default password should not be stored on the base of the camera. Also, linking it to the encryption password (since they are both the same by default) so that it gets changed when the encryption password is changed within the app would help reduce the risk posed by weak credentials.

Future work testing the EZVIZ security camera would be best spent on the iVMS software. We did not discover it until late in our pentesting and may still hold some tricks we did not think to exploit yet. It is very powerful in the realm of consumer grade EZVIZ security cameras (considering the software was built to help people manage commercial grade Hikvision devices). Beyond that, we tested a plethora of attack vectors and further research would be well spent on different security cameras. This would lead to a better idea of what vulnerabilities may exist across multiple camera models or even different brands.