

**Security Research of IoT Devices:
Penetration Testing of an IoT Camera**

Luke D. L. Katsel

University of Arizona

CYBV 498: Senior Capstone – Engineering

Professor Jordan VanHoy

April 7, 2024

Abstract

Internet of Things (IoT) devices inherently have vulnerabilities that expose users to unnecessary risks when the devices are integrated into their network. These vulnerabilities stem from poor security practices in the design and development process. Regulations can be imposed on IoT device manufacturers to bring all IoT devices in compliance with a baseline of cybersecurity capabilities. To demonstrate the vulnerabilities that these devices can have, a penetration test of an IoT camera will be performed. The goal of the penetration test is to compromise the device in a way that grants full unauthorized access to the device's root terminal. The goal was achieved through the exploitation of the camera's manual firmware update mechanism.

Table of Contents

Convenience versus risk	5
Literature Review	6
NISTIR 8259A.....	6
NISTIR 8259.....	8
NIST Special Publication 800-213	9
Analysis	10
Risks Associated with IoT Devices	11
Regulations and Standardization	12
Penetration Testing Framework.....	13
Penetration Test of IoT Camera	14
Reconnaissance.....	15
Scanning.....	16
Webpage.....	17
Script development	21
Exploitation	24
Summary and Mitigations.....	25
Conclusion	26
References	28

Table of figures

Figure 1: Nmap port scan of IoT Camera	17
Figure 2: Webpage in Internet Explorer mode.	18
Figure 3: Login page with Default Credentials	19
Figure 4: IoT Camera Manual Update Page	21
Figure 5: Output from Unpacking the Binary Update file.....	22
Figure 6: Contents of /etc/passwd.txt.	23
Figure 7: Modified InstallDesc File	24
Figure 8: Telnet Connection to IoT Camera.....	25

Convenience versus risk

Parents have an innate need to protect their child, but parents cannot always be in the same room as their baby. This is a problem that technology has been able to solve through the invention of baby monitors. For decades one of the first items new parents buy is a baby monitor to watch over their baby, but recently these devices have transformed from simple microphones to complex Wi-Fi cameras with two-way communication. This has been a comfort to many parents, but in some cases these devices have put children and their parents in a terrifying situation. Kurin Adele is one of these mothers who bought a baby monitor camera to watch over her baby, only to find out that it put her child in a horrible situation (Kato, 2023). Kurin found out that a stranger had been able to view her Wi-Fi camera's live stream, wake up her child, and speak to him through the built-in microphone and speaker. This egregious incident is not a rare occurrence because these types of devices introduce more risk than they eliminate.

These baby monitor cameras are part of something called the Internet of Things (IoT). IBM defines IoT as "A network of physical devices, vehicles, appliances, and other physical objects that are embedded with sensors, software, and network connectivity, allowing them to collect and share data" (What Is the Internet of Things (IoT)? | IBM, n.d.). The use of these devices is widespread; the most common examples are smart devices like smart phones, smart cameras, and smart speakers. These devices are designed to make life easier and in the case of IoT cameras, safer. IoT cameras are marketed as home security devices, pet monitoring devices, or baby monitors. They're supposed to make you and your family safer and less vulnerable, but this isn't the case. As Internet of Things (IoT) devices become increasingly integrated into homes

worldwide, their widespread usage raises concerns due to inherent vulnerabilities, leaving them susceptible to potential cyberattacks.

Literature Review

The three articles to be reviewed in this section are published by the National Institute of Standards and Technology (NIST). NIST develops cybersecurity standards, guidelines, and best practices that are widely adopted by both private and public sector organizations. These three publications are directly related to the security of IoT devices, and they are targeted towards both private and public organizations.

NISTIR 8259A

The first publication that will be reviewed is NISTIR 8259A. Titled “IoT Device Cybersecurity Capability Core Baseline” (Fagan et al., 2020b), it focuses on the minimum-security requirements that an IoT device should have. This publication is directed towards manufacturers of IoT devices, but it also offers valuable insight for people who are purchasing or utilizing an IoT device. The recommendations in this report come from NIST’s Information Technology Laboratory (ITL), which researched “common cybersecurity risk management approaches and commonly used capabilities for addressing cybersecurity risks to IoT devices” (Fagan et al., 2020b, p. 1). This publication lists 6 capabilities that an IoT device should have to be minimally secure. NIST does not provide a recommendation on how to achieve these capabilities, just the rationale behind it.

The first recommendation made in this publication is “Device Identification: The IoT device can be uniquely identified logically and physically” (Fagan et al., 2020b, p. 5). This capability is an important feature that allows the user to distinguish the specific device from other IoT devices. Device identification is an important part of vulnerability management. The

next recommended capability is “Device Configuration: The configuration of the IoT device’s software can be changed, and such changes can be performed by authorized entities only” (Fagan et al., 2020b, p. 6). This recommendation has two parts to it; the ability to change the configuration and the ability to only allow authorized changes. Unauthorized configuration changes are a security risk that can carry serious consequences. This is the third recommendation “Data Protection: The IoT device can protect the data it stores and transmits from unauthorized access and modification” (Fagan et al., 2020b, p. 7). Data protection is an important principle of cybersecurity. All devices that handle people’s data should have the ability to protect this data.

The next capability has to do with the device’s network interfaces, “Logical Access to Interfaces: The IoT device can restrict logical access to its local and network interfaces, and the protocols and services used by those interfaces, to authorized entities only” (Fagan et al., 2020b, p. 8). Restricting access to the device’s network interface to authorized entities works to shrink the attack surface of the device, which significantly increases the security of the device. The fifth capability is as follows, “Software Update: The IoT device’s software can be updated by authorized entities only using a secure and configurable mechanism” (Fagan et al., 2020b, p. 9). This recommendation protects the user from any unwanted device behavior. Updates protect the device from vulnerabilities, and allowing only authorized updates protects the device from being changed in a malicious way. The final recommendation is “Cybersecurity State Awareness: The IoT device can report on its cybersecurity state and make that information accessible to authorized entities only” (Fagan et al., 2020b, p. 10). This capability will allow the authorized users to know what kind of security state the device is in, without letting any unauthorized parties become aware.

These IoT device capabilities by NIST are a baseline they recommend manufacturers implement to have a minimally secure device. These recommendations also apply to users of IoT

devices, when deciding which devices to use. A device that does not implement one or multiple of these capabilities would not be considered secure. NISTIR 8259A does not provide any recommendations for how manufacturers should implement these capabilities. This allows the manufacturers the freedom to implement these capabilities in whatever method they wish, however, this does not encourage any sort of standard implementation of these capabilities. The methods that manufacturers will use to implement these recommendations will vary from device to device, allowing for varying degrees of security.

NISTIR 8259

This publication by NIST's Information Technology Laboratory, titled: "Foundational Cybersecurity Activities for IoT Device Manufacturers" (Fagan et al., 2020a) is a recommended course of actions that IoT device manufacturers should take to create a secure device. These actions are designed to help create a secure device from the ground up, which means that these recommendations only apply to devices that have not been created yet. This publication can also be helpful to IoT device users as it can help them understand the cybersecurity features their device may offer. There are 6 actions that this publication recommends, and they are broken into 2 categories. The first 4 activities are actions that will have a pre-market impact, while the last 2 actions will have a post-market impact.

The first activity is "Identify Expected Customers and Define Expected Use Cases" (Fagan et al., 2020a, p. 6). The first step in creating a secure device is knowing who the customer is, and therefore what their needs are. The second action is "Research Customer Cybersecurity Needs and Goals" (Fagan et al., 2020a, p. 7). This activity expands on the first action by identifying the specific cyber security needs for each type of customer. The next logical step is shown in the third activity "Determine How to Address Customer Needs and Goals" (Fagan et al., 2020a, p. 11). Once the customer's needs are identified the next action to take is to address

these needs. This is where the manufacturer will determine the specific method, they will use to implement the user's security needs. The final pre-market activity is "Plan for Adequate Support of Customer Needs and Goals" (Fagan et al., 2020a, p. 14). This step is where the manufacturer will create a plan for the future needs of the customer, such as the resources needed to provide continuing cybersecurity support.

The last two activities are post-market activities, which means that these are steps the manufacturers should take to provide security for the customer even after they have sold the product to the end user. The fifth activity is "Define Approaches for Communicating to Customers" (Fagan et al., 2020a, p. 17). Customers will benefit if they are aware of their cybersecurity options, so the manufacturer should find a way to inform the user of their options. The final activity NIST recommends to manufacturers is "Decide What to Communicate to Customers and How to Communicate It" (Fagan et al., 2020a, p. 18). Once the company decides how to communicate with the customer, they decide what the customer will benefit the most from knowing.

NISTIR 8259 provides several activities that it recommends manufacturers perform to develop a secure IoT device. Each activity is accompanied by several questions to guide the manufacturers through the process. This publication by NIST establishes a thorough framework but does not provide any specifics guidance on how to implement its recommendations. When you pair this paper with NISTIR 8259A, manufacturers should have a solid understanding of how to develop a secure IoT device.

NIST Special Publication 800-213

The final publication to be reviewed is NIST Special Publication 800-213, titled: "IoT Device Cybersecurity Guidance for the Federal Government: Establishing IoT Device

Cybersecurity Requirements” (Fagan et al., 2021). This publication provides guidance and recommendations for integrating IoT devices into an organization’s information systems. These recommendations are targeted towards people in private or public organizations who are responsible for ensuring the security of their organizations systems. This publication addresses the risks these devices present and then presents the requirements needed to mitigate these risks. NIST Special Publication 800-213 ultimately produces recommendations based on how to mitigate the unique risks IoT devices present.

Incorporating IoT devices into an organization’s systems introduces additional risks that the organization must overcome or accept. Even with proper manufacturing processes that ensure good security, introducing an IoT device to an information system produces additional risk. NIST SP 800-213 presents several questions that will help an organization determine what kind of risk the device presents such as “Does the IoT device have known security and/or privacy vulnerabilities?” (Fagan et al., 2021, p. 15). These questions are an essential part of the process for determining risk. With the guidance provided by these questions, this publication provides some recommendations, and they provide an example of the use of a smart speaker in an office setting. In this situation the speaker will require proper authentication and authorization along with the ability to terminate connections within the organizational policy (Fagan et al., 2021, p. 24).

Analysis

All three of these publications by the National Institute of Standards and Technology emphasize the importance of security when dealing with IoT Devices. NISTIR 8259 and NISTIR 8259A provide guidance for manufacturers because IoT device security should be considered from the beginning. NIST SP 800-213 provides guidance for organizations that are integrating IoT devices into their information systems, because they inherently bring additional risk. All

these publications provide guidance but offer no answers on how the security measures should be implemented. This allows these publications to be applicable to all types of IoT devices but misses out on an opportunity to standardize how security measures are implemented. Some implementations will be better than others, so while two comparable devices may address the same security risks, one device may do it more effectively than the other.

Risks Associated with IoT Devices

IoT devices have widespread applications in both business and home settings. When it comes to IoT device security there are no standards that the manufacturers must follow, but in a business setting there is usually a security team that can manage the risks associated with these devices. In a home setting, the users are likely not prepared, or even aware of the risks that these devices present. In a talk given by Julie Haney, she discusses insights gained from three NIST publications, and during her research she performed interviews and surveys with numerous smart home device users. The results presented in this talk were concerning, the average user of smart home devices have little concern about data privacy and security. When the participants were asked why they are not concerned about the security of their IoT device, the responses fell into 4 general categories. Benefits outweigh the risks, the chances of being hacked are low, I trust the manufacturer, and my data/devices aren't that interesting (Haney, 2023). These explanations are anecdotal but show that the average user is unaware that their device brings additional risk into their home.

While the risks of IoT devices may not be apparent to the average home user, the vulnerabilities of these devices are well documented. In a survey published by the Institute of Electrical and Electronics Engineers the authors surveyed over 100 IoT-specific research publications and synthesized a list of 9 categories that each vulnerability falls into. These

categories are deficient physical security, insufficient energy harvesting, inadequate authentication, improper encryption, unnecessary open ports, insufficient access control, improper patch management capabilities, weak programming practices, and insufficient audit mechanisms (Neshenko et al., 2019). Some of the vulnerabilities that fall into these categories are more common than others, for instance it is very common for IoT devices to have a complete lack of logging, which falls under the insufficient audit mechanisms category. Each of these vulnerability categories can have dozens or hundreds of threats associated with them, which shows that each IoT device introduces a large attack surface to a system.

Regulations and Standardization

“As IoT grows and becomes more critical, security must be a top priority for government regulation to ensure safety and reliability of the industry” (Nidagundi, 2022). This quote from Padmaraj Nidagundi’s book, *The IoT product manager*, states the critical importance of IoT device regulation. IoT is a relatively new phenomenon which means that there are few regulations or standardizations imposed upon the industry. This creates a market for IoT devices that has a wide range of products that can differ greatly from one another. The number of options can be desirable to consumers, but it can also lead to very insecure devices being marketed to consumers. While the manufacturers are under no obligation to meet any security standard for their IoT device, there will be some products on the market that are easily exploited. The problem for consumers is the difficulty differentiating a secure device from an insecure one because most home users have no experience doing security testing or research. Once IoT device manufacturers are required to meet a minimum-security baseline, then all devices on the market will have at least a minimal amount of security.

IoT is under regulated all around the world, but it is not going unnoticed. Experts In the field of IoT are aware of the under regulation and are trying to change the regulations to benefit the users. Thirty-two interviews with subject matter experts were conducted for a research paper called “Consumer IoT and its under-regulation: Findings from an Australian study” (Harkin et al., 2022). The move to regulate IoT is highlighted by the quotes taken from these interviews such as “IoT's have been raced to market. They're not even necessarily secure over Wi-Fi” (Harkin et al., 2022). This quote illustrates the idea that security is not always taken seriously at the manufacturing level. IoT devices can be perceived differently depending on who is looking at them. From the perspective of a device manufacturer, it is a product they can produce quickly and make money on. This type of view doesn't allow for any consideration for protecting the user's data and privacy.

Penetration Testing Framework

Before a penetration test on an IoT camera can be performed some research needs to be conducted to find a framework to guide the testing. There are many ways to perform a penetration test, and there are many frameworks that a security researcher can follow. One of these frameworks is published by the Open Worldwide Application Security Project (OWASP), and it is called OWASP IoT Security Testing Guide (OWASP, n.d.). This framework starts by presenting the “IoT Device Model” this model breaks down the IoT into its hardware and software parts. By breaking down the device into its individual components, the security researcher can target different components with different testing techniques. The next part of the OWASP IoT Security Testing Guide is the “Attacker Model”, this model looks at the IoT device and selects specific threats that are more common with that device. This approach streamlines the testing process by starting with the most common attacks targeting that device. The framework then goes on to present several test cases, which are derived from the preceding IoT device and

attacker models. One of these test cases is for the firmware of an IoT device, which is presented as a source of information gathering for the device. In the following penetration test, the firmware is also used as an information source, but the update mechanism for the IoT device is also exploited in such a way that it compromises the device.

Along with the OWASP IoT Security Testing Guide there is a penetration testing framework published by the Institute of Electrical and Electronics Engineering called Penetration Testing Framework for IoT (Yadav et al., 2019). This penetration testing guide presents a testing methodology for both manual and automated security testing. Only the manual security testing methodology applies in this case. In either case the process has 4 steps: planning of test, discovery, attack, and reporting. The first three steps are where the penetration test happens, and the last step is where it gets written up. The planning phase involves research into the device and company that produces it. The discovery phase involves scanning the network and capturing data. The attack phase is where the vulnerability is exploited and then the test concludes. After this, the only thing left to do is to write the report about the penetration test.

Penetration Test of IoT Camera

The first step in this penetration test was to identify and purchase a device. Identifying the type of IoT device to perform a penetration test on was simple. One of the most common types of IoT device that is made for home use is the Wi-Fi camera. This type of device doesn't inherently have more vulnerabilities than other IoT devices, but the consequences for an IoT camera being compromised can be more detrimental to the user. Cameras tend to capture more data about a person than other IoT devices, they also present a unique kind of privacy risk. This is why an IoT camera was chosen as the target device, but the specific brand of IoT camera is less consequential. The purpose of this penetration test is to demonstrate that IoT cameras in

general are not secure, or rather they aren't as secure as they could be. The criteria for choosing the IoT camera model were relatively non-restrictive. The first criterion was that it had to be Wi-Fi capable. The second criterion was that it had to be a camera that a moderate number of people purchased. The third criterion was that it had to receive generally positive reviews. All it took to find an IoT device that met these criteria was a quick search on Amazon. I purchased the first Wi-Fi camera that had over one thousand reviews and had at least four stars. The device is called Qilmy Pan Tilt Security Camera.

Through the course of this penetration test there were a couple of tools used, all of which are commonly used during a security test. Two of the tools were used during the scanning phase of the penetration test, and one tool was used during the exploitation phase. While this penetration test required a relatively small number of tools there are many more tools that are useful for a penetration test. "Practical IoT Hacking" is a book that has a chapter called "Tools for IoT Hacking" that has a comprehensive list of commonly used tools for IoT penetration testing (Chantzis et al., 2021). This list also includes a short explanation of the tool and an example of how it can be used. One of the two tools used in the scanning phase of the penetration test was nmap. This was an essential tool as it revealed a lot of information that was necessary to complete the penetration test. The other tool used in the scanning phase was Wireshark, which was used to analyze the network packets that were sent to and from the camera. The final tool used in the penetration test was Binwalk, which was used in the exploitation phase to analyze a binary file. The overall goal of this penetration test was to see if an unauthorized person could gain root shell access to the IoT device.

Reconnaissance

The first thing to do when starting a penetration test is to perform some background research. There are two things that need to be researched, the company that makes the IoT device

and the device itself. The amount of information available about either subject can vary greatly for each device. In this case there was not a robust amount of information available for either the company or the device. When researching the company Qilmy there was nothing to be found online. The only references to this company were from Amazon where the camera was purchased from. The device itself was a model that could be screwed into a light bulb socket and the camera angle could be controlled remotely. The camera required the use of a proprietary mobile application and offered an optional cloud storage subscription.

Getting the device set up was not a complicated procedure. The camera came with a simple set of instructions that included a QR code that led to the download page of their mobile app. After downloading the app and entering in the Wi-Fi information the camera connected to the app easily and the livestream began. The next step from here is to become familiar with the mobile application and the features of the camera. After several days of using the camera as normal it was time to start the testing.

Scanning

The first goal that needed to be met during the scanning phase was to identify the IP address for the device. The tool that was used to achieve this was Nmap with this command: `Nmap -sP 10.0.0.1/24`. The IP addresses for the devices on my home Wi-Fi is in the 10.0.0.1 to 10.0.0.255. The Nmap command located all the devices connected to my home Wi-Fi and displayed the IP address and a name for the device. After a process of elimination, I found that the name associated with the IoT camera was (AltoBeam) China. The IP address associated with the device was 10.0.0.168. Now that the first goal was met the next goal was to identify all the open ports on the device.

The process to identify the open ports is similar to the process of identifying the IP address. Nmap is used again but this time this command is used: `Nmap -n -v -sT -p- 10.0.0.168`. Below the full output from the Nmap scan can be seen.

```
(cyber@kali)-[~]
$ nmap -n -v -sT -p- 10.0.0.168
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 15:48 MST
Initiating Ping Scan at 15:48
Scanning 10.0.0.168 [2 ports]
Completed Ping Scan at 15:48, 0.01s elapsed (1 total hosts)
Initiating Connect Scan at 15:48
Scanning 10.0.0.168 [65535 ports]
Discovered open port 554/tcp on 10.0.0.168
Discovered open port 80/tcp on 10.0.0.168
Discovered open port 34567/tcp on 10.0.0.168
Discovered open port 8899/tcp on 10.0.0.168
Discovered open port 23000/tcp on 10.0.0.168
Completed Connect Scan at 15:48, 12.66s elapsed (65535 total ports)
Nmap scan report for 10.0.0.168
Host is up (0.032s latency).
Not shown: 65530 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
554/tcp   open  rtsp
8899/tcp  open  ospf-lite
23000/tcp open  inovaport1
34567/tcp open  dhanalakshmi

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 12.80 seconds
```

```
HTTP/1.0 200 OK
Content-type: text/html
Expires: 0

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/
html" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; ch
  <meta http-equiv="X-UA-Compatible" content="IE=edge" /
  <link rel="stylesheet" type="text/css" media="screen"
  <title>NETSurveillance WEB</title>
  </head>
```

Figure 1: Nmap port scan of IoT Camera

The most notable port that is open is port 80, which is typically used by http. Before going any further the next step is to run a UDP port scan using Nmap. This type of scan takes a lot longer so only the one thousand most common ports were scanned. The results of the UDP scan showed that none of the one thousand ports were open.

Webpage

After using Nmap to identify that port 80 is open the most logical next step was to see if the camera could be accessed from a web browser. This is done by typing <http://10.0.0.168> into the search bar. The webpage that pops up states that the web browser is too new, and a different browser must be used. After a little research into this error, it becomes apparent that this page only works in Internet Explorer, which is no longer supported by Microsoft. This is a red flag, but luckily there is a way around this problem. Microsoft edge can open webpages in “Internet

Explorer mode”, which is just as good as using Internet Explorer. When this page is opened in Internet Explorer mode a new error message appears. Below is a screenshot of the webpage.

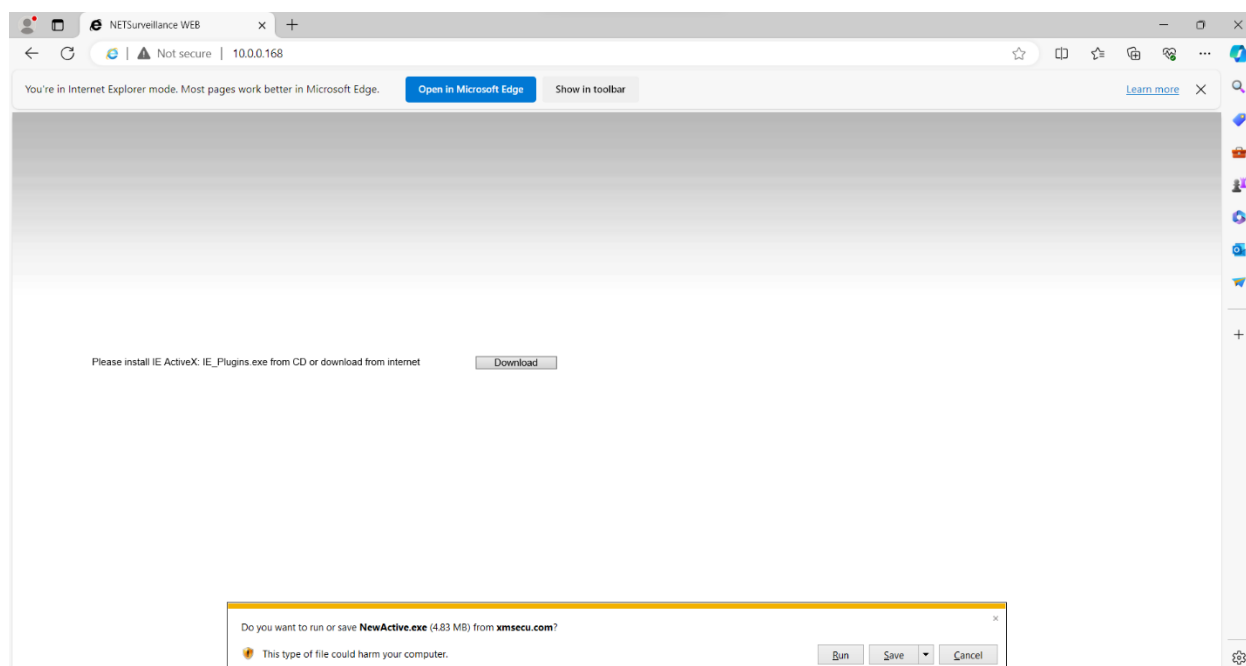


Figure 2: Webpage in Internet Explorer mode.

The webpage requires a specific program to be downloaded onto the computer to open properly. The next steps were performed on a malware analysis laptop out of an abundance of caution. The program NewActive.exe was downloaded and then the webpage reloaded. The next webpage that was loaded was a login page. It appeared that accessing the IP address of the camera might allow the user to view the camera’s live stream. During the setup phase of the camera there was no mention of a username and password for accessing the camera. There was a username and password to log into the mobile app, but these credentials did not work for the web page login screen. The next step was to guess the default username and password, but this could be a time-consuming process. There could be millions of different usernames and password combinations, but for this IoT device it took less than five tries to guess the default credentials.

The default credentials were admin: no password. This is the login page with the default credentials.



Figure 3: Login page with Default Credentials

Once past the login screen the live stream from the IoT camera could be viewed. From the webpage there was total control over the camera, the user could move the camera angle, use the speaker, and even turn on the light built into the camera. After getting familiar with the webpage interface the next step was to look at the network traffic between the laptop and the camera. Wireshark is the tool that was used to analyze the network traffic between the IoT device, and the laptop used to access the webpage live stream. The goal was to catch any unencrypted credentials sent over the network. After several tests and examining countless network packets, it became clear that there were no plain text credentials being sent over the network.

Now that it was proven that unauthorized access to the camera live stream and setting could be done, it was time to move on to the goal of the penetration test, gaining root shell access to the camera. This could be achieved through multiple means, but the simplest method is through Telnet. The only problem is that the camera does not have any Telnet ports open, so the next goal would be to find a way to open port 23. The process of determining the exact method that will be used to open the telnet port was the longest process in the testing procedure, but ultimately the firmware update mechanism was chosen. While examining the webpage it was discovered that there was a portal where a user could submit an update file to manually update the camera's firmware. This discovery along with extensive research to uncover a valid update file led to a plan of execution. If the update file could be changed in such a way that it opens a telnet port, then shell access could be granted. The update file was confirmed to be valid because the camera continued to work after the update file was manually submitted. This is a screenshot showing the manual update form.

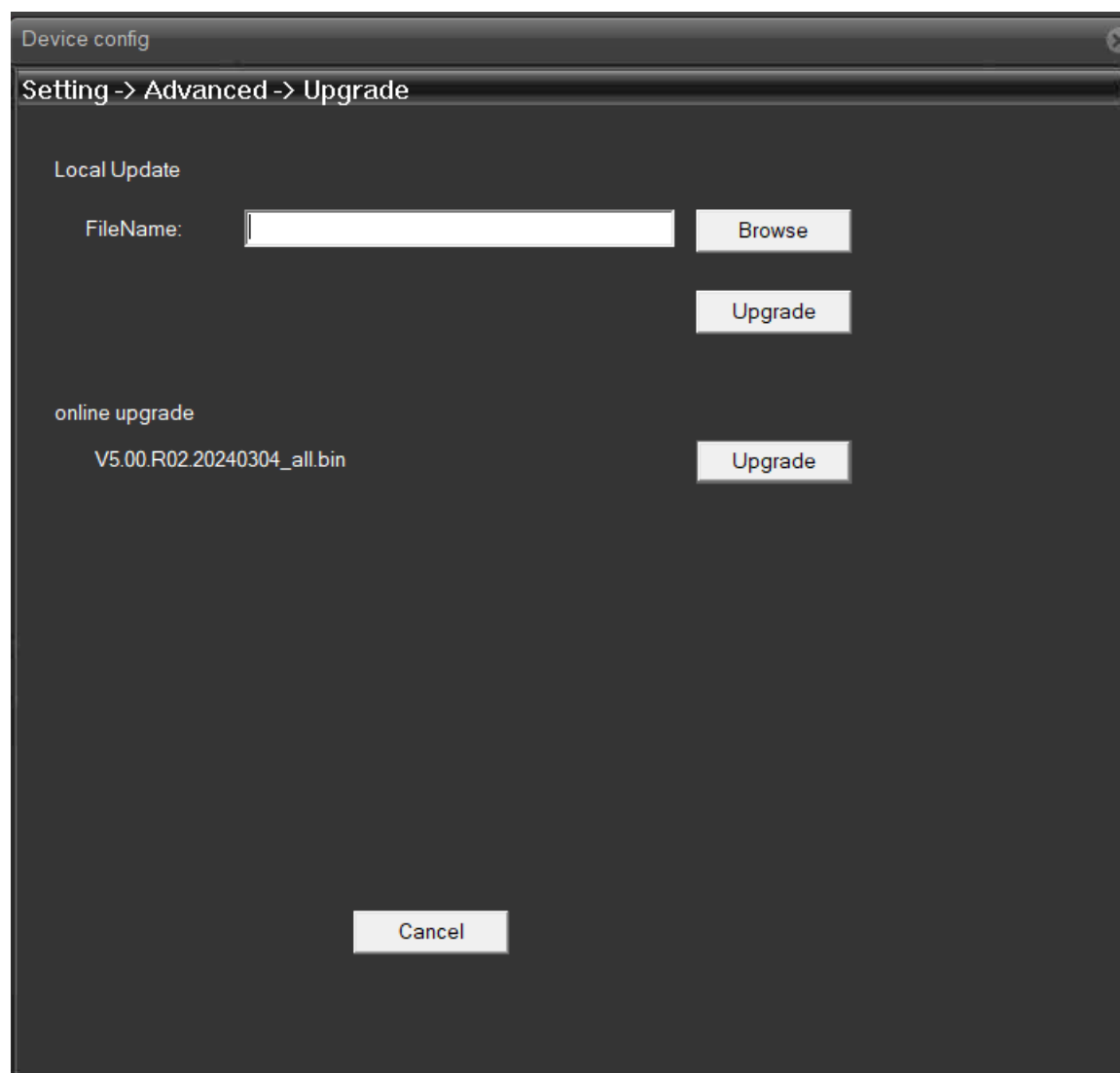
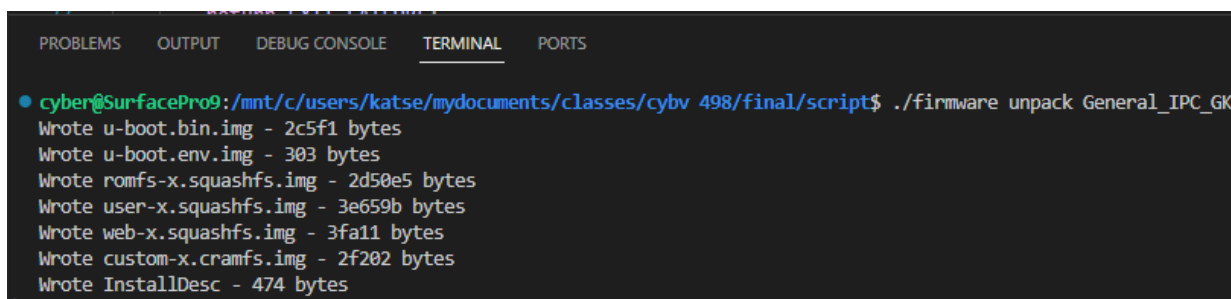


Figure 4: IoT Camera Manual Update Page

Script development

Reverse engineering the firmware update file was the most difficult process in the penetration test because binary files have no standard for packing and unpacking. This meant that the .bin file needed to be unpacked and then repacked very carefully. The tool used to determine what was contained inside the binary update file was Binwalk. Binwalk was able to identify the different sections within the .bin file along with their offset from the beginning of the file. Using this information, a c program was made to manually slice the .bin file into several individual files

containing each part. The program is simple and took 2 command line arguments, the first was either pack or unpack, and the other was the filename. The full c program can be found at this GitHub repository: <https://github.com/LukeKatsel/Firmware>. This screenshot shows the execution of the program.

A screenshot of a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. The terminal shows a command prompt for a user named 'cyber' on a machine named 'SurfacePro9'. The command executed is './firmware unpack General_IPC_GK'. The output shows seven files being written with their respective sizes in bytes: 'u-boot.bin.img' (2c5f1 bytes), 'u-boot.env.img' (303 bytes), 'romfs-x.squashfs.img' (2d50e5 bytes), 'user-x.squashfs.img' (3e659b bytes), 'web-x.squashfs.img' (3fa11 bytes), 'custom-x.cramfs.img' (2f202 bytes), and 'InstallDesc' (474 bytes).

```
cyber@SurfacePro9:/mnt/c/users/katse/mydocuments/classes/cybv 498/final/script$ ./firmware unpack General_IPC_GK
Wrote u-boot.bin.img - 2c5f1 bytes
Wrote u-boot.env.img - 303 bytes
Wrote romfs-x.squashfs.img - 2d50e5 bytes
Wrote user-x.squashfs.img - 3e659b bytes
Wrote web-x.squashfs.img - 3fa11 bytes
Wrote custom-x.cramfs.img - 2f202 bytes
Wrote InstallDesc - 474 bytes
```

Figure 5: Output from Unpacking the Binary Update file.

The update file contained seven sections, so seven files were created. Six were system images, and one file did not have an extension. While exploring the system images, a lot was learned about the inner workings of the IoT camera for instance it was confirmed that the operating system is a Linux based system which is usually the case. I was able to access a file called passwd.txt that contained a hashed password for root. Here is a screenshot of the file system and the contents of the passwd.txt file.

Name	Size	Packed Size	Modified	Mode	User ID	Group ID	Folders	Files
init.d	5 153	0	2014-10-22 21:45	drwxrwxrwx	1057	1057	0	2
ppp	479	0	2015-01-26 23:36	drwxrwxrwx	1057	1057	1	2
udev	13 975	0	2014-10-22 21:22	drwxrwxrwx	1057	1057	1	14
automount.sh	1 221	0	2022-12-27 18:56	-rwxr--r--	1057	1057		
fs-version	30	0	2006-08-10 20:49	-rwxrwxrwx	1057	1057		
fstab	95	0	2011-12-20 07:47	-rwxrwxrwx	1057	1057		
group	9	0	2006-04-19 01:05	-rwxrwxrwx	1057	1057		
inittab	3 399	0	2009-02-17 19:47	-rwxrwxrwx	1057	1057		
localtime	25	25	2022-12-27 18:56	lrwxrwxrwx	1057	1057		
mdev.conf	218	0	2022-12-27 18:56	-rwxr--r--	1057	1057		
mtab	101	0	2006-04-18 21:05	-rwxrwxrwx	1057	1057		
passwd	59	0	2014-10-22 03:58	-rwxrwxrwx	1057	1057		

C:\Users\katse\AppData\Local\Temp\7zOC365F587\passwd - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

General IPC_GK7202V300_G3-WR_S38.713g.Nat.dss.Onvifs_V5.00.R02.20221228_all.bin x passwd x

1 root:\$1\$RYIwEiRA\$d5iRRVQ5ZeRTrJwGjRy.B0:0:0:root:/:/bin/sh

2

Figure 6: Contents of /etc/passwd.txt.

There are multiple ways of altering the images of the camera's file system to enable telnet, but there seemed to be a much simpler method. The last section of the binary update file is called InstallDesc, and it is a document that contains commands that seem to run during the firmware update. It is simpler to edit the InstallDesc file and then repack the binary file rather than edit the file system images and then need to repack both the image files and the binary file. After examining the contents of InstallDesc and researching how to open a Telnet port from the command line it was clear that only a few lines of text needed to be added to the InstallDesc file. Here is an image of InstallDesc after the text was added.

```

1  {
2      "UpgradeCommand":  [{
3          "Command":  "Burn",
4          "FileName":  "u-boot.bin.img"
5      }, {
6          "Command":  "Burn",
7          "FileName":  "u-boot.env.img"
8      }, {
9          "Command":  "Burn",
10         "FileName":  "romfs-x.squashfs.img"
11     }, {
12         "Command":  "Burn",
13         "FileName":  "user-x.squashfs.img"
14     }, {
15         "Command":  "Burn",
16         "FileName":  "web-x.squashfs.img"
17     }, {
18         "Command":  "Burn",
19         "FileName":  "custom-x.cramfs.img"
20     }, {
21         "Command":  "Shell",
22         "Script":    "/bin/busybox telnetd"
23     }],
24     "Hardware":  "IPC_GK7202V300_G3-WR_S38",

```

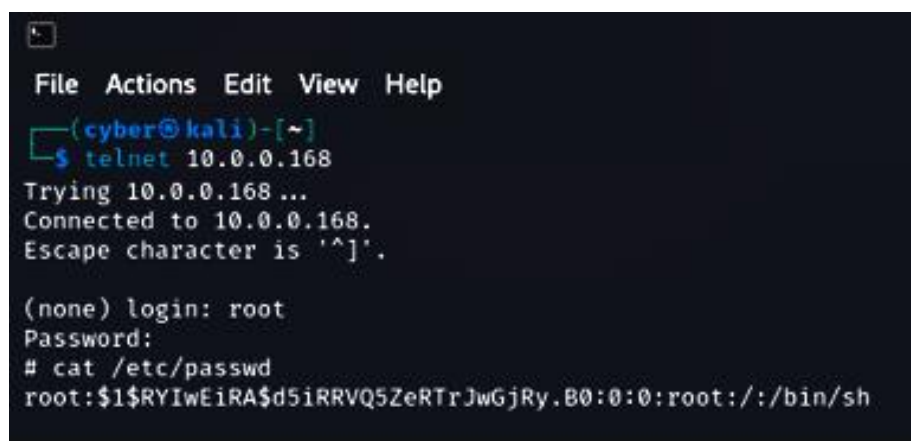
Figure 7: Modified InstallDesc File

Now that InstallDesc has been successfully modified, the next step is to use the same C program that was written earlier to repack the binary file and submit that file to the manual update portal.

Exploitation

After uploading the modified firmware update file to the IoT camera it restarted. Once it came back online an Nmap scan was run on the IP address. It was revealed that port 23 was open, which is a port used by Telnet. This meant that a Telnet connection could be attempted. The connection was made with this command: Telnet 10.0.0.168. Once the connection was made it asked for login credentials, but the username is already known. Root is the default superuser

username for Linux systems, and it was the only username in the passwd.txt file, so therefore the username must be root. The password was harder to get correct, but after manually brute forcing the most common default passwords for a while, the password was found to be 123456. While connected to the IoT camera through Telnet, compromising the passwd.txt file again was performed as a way of confirming the file was the same. Here is a screenshot of the telnet connection.



```
File Actions Edit View Help
(cyber@kali)-[~]
$ telnet 10.0.0.168
Trying 10.0.0.168 ...
Connected to 10.0.0.168.
Escape character is '^]'.

(none) login: root
Password:
# cat /etc/passwd
root:$1$RYIwEiRA$d5iRRVQ5ZeRTrJwGjRy.B0:0:0:root:/:/bin/sh
```

Figure 8: Telnet Connection to IoT Camera.

Summary and Mitigations

The goal of the penetration test has been achieved; it has been shown that a motivated attacker can gain root shell access to this IoT camera. This IoT device has shown some serious lapses in good security practices, but this IoT camera did not have the Telnet port open by default, which is a good practice. Some of the bad security practices that this device has exhibited are egregious and basic. In two separate places, the manufacturers of this IoT camera did not change the default credentials. The first place this problem was encountered was on the live stream webpage. Normally changing the default credentials for this type of account would fall to the user/owner of the device, but after several days of normal use it was not even known

that this device offered a webpage to view the live stream. The mobile application is the primary interface with the camera, but there is no option to change the username or password for the web interface.

The second location of the default password is for the IoT camera's operating system. The default user is root, which is very common to keep as the default, but it is bad security practice to leave the default password as something so simple as 123456. The second type of security issue this device had is the lack of firmware update verification. A modified firmware update should not be accepted or allowed to make changes to the system. The device manufacturers should develop some sort of update verification system to only allow signed updates to make changes to the system. This vulnerability can be tested by adding a single null value to the end of the firmware update binary and submitting the update. If it is accepted, then the device does not have update verification checks. Additional security testing can be performed based around the system images that are contained within the firmware updates.

Conclusion

IoT devices inherently have vulnerabilities that can be exploited, and their widespread usage raises cybersecurity concerns regarding user's data and privacy. The security of an IoT device starts at the design and manufacturing step, but it is also the responsibility of every person involved on the way to the end user. With regulations and standardizations imposed on the IoT device industry, IoT devices can be developed with a standard baseline of cybersecurity capabilities that will benefit the users. This penetration test of an IoT device has shown one method that an IoT device can be compromised to expose user's data and compromise their privacy. While the penetration test demonstrated in this report was successful in achieving the goal, there are still many ways to develop and improve upon the testing. One of the ways that

this testing procedure can be expanded upon is by conducting security testing on the mobile application, which is the main interface that users interact with the IoT device. This IoT device is not unique in its lack of security, the IoT market is full of devices that are designed with little regard for protecting user's data and privacy.

References

- Chantzis, F., Stais, I., Calderon, P., Deirmentzoglou, E., & Woods, B. (2021). *Practical IoT hacking: The Definitive Guide to Attacking the Internet of Things*. No Starch Press.
- Fagan, M. J., Marron, J., Brady, K. P., Cuthill, B., Megas, K. N., & Herold, R. (2021). *IoT device cybersecurity guidance for the federal government* : <https://doi.org/10.6028/nist.sp.800-213>
- Fagan, M. J., Megas, K. N., Scarfone, K., & Smith, M. (2020a). *Foundational cybersecurity activities for IoT device manufacturers*. <https://doi.org/10.6028/nist.ir.8259>
- Fagan, M. J., Megas, K. N., Scarfone, K., & Smith, M. (2020b). *IoT device cybersecurity capability core baseline*. <https://doi.org/10.6028/nist.ir.8259a>
- Haney, J. (2023, December 18). *Tradeoffs, transparency, and shared Responsibility: Exploring users' perceptions of smart home security and privacy in the U.S.* | NIST. NIST. <https://www.nist.gov/publications/tradeoffs-transparency-and-shared-responsibility-exploring-users-perceptions-smart-home>
- Harkin, D., Mann, M., & Warren, I. (2022). Consumer IoT and its under-regulation: Findings from an Australian study. *Policy & Internet*, 14(1), 96–113. <https://doi.org/10.1002/poi3.285>
- Kato, B. (2023, May 9). My baby cam was hacked — it was terrifying for me and my infant son. *New York Post*. <https://nypost.com/2023/05/09/my-baby-cam-was-hacked-it-was-terrifying-for-me-and-my-infant-son/>
- Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT Security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-Scale IoT exploitations. *IEEE Communications Surveys and Tutorials*, 21(3), 2702–2733. <https://doi.org/10.1109/comst.2019.2910750>

Nidagundi, P. (2022). Government regulation on IoT. In *Apress eBooks* (pp. 161–178).

https://doi.org/10.1007/978-1-4842-8631-9_7

OWASP IoT Security Testing Guide - OWASP IoT Security Testing Guide. (n.d.).

<https://owasp.org/owasp-istg/>

What is the Internet of Things (IoT)? | IBM. (n.d.). <https://www.ibm.com/topics/internet-of-things>

Yadav, G., Allakany, A., Kumar, V., Paul, K., & Okamura, K. (2019). Penetration Testing Framework for IoT. *8th International Congress on Advanced Applied Informatics (IIAI-AAI)*. <https://doi.org/10.1109/iiiai-aa.2019.00104>