

Analysis of Security in IP-based Camera Surveillance System

Survya Pratap Singh | Email: singh.143@wright.edu

CEG 6420, Fall 2016

Abstract

Using surveillance camera as a preventive security control measure for monitoring some specific area, is not new to this world. We have always seen some sort of camera used for monitoring the public places. Most of the camera used previously were isolated analog CCTV camera, but with the advent of new technologies analog, CCTV cameras are being replaced by the IP-based cameras. IP-based cameras fall under the category of IoT (“Internet of things”) and due to low cost, it is widely used not only for monitoring the public places but is also used in homes, private offices, shops, and various other places.

IP based cameras are embedded device, which has some operating system installed on it and can be considered as a mini-computer system. The presence of this low-cost system in public web network exposed them to various type of IT security threats. This paper discusses commonly used design architecture of IP-based camera, and components used in it. The paper will review and describes an in-detail explanation of various security attack techniques, tools and malware developed by researchers for successfully exploiting the existing vulnerabilities in the IP based camera and discuss prevention measures that should be enforced on IoT devices to protect them from such attacks.

Keywords: Shodan searches engine, nmap, hydra, cgi-attack, uClinux, Mirari, and BashLite malware, XSS, CSRF, CDIS, NVR, DVR, Low orbit Ion Cannon, DOS.

1. Introduction

Easy access and availability of Internet has not only connected the different parts of the world but has also led to the invention of new technologies such as IoT. Internet of things are the embedded devices which can share the information with other devices present on the web, one of the commonly used IoT devices is IP-based surveillance camera used for remotely monitoring the secured area. Shifting from CCTV based analog camera to IP-based camera has exposed the same security risk, which is applicable for any IT department.

Based on the monitoring implementation, the IP-based camera can be divided into “centralized” and “decentralized” IP-camera. In centralized IP camera, the streaming video is encoded by the camera and is sent to NVR (Network video recorder) which is a central video server where all the streaming video is processed. In decentralized IP camera, all the intelligence is embedded in the camera itself, stream video is encoded and is stored in the digital storage media mounted inside the camera. Decentralized IP camera comes with pre-configured web application hosted on light-version of the web server. Detail design architecture of decentralized IP camera, the operating system used and type of web server deployed on these cameras will be discussed in section 2. Most of the IP camera present in the market are vulnerable because they fail to implement the industry best practices and due to low cost these devices does not go through intensive vulnerability penetration test. One of the biggest vulnerability of this IP-based camera is present of default username and password for accessing the video stream. The paper discusses how using “Shodan search engine”, a hacker

can discover vulnerable IP cameras. As per the research conducted by security firm Qualys, it is estimated that around 20% camera found online using Shodan were accessible using default id and password. The paper also describes how brute force attack can be done using “hydra tool” to automate the task of finding the vulnerable camera.

Even though this device allows the user to change the default username and password for a web application that ship with the product, these systems can still be accessed by using “SSH” or “Telnet” by exploiting the hardware and firmware vulnerability. The paper discusses some of the vulnerability and attack related to hard coding the operating system password in the firmware. IP based cameras are pre-configured with a web-based application, various security attacks which are applicable for a web application can be performed on these systems to reveal confidential information. This paper discusses how, some of the famous web application attack (XSS, CSRF, CDIS) were performed on this system. The presence of poorly configured “Lighttpd” web server also make these systems vulnerable against CGI based scripting attack. The paper also discusses some of the attack performed by researchers by analyzing the binary code of firmware and web server program installed on this system by using tools like IDA pro and Binwalk. “Foscam” being one of the well know IP camera product uses uClinux as an operating system which in the modified version of Linux v2.4, paper discuss various attack methodology performed by researchers for hijacking this product and how to use “getmecatool” to hack such vulnerable products.

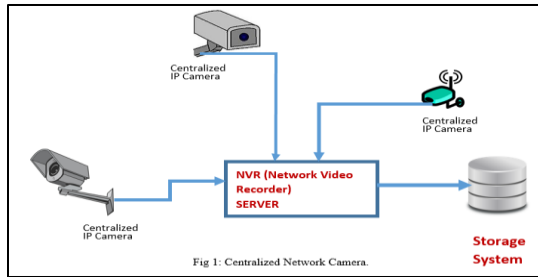
One of the largest cyber-attack in the history which occurred on 21st-Oct-2016, on “Dyn servers - a company which controls most of DNS infrastructure” resulted in bringing down the most famous websites like Twitter, Netflix, CNN, and many others for a day was caused by DDOS attack launched by the botnet network of IP-based camera. The paper describes in detail working of “Mirari” malware which was used for exploiting the

vulnerabilities in IP-based camera and form a network of a botnet.

Based on the various type of attack that is discussed, this paper will discuss a preventive measure that can be adopted in future to make IoT devices more secure.

2. IP Camera Architecture

IP camera is a combination of computer and camera, with the ability to convert the analog video signal to digital and streaming it over an IP-based network. In comparison to analog CCTV camera which uses dedicated point-to-point analog cabling, IP camera uses IP web network as a backbone for transporting the digital data and has several advantages over analog CCTV camera. IP camera provides remote accessibility for accessing the video streaming and configuring the device remotely. IP based camera are very much scalable and flexible, and can easily be integrated into pre-existing IP web network thus making them easy to plug and play. Unlike analog CCTV camera which has one-directional transportation of analog video streaming, the IP-based camera is equipped with bi-directional functionality enabling the user to send and receive data at the same time. IP Camera can be divided into 2 broad categories centralized and de-centralized system [3]. Centralized IP camera falls into the category which just performs the task of recording and converting analog video data to digital data and sends it over the web network, they do not have any storage system or web application for processing of video streaming and requires centralized NVR server for processing of digital data. NVR (Network Video Recorder) is a specialized software program used in the implementation of centralized based IP camera and is used for collecting the digital data from the various centralized camera and store it on the digital storage media.



One of the advantages of NVR over DVR (Digital Video recorder) which is also used as a centralized video processing server is that NVR collects data from IP web network, but in DVR implementation camera is directly connected to DVR for data transmission. Decentralized IP camera which is the focus of this paper are intelligent device and does not require NVR server. It has built in feature of recording the video and storing it on the digital storage media and are equipped mostly with the Linux-based operating system. Decentralized IP camera has basically three main components camera, encoder and web server.

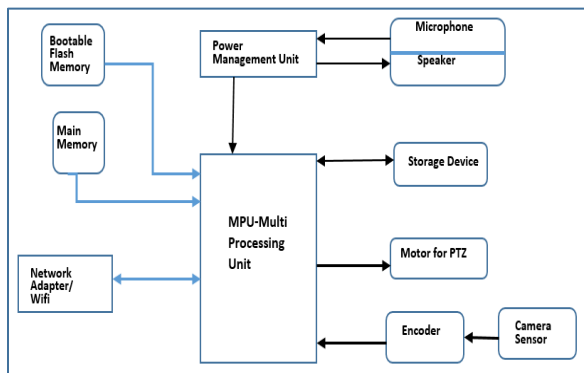


Fig 2: Decentralized Camera Components [7].

2.1 Data Flow process between User and IP Camera

Camera and microphone sensor are input devices which capture image and sound in the form of analog electrical signal. The encoder is used to convert the analog signal to digital signal and compress the video size. One of the most commonly used compression video codec is M-JPEG (Motion JPEG). The compression technique is used by this camera to increase the efficiency of video

streaming by decreasing the resolution of video by removing unwanted data. Audio is also encoded using an audio codec such as MP3, once audio and video feed are encoded they are assembled using various type of wrapper technique such as AVI. After wrapping is done it is stored in the storage media device present on the decentralized camera. As discussed earlier most commonly used the operating system on this type of system are based on modified version of Linux kernel (2.4, 2.6, 3.2). The operating system used in the camera are configured with web-server most of the devices has some version of “Lighttpd” web server. This web server is pre-configured with web-application which is access by the end user for seeing the video stream. Once the user gets connected to IP address of the camera he/she will be presented with the web application login authentication page. After getting authenticated, the user will be able to stream the live video and can modify some of the settings of the camera. IP cameras mostly use RTSP (real time streaming protocol), so that user can control the video streaming session and serves as a remote control for the user. The transmission of streaming video is done by using the RTP (Real-time transport protocol) which work like UDP [2]. After the user gets authenticated on a web application which is hosted on the IP Camera webserver. The web application on user PC will receive the meta-file which has information for creating the streaming session with the camera using RTSP and RTP protocol for streaming the video directly from the camera [4]. Once the streaming packet reaches to user PC, it is decoded by using the similar codec which was used by the camera for decoding.

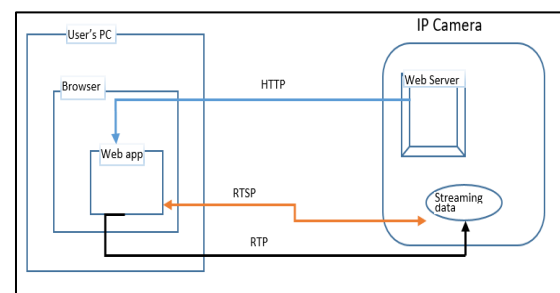


Fig 3: Communication between IP camera and User [2].

Some of the IP camera products are equipped with PTZ (pan-tilt-zoom) functionality, it gives the user to control the motion of camera by issuing the commands from a web application. When data transmission occurs between user and camera two type of data packets are sent control packets and video stream packet [2]. The control packet is generally small packet which takes command of handling the motion of PTZ and can contain confidential information such as username and password of the admin user. Most of the decentralized camera which is used in homes and for private uses works on insecure HTTP protocol. Section 3 contains the discussion on the various type of attack that can be exploited on IP-based camera based on flaws they have in their design.

3. Exploiting Design Flaws in IP Camera

Based on design implementation of the IP-based camera and especially the insecure configuration of web server gives sufficient opportunity to the hacker for exploiting this camera. Below are some of the attack technique which is targeted towards flaws in the design of IP camera.

3.1 Web application authentication Vulnerability

One of the entry points which gives motivation to hackers for trying to attack IP based camera is the presence of poorly configured web based application. Most of the IP cameras that are used in public places have static public IP address which means anyone having IP address of that system can access the web application of that devices [11]. Below are the attacks which I tried remotely on IP-based camera.

3.1.1 Shodan Tool: “Shodan” a search engine which is used for finding specific devices and device type is a great tool which is used mostly by a hacker to find the target for exploiting the vulnerability and infect it with

backdoor or malware [14]. Shodan works by scanning the entire public IP address available and tries to get banner information from scanned IP address. Shodan has been used by the researcher to identify the IP address of billions of IoT devices connected to the internet and determine the vulnerability based on the banner information. Shodan also gives the functionality of searching the web based on the query like if you want to search for IP camera which used “boa web server” and is in the United States, you can give the query in Shodan as “boa camera country:us”.

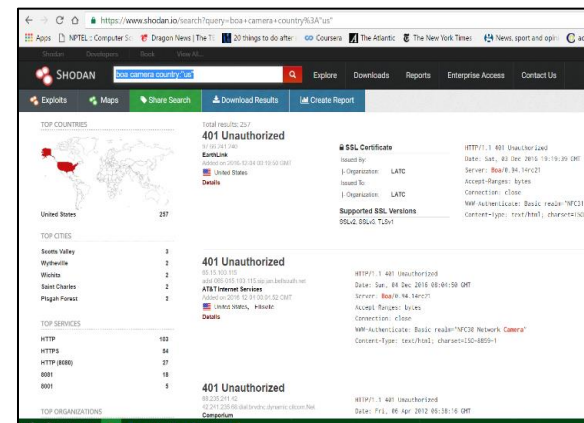


Fig 4: Shodan Search Engine

As per the research conducted by researchers worldwide, it is estimated that about 20% of the IP based camera searched by Shodan have default username and password. Getting motivated by the above data, I passed the query in Shodan as “IP Camera” and filtered it based on “Linux 2.4.x” operating system, and was surprised to find out that the first IP address which I got could authenticate by default ID as “admin” and password as “admin”. Apart from it, I could get the critical banner information such as what web server along with version number is being used to host the application inside the camera.

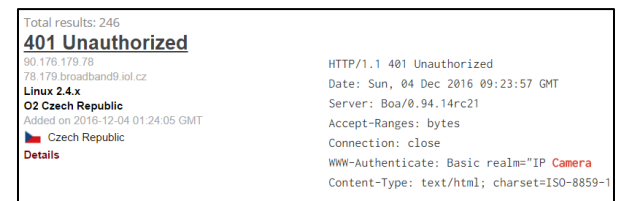


Fig 5: Banner Information from Shodan.

Using the above banner information attacker can search various attacker can search various vulnerability management websites like <https://cve.mitre.org> to find if any vulnerability exists already for the specified version and if vulnerability exist attacker can either create an exploit or can search existing exploit from websites like <https://www.exploit-db.com/>.

Prevention Control Against Shodan:

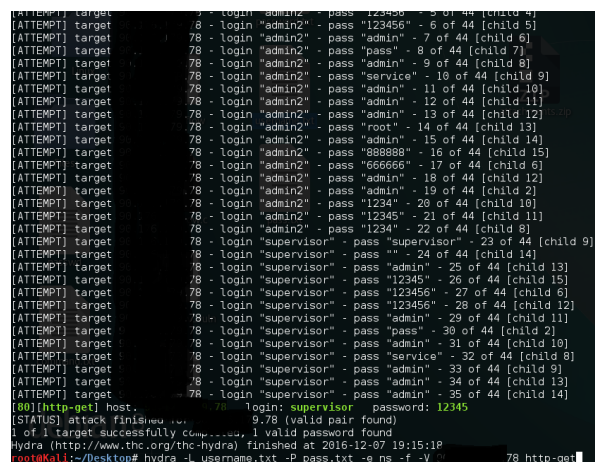
As discussed above, Shodan works by scanning entire public IP address available and tries to get banner information available with services. By default, web server displays their banner information and is unwanted data. Other than Shodan there is a tool like Netcat which is used to grab the banner information of web server. In my opinion giving information related to services running on IoT devices that too with the version number is a threat. IoT devices such as IP camera should be configured in a way that leaking of banner information and information about open ports should be avoided. Even if there is a vulnerability in IP-based web server not revealing the banner information will make the hacker task more complicated.

3.1.2 Brute Force Attack Using Hydra Tool:

Brute force attack is a technique of attacking the insecure web application login authentication system by using different username and password. I was surprised to see that none of the web application of the IP-based camera, which I tested for brute force attack has implemented the “minimum login attempts” which is one of the industry best practices. By searching in the Google, one can easily get default username and password of all the IP-based camera available in the market and a list of password file for launching the dictionary based attack. Using Hydra tool which is available in Kali Linux, I searched the IP address of camera using Shodan and tried to launch the brute force attack and could get the authentication credentials. Even I found that this web application does not except special character in username and password and have maximum fixed size of the password as 12, thus making it, even more, easier for a hacker to launch brute force attack.

Below is the screenshot of successful brute force attack performed on targeting the web application of one of the IP-based camera searched by Shodan.

Hydra command for brute force “hydra -L <username.txt> -P<password.txt> -e ns -f -V <IP address> http-get” [15].



```
[*] (http-get) host: 78 - login: 'admin2' - pass: '123456' - 5 of 44 [child 4]
[ATTEMPT] target 78 - login: 'admin2' - pass: '123456' - 6 of 44 [child 5]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 7 of 44 [child 6]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'pass' - 8 of 44 [child 7]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 9 of 44 [child 8]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'service' - 10 of 44 [child 9]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 11 of 44 [child 10]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 12 of 44 [child 11]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 13 of 44 [child 12]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'root' - 14 of 44 [child 13]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 15 of 44 [child 14]
[ATTEMPT] target 78 - login: 'admin2' - pass: '888888' - 16 of 44 [child 15]
[ATTEMPT] target 78 - login: 'admin2' - pass: '666666' - 17 of 44 [child 6]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 18 of 44 [child 12]
[ATTEMPT] target 78 - login: 'admin2' - pass: 'admin' - 19 of 44 [child 2]
[ATTEMPT] target 78 - login: 'admin2' - pass: '1234' - 20 of 44 [child 10]
[ATTEMPT] target 78 - login: 'admin2' - pass: '12345' - 21 of 44 [child 11]
[ATTEMPT] target 78 - login: 'admin2' - pass: '1234' - 22 of 44 [child 8]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'supervisor' - 23 of 44 [child 9]
[ATTEMPT] target 78 - login: 'supervisor' - pass: '1' - 24 of 44 [child 14]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'admin' - 25 of 44 [child 13]
[ATTEMPT] target 78 - login: 'supervisor' - pass: '12345' - 26 of 44 [child 15]
[ATTEMPT] target 78 - login: 'supervisor' - pass: '123456' - 27 of 44 [child 6]
[ATTEMPT] target 78 - login: 'supervisor' - pass: '123456' - 28 of 44 [child 12]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'admin' - 29 of 44 [child 11]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'pass' - 30 of 44 [child 2]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'admin' - 31 of 44 [child 10]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'service' - 32 of 44 [child 8]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'admin' - 33 of 44 [child 9]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'admin' - 34 of 44 [child 13]
[ATTEMPT] target 78 - login: 'supervisor' - pass: 'admin' - 35 of 44 [child 14]
[00] (http-get) host: 78 - login: 'supervisor' - password: '12345'
[STATUS] attack finished on 78 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
hydra (http://www.thc.org/thc-hydra) finished at 2016-12-07 19:15:18
root@kali:~/Desktop# hydra -L username.txt -P pass.txt -e ns -f -V 78 http-get
```

Fig 6: Hydra tool demonstrating Brute Force Attack on IP Camera.

Prevention control Against Brute Force Attack:

There are various security best practices that can be implemented to prevent Brute force attack on IoT devices. IP camera manufacturer should force the user to change default user id and password and one way of doing this can be by assign unique default password for each device. When a user tries to access streaming video for the first time he will be forced to change user id and password by verifying the devices unique password and will be allowed only to set a complex password with alphanumeric digits. To prevent brute force attack, the web server should implement maximum account login attempt and should lock the account for certain time (around thirty minutes), if maximum account login attempt has been reached or user has to physically restart the camera for logging into the web application of Ip camera. I think by implementing this into web application will make it impossible for hacker/attacker to try brute force attack remotely on IoT camera devices.

3.2 Attacking Availability of IP Camera

Availability being one of the 3 pillars in CIA triad, can easily be compromised in IoT devices, which usually have small RAM size and virtual memory. IP based camera are like any computer, the only difference is that they are dedicated to performing specific task thus has very low system configuration. So, any attack is possible on the normal system can easily be performed on IP-based camera. Motivated by this idea I thought of performing the DOS attack (Denial of Services) on one of the publicly available IP camera which I searched using Shodan search engine. Most of the IP camera in market work on the concept of “multicasting” that is more than one user can connect with a camera to stream the live video and that I think is one of the vulnerabilities which invites attacker to perform DOS attack. Denial of services attack works by sending many packets to open the partial connection with the devices in a short span of time. One of the tools which are quite famous in hacker’s community is “Low orbit Ion Cannon [9]” and this tool was used previously to launch DDOS attack on companies’ like Sony, PayPal, MasterCard, and Visa. This tool allows the user to send Http/TCP/UDP packets and allows the user to set threads to run for initiating the connection with the targeted device. Below is the screenshot of live streaming shown by one of the cameras searched by Shodan.

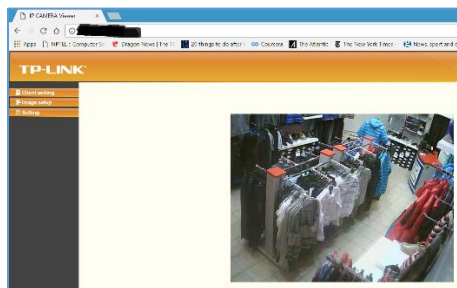


Fig 7: Live Streaming before DOS attack.

We can see that the above IP camera is being used for monitoring the shop, suppose if the attacker was to perform theft at this store and his is aware that this shop is monitored by IP camera, he can easily launch the DOS attack, thus compromising the main functionality of this device that is monitoring. I did launch the DOS attack

for the above IP camera and while the attack was in process, I tried to stream the video and page failed to load.

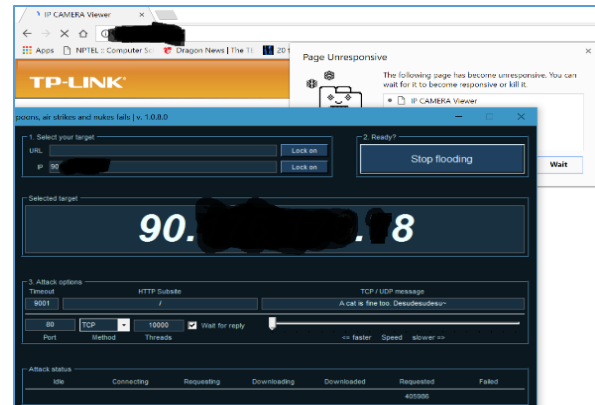


Fig 8: Page Unresponsive due to DOS attack on IP Camera using LOIC.

A creating tool like is LOIC is not a rocket science anyone with good programming skill can create a program for initiating the partial connection over HTTP which used port 80 for communication and is generally open in all web server.

Prevention Control Against DOS attack

Even for the normal system/computer with good hardware configuration, it is challenging task to protect them from DOS attack. There are many products in the market like IDS, Firewall which helps to prevent against such attack but is expensive to use for IoT devices like decentralized IP camera which has inbuilt web server for hosting web application. One of the techniques that I think can prevent DOS attack IP camera is by using unicast video streaming from the server rather than multicast streaming. The web server should only allow one user to connect at a time and discard any connection request while the web application is being used by the user. Even though implementing unicasting will limit the streaming availability to multiple users but it will surely help in preventing the DOS attack. Preventing IP camera along with multicasting ability should be the area of future research as DOS attack are easy to launch

not only on IP camera but on any “Internet of things” devices which are connected to public web network.

3.2 CGI-Based Attack on IP Camera

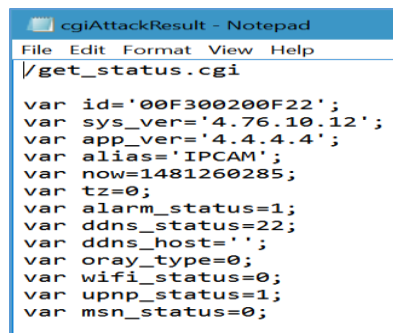
As per the research conducted by “Aditya K Sood” and “Bipin Gajbhiye” [1], which focused on exploiting the vulnerability present in the web application interface hosted on the web server of the IP-based decentralized camera. The paper presented by them shows attack techniques which are also a part of OWASP Top 10 vulnerability for any web application. The first attack which is discussed in the paper is based on RFI (remote file inclusion) attack. RFI is an attack technique which is used for exploiting the web application which has “dynamic file include” functionality [8]. “Dynamic file include” technique is provided by languages like PHP and Java and is mostly used by web developer for including dynamic header and footer in a web application. The researchers could identify one of the IP camera product which was using the video streaming page as included URL in main web application and were launch CDIS (cross domain image streaming) attack. In this attack, they created a counterfeit video streaming page and by using JavaScript they could modify the content of original web page by replacing the included file name with the counterfeit one for e.g. if real page has “include(\$file)” attacker can modify this file with their fake file and if user will try to view the streaming video, he will be presented with the video from fake domain. In their paper, they also demonstrated XSS and CSRF attack which are based on vulnerability related to JavaScript. In XSS attack, the attacker will craft the malicious JavaScript code aimed at stealing the credential/session/cookies from the already authenticated user, but the drawback of this vulnerability is that attacker should use some social-engineering trick so that he can make the user run the malicious code on this web application. Cross-site request forgery is also a scripting based attack which is demonstrated in the paper, in this attack attacker will force already authenticated the user to executed an unwanted action on a web application. The researchers could conduct above attack because the web application

was not validating the input command which is normally done by the secured web application. One of the other attacks which were also demonstrated in this paper was based on the vulnerability present in web server configuration. CGI (common gateway interface) is a technology used for communication between web server and web application and is used by almost all the IP camera product available in the market for dynamic execution of web server program [1]. A most common attack that can be performed on IP camera devices using CGI are remote command execution and bypassing web application authentication mechanism. Insecure CGI design in IP camera provides an attacker with the ability of remotely rebooting the device (`http://<ip of camera/reboot.cgi>`) or gaining the admin user and password (`http://<ip of camera>/check_user.cgi`) with any authentication. The paper has not mentioned any details about how they tested for the different CGI command to find vulnerabilities. To speed up the process of checking the CGI command vulnerability using URL, I have created a tool in python which will remotely check for the CGI attack on the camera. The tool is command line tool which takes 3 parameter IP address of the target, the port number to communicate and CGI script file which will have all the CGI command available for that IP camera brand, which is easily available on the Internet. Below is the screen shot of the tool being used in CMD for testing of CGI attack.

Fig 9: CGI attack testing tool.

We can see in the screenshot, that one successful CGI attack was found by the tool and the result of the successful CGI command will get stored in the file “cgiAttackResult.txt”. The tool works by sending the

URL request with the CGI command mention in the file and if the response is received, it gets stored in the output file.



```
File Edit Format View Help
/get_status.cgi

var id='00F300200F22';
var sys_ver='4.76.10.12';
var app_ver='4.4.4.4';
var alias='IPCAM';
var now=1481260285;
var tz=0;
var alarm_status=1;
var ddns_status=22;
var ddns_host='';
var oray_type=0;
var wifi_status=0;
var upnp_status=1;
var msn_status=0;
```

Fig 10: CGI attack output.

As we can see “get_status.CGI” CGI command was executed without asking for login credentials, which is a major flaw in the design of IP web camera and can be found in many IP camera products in the market.

Prevention Control against CGI attack

The paper does not provide any information related to prevention against the attacks and fails to describe in detail how they exploited the vulnerabilities [1]. One of the prevention control that can be used against CGI based attack is by preventing normal user to run CGI command on these devices. CGI command can only be executed by admin user and each CGI command should pass through secure login authentication procedure before being executed.

3.3 IP Camera Video Streaming Attack

Due to the low-cost availability of IP camera about 95% of available devices transmit streaming data over Http, without using any encryption method. One of the reasons for not using the encryption is to increase the performance of streaming data between the end user and device. Transmitting plain data gives an opportunity to an attacker to launch MITM (Man-in-the-middle) attack, thus compromising the confidentiality of IP camera devices.

3.3.1 MITM Attack

MITM attack mostly takes place when the attacker is also a part of target’s local web network and any communication between device and end user must pass through the attacker system. MITM attack can be easily done by using “arp spoof” tool which is a part of kali Linux. Passing information in plain text over HTTP can also be compromised by sniffing attack, an attacker can use tool like “Wireshark” in promiscuous mode to sniff the packet transmission between devices and end user. Below is the screenshot of sniffing attack which I performed on one of the IP cameras searched from Shodan to check the how username and password for IP camera login are being sent in plain text.

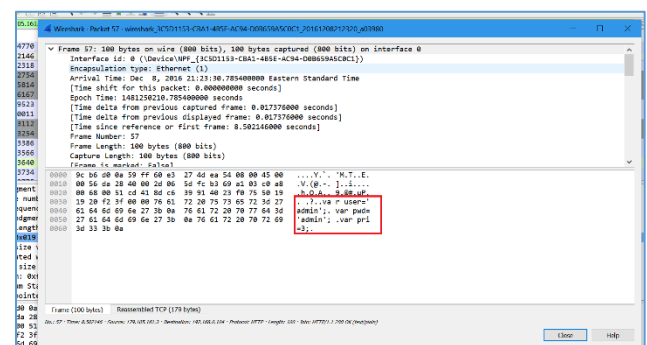


Fig 11: Sniffing attack showing Login credentials of IP camera in Plain text.

Prevention against MITM/Sniffing attack [2]

Encrypting the video streaming data is an obvious solution to maintain the confidentiality of data, but in IP-based camera encrypting data will affect the performance of live video streaming functionality. Referring to a research paper written by “Liu, Peng and Zheng [2]”, where they have demonstrated how encryption can be achieved without decreasing the performance. They gave the idea of using hardware-based random number seed generator along with the pseudo-random generator to generate a sequence of random number. This generated random number will then be used with XOR operator on the video stream data. The random key and XOR-operated data then will

be transmitted to the end user. The user will again use XOR with the provided key on already XOR-operated data to get the video stream. Researchers have to use symmetric stream cipher RC4 for encrypting the random key which is getting transmitted to the end user. For sharing the public key of RC4 cipher they have to use the public key cryptosystem. The idea present by them seems feasible as they are not trying to encrypt entire communication between devices and end user but are only using XOR operator on video streaming data. They have not discussed in the paper what kind of public key cryptosystem they are using for sharing the public key. I think using Diffie-Hellman key exchange would really help in this scenario as it is mostly used for creating shared secret between two parties.

3.3.2 OS Default Hard-Coded Password Attack

In the research paper written by “Craig Heffner [5]” where he demonstrated remote exploitation of 0-day vulnerabilities on various brands of IP-based camera. The methodology which he used for finding the vulnerabilities on this device was simple. He downloaded the firmware image used in the IP camera from the product website which is available for free to anyone. Then he used IDA Pro tool which is a software disassembler tool for identifying the bugs in a specific module. He then used Binwalk and Firmware-mod-kit for creating modified version of the same firmware which he later used for upgrading the devices with his modified firmware by exploiting the CGI based vulnerabilities present in the web application of cameras. Most of the IP camera displays their model number and other details on login authentication page, once can easily use this information for finding the firmware from the internet and follow the above steps for finding the vulnerabilities. The vulnerability he found using this technique on the various product were related to unauthenticated access to CGI command, use of unsafe functions like “strcpy and sprintf” which can cause a buffer overflow. Out of all the research paper I have read related to finding vulnerabilities in IP based, this paper gave a proper methodology for finding the vulnerability in IP camera remotely. Most of the IP-

based cameras use hardcoded password for operating system being used on the device and have their telnet port open. Using this knowledge attacker can easily gain access to the root shell command of the operating system. Once he gains access to the shell, can execute Linux command to modify the operating system files remotely. One of the attacks shown in the paper was to modify the video feed once an attacker gains access to root shell. In most IP camera “mjpg.cgi” or “videostream.cgi” is used for streaming the data, attacker can run “top” command in the shell and can see all the running process and if he kills the “mjpg.cgi” or “videostream.cgi” the video stream on the end user screen will be paused and user will think that he is receiving the normal video streaming, until he clicks the refresh button on browser. The attack technique described in this paper is impressive and more research can be conducted following the same process.

Prevention against OS default Coded Password attack

Generally, IP camera product manufacturer used Linux based operating system which comes with the default username and password, and due to ignorance of security concern in the manufacturing process, they keep the same default password for each of the devices. One simple way of avoiding this type of attack is to set unique OS password for each product during manufacturing process, and unwanted ports like 23(telnet),21(SSH) should never be enabled in the IP-based de-centralized camera.

4. Mirari Malware and IP Camera

One of the biggest DDOS attacks of Internet history that occurred on 21-oct-2016 against “DYN servers” an Internet performance management company which offer it services to websites like Twitter, Netflix, Google and many other popular websites was caused by botnets formed by compromising above-discussed vulnerabilities in IoT devices especially IP-based cameras using Mirari malware. And on top of it the

[illegible]

4.1 Mirari Working

And it is estimated that around 500,000 IoT devices at present are vulnerable to Mirari malware, and

5. Future Research

6. Conclusion

10 | Page

References

- [1] Sood, A. K., & GAJBHIYE, B., “Design Flaws in IP Surveillance Cameras Exploiting Web Interfaces”. Retrieved from https://www.cigital.com/papers/download/design_flow_s_IP_surveillance_cameras_adityaks_bipin.pdf.
- [2] Zhaoyu Liu, Dichao Peng, Yuliang Zheng and J. Liu, "Communication protection in IP-based video surveillance systems," *Seventh IEEE International Symposium on Multimedia (ISM'05)*, 2005, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1565815&isnumber=33218>
- [3] “Design and Implementation of an IP-Based Security Surveillance System”. Retrieved from <http://www.ijcsi.org/papers/IJCSI-9-5-1-391-400.pdf>
- [4] IP-Surveillance design guide. (n.d.). Retrieved from http://www.imctv.com/pdf/ipcamera/IP_Surveillance_Design_Guide.pdf
- [5] “EXPLOITING SURVEILLANCE CAMERAS.” Retrieved from <https://media.blackhat.com/us-13/US-13-Heffner-Exploiting-Network-Surveillance-Cameras-Like-A-Hollywood-Hacker-WP.pdf>.
- [6] “To Watch Or To Be Watched Turning Your Surveillance Camera against You.” Retrieved from <https://conference.hitb.org/hitbsecconf2013ams/materials/D2T1%20-%20Sergey%20Shekhan%20and%20Artem%20Harutyunyan%20-%20Turning%20Your%20Surveillance%20Camera%20Against%20You.pdf>
- [7] “IP camera application block diagram. (n.d.). IP Camera Application Block Diagram.” Retrieved from http://www.nxp.com/files-static/graphic/block_diagram/APLSECURITYENHANCEDCAMERA_BD1.html?&keepThis=true&TB_iframe=true&height=480&width=75
- [8] “Video Streaming Protocols. Retrieved from” http://www.cohuhd.com/Files/white_papers/CohuHDV_ideoStreamingProtocolsWhitePaper.pdf
- [9] “LOIC Tool usage” [online] <http://resources.infosecinstitute.com/loic-dos-attacking-tool/>
- [10] "IP Video Standards.”. Retrieved from https://www.security.honeywell.com/documents/IP_Video_Standards.pdf
- [11] “Mirari Malware” [online] <https://www.incapsula.com/blog/malware-analysis-Mirari-ddos-botnet.html>
- [12] “Mirari Source code” [online] <https://github.com/jgamblin/Mirari-Source-Code>
- [13] “IP Camera Standard/Protocol – ONVIF” [online] <http://www.hkvstar.com/technology-news/ip-camera-standard-protocol-onvif.html>
- [14] “Using Shodan” [online] <https://danielmiessler.com/study/shodan/#gs.1m9fA1E>
- [15] “Hydra Tool usage” [online] <http://tools.kali.org/password-attacks/hydra>