Pentesting Report: Ebode IP Camera

Group 2

April 3, 2020

1 Introduction

During the IoT analysis, we read the IP Camera packaging and manuals to understand the physical and commercial properties of the product. Then we used an Ethernet connection to observe the data the device produced, this was then scanned and analysed using Wireshark and other tools. During the process of reverse-engineering the firmware structure, some common vulnerabilities were discovered and then analysed. The device was finally connected fully to the network and network and web analysis of the packets was completed.

1.1 Outline

Our security analysis is based on networking, firmware and mobile app security. We were able to find some vulnerabilities in the web and firmware sections, but we were not able to fully analyse the app due the payment requirement. After completing our analysis, we shall confirm whether this device is a security risk and is `not secure`.

1.2 An Overview of the Weaknesses Found

A brief summary of the vulnerabilities we discovered during the penetration process.

- After the camera is initialized, the camera uses the default /admin account and password. Since many users do not modify their default passwords after setting up the product, this can lead to intrusion by malicious attackers. The design should force the user to set a new password when the user first logs and specify the strength needed for the password, for example: not less than 8-bit characters with English capital and special characters mixed.
- HTTPS protocol should be used to protect user communication security. However, the device uses the unencrypted HTTP protocol.
- All user account passwords of the device are stored in plain text. Accessing `get_status.cgi` with the login state will return the user's plain text account password. The server should only store encrypted user passwords to prevent attackers from deciphering them if acquired.
- Attackers can brutally enumerate and search DDNS

2 Investigating the device

After being given the IP camera we saw several notable things on the box, such as the company name and model number. Searching online brought up sale pages that provided a user manual and details that can be found on the box, which would be useful for members who did not have the device in their possession.

	Details
Name	Ebode IP Camera IPV68
Firmware Number	17.35.2.49
UI Numbe	20.8.5.38

2.1 Analysis of Firmware Security

The network function of the camera needs to be set up using a wired network and then connected to WLAN. This allows for wireless connection and control.

2.1.1 Find the Firmware and Download

Once the IP was configured, we can access the web management system of the camera. Then a hidden file was found through this link: `http://100.84.124.183/get_params.cgi`

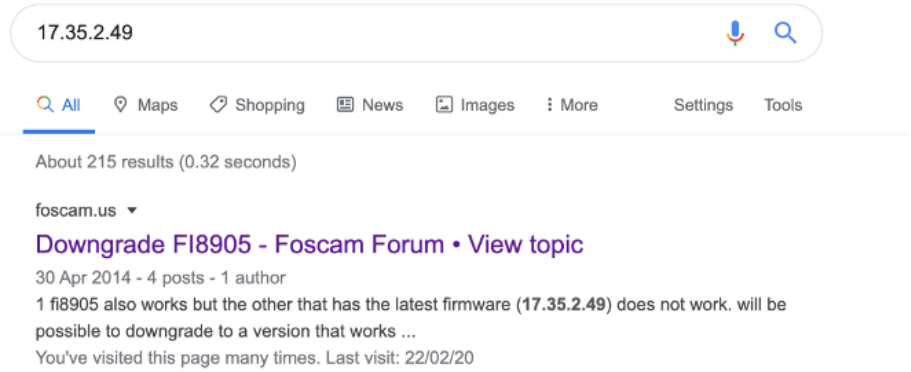
Figure 1: Firmware Version

```
HTTP/1.1 200 OK
Server: Netwave IP Camera
Date: Thu, 01 Jan 1970 00:03:36 GMT
Content-Type: text/plain
Content-Length: 4370
Cache-Control: no-cache
Connection: close

var id='54CDEE003896';
var sys_ver='17.35.2.49';
var app_ver='20.8.5.38';
var alias='';
var now=216;
var tz=0;
var daylight_saving_time=0;
var ntp_enable=1;
var ntp_svr='time.nist.gov';
var user1_name='admin';
var user1_pwd='admin';
var user1_pri=2;
```

Then we used Google search engine to search version number and found the following information.

Figure 2: Version Number



In order to get the firmware, luckily we found that the firmware of the Ebode camera is the same as `foscam` camera, a different manufacture.The firmware of our camera uses the same firmware as a camera called `foscam`. By looking through the documentation [2] on the foscam official website, we found that foscam OS systems have many version numbers, such as x.x.2.41.

Then, we downloaded the firmware for our camera version from:
`http://www.foscam.eu/index.php/productattachments/index/download?id=129`

2.1.2 Brute force enumeration search DDNS

We read the documents and know that the product has many domain names connected to the outside network. The cameras of this brand will be assigned a 6-character DDNS domain name for remote access to users. The DNS

domain name is in the following form: `******.Myfoscam.org`. Because of this structure an attacker can very easily brute force the correct domain name.

As a defense solution, the manufacturer could be allowed to choose whether to place the domain name resolution records in the program according to the actual situation, reducing the risk of the device's DNS domain name being hijacked.

Manufacturers can also use DNS over TLS (DoT). Using the TLS protocol can ensure the integrity and confidentiality of the DNS query process and prevent DNS requests from being hijacked.

2.1.3 Analyzing the Firmware

In order to analyse the firmware, there are several several following commands were run to go in depth and extract its content:

- Use command **% hexdump lr_cmos_11_35_2_49.bin | less**.

  We can find that the fourth byte of the second line of the file is `50 4b 03 04`, whose corresponding ASCII characters are `PK`. Details from the figure 17 in the appendix.

- Use command **dd if=lr_cmos_11_35_2_49.bin of=test.zip skip=0x14 count=764084 bs=1** .

  These characters we knew from last step are the *Magic Number* of the compressed file, so we can get the contents of the zip file through the `dd` command.

Figure 3: Zip file format

End of central directory record:

Offset	Length	Contents
0	4 bytes	End of central dir signature (0x06054b50)
4	2 bytes	Number of this disk
6	2 bytes	Number of the disk with the start of the central directory
8	2 bytes	Total number of entries in the central dir on this disk
10	2 bytes	Total number of entries in the central dir
12	4 bytes	Size of the central directory
16	4 bytes	Offset of start of central directory with respect to the starting disk number
20	2 bytes	zipfile comment length (c)
22	(c)bytes	zipfile comment

- Get the **linux.bin** file. we can identify from the remaining bytes that this is a `romfs` file system.

Figure 4: Linux.bin File

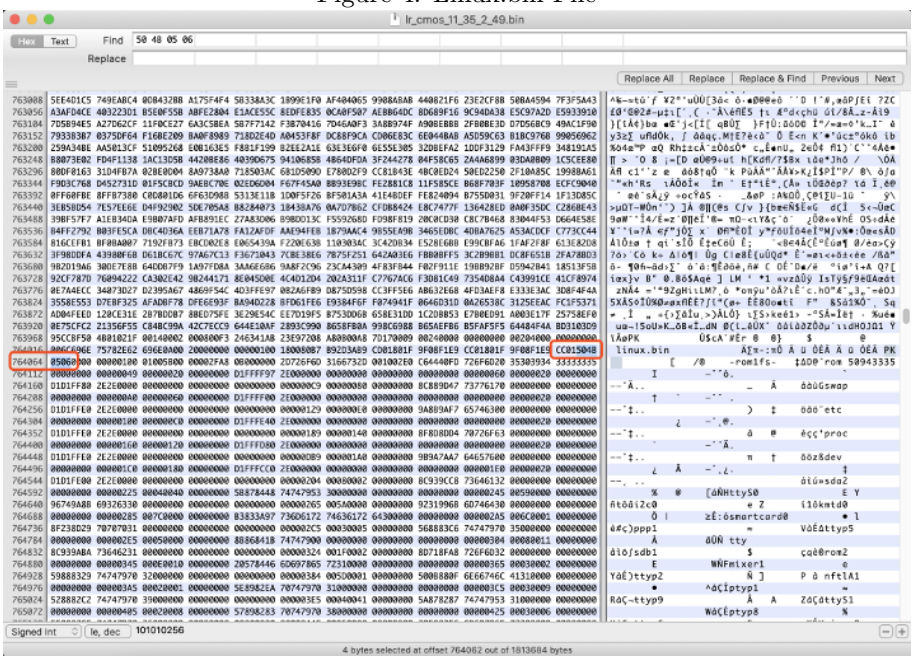
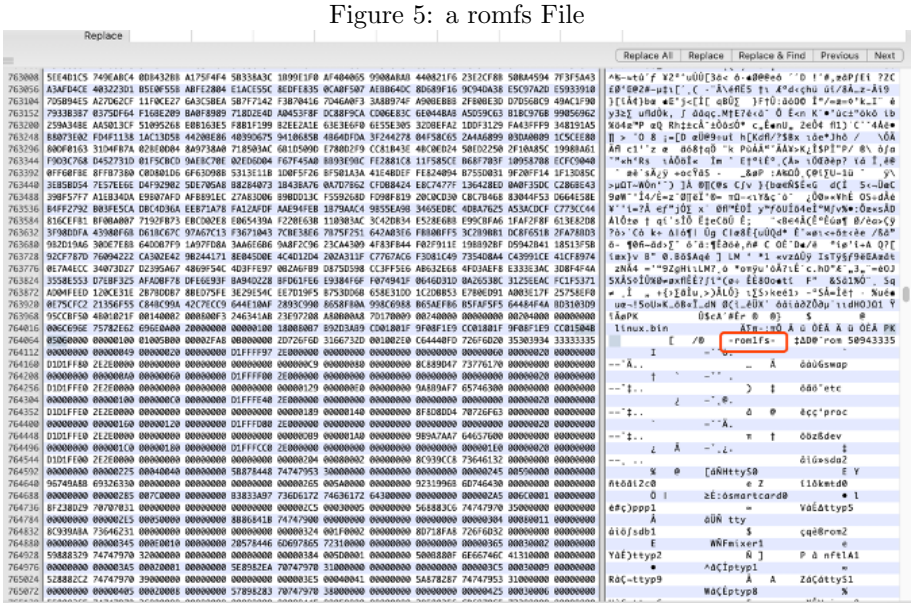


Figure 5: a romfs File



- Use command **mount -o loop romfs.img**

  After mounting Linux.bin, we can **extract the camera's bin file** from the directory.

  The camera bin file contains (or includes)http server and it controls the camera

- Use command **cat linux.text linux.data > linux.bin**

  We found this to be a `bflt` file by reading the literature.

  The file format is to extract `text` and `data` parts from the `ELF` file in Linux into a new file.

- **Analyze the program file of camera**

  IDA provides a python script that helps parsing bflt format, however, the script uses `coolfire mode` by default. So we modified the script mode to arm, which is used by the camera.

2.2 Analysis of Network Security

The camera was initially connected to a laptop Ethernet port. A Wireshark capture of this connection shown below displays the device attempting to receive an IP number from a DHCP server.

Figure 6: A Wireshark capture

Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000…	0.0.0.0	255.255.255.255	DHCP	318	DHCP Discover - Transaction ID 0x7af22581
2	0.028796…	::	ff02::16	ICMP…	90	Multicast Listener Report Message v2
3	0.158842…	::	ff02::1:ff56:1…	ICMP…	86	Neighbor Solicitation for fe80::9b93:4aba:5b56:13da
4	0.408742…	::	ff02::16	ICMP…	90	Multicast Listener Report Message v2
5	2.508743…	fe80::9b93:4ab…	ff02::16	ICMP…	90	Multicast Listener Report Message v2
6	2.668800…	fe80::9b93:4ab…	ff02::16	ICMP…	90	Multicast Listener Report Message v2
7	2.953104…	0.0.0.0	255.255.255.255	UDP	60	2049 → 10000 Len=1
8	3.172664…	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x5430ae10
9	5.225504…	fe80::9b93:4ab…	ff02::2	ICMP…	62	Router Solicitation
10	6.673607…	0.0.0.0	255.255.255.255	DHCP	318	DHCP Discover - Transaction ID 0xecff4923
11	7.184345…	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x5430ae10
12	9.227439…	fe80::9b93:4ab…	ff02::2	ICMP…	62	Router Solicitation
13	15.27384…	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x5430ae10
14	15.48577…	0.0.0.0	255.255.255.255	DHCP	318	DHCP Discover - Transaction ID 0xf1223a9c
15	31.42302…	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x5430ae10
16	31.49959…	0.0.0.0	255.255.255.255	DHCP	318	DHCP Discover - Transaction ID 0x868e7d11

It was believed that it may be necessary to set up our own DHCP server or even a full network to effectively test this device but we found that on most modern OSs there is an option to set up a bridge between two network interfaces; allowing the IP camera to connect through to the wireless network the laptop is also connected to.

After connecting, a tool was used that came with the camera (called IP camera) which searched online for the company website linked with the camera to allow us to set the IP address of the camera. We were now able to connect to this camera via the web link `http://ip:80`.

This didn't allow all group members to connect however, an initial VPN server was tested but could not be made to work in time and so had to be scrapped.

2.2.1 IP Camera network analysis

One of the serious network security vulnerabilities in the IP camera is that it uses port 80 to forward traffic by default, to forward traffic, using the less secure HTTP rather than HTTPS. The attacker can easily get the information by sniffing the traffic between the camera and desktop using Wireshark. Because the version of requests used is `HTTP/1.1`, this again proves the traffic is not encrypted.

Figure 7: A HTTP packet capture

```
  TCP payload (474 bytes)
v Hypertext Transfer Protocol
  v GET / HTTP/1.1\r\n
    v [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
        [GET / HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: 172.22.200.222\r\n
    Connection: keep-alive\r\n
  v Authorization: Basic YWRtaW46\r\n
      Credentials: admin:
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,ar;q=0.7\r\n
```

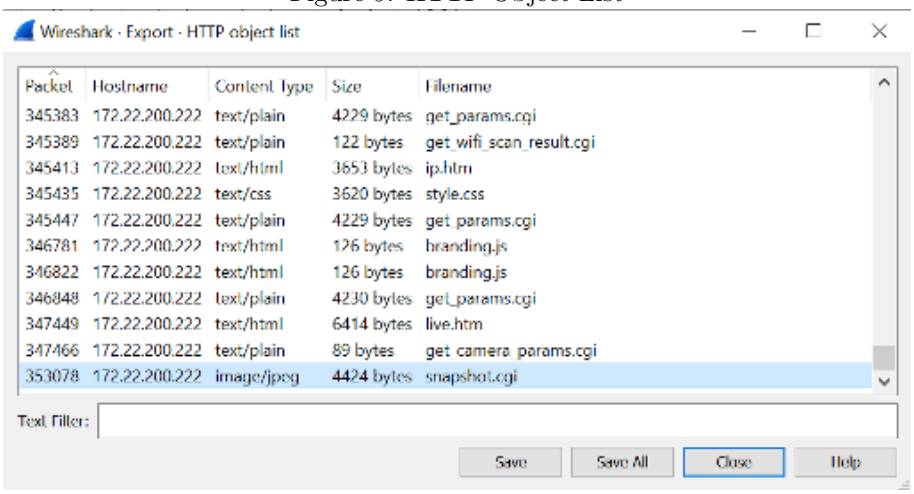
We found that the type of the authorization is weak, credentials travel through a header encoded only in Base 64. Decoding the `YWRtaW46` string gives `admin:` and no other data, leading us to the fact that there is no password.

Figure 8: Encoded in Base 64

```
Authorization: Basic YWRtaW46\r\n
    Credentials: admin:
```

It is also possible for the attacker to intercept live streams and images from incoming traffic. From the HTTP request within this traffic, it contains `images/jpeg`which can capture from the camera.The attacker is then able to extract this image and export it (with the `.jpeg` extension) to their desktop.

Figure 9: HTTP Object List



It's also possible to link to a livestream from the camera by accessing the TCP stream.

Figure 10: TCP Stream

```
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,ar;q=0.7\r\n
\r\n
[Full request URI: http://172.22.200.222/live.htm]
[HTTP request 1/1]
[Response in frame: 209309]
```

2.2.2 Web application analysis

We use an intermediate proxy server to intercept the data packets, and then analyze the packets. Then we analyzed the packets with the reverse code of the firmware (section 3.1) and found the following vulnerabilities.

- **CSRF (one-click attack)** - `High Risk`

  The attacker deceived the user's browser through some technical means to visit a website that he had authenticated and run some operations. Since the browser has been authenticated, the website visited will be considered as a real user operation and run. This takes advantage of a loophole in user authentication in the web: simple authentication can only guarantee that the request is sent from a user's browser, but it cannot guarantee that the request itself is issued by the user voluntarily [1].

  The attacker sends the link to the administrator or embeds the link in another HTTP page. When the administrator browses to this page, the attacker can modify the password of the camera through the corresponding malicious code.

  Attack Link `http://100.84.124.155/set_users.cgi?next_url=rebootme.htm&user1=admin&pwd1=admin&pri1=2&user2=&pwd2=&pri2=0&user3=&pwd3=&pri3=0&user4=&pwd4=&pri4=0&user5=&pwd5=&pri5=0&user6=&pwd6=&pri6=0&user7=&pwd7=&pri7=0&user8=&pwd8=&pri8=0`

Possible Security Risks

- Attackers can modify the security settings of users and administrator accounts, or perform dangerous operations to deceive users.
- This vulnerability can also be combined with XSS vulnerabilities to increase risk.

Repair suggestions

- Verify the `HTTP Referer`.
- Add `token` verification to the request packet.
- Client attribute verification in `HTTP header`.
- Use graphical verification codes or SMS verification codes for sensitive operations.

- **HTTPS is not enabled** – `High Risk`

  Since HTTP is a unencrypted protocol, the protocol does not guarantee the following security: Confidentiality, Completeness, Replay Attacks defense and Certification.

  Nodes that HTTP data packets pass through (such as public WIFI or operators) can capture plain text data packets, know the contents of the data packets and modify them.

Figure 11: HTTP protocol

Repair suggestions : Enable HTTPS protocol.

- **Server Header is not enabled X-FRAME-OPTIONS** – `High Risk`

  An attacker can use `a transparent iframe` to overlay on a normal web page, and then induce users to operate on the web page. When a user clicks on a transparent iframe page without knowing it, the user 's operation has been hijacked onto a malicious button or link designed by the attacker in advance

Figure 12: HTTP response header

```
HTTP/1.1 200 OK
Server: Netwave IP Camera
Date: Wed, 18 Mar 2020 17:12:21 GMT
Content-Type: text/plain
Content-Length: 350
Cache-Control: no-cache
Connection: close
```

Possible Security Risks

- Fishing attack.
- Users can be induced to modify security settings without their knowledge.
- Can induce users to carry out dangerous operations.
- It can be combined with XSS, CSRF, etc. to increase the vulnerability.

Repair suggestions

- The server turns on X-FRAME-OPTIONS.
- Verification code operation authentication for sensitive click operations

- **Sensitive Information Leakage** - `Low Risk`

  The attacker can access the link without authentication: `http://100.84.124.155/get_status.cgi`

  The API can obtain system information of the camera, such as system firmware version number and other information.

Figure 13: Camera System Information

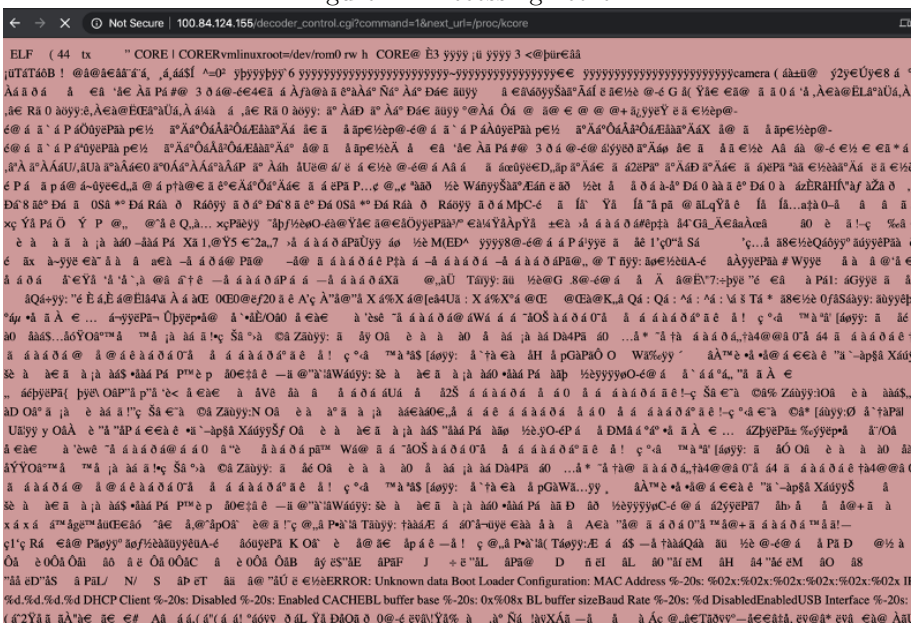
```
HTTP/1.1 200 OK
Server: Netwave IP Camera
Date: Wed, 18 Mar 2020 17:12:21 GMT
Content-Type: text/plain
Content-Length: 350
Cache-Control: no-cache
Connection: close

var id='54CDEE003896';
var sys_ver='11.35.2.49';
var app_ver='20.8.5.38';
var alias='';
var now=1584551541;
var tz=0;
var alarm_status=0;
var ddns_status=0;
var ddns_host='';
var oray_type=0;
var upnp_status=0;
var p2p_status=0;
var p2p_local_port=20669;
var msn_status=0;
var wifi_status=0;
```

- **Arbitrary File Read** - `High Risk`

  We found that the link `http://100.84.124.155/decoder_control.cgi?command=1&next_url=/proc/kcore` can read any file. By accessing `kcore`, we can get all the information in memory, including the user account name and password.

Figure 14: Accessing kcore



Then we search for keywords related to sensitive information. After searching, we found the password and admin account in memory.

Figure 15: Search PASSWORD

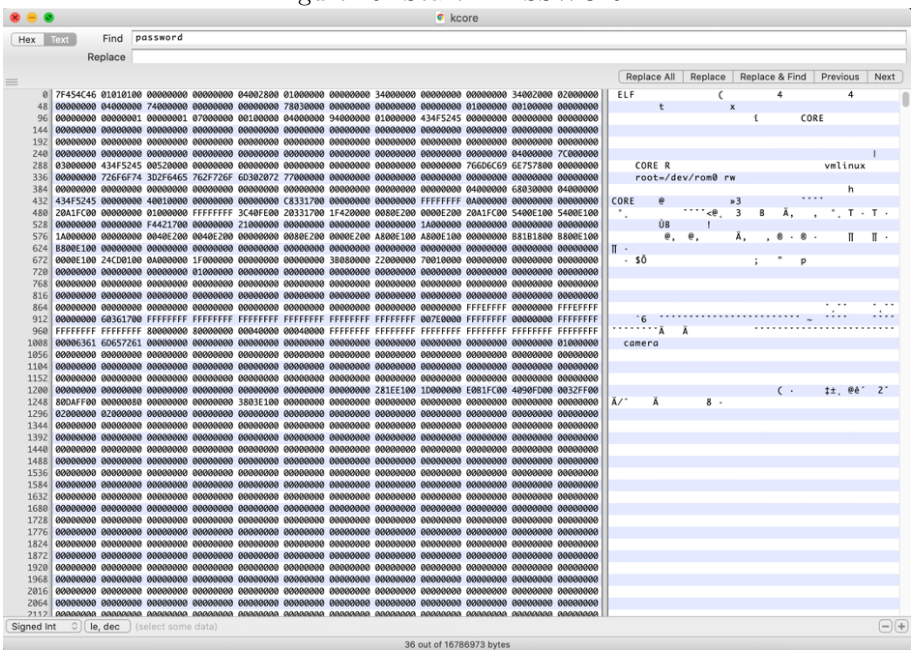
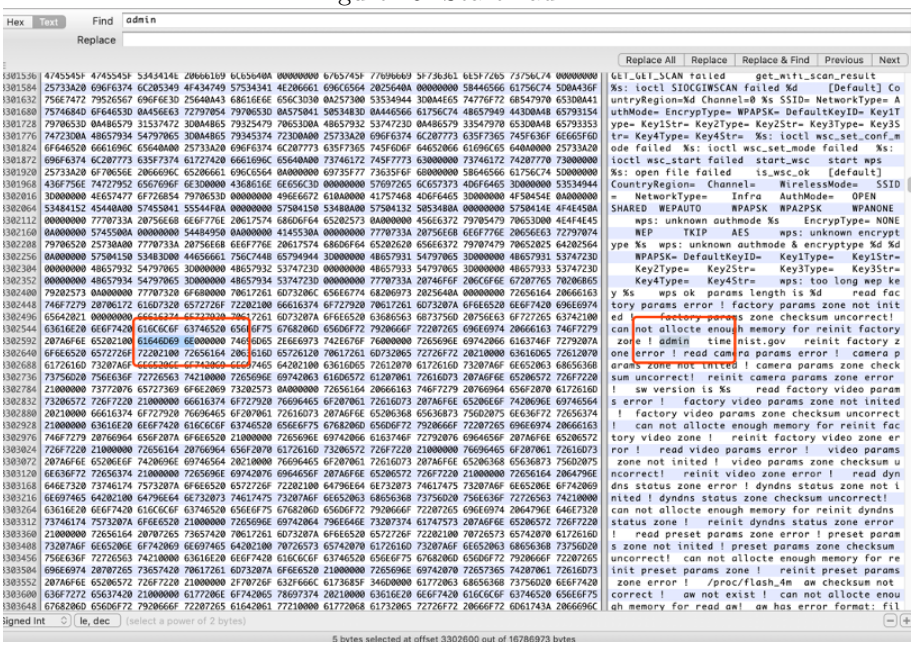


Figure 16: Search admin



2.3 Analysis of Mobile App Security

IP camera has its own ebode camera viewer, so it can be easily view and control the IP camera but unfortunately its a paid app and it is only available in iOS. alternatively, we checked some open source IO camera viewers apps in Android and we tried to connect it with camera, but it did not work, however, we assumed there are some reasons that made the camera or the app unable to connect with each other: the camera firmware or software may out of date and the application may have been fixed by an update. We can assume however that because we have found so many vulnerabilities within other areas of study that the app would be fraught with issues and exploits.

3 Conclusion

We believe that after fully testing the device and evaluating it's security, that we can safely say the device is insecure. As previously stated, the device uses a default blank password, does not correctly implement the functions of the program resulting in an Arbitrary File Read vulnerability and also allows attacks to this device from the local network or even the Internet through various attack methods such as DNS hijacking.

Overall this device is a threat to information security and should be extensively patched to ensure that it is not to the detriment of the overall system or network it is a part of.

Considering how prevalent security cameras are in business situations and how there has been an introduction of GDPR and heavy fines of those who don't keep data secure; having a device that can be used as a pivot to attack data servers or other important systems should be out of the question.

4 Appendix

We find that the fourth byte of `the second` of the file is `50 4b 03 04`, whose corresponding ASCII characters are `PK`.

Figure 17: Find the Bytes

```
0000000 42 4e 45 47 01 00 00 00 01 00 00 00 a0 a8 0b 00
0000010 00 04 10 00 50 4b 03 04 14 00 02 00 08 00 f3 24
0000020 63 41 ab 23 e9 72 08 a8 0b 00 a8 7d 17 00 09 00
0000030 00 00 6c 69 6e 75 78 2e 62 69 6e ec fd 7f 7c 54
0000040 57 99 07 8e 9f 3b 3f 92 21 0c 70 f3 8b 04 48 cb
0000050 05 d2 36 b6 69 7b 81 b4 4d 31 2d c3 8f 56 14 b4
0000060 c3 8f b6 a8 a8 69 4b 2b 2a b5 69 8b 8a bb 68 27
0000070 c9 04 52 36 d0 00 e1 47 69 da 4c 5b d6 c5 2e ee
0000080 a2 56 97 ad e8 4e 0b 55 44 6a 69 4b 15 6b d5 3b
0000090 33 5c 13 32 68 a3 a2 62 45 e6 f3 7e 9f 73 26 33
00000a0 49 a9 d6 dd fd 7c be 7f 7c 37 2f 0e 67 ee b9 e7
00000b0 9e 9f cf 79 ce f3 3c e7 39 cf 23 e2 b1 c4 37 cc
00000c0 58 ea 15 11 4b 89 81 49 89 9a f0 43 c9 c6 29 5b
00000d0 fb 44 28 96 0a 88 9b 13 be 50 5b dd d9 4c 66 46
00000e0 b5 d5 e3 0a 6b ad fb 0d 0b f9 ac 07 5d b1 37 96
00000f0 ba d1 10 fd f1 bf 54 88 b0 a7 52 74 fd 65 9c 58
0000100 80 58 58 15 c2 98 66 4e bb 58 88 b4 10 33 52 4d
0000110 22 e6 9d 24 44 d1 55 f2 f9 9a d4 6d 78 be 00 cf
0000120 96 f3 50 32 38 b0 3e 19 8f 08 b1 04 df b1 1c 96
0000130 c1 b2 58 ce 92 cb 2b 64 59 35 ce 44 77 f4 40 2c
0000140 21 de e6 6f ae 7d 53 f2 f4 a4 2d 7d 22 de 22 db
0000150 e4 17 4b 12 be bd 51 d9 66 d1 b9 d6 15 e6 3a d7
0000160 63 c5 91 de de 37 1e ed 15 22 96 f8 4b 26 d3 6f
0000170 a3 9f 91 82 ad 7d 45 e1 1b 1c 7f d3 3a c7 1f 7b
0000180 d0 69 34 76 f6 8d 68 2c 70 0a c4 55 09 d4 69 54
0000190 c7 d6 27 0b c5 87 12 67 32 99 67 44 2c 96 62 3b
00001a0 8e 16 19 62 c5 08 43 d8 cd 42 f0 77 60 a4 21 aa
00001b0 42 37 25 4f 89 ae 3e 7f 78 ae e3 0f ad 95 e5 89
00001c0 70 97 6b a0 1c 1f fa 5a 1f 8a a2 9c 25 89 3f b1
00001d0 9c 26 55 8e 23 ba 4e 5e f2 f9 58 42 f5 6d 20 c3
00001e0 76 9d 2f 8c c2 fc 88 e3 97 9f f4 ed 5d 90 0c 5a
00001f0 f3 93 1e 3b 96 08 1c 7d 7f d2 27 a6 b8 65 e6 fc
0000200 a4 17 79 06 de e7 c7 d8 86 53 06 e7 70 f7 84 11
0000210 65 d6 04 d7 67 73 be 5a 5d 3c 9f b4 57 57 0a 96
0000220 d3 84 72 96 87 7b 5c 96 25 1a 59 fe 56 f7 ea 1b
0000230 f0 6d 28 96 f0 59 78 36 b7 ba 05 28 af 9c 69 a8
0000240 c7 87 be 08 b1 38 e5 43 9a 1f fd a8 45 99 81 70
0000250 1b ea bc 39 f1 06 c7 17 f5 35 a1 fc 31 0e f2 0e
0000260 6c 75 75 1b 4a 4f 67 32 45 4c 6f 2a 37 44 e5 58
0000270 43 d6 3d f0 b3 cb 4f 06 66 f5 b8 c6 8c 58 ca 67
0000280 a0 6d 76 54 b6 e3 d3 f3 45 fa 4c 23 da 14 46 fd
0000290 f6 56 b7 d5 66 bb 16 a7 84 dd e6 16 a1 4e 11 da
00002a0 e6 1e 27 ec d9 9b 5d 2f c6 10 f3 5a c4 76 33 fd
00002b0 70 88 ed dc e6 fa c4 62 b6 a7 68 b7 ce 37 e5 b6
00002c0 bd 89 27 59 8e dd e2 16 46 62 a9 b1 28 a7 ab a9
00002d0 c7 f5 c4 98 b6 d3 2d b0 5b 11 af 47 19 db 91 7f
00002e0 fb 60 b9 2b 30 36 aa ec ed ee d7 65 d9 db 65 d9
00002f0 68 57 b9 11 99 97 2c b0 0f 14 a3 5d e8 d5 6a f4
:
```

References

[1] *Cross-site request forgery.* URL: `https://en.wikipedia.org/wiki/Cross-site_request_forgery`. (accessed: 12.03.2020).

[2] *IP Camera CGI V1.27.* URL: `https://www.foscam.es/descarga/ipcam_cgi_sdk.pdf`. (accessed: 24.02.2020).