

Penetration Testing Foscam IP Cameras

University of Missouri

Daniel Dunn

Introduction

The emergence of the Internet of Things and so called IoT devices has spawned panic in the cybersecurity community. Experts define the Internet of Things as a network of physical devices, vehicles, home appliances, and other devices able to connect to the internet. A substantial number of these internet connected smart devices contain major design flaws resulting in serious security breaches ranging from privacy invasion to massive-scale botnets. For example, the Mirai botnet, which targeted insecure IoT devices, successfully attacked and infected over 600,000 devices for use in a Distributed Denial of Service (DDoS) attack.[1] Furthermore, experts predict that up to 30 million IoT devices will be connected to the internet by 2020, resulting in a global market for IoT reaching \$7.1 billion.[2,3] As staggering as these numbers are, they represent the target being placed on the Internet of Things and the challenge cybersecurity researchers and professionals face to keep the internet secure.

IP cameras are a category of IoT devices that, if containing security vulnerabilities, could host significant security and privacy implications. One such brand of these cameras is Foscam, a Chinese-based video product manufacturer. The objective of this research project was to penetration test a range of Foscam IP cameras to discover any vulnerabilities and potential exploits. Industry professionals actively conduct research in

IP camera security; in particular, F-Secure released a report detailing their own research that resulted in the discovery of several security vulnerabilities in the Foscam C2 and Opticam i5.[4] Building on the findings outlined in the F-Secure report, this research, discussed in the remainder of this report, succeeded in discovering numerous vulnerabilities and attack vectors in the Foscam C2, R2, FI9803P, and FI9831P.

Selected Devices

Foscam, an IP camera manufacturer based out of Shenzhen, China, boasts of “distribution channels in more than 30 countries and regions, including Germany, the United States, Britain, Italy, Singapore, India, France and Canada.” Their company website also states that “100 million Foscam products have been sold to over 60 countries.”[5] As such, a range of Foscam IP cameras were selected in an attempt to experience greater differentiation in software and firmware between devices and explore the similarities and differences between any discovered vulnerabilities. The table below lists the selected devices and their firmware versions:

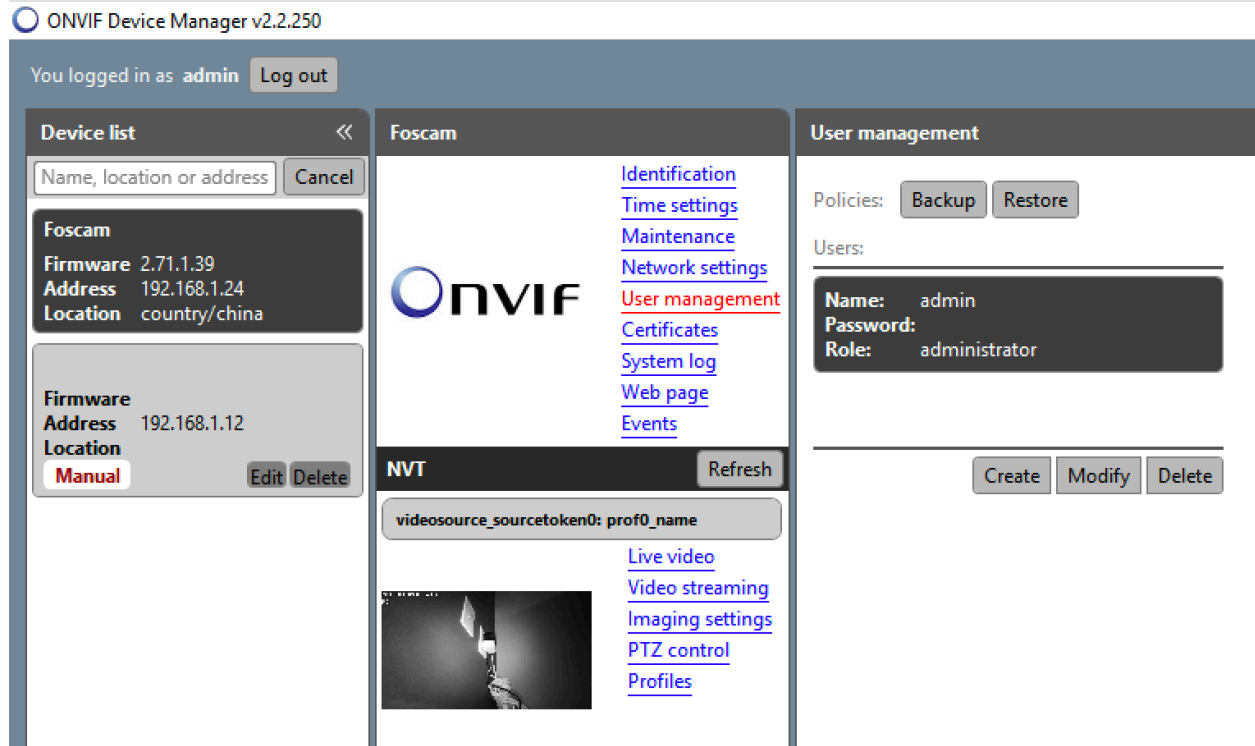
Model Name	System Firmware Version	Application Firmware Version
Foscam C2	1.11.1.8	2.72.1.32
Foscam R2	1.11.1.8	2.71.1.39
Foscam FI9803P	1.9.3.17	2.54.2.37
Foscam FI9831P	1.5.3.19	2.21.2.27

Discovered Vulnerabilities

Seven vulnerabilities and weaknesses, ranging in severity from the release of sensitive information all the way to the possibility of an attacker being able to gain full control of the device, were discovered during this research. Additionally, every camera tested contained at least five out of the seven vulnerabilities. These findings are detailed below.

I. Insecure Factory Default Credentials

Foscam released these cameras from the factory with default credentials of admin:<blank> for the username and password respectively. As the name suggests, this is an administrator account with full access to and control of the camera. Furthermore, the devices do not require a new account to be created to remove the admin account or a password to be set for the account when the user configures the devices. Not until the user attempts to login to the cameras with the web application or mobile application are they prompted to create a new account; this process also disables the admin account as well. However, the devices do not initiate this activity automatically and if the user instead opts to utilize an ONVIF (Open Network Video Interface Forum) application or RTSP-capable (Real Time Streaming Protocol) media player then they may never need to login to the web or mobile application. This thereby leaves the insecure administrator account intact for any malicious actor to exploit.



II. Firewall Limitations

Investigation reveals that the standard firewall, enabled in the web application, implements limited defensive measures. In fact, it acts only as an IP filter for the web application on ports 88 and 443. By extension, it does not perform any other defensive actions, nor does it block IP addresses from accessing any other ports, such as ONVIF (888), FTP (50021), or RTSP (65534). This means that even if a network administrator identifies and attempts to block a potential threat using the device's firewall, an attacker can still communicate with the device through other ports and protocols.

III. Web Application Does Not Require HTTPS

The standard web application provided with these models is accessible through two ports: 88 and 443. While port 443 forces the use of HTTPS and TLS for secure, encrypted communication (albeit with a certificate error), navigating to port 88 does not require HTTPS or TLS. While the application may use some alternate form of encryption or obfuscation for the password, it sends the username in plaintext. Regardless, it

remains unwise to refrain from requiring HTTPS and TLS on all web pages which send sensitive data.

The image shows a Wireshark packet capture interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. The main window is divided into three panes. The top pane shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, and Length. The middle pane shows the details of the selected packet (No. 112), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
111	8.552534705	192.168.1.18	192.168.1.12	TCP	60	49892 → 88 [ACK] Seq=375 Ack=36239 Win=262144 Len=0
112	8.722354749	192.168.1.18	192.168.1.12	TCP	473	49892 → 88 [PSH, ACK] Seq=375 Ack=36239 Win=262144 Len=419
119	8.743148418	192.168.1.18	192.168.1.12	TCP	60	49892 → 88 [ACK] Seq=794 Ack=50839 Win=262144 Len=0
120	8.744043242	192.168.1.18	192.168.1.12	TCP	60	49892 → 88 [ACK] Seq=794 Ack=55885 Win=262144 Len=0
121	8.814123153	192.168.1.18	192.168.1.12	TCP	66	49893 → 88 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=0
123	8.817601022	192.168.1.18	192.168.1.12	TCP	60	49893 → 88 [ACK] Seq=1 Ack=1 Win=524288 Len=0
124	8.819347067	192.168.1.18	192.168.1.12	TCP	133	49893 → 88 [PSH, ACK] Seq=1 Ack=1 Win=524288 Len=79
126	8.940890940	192.168.1.18	192.168.1.12	TCP	66	49894 → 88 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=0
128	8.949163958	192.168.1.18	192.168.1.12	TCP	60	49894 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
129	8.949573147	192.168.1.18	192.168.1.12	TCP	161	49894 → 88 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=107
133	8.992109933	192.168.1.18	192.168.1.12	TCP	60	49894 → 88 [ACK] Seq=108 Ack=239 Win=65280 Len=0
134	8.992272471	192.168.1.18	192.168.1.12	TCP	60	49894 → 88 [FIN, ACK] Seq=108 Ack=239 Win=65280 Len=0

Frame 112: 473 bytes on wire (3784 bits), 473 bytes captured (3784 bits) on interface 0
Ethernet II, Src: PcsCompu_a7:71:f0 (08:00:27:a7:71:f0), Dst: 00:62:6e:6d:9f:70 (00:62:6e:6d:9f:70)
Internet Protocol Version 4, Src: 192.168.1.18, Dst: 192.168.1.12
Transmission Control Protocol, Src Port: 49892, Dst Port: 88, Seq: 375, Ack: 36239, Len: 419

00f0 55 53 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 US..Acce pt-Encod
0100 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 ing: gzi p, defla
0110 74 65 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 te..User -Agent:
0120 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e Mozilla/ 5.0 (Win
0130 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57 4f dows NT 10.0; W
0140 57 36 34 3b 20 54 72 69 64 65 6e 74 2f 37 2e 30 W64; Tri dent/7.0
0150 3b 20 72 76 3a 31 31 2e 30 29 20 6c 69 6b 65 20 ; rv:11. 0) like
0160 47 65 63 6b 6f 0d 0a 48 6f 73 74 3a 20 31 39 32 Gecko..H ost: 192
0170 2e 31 36 38 2e 31 2e 31 32 3a 38 38 0d 0a 43 6f .168.1.1 2:88..Co
0180 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 nnection : Keep-A
0190 6c 69 76 65 0d 0a 43 6f 6f 6b 69 65 3a 20 6c 61 live..Co okie: la
01a0 6e 67 75 61 67 65 3d 45 4e 55 3b 20 75 73 65 72 nguage=E NU; user
01b0 4e 61 6d 65 3d 44 61 6e 69 65 6c 3b 20 72 65 6d Name=Dan iel; rem
01c0 65 6e 62 65 72 3d 3b 20 70 77 64 3d 3b 20 6d 75 enber=; pwd=; mu
01d0 6c 74 69 3d 31 0d 0a 0d 0a lti=1... .

wireshark_eth0_20180108165006_E8v7hE Packets: 212 · Displayed: 79 (37.3%) · Dropped: 0 (0.0%) Profile: Default

IV. Device Does Not Limit Number of Login Attempts

None of the services identified on these devices enforce any form of limit on the number of login attempts, whether from an IP address or on a particular account. This includes the web application, ONVIF, and RTSP on all models, as well as FTP on the C2, R2, and FI9831P models. This behavior allows for an attacker to launch a brute force or dictionary attack against a camera to obtain valid credentials and a foothold on the device.

V. ONVIF Protocol Transmits Unencrypted Credentials

The ONVIF protocol, supported by all the devices selected for this study, does not inherently require credentials to be transmitted in an encrypted manner. As a result, while operating an ONVIF application, ONVIF Device Manager[6], credentials are sent in plaintext, and thus may be intercepted and read by an eavesdropper on the network.

VI. FTP Protocol Transmits Unencrypted Credentials and Data

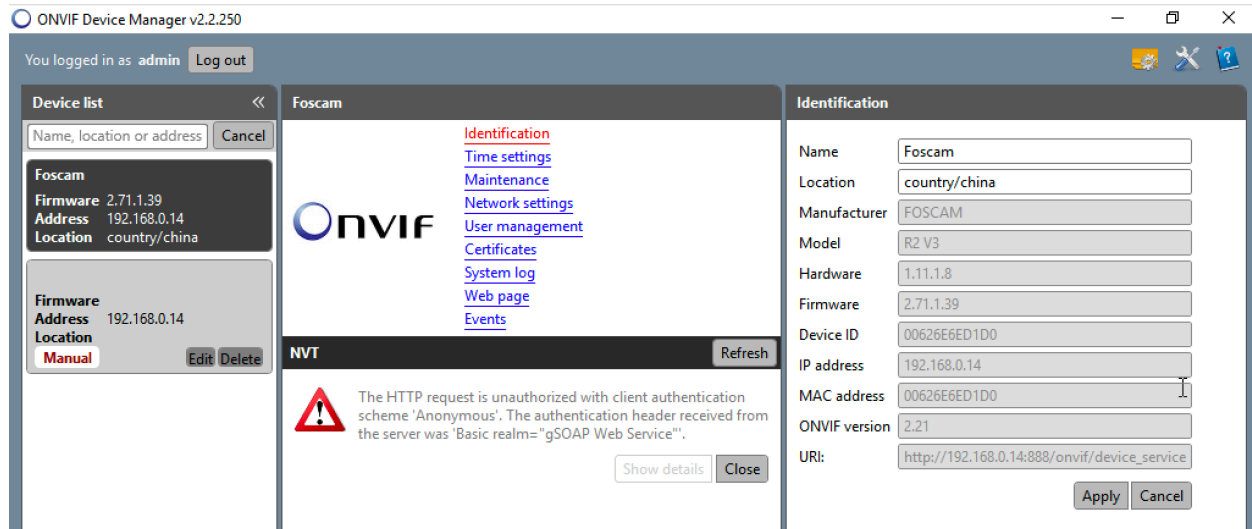
Similar to the ONVIF vulnerability that sends unencrypted credentials over the network, these devices send FTP credentials and data in unencrypted packets as well. This vulnerability exists due to the devices leveraging an FTP service rather than SFTP, which encrypts all transmissions. While the FI9803P does not have functional FTP capabilities and therefore does not possess this weakness, the other models all contain this vulnerability.

Note: This vulnerability was found in the C2, R2, and FI9831P models.

VII. ONVIF Reveals Maintenance and Configuration Information

Some devices leak information through the ONVIF protocol, including potentially sensitive maintenance and configuration data. A user can 'login' to the device using the admin:<blank> credentials discussed in the first vulnerability, even if the account has been removed through the web application. The leaked information includes, but is not limited to, the camera name, location, manufacturer, model, system firmware version, application firmware version, MAC address, ONVIF version, DHCP settings, DNS settings, and more. An attacker could utilize this information leak during their reconnaissance phase to discover a trove of valuable, sensitive information.

Note: This vulnerability was found in the C2, R2, and FI9831P models.



Potential Attack Vectors

This section presents a sample of several potential attack vectors that could be carried out by a malicious actor. The attack vectors are split into two general categories: not configured and configured, indicating whether the admin:<blank> account detailed in part I of the Discovered Vulnerabilities section remains active on the device. The primary tools utilized in these attacks are VirtualBox[7] to host Kali Linux[8] and Windows 10[9], Nmap[10], Wireshark[11], VLC Media Player[12], and ONVIF Device Manager[6].

Not Configured

i. View the live RTSP video feed

These devices allow a user to view the live camera stream by using the RTSP protocol. If an attacker's goal involves gathering physical intelligence about a location or user for some other nefarious purpose, the attacker can easily exploit the admin:<blank> account to view the live video feed using commercial media software capable of viewing a network stream.

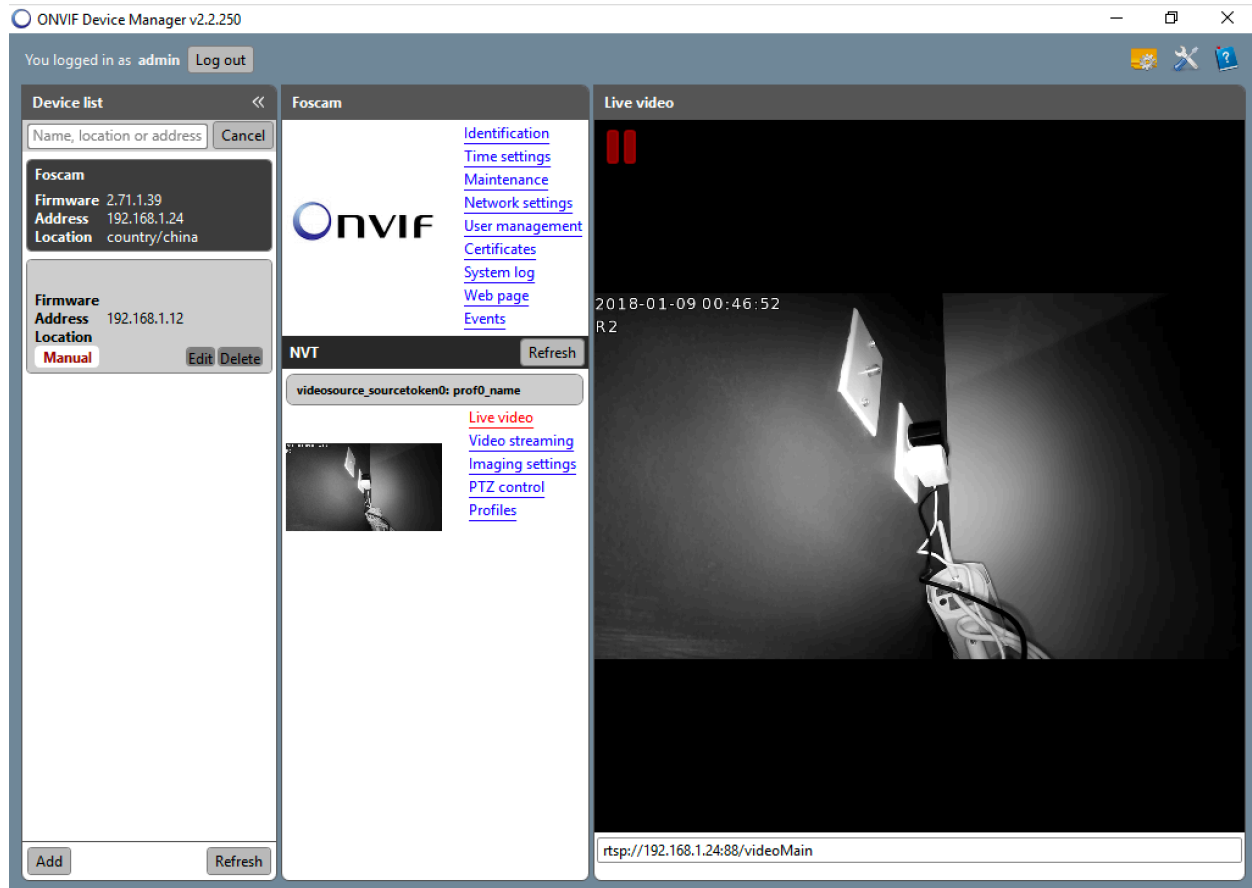
ii. Utilize FTP server

An attacker can use the admin:<blank> account to login to the FTP service on these cameras allowing them to upload and download any files of their choosing.

Furthermore, the F-Secure report[4] states that this behavior allows an attacker to enable a hidden telnet service that would allow them to upload their own payload for covert, persistent access to the device. Furthermore, this allows an attacker not only access to the device itself, but also the ability to pivot to the rest of the network.

iii. Create administrator-level user in ONVIF

An attacker can login to the ONVIF management console, giving them full control of the device, by again using the admin:<blank> credentials that the cameras are shipped with. Account creation is one action, among a multitude of others, that could be taken. An attacker could simply add their own administrator-level account and retain full access to and control of the device, even if the owner of the camera proceeds to configure the device and remove the admin:<blank> account. This control includes the ability to manipulate accounts, modify maintenance information, view the live video feed, operate the PTZ (Pan, Tilt, Zoom) functionality if present, and more.



Configured

i. Execute a Man-in-the-Middle attack against ONVIF

As discussed in part V of the Discovered Vulnerabilities section, the ONVIF protocol does not require credentials to be encrypted before transmission. An attacker can intercept communication between a legitimate user and their device by performing arp-cache poisoning. During the authentication process, the user's ONVIF application may send the credentials in plaintext, allowing an attacker intercepting these packets to analyze them using Wireshark and quickly uncover the user's credentials.

622	8.970810366	192.168.0.24	192.168.0.14	TCP	66	[TCP Dup
623	8.970818480	192.168.0.24	192.168.0.14	TCP	66	[TCP Dup
624	8.970833204	192.168.0.24	192.168.0.14	TCP	176	49799 →
625	8.970835396	192.168.0.24	192.168.0.14	TCP	176	[TCP Ret
630	9.032599770	192.168.0.24	192.168.0.14	TCP	66	49799 →
631	9.032607400	192.168.0.24	192.168.0.14	TCP	66	[TCP Dup
634	9.039621747	192.168.0.24	192.168.0.14	TCP	60	49799 →
635	9.039624764	192.168.0.24	192.168.0.14	TCP	54	[TCP Dup
636	9.039966959	192.168.0.24	192.168.0.14	TCP	66	[TCP Dup
637	9.039974323	192.168.0.24	192.168.0.14	TCP	66	[TCP Dup
640	9.043546223	192.168.0.24	192.168.0.14	TCP	60	49799 →
641	9.043550774	192.168.0.24	192.168.0.14	TCP	54	[TCP Dup

Destination Port: 88
[Stream index: 19]
[TCP Segment Len: 122]
Sequence number: 1 (relative sequence number)
[Next sequence number: 123 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
0101 = Header Length: 20 bytes (5)
► Flags: 0x018 (PSH, ACK)
Window size value: 256
[Calculated window size: 65536]
[Window size scaling factor: 256]
Checksum: 0xf6dc [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
► [SEQ/ACK analysis]
TCP payload (122 bytes)
TCP segment data (122 bytes)

0000	08 00 27 09 6b e4 08 00	27 a7 71 f0 08 00 45 00	..'.k...'.q...E.
0010	00 a2 41 70 40 00 80 06	37 6f c0 a8 00 18 c0 a8	..Ap@... 7o.....
0020	00 0e c2 87 00 58 57 fb	a0 97 0e ae 84 df 50 18XW.P.
0030	01 00 f6 dc 00 00 47 45	54 20 2f 63 67 69 2d 62GE T /cgi-b
0040	69 6e 2f 43 47 49 50 72	6f 78 79 2e 66 63 67 69	in/CGIPr oxy.fcgi
0050	3f 75 73 72 3d 44 61 6e	69 65 6c 26 70 77 64 3d	?usr=Dan iel&pwd=
0060	53 70 61 72 74 61 6e 35	26 63 6d 64 3d 73 6e 61	Spartan5 &cmd=sna
0070	70 50 69 63 74 75 72 65	32 20 48 54 54 50 2f 31	pPicture 2 HTTP/1
0080	2e 31 0d 0a 48 6f 73 74	3a 20 31 39 32 2e 31 36	.1..Host : 192.16
0090	38 2e 30 2e 31 34 3a 38	38 0d 0a 43 6f 6e 6e 65	8.0.14:8 8..Conne
00a0	63 74 69 6f 6e 3a 20 43	6c 6f 73 65 0d 0a 0d 0a	ction: C lose....

ii. Execute a Man-in-the-Middle attack against FTP

Vulnerability VI in the Discovered Vulnerabilities section outlines how the FTP server also sends and receives credentials over the network in plaintext, similar to the ONVIF protocol. Moreover, it sends data and other potentially sensitive information in such a manner as well. This allows an attacker to launch a man-in-the-middle attack, virtually positioning themselves between the camera and an unsuspecting user by employing arp-cache poisoning. Then, the attacker can analyze intercepted packets in Wireshark to discover not only the user's login information, but also the user's commands and the camera's response codes.

```

daniel@kali:~$ ftp 192.168.0.15 50021
Connected to 192.168.0.15.
220-Welcome to Pure-FTPd [privsep] -----
220-You are user number 1 of 50 allowed.
220-Local time is now 00:26. Server port: 50021.
220-This is a private system. No anonymous login
220 You will be disconnected after 15 minutes of inactivity.
Name (192.168.0.15:daniel): user1
331 User user1 OK. Password required
Password:
230 OK. Current directory is /
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l ./.k..b ng6c..E.
200 PORT command successful
150 Connecting to port 41825
drwxr-sr-x 2 1001 ftpuser1 160 Feb 17 21:19 testdir
drwxr-sr-x 2 1001 ftpuser1 160 Feb 17 21:20 testdir2
226 Options: -l
226 2 matches total
ftp> exit
221 Goodbye. You uploaded 0 and downloaded 0 kbytes.
221 Logout.

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src==192.168.0.15 || ip.dst==192.168.0.15

No.	Time	Source	Destination	Protocol	Length	Info
111	14.861688122	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=1 A
152	22.065987911	192.168.0.21	192.168.0.15	TCP	78	35826 → 50021 [PSH, ACK] Se
153	22.068833570	192.168.0.15	192.168.0.21	TCP	66	50021 → 35826 [ACK] Seq=263
154	22.069066821	192.168.0.15	192.168.0.21	TCP	104	50021 → 35826 [PSH, ACK] Se
155	22.069147461	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=13
206	26.326636098	192.168.0.21	192.168.0.15	TCP	82	35826 → 50021 [PSH, ACK] Se
207	26.351562191	192.168.0.15	192.168.0.21	TCP	98	50021 → 35826 [PSH, ACK] Se
208	26.351731478	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=29
209	26.351900068	192.168.0.21	192.168.0.15	TCP	72	35826 → 50021 [PSH, ACK] Se
210	26.357999486	192.168.0.15	192.168.0.21	TCP	85	50021 → 35826 [PSH, ACK] Se
211	26.401378071	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=35

Frame 152: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_09:6b:e4 (08:00:27:09:6b:e4), Dst: 00:62:6e:67:36:63 (00:62:6e:67:36:63)
 ▶ Internet Protocol Version 4, Src: 192.168.0.21, Dst: 192.168.0.15
 ▶ Transmission Control Protocol, Src Port: 35826, Dst Port: 50021, Seq: 1, Ack: 263, Len: 12
 ▶ Data (12 bytes)

```

0000  00 62 6e 67 36 63 08 00 27 09 6b e4 08 00 45 10  .bng6c..'.k...E.
0010  00 40 a7 fd 40 00 40 06 11 36 c0 a8 00 15 c0 a8  .@...@. .6.....
0020  00 0f 8b f2 c3 65 ce b4 90 8e 07 f8 82 db 80 18  .....e..
0030  00 ed 81 a7 00 00 01 01 08 0a 1b 5f d7 5c 06 31  ..... \.1
0040  89 8a 55 53 45 52 20 75 73 65 72 31 0d 0a      ..USER u ser1..

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src==192.168.0.15 || ip.dst==192.168.0.15

No.	Time	Source	Destination	Protocol	Length	Info
111	14.861688122	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=1
152	22.065987911	192.168.0.21	192.168.0.15	TCP	78	35826 → 50021 [PSH, ACK] Seq=1
153	22.068833570	192.168.0.15	192.168.0.21	TCP	66	50021 → 35826 [ACK] Seq=2
154	22.069066821	192.168.0.15	192.168.0.21	TCP	104	50021 → 35826 [PSH, ACK] Seq=1
155	22.069147461	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=1
206	26.326636098	192.168.0.21	192.168.0.15	TCP	82	35826 → 50021 [PSH, ACK] Seq=1
207	26.351562191	192.168.0.15	192.168.0.21	TCP	98	50021 → 35826 [PSH, ACK] Seq=2
208	26.351731478	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=2
209	26.351900068	192.168.0.21	192.168.0.15	TCP	72	35826 → 50021 [PSH, ACK] Seq=1
210	26.357999486	192.168.0.15	192.168.0.21	TCP	85	50021 → 35826 [PSH, ACK] Seq=1
211	26.401378071	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=3
212	26.643648062	192.168.0.21	192.168.0.15	TCP	82	35826 → 50021 [PSH, ACK] Seq=1

Frame 206: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0

Ethernet II, Src: PcsCompu_09:6b:e4 (08:00:27:09:6b:e4), Dst: 00:62:6e:67:36:63 (00:62:6e:67:36:63)

Internet Protocol Version 4, Src: 192.168.0.21, Dst: 192.168.0.15

Transmission Control Protocol, Src Port: 35826, Dst Port: 50021, Seq: 13, Ack: 301, Len: 16

Data (16 bytes)

```

0000  00 62 6e 67 36 63 08 00 27 09 6b e4 08 00 45 10  .bng6c..'.k...E.
0010  00 44 a7 ff 40 00 40 06 11 30 c0 a8 00 15 c0 a8  .D..@.@. .0.....
0020  00 0f 8b f2 c3 65 ce b4 90 9a 07 f8 83 01 80 18  ....e.. ....
0030  00 ed 81 ab 00 00 01 01 08 0a 1b 5f e8 00 06 31  ....._...1
0040  8c 5b 50 41 53 53 20 70 61 73 73 77 6f 72 64 31  .[PASS p assword1
0050  0d 0a                                     ..

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
ip.src==192.168.0.15 ip.dst==192.168.0.15						
No.	Time	Source	Destination	Protocol	Length	Info
210	26.357999486	192.168.0.15	192.168.0.21	TCP	85	50021 → 35826 [PSH, ACK] Seq=333
211	26.401378071	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=35 Ack=38
241	34.642648062	192.168.0.21	192.168.0.15	TCP	92	35826 → 50021 [PSH, ACK] Seq=35
242	34.677910609	192.168.0.15	192.168.0.21	TCP	95	50021 → 35826 [PSH, ACK] Seq=352
243	34.677987984	192.168.0.21	192.168.0.15	TCP	66	35826 → 50021 [ACK] Seq=61 Ack=38
244	34.678069684	192.168.0.21	192.168.0.15	TCP	75	35826 → 50021 [PSH, ACK] Seq=61
245	34.680761900	192.168.0.15	192.168.0.21	TCP	74	20 → 41825 [SYN] Seq=0 Win=14600
246	34.680798045	192.168.0.21	192.168.0.15	TCP	74	41825 → 20 [SYN, ACK] Seq=0 Ack=
247	34.682616846	192.168.0.15	192.168.0.21	TCP	66	20 → 41825 [ACK] Seq=1 Ack=1 Win=
248	34.683183594	192.168.0.15	192.168.0.21	TCP	96	50021 → 35826 [PSH, ACK] Seq=381
249	34.684802995	192.168.0.15	192.168.0.21	FTP-DA...	209	FTP Data: 143 bytes
▶ Frame 249: 209 bytes on wire (1672 bits), 209 bytes captured (1672 bits) on interface 0 ▶ Ethernet II, Src: 00:62:6e:67:36:63 (00:62:6e:67:36:63), Dst: PcsCompu_09:6b:e4 (08:00:27:09:6b:e4) ▶ Internet Protocol Version 4, Src: 192.168.0.15, Dst: 192.168.0.21 ▶ Transmission Control Protocol, Src Port: 20, Dst Port: 41825, Seq: 1, Ack: 1, Len: 143 FTP Data (drwxr-sr-x 2 1001 ftpuser1 160 Feb 17 21:19 testdir\r\ndrwxr-sr-x 2 1001						
0000	08 00 27 09 6b e4 00 62	6e 67 36 63 08 00 45 08	..!.k..b ng6c..E.			
0010	00 c3 5a 00 40 00 40 06	5e b8 c0 a8 00 0f c0 a8	..Z.@.@. ^.....			
0020	00 15 00 14 a3 61 a3 74	c8 b7 7e 98 d9 b8 80 18a.t ..~.....			
0030	03 91 07 20 00 00 01 01	08 0a 06 31 91 48 1b 601.H.			
0040	08 a3 64 72 77 78 72 2d	73 72 2d 78 20 20 20 20	..drwxr- sr-x			
0050	32 20 31 30 30 31 20 20	20 20 20 20 20 66 74 70	2 1001 ftp			
0060	75 73 65 72 31 20 20 20	20 20 20 20 20 20 20 31	user1 1			
0070	36 30 20 46 65 62 20 31	37 20 32 31 3a 31 39 20	60 Feb 1 7 21:19			
0080	74 65 73 74 64 69 72 0d	0a 64 72 77 78 72 2d 73	testdir. .drwxr-s			
0090	72 2d 78 20 20 20 20 32	20 31 30 30 31 20 20 20	r-x 2 1001			
00a0	20 20 20 20 66 74 70 75	73 65 72 31 20 20 20 20	ftpu ser1			
00b0	20 20 20 20 20 31 36	30 20 46 65 62 20 31 37	16 0 Feb 17			
00c0	20 32 31 3a 32 30 20 74	65 73 74 64 69 72 32 0d	21:20 t estdir2.			
00d0	0a					

iii. Exploit ONVIF information leak, launch brute-force attack against FTP

An attacker who has gained access to a network, but is unaware of the devices on the network, could use an ONVIF application to exploit the information leak from part VII of the Discovered Vulnerabilities section to learn the manufacturer, model, and firmware versions of the camera. In addition to being able to perform both of the aforementioned attacks, this information can be used by the attacker to launch a brute-force attack against the FTP server. The lack of restrictions on login attempts described in vulnerability IV allows this attack. After discerning a valid set of credentials, the attacker can gain a foothold on the device.

Conclusion

This research project successfully uncovered a myriad of vulnerabilities in a variety of Foscam IP cameras, ranging from inadvertent leaking of sensitive device information to severe bugs that can allow an attacker to acquire full control of a device. The devices that were tested share the majority of these vulnerabilities, since they exist in the generic software and applications loaded onto the devices. The privacy and security implications of these findings should concern not only cybersecurity researchers and professionals, but consumers as well. By exploiting these vulnerabilities, attackers can view the live video feed of cameras owned by individuals and organizations, can infect these devices with malware to add them to a botnet, or can compromise them to pivot to other devices on the network. This research serves as a warning to tread carefully in the Internet of Things during its infancy and be aware of potential security issues, especially in devices of a sensitive nature.

*Dr. William Harrison, whose advice and support made this research possible, deserves special recognition for his contributions to this project.

References

1. "Inside the Infamous Mirai IoT Botnet: A Retrospective Analysis." CloudFlare, 14 Dec. 2017, blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/.
2. Nordrum, Amy. "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated." IEEE Spectrum: Technology, Engineering, and Science News, IEEE, 18 Aug. 2016, spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated.
3. Hsu, Chin-Lung, and Judy Chuan-Chuan Lin. "An Empirical Examination of Consumer Adoption of Internet of Things Services: Network Externalities and Concern for Information Privacy Perspectives." *Computers in Human Behavior*, vol. 62, 2016, pp. 516–527., doi:10.1016/j.chb.2016.04.023.

4. "Vulnerabilities in Foscam IP Cameras." F-Secure. http://images.news.f-secure.com/Web/FSecure/%7B43df9e0d-20a8-404a-86d0-70dcca00b6e5%7D_vulnerabilities-in-foscam-IP-cameras_report.pdf
5. "Company Profile." Foscam About Us System, Foscam, www.foscam.com/company/about-us.html.
6. Akolomentsev, et al. "ONVIF Device Manager." SourceForge, 2.2.250, 15 Nov. 2016, sourceforge.net/projects/onvifdm/.
7. "VirtualBox." Oracle VM VirtualBox, 5.1.30, Oracle, 16 Oct. 2017, VirtualBox.org.
8. "Kali Linux." Kali Linux, 4.13.0, Offensive Security, kali.org.
9. "Windows 10 Home." Windows 10 - Microsoft Store, 1607, Microsoft, www.microsoft.com/en/us/store/b/windows.
10. Lyon, Gordon "Fyodor". "Nmap Security Scanner." Nmap.org, 7.60, Insecure.com LLC, 1 Aug. 2017, Nmap.org.
11. Combs, Gerald, et al. "Wireshark." Wireshark, 2.4.2, Riverbed, 10 Oct. 2017, Wireshark.org.
12. "VLC Media Player." VideoLAN, 2.2.6 Umbrella, VideoLAN, videolan.org/vlc/index.html.