RAMPARTS

Exploiting Foscam IP Cameras

contact@rampartssecurity.com

Contents

1. Introduction 2

2. Finding the Cameras 3

2.1 Scanning the Address Space 3

2.1.1 Results from Live Scan 3

2.2 The Foscam Fingerprint 4

3. Attack Vectors 5

3.1 Prior Vulnerabilities 5

3.2 Password Attacks 5

3.3 CVE-2014-1911 5

3.4 CVE-2014-1849 (DDNS Poisoning) 5

4. Recommendations 9

4.1 End User 9

4.2 Vendor 9

5. Timeline 11

1. Introduction

Over the last several years IP security cameras have gained wide acceptance for business and home use. One of the leading providers of cameras in this space is Shenzhen Foscam Intelligent Technology Corporation, or simply Foscam. Foscam's product line includes indoor and outdoor pan/tilt cameras.

Prior work has been done by Shekyan and Harutyunyan of Qualys[1]. This research expands upon their work and reports two new attack methods.

[1] http://www.slideshare.net/SergeyShekyan/d2-t1-sergey-shekyan-and-artem-harutyunyan-turning-your-surveillance-camera-against-you

2. Finding the Cameras

2.1 Scanning the Address Space

Normally network scanning for a particular device, operating system, etc. would involve scanning a large portion of IP space and hoping an endpoint device replies with a response that distinguishes the device as a unique type. Scanning tools exist, such as nmap, which attempt to make a determination of a device using a built-in database of fingerprints.

Foscam provides their customers a value-added service with a dynamic domain name system (DDNS) for its cameras. The names follow a rigid naming scheme of the form "XXYYYY.myfoscam.org", where XX are alphabetic characters and YYYY is a 4 digit number ranging from 0000 to 9999. For example, the first camera off the assembly line presumably came with a domain name of AA0000.myfoscam.org. With this scheme it is conceivable for Foscam to have 6,760,000 cameras (26 * 26 * 10,000).

2.1.1 Results from Live Scan

Given the above, it is possible to walk the entire DNS range to learn the IP addresses of all cameras which have communicated with the Foscam DDNS servers. Over the course of several weeks we walked the DDNS space with our custom software and came up with some interesting results.

Summary	Count
Total DDNS Space	6,760,000
DNS Entries with a Valid IP	404,686
IPs Reachable on Port 80	41,893
Cameras Fingerprinted	15,209
Cameras Running 11.x.y.z Firmware[2]	14,826
Fingerprinted Cameras Running Latest 11.x.y.z Firmware	**0**
Cameras Vulnerable to DNS Poisoning	**> 15,209**
Cameras Vulnerable to CVE-2014-1911	**unknown**

We found exactly zero cameras in the wild which run the latest firmware offered by Foscam. This could indicate end users who know to patch also know better than to hook up an IP camera to the Internet, or it could indicate that no one patches their cameras.

[2] Either 11.25.y.z or 11.35.y.z (fixed camera), or 11.22.y.z or 11.37.y.z (pan/tilt camera) according to http://foscam.us/downloads/

2.2 The Foscam Fingerprint

At this point an attacker has a starting point of possible accessible cameras. Now the attacker will want to fingerprint the cameras to determine which are exploitable.

As shown by Shekyan and Harutyunyan[3], the Foscam cameras have an unauthenticated page, get_status.cgi, available. This page provides a wealth of information to an attacker. Of particular concern are id, sys_ver, app_ver, and alias. ID is the MAC address of the wired network adapter. Sys_ver is the firmware version. App_ver is the web user interface version. Alias is a name (e.g. office, attic, etc.) given to the camera by the end user. As of firmware version 11.37.2.54 and web UI version 2.0.10.7, it appears Foscam has patched this hole. Unauthenticated requests to this page are met by a "401 Unauthorized" response.

It is still possible, however, to fingerprint using one of two methods:

1. Make an unauthenticated request to the vars.htm page. Unfortunately, this page returns only the id and alias. Given that an attacker can't access get_status.cgi, but can access vars.htm, the attacker can assume the camera is running a newer firmware. Additionally, as of firmware 11.37.2.54, the web server returned in the header is "Boa/0.94.13" instead of "Netwave IP Camera".

2. Make an "authenticated" request to the get_status.cgi page using an empty username in an HTTP digest access authentication. This may bypass the "401 Unauthorized" message and present an attacker with the same information as before. Please see section 3.3 for more details.

[3] https://github.com/artemharutyunyan/getmecamtool/blob/master/misc/scanner.go

3. Attack Vectors

3.1 Prior Vulnerabilities

Now that an attacker has done the appropriate reconnaissance, it is time for him to attack. Previous attacks have been documented in CVE-2012-3002, CVE-2013-2560, and CVE-2013-5215.

3.2 Password Attacks

The Foscam cameras are subject to password attacks. We have made the following observations:

1. The number of login attempts before locking out an IP address is 10. It appears the lockout remains in effect until reboot.
2. The cameras ship with a default login of username admin and an empty password. Shekyan and Harutyunyan previously showed 2 out of 10 cameras have default credentials in the wild. We did not replicate this research as we have no reason to doubt the trend has changed.

3.3 CVE-2014-1911

We discovered in firmware 11.37.2.54, an attacker can view a video stream remotely unauthenticated via videostream.cgi or snapshot.cgi. This is done by directly requesting the pages and passing an empty username and password in the http digest authentication handshake. The caveat is the camera must currently have fewer users than the maximum it has ever had. In other words a user must have been provisioned and then removed. For example, if the maximum configured users in the past was 3, and the camera currently has 2 users configured, then the empty credential login attack will work.

An end user can mitigate this attack by filling out all 8 user/password slots on the camera.

This vulnerability was first discovered by Ramparts and reported through US-CERT[4].

3.4 CVE-2014-1849 (DDNS Poisoning)

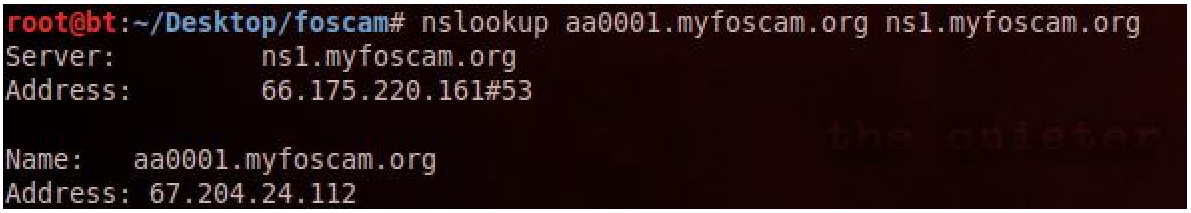
If an attacker is unable to guess the password and has no remote exploit, then he can use the method we describe here[5] to acquire credentials. This attack requires patience on the part of the attacker as he must wait for the end user to login to their camera remotely.

[4] http://www.kb.cert.org/vuls/id/525132

[5] This was independently and concurrently discovered by Shekyan and Harutyunyan: http://blog.shekyan.com/2014/05/cve-2014-1849-foscam-dynamic-dns-predictable-credentials-vulnerability.html

As described above Foscam runs a DDNS for their customers. An attacker can take the following steps to acquire end user credentials:

1. Select a target DNS entry from the previous DDNS scan that is reachable and can be fingerprinted.

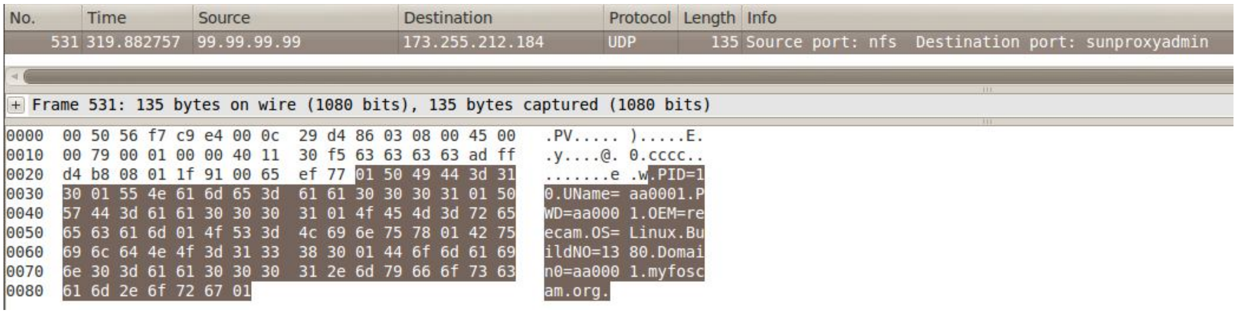


```
root@bt:~/Desktop/foscam# nslookup aa0001.myfoscam.org ns1.myfoscam.org
Server:         ns1.myfoscam.org
Address:        66.175.220.161#53

Name:   aa0001.myfoscam.org
Address: 67.204.24.112
```

2. Save off the associated DDNS IP for possible restoration later.

3. Update Foscam DDNS with a spoofed DDNS update from an IP that the attacker controls. In the below screenshot we chose 99.99.99.99, but our ISP altered it in transit.

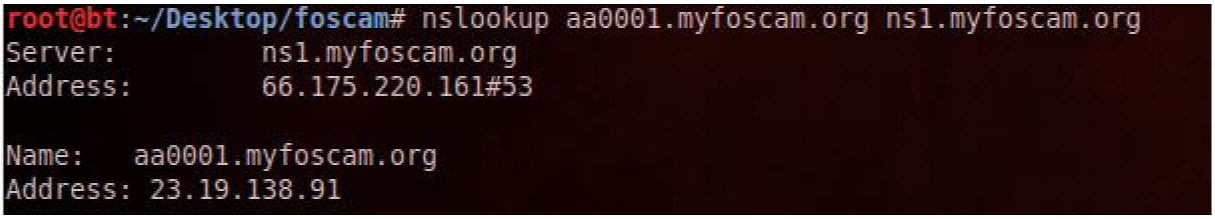


```
No.     Time         Source        Destination       Protocol Length Info
    531 319.882757   99.99.99.99   173.255.212.184   UDP         135 Source port: nfs  Destination port: sunproxyadmin

+ Frame 531: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits)

0000  00 50 56 f7 c9 e4 00 0c  29 d4 86 03 08 00 45 00   .PV..... ).....E.
0010  00 79 00 01 00 00 40 11  30 f5 63 63 63 63 ad ff   .y....@. 0.cccc..
0020  d4 b8 08 01 1f 91 00 65  ef 77 01 50 49 44 3d 31   .......e .w.PID=1
0030  30 01 55 4e 61 6d 65 3d  61 61 30 30 30 31 01 50   0.UName= aa0001.P
0040  57 44 3d 61 61 30 30 30  31 01 4f 45 4d 3d 72 65   WD=aa000 1.OEM=re
0050  65 63 61 6d 01 4f 53 3d  4c 69 6e 75 78 01 42 75   ecam.OS= Linux.Bu
0060  69 6c 64 4e 4f 3d 31 33  38 30 01 44 6f 6d 61 69   ildNO=13 80.Domai
0070  6e 30 3d 61 61 30 30 30  31 2e 6d 79 66 6f 73 63   n0=aa000 1.myfosc
0080  61 6d 2e 6f 72 67 01                               am.org.
```

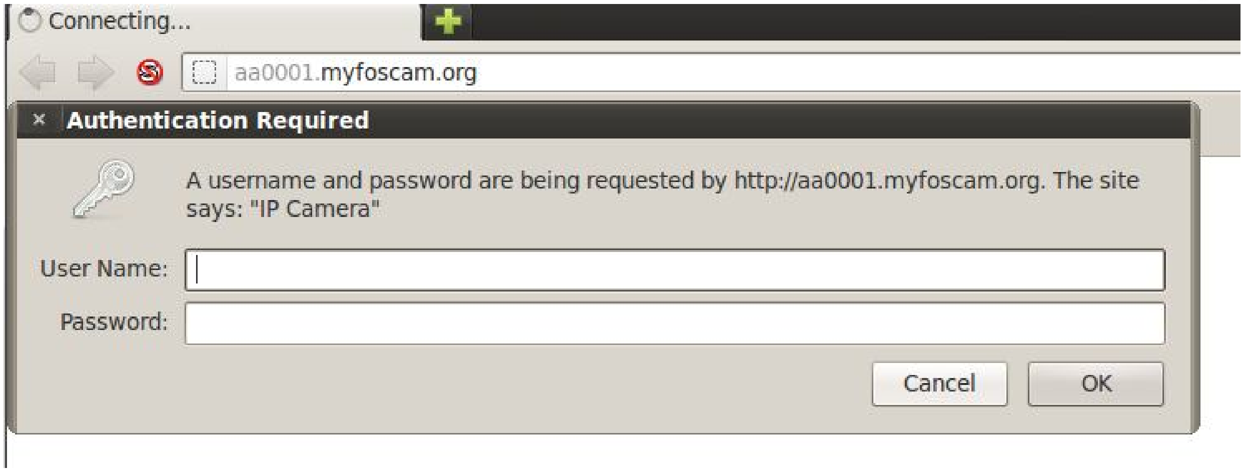
4. Wait for the unwitting user to connect to their camera using their assigned DDNS name. The user is returned a DNS record pointing to the attacker's IP.



```
root@bt:~/Desktop/foscam# nslookup aa0001.myfoscam.org ns1.myfoscam.org
Server:         ns1.myfoscam.org
Address:        66.175.220.161#53

Name:   aa0001.myfoscam.org
Address: 23.19.138.91
```

5. The attacker presents the user with a login page.



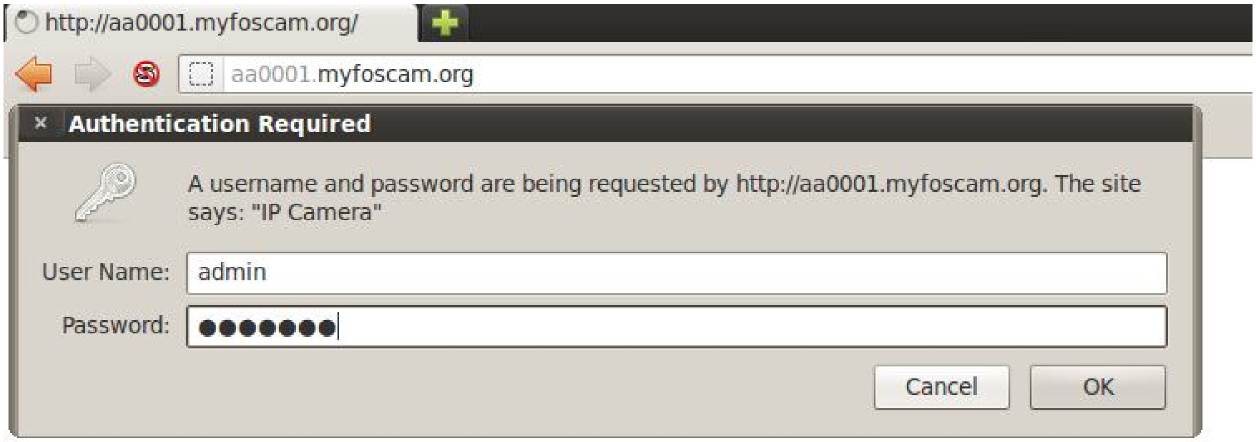
Authentication Required

A username and password are being requested by http://aa0001.myfoscam.org. The site says: "IP Camera"

User Name:

Password:

6. The user submits their credentials.



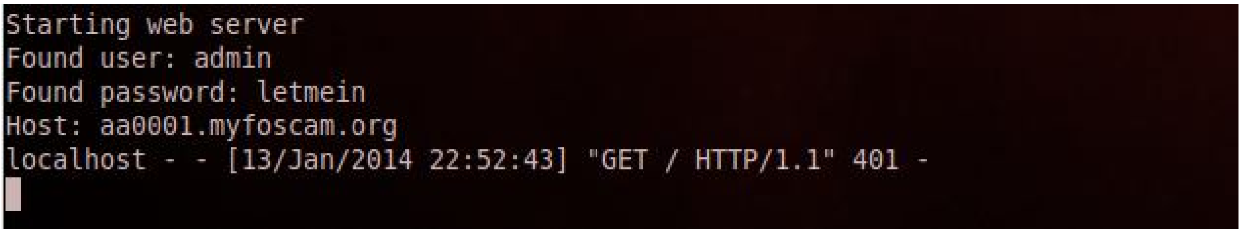
Authentication Required

A username and password are being requested by http://aa0001.myfoscam.org. The site says: "IP Camera"

User Name: admin

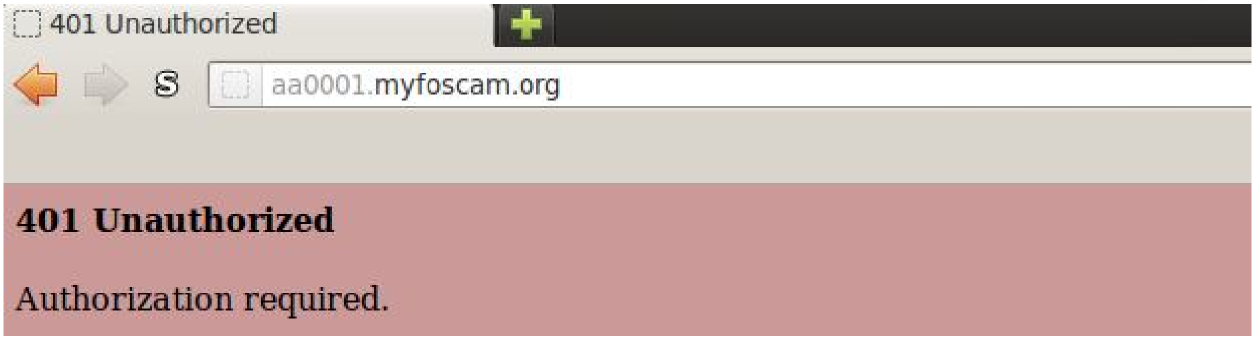
Password: ●●●●●●●

7. The attacker extracts the HOST header field indicating which DDNS entry corresponds to the credentials.



```
Starting web server
Found user: admin
Found password: letmein
Host: aa0001.myfoscam.org
localhost - - [13/Jan/2014 22:52:43] "GET / HTTP/1.1" 401 -
```

8. The attacker sends a 30x redirection or a 401 unauthorized. The user is led to believe they mistyped something.



401 Unauthorized

Authorization required.

9. The attacker possibly restores the DDNS entry to its prior value.

For all of these queries we used ns1.myfoscam.org (66.175.220.161) as the DNS server. This technique did not appear to work (DNS flags 0x8503) on records which were returned as “No such name”. This technique did work (DNS flags 0x8500) in records returned as “127.0.0.1” and records returned with regular, publically routable IP addresses.

The IP used to update a DDNS record with a new IP address is 173.255.212.184. As can be seen in step 3 above, both the username and password are set to the name of the DDNS record being updated.

At boot the camera sends a DDNS update to indicate its IP. After that it sends some type of heartbeat (consisting of 11 bytes) every 60 seconds to port 8082 on 173.255.212.184. It is unclear what the purpose of the message is, but it appears to have no impact on the effectiveness of the poisoning attack.

4. Recommendations

4.1 End User

Unfortunately the end user is the one who has to deal with the security holes in these cameras. We recommend the following at a minimum.

1. Do not make Foscam cameras available from the Internet.

2. If you must make a Foscam camera available from the Internet, then only allow access to the camera while connected through a VPN. This is beyond the technical ability of most end users, so see recommendation one.

3. Update your camera's firmware like you would any other computing device even if it is not directly accessible from the Internet.

4.2 Vendor

Foscam needs to seriously reevaluate the security around their products. We recommend the following measures at a minimum.

1. Randomize the DDNS names

By randomizing the DDNS names given to consumers, Foscam limits an attacker from quickly generating a list of target IP addresses.

2. No unauthenticated access

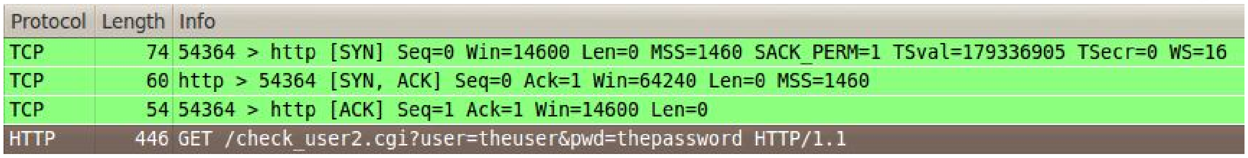
No web pages on the cameras should be accessible without credentials. This prevents an attacker from fingerprinting the model and firmware revision of a camera.

3. No anonymous DDNS updates

The DDNS system should not allow DDNS updates from cameras without a cryptographically sound method in place. This could consist of an endpoint digital signature distributed with the camera. Any DDNS update sent to the Foscam servers would be digitally signed and thus theoretically invulnerable to a poisoning attack.

4. Don't pass credentials via GET parameters

This is poor practice as the URL is liable to be logged in any number of places between the customer and the camera. The credentials should be passed in the body of a POST request (under SSL of course). See the screenshot below for an example.



Protocol	Length	Info
TCP	74	54364 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=179336905 TSecr=0 WS=16
TCP	60	http > 54364 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
TCP	54	54364 > http [ACK] Seq=1 Ack=1 Win=14600 Len=0
HTTP	446	GET /check_user2.cgi?user=theuser&pwd=thepassword HTTP/1.1

5. Timeline

1/6/2014	Vulnerabilities discovered
1/8/2014	Submitted to US-CERT
1/25/2014	Paper published and vulnerabilities publically disclosed via http://krebsonsecurity.com/2014/01/bug-exposes-ip-cameras-baby-monitors/