Hacking the Canon 300D digital camera

Lex Augusteijn

November 2018

www.lex-augusteijn.nl

lex.augusteijn@gmail.com

History

- March 2003: Canon 10D was introduced
- $1500 DSLR
- 6Mpix CMOS sensor

- August 2003: Canon 300D was introduced
- $900
- Same image sensor, same image quality
- Reduced feature set

- I bought mine June 2004

10D vs 300D

ISO 100-3200
Mirror lock-up
Servo AF
Programmable SET button
Flash sync 1/200 in AV
½ or 1/3 stop exposure increments
Flash exposure compensation
Raw + Any Jpeg
Programmable curtain sync

ISO 100-1600
No mirror lock-up
Servo AF only in sports mode
SET button only for menu selection
Shutter speed variable in AV
1/3 stop exposure increments
Fixed flash exposure
Raw + Medium Jpeg
Flash on first curtain

300D Hommage: https://www.youtube.com/watch?v=HVvaUAT4c-8

History

- 2004: Wasia firmware hack

ISO 100-3200

Mirror lock-up with programmable delay

Servo AF only in sports mode

Programmable SET button

Flash sync 1/200 in AV

1/3 stop exposure increments

Flash exposure compensation

Raw + Any Jpeg

Flash on first curtain

History

Wasia disappeared

No material apart from the firmware binary left behind

Alex Bernstein: Firmware decryptor

Michael Tan : Yahoo group about Canon camera hacking

- To distribute firmware
- Many camera models + topics, e.g. Tetris on Powershots

Reconstructing firmware: me and some other dutchman

Steve Yeager: Private forum for developers + 40 beta testers

- Beta leaking elsewhere on the web

https://groups.yahoo.com/neo/groups/canondigicamhacking/info

http://www.digitalrebel.nl/firmware.html

History

- 2004: Wasia firmware hack

ISO 100-3200

Mirror lock-up with programmable delay

Servo AF only in sports mode

Programmable SET button

Flash sync 1/200 in AV

1/3 stop exposure increments

Flash exposure compensation

Raw + Any Jpeg

Flash on first curtain

- 2005: UnDutchables firmware hack

ISO 100-3200

Mirror lock-up with programmable delay

Servo AF in all modes

Programmable SET button

Flash sync 1/200 in AV

1/3 stop exposure increments

Flash exposure compensation

Raw + Any Jpeg

Flash on first curtain

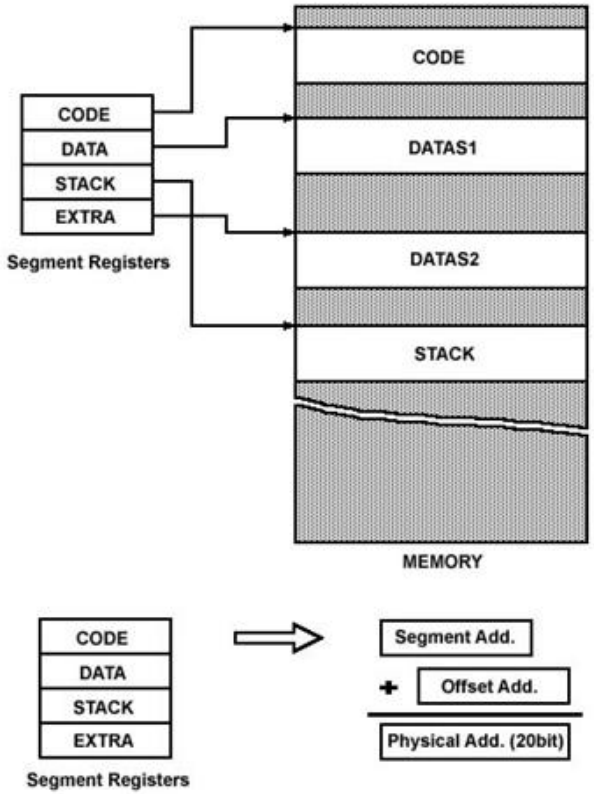
Custom settings in all modes

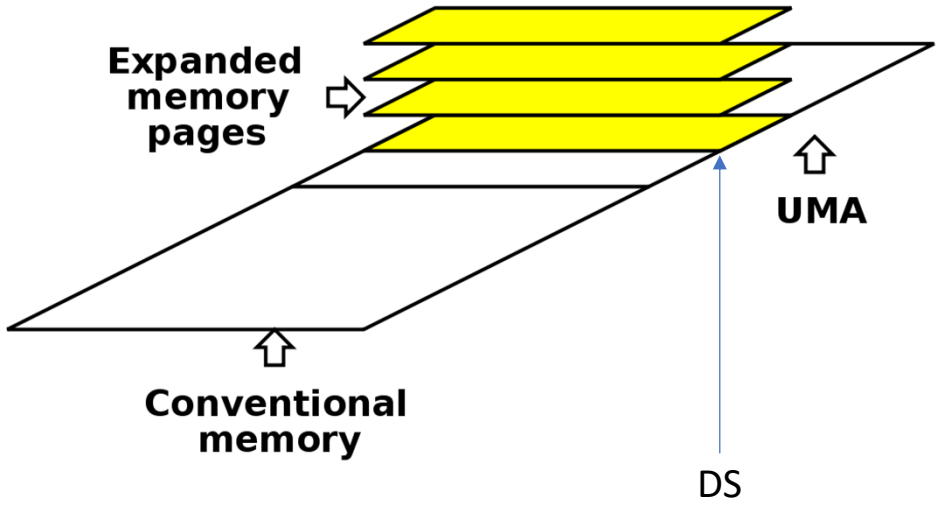
Raw in all modes

Architecture

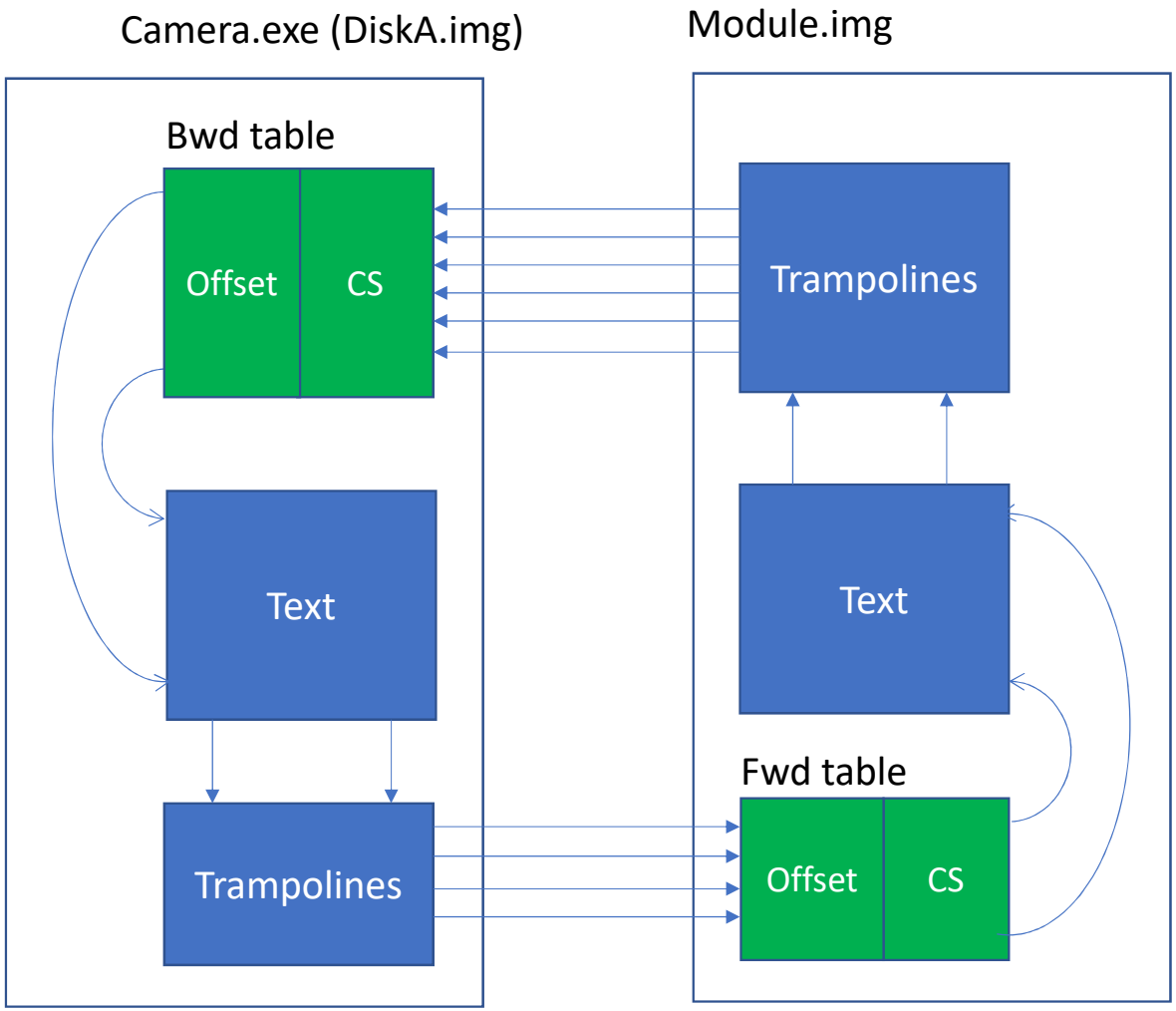
- 80186
  - Running DOS
  - Menu handling and camera settings
  - Segmented memory: 16 bit registers, 20 bit address space
  - Bank switching
  - 20 bit pointers not unique
  - Nightmare for reverse engineering
- Mips
  - Hardware control, e.g. metering, lens control
  - Deeply embedded
  - Memory dump obtained through debug port (Jtag?)
  - Disassembled

Bank switching





Bank switching




```
backward trampoline 18 - print line with parameter:
0xec2e5     far target, 0xec1b:0x135
0xec2e5 TEC-S-          c8 04 00 00         enter   $0x4,$0x0
0xec2e9 T-----          8c 5e fe            mov     %ds,-0x2(%bp)
0xec2ec T-----          c7 46 fc e6 01      movw    $0x1e6,-0x4(%bp)
0xec2f1 T-----          ff 5e fc            lcall   *-0x4(%bp)
0xec2f4 T-----          c9                  leave
0xec2f5 T-----          cb                  lret
…..
0xeced1 T-----          9a 35 01 1b ec      lcall   $0xec1b,$0x135
(0xec2e5) backward trampoline 18 - print line with parameter
```

```
print line with parameter:
0xdc01b     ; WB backward function pointer 18
```

How was it done

- Canon released firmware update 1.1.1 to replace shipped 1.0.2
  - E3kr111.fir file
  - Firmware uploader to copy fir file to CF card in camera through USB
- Decryptor from Alex Bernstein: Cyclic XOR with 512 and 513 tables
- Firmware unpacker
- S10sh: camera control through USB
  - http://s10sh.sourceforge.net/
  - Control camera through USB port
  - Take pictures
  - Download pictures
  - Set camera parameters (my contribution, 2006)
  - Debug log through USB

Debug log to USB

[TX]IN 01 03 10 00

[TX]R1:11

[TX]IN 01 00 91 1C 00 1C 00 4B

[TX]R1:12

[TX]IN 01 20 58 00 1C

[Mc]AvoChanged!

[MC]R3:04

[MC]R3:06

[TX]R1:82

[TX]OUT82 11 20 31 40 50 61 70 80 90 A0 B0 C1 D0 E0 F0 00 10

[TX]R1:81

[TX]OUT81 01 01 01 08 00 50 30 18 00 00

[MC]R2:19

[TX]R1:19

[TX]IN 01 18

[MC]R2:40

[MC]R2:40

[MC]tvchg:75

[AE]Tv:0075

[MC]aemodchg:01

[MC]R2:40

[MC]R2:04

[TX]R1:16

[TX]IN 01 75 FF

[TX]R1:12

[TX]IN 01 20 58 00 1C

[TX]R1:12

[TX]IN 01 20 58 00 1C

[WB] Page 15 / Offset 11180

[CPU] CPU24MHz

[CPU] CPU48MHz

[CPU] CPU24MHz

[CPU] CPU48MHz

[WD]START;#1

[SUP]Time:#1400

[TX]OUT82 11 20 31 40 50 61 70 80 90 A0 B0 C1 D0 E0 F0 00 10

[TX]R1:81

[TX]OUT81 01 01 01 08 00 50 30 18 00 01

[SwDrv] ModeD: 0008

[CamUI] sw int 0105:000C, post.

[CamUI] QREV _Stop

[TX]R1:19

[TX]IN 01 18

[TX]R1:16

[AVS]CalcAvailShot

[TX]IN 01 75 FF

[AE]Tv:0075

[AE]Av:00FF

[AVS]CalcSize, LNSN

[AVS]Shot #262

[CAMUI] AvSht3:262

[CamUI] sw int 0102:0000, post.

[CamUI] sw int 0102:0002, post.

[CamUI] sw int 0102:0004, post.

[CamUI] sw int 0102:0006, post.

[CamUI] sw int 0102:0008, post.

[CamUI] sw int 0103:0002, post.0000

[CamUI] sw int 0104:0000

[CamUI] sw int 0104:0002

[CamUI] sw int 0104:0006

[CamUI] sw int 0104:0008

[CamUI] sw int 0104:000A

[CamUI] sw int 0104:000C

[TX]R1:16

[CamUI] Init End.

[TX]IN 01 75 FF

[CAMUI-R]ID:0214(0000)B:0000GS:0000CS:00//

[TX]IN 01 20 58 00 1C

[CAMUI]//GS:0000CS:0000

[MC]R3:20

[MC]R3:21

[CAMUI-R]ID:0213(0000)B:0000GS:0000CS:00//

[CAMUI]//GS:0000CS:0000

[CamUI] ShtUI GotoCapture !

[MC]R3:20

[TX]R1:16

[AVS]CalcAvailShot

[TX]IN 01 75 FF

[AE]Tv:0075

[AE]Av:00FF

[AVS]CalcSize, LNSN

[AVS]Shot #262

[CAMUI] AvSht4:262

[MC]R3:28

[CAMUI-R]ID:0214(0000)B:0000GS:0000CS:00//

[CAMUI]//GS:0000CS:0000

[MC]R3:20

[MC]R3:21

[CAMUI-R]ID:0213(0000)B:0000GS:0000CS:00//

[CAMUI]//GS:0000CS:0000

[CamUI] ShtUI GotoCapture !

[MC]R3:20

[TX]R1:16

[AVS]CalcAvailShot

[TX]IN 01 75 FF

[AE]Tv:0075

[AE]Av:00FF

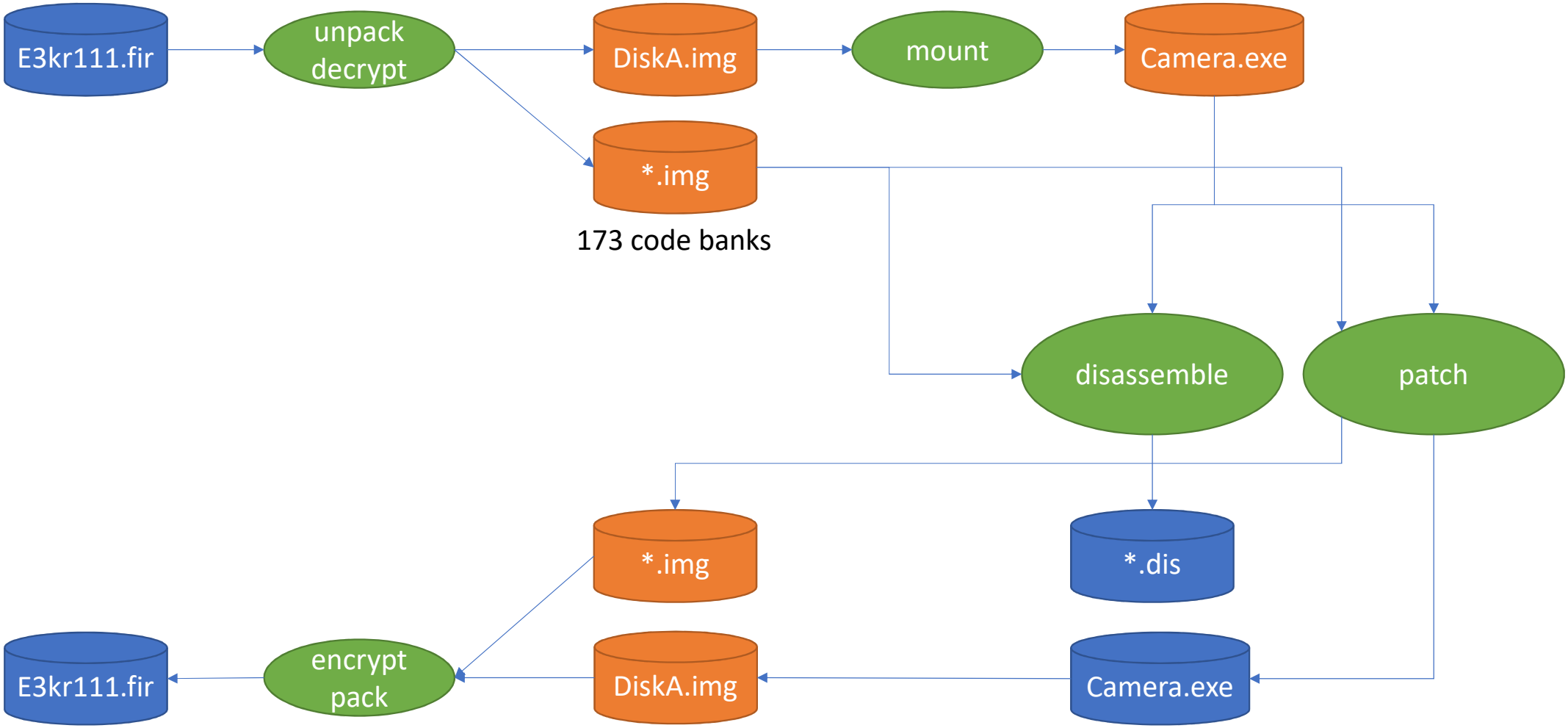
[AVS]CalcSize, LNSN

[AVS]Shot #262

[CAMUI] AvSht4:262

[MC]R3:28

Work flow

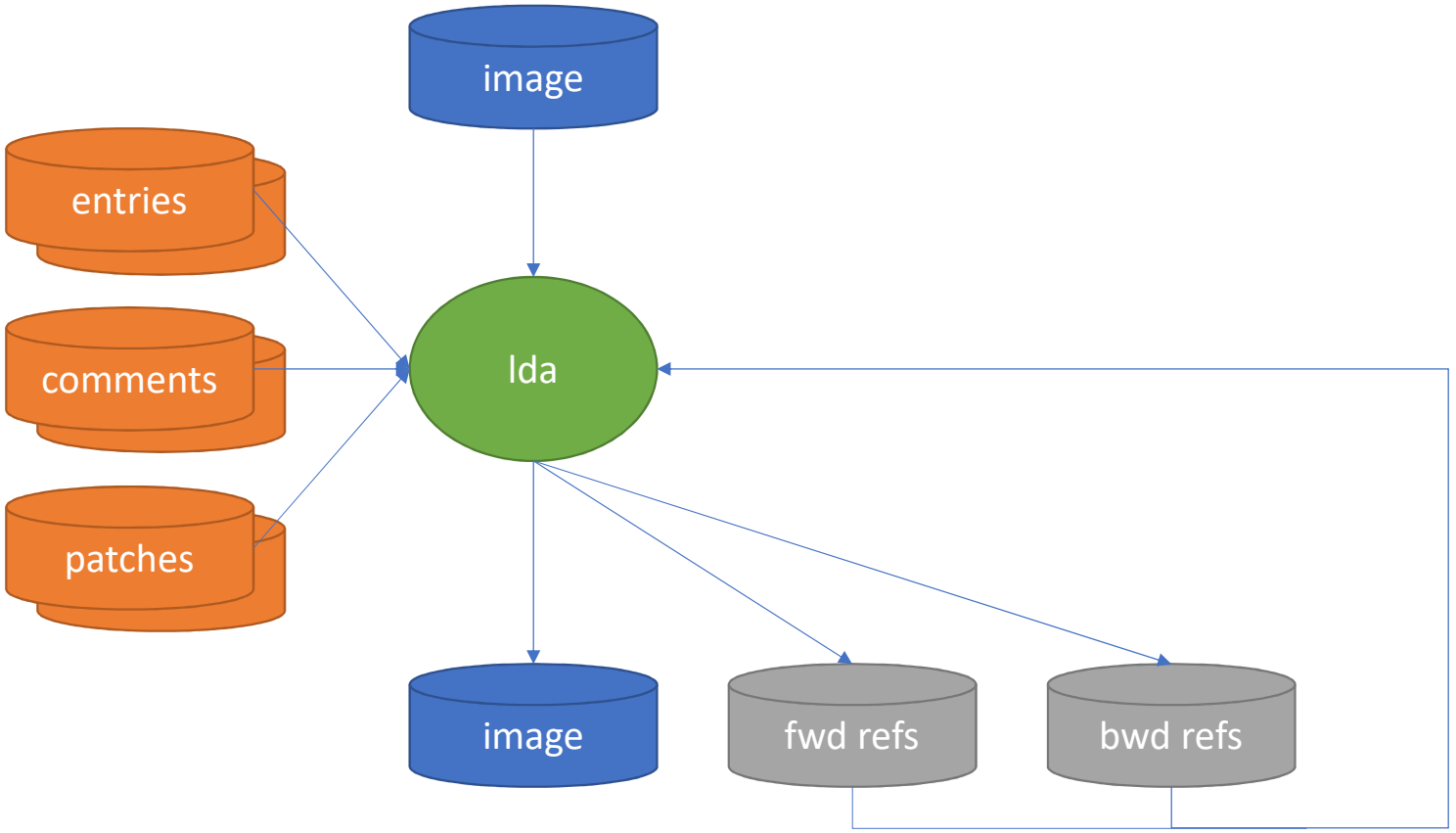

E3kr111.fir
unpack
decrypt
DiskA.img
mount
Camera.exe
*.img
173 code banks
disassemble
patch
*.img
*.dis
E3kr111.fir
encrypt
pack
DiskA.img
Camera.exe

Tooling

- LDA (Lex' DisAssembler)
- Using binutils for disassembling
- Distributed hacking
- Hacker can specify
  - Comments
  - Entry points
  - Patches
- Automatic cross references
- Detect bank-switching, xrefs accross banks through trampolines
- No code movement: just modifying instructions without changing size
- Most 10D code was still there: needed activation

- Needed 2 hours to revive tooling from 13 years ago (old Redhat to latest Ubuntu).

Tooling



Disassembly

Text
Callee
Start

```
0xac3d6 .. 0xac3df "\r\n[AE]Tv:"

0xac3e0     CALLEE
            XREFS 0xac2ec
0xac3e0 T-C-S-          56                  push    %si
0xac3e1 T-----          9a ad 1f 2e b4      lcall   $0xb42e,$0x1fad (0xb628d)
0xac3e6 T-----          8b f0               mov     %ax,%si
0xac3e8 T-----          9a 46 01 84 c0      lcall   $0xc084,$0x146 (0xc0986)
0xac3ed T-----          0a c0               or      %al,%al
0xac3ef T-----          74 2e               je      0x000ac41f
0xac3f1 T-----          9a a5 1e 2e b4      lcall   $0xb42e,$0x1ea5 (0xb6185)
0xac3f6 T-----          3d 02 00            cmp     $0x2,%ax
0xac3f9 T-----          75 24               jne     0x000ac41f
0xac3fb T-----          6a 02               push    $0x2
0xac3fd T-----          9a 82 01 2e b4      lcall   $0xb42e,$0x182 (0xb4462)
0xac402 T-----          59                  pop     %cx
0xac403 T-----          0a c0               or      %al,%al
0xac405 T-----          75 18               jne     0x000ac41f
0xac407 T-----          83 fe 75            cmp     $0x75,%si
0xac40a T-----          74 2b               je      0x000ac437
0xac40c T-----          81 fe ff 00         cmp     $0xff,%si
0xac410 T-----          74 25               je      0x000ac437
0xac412 T-----          9a d8 02 2e b4      lcall   $0xb42e,$0x2d8 (0xb45b8)
0xac417 T-----          c6 06 a5 03 01      movb    $0x1,933
0xac41c T-----          e9 99 00            jmp     0x0000c4b8 (0xac4b8)
0xac41f     TARGET
            XREFS 0xac405 0xac3f9 0xac3ef
0xac41f T--J--          80 3e a5 03 01      cmpb    $0x1,933
0xac424 T-----          75 11               jne     0x000ac437
0xac426 T-----          c6 06 a5 03 00      movb    $0x0,933
0xac42b T-----          6a 00               push    $0x0
0xac42d T-----          6a 00               push    $0x0
0xac42f T-----          9a 7b 02 2e b4      lcall   $0xb42e,$0x27b (0xb455b)
0xac434 T-----          83 c4 04            add     $0x4,%sp
0xac437     TARGET
            XREFS 0xac424 0xac410 0xac40a
0xac437 T--J--          0e                  push    %cs
0xac438 T-----          68 0c 01            push    $0x10c
0xac43b T-----          9a 28 00 47 d1      lcall   $0xd147,$0x28 (0xd1498)
0xac440 T-----          83 c4 04            add     $0x4,%sp
0xac443 T-----          56                  push    %si
0xac444 T-----          9a 62 00 47 d1      lcall   $0xd147,$0x62 (0xd14d2)
0xac449 T-----          59                  pop     %cx
0xac44a T-----          9a a5 1e 2e b4      lcall   $0xb42e,$0x1ea5 (0xb6185)
```

Text
Jump target

Entries

0x8d26e
0x8d2b8
0x8d400
0x8d524
0x8d70c
0x8d908
0x8d984
0x8d9b6
0x940dd
0x907f7
0x90e31
0x9190e
0x979d4
0x97bb9
0x97ff5
0x98238
0x985e5
0x98c44
0x98e56
0x98ff0
0x99123
0x992f7
0x9970c
0x99e
0x9a208
0x9a3b3
0x9a69f
0x9ab82
0x9ae68
0x9aff3
0x9b152
0x9b2f3
0x9b6d4
0x9bc0b
0x9bd4e
0x9bedf
0x9c13f
0x9c344
0x9c569
0x9ca2f
0x9d0d5
0x9d2fc
0x9d500
0x9d88c
0x9dba3
0x9dda1
0x9e060
0x9e4e9
0x9e99c
0x9ebc6
0x9edfc
-0x9d293
-0x93dd5
-0x9d831
-0x9efee
-0x9f523
-0xa5930
-0xa5938
-0xab64e
-0xad8bb

Comments

```
# AV / TV
; 0xac4ba "\r\n[AE]Av:"
: 0xac3d6 "\r\n[AE]Tv:"
; 0xac3ec 2 = AV
; 0xac3f1 2 Flash sync speed in AV mode
; 0xac3fb Flash sync CF-3 = 1= -> TV fixed to 1/200 sec
: 0xb45b8 Copy of Cf memory 29719-29727 to 29674-29682; sets 29676=0x31, CF5=3=Emits/does not fire, CF6=1=1/3
stops increment
; 0xac40d Set memory copied, 29676 changed
; 0xac415 Check if memory was copied and 29676 was changed, 0 = not changed
: 0xb455b Copy of Cf memory 29719-29727 to 29674-29682; no changes to 29676
: 0xac41c Set memory copied, 29676 unchanged
; 0xac498 Check if memory was copied and 29676 was changed, 0 = not changed
```

comment
; address comment at end of line
< address comment before
> address comment after
: address label

Disassembly

```
0xac3d6 .. 0xac3df "\r\n[AE]Tv:"

0xac3e0       CALLEE
              XREFS 0xac2ec
0xac3e0 T-C-S-              56                  push   %si
0xac3e1 T-----              9a ad 1f 2e b4      lcall  $0xb42e,$0x1fad (0xb628d) Get TV value, 0xFF=Auto else if testb $0x1,30992=1 -> ret ([9ca] || 0x100) else ret [9ca]
0xac3e6 T-----              8b f0               mov    %ax,%si
0xac3e8 T-----              9a 46 01 84 c0      lcall  $0xc084,$0x146 (0xc0986) if ([30993] && 0xf0) == 0x30 al=1 else al=0
0xac3ed T-----              0a c0               or     %al,%al
0xac3ef T-----              74 2e               je     0x00000000000ac41f
0xac3f1 T-----              9a a5 1e 2e b4      lcall  $0xb42e,$0x1ea5 (0xb6185) Check Camera Mode (0-5= Advanced mode;  >6 Basic mode) ; 2 Flash sync speed in AV mode
0xac3f6 T-----              3d 02 00            cmp    $0x2,%ax
0xac3f9 T-----              75 24               jne    0x00000000000ac41f
0xac3fb T-----              6a 02               push   $0x2 ; Flash sync CF-3 = 1= -> TV fixed to 1/200 sec
0xac3fd T-----              9a 82 01 2e b4      lcall  $0xb42e,$0x182 (0xb4462) Retrieve value for CF setting : 1 byte as input arg (0-16)
0xac402 T-----              59                  pop    %cx
0xac403 T-----              0a c0               or     %al,%al
0xac405 T-----              75 18               jne    0x00000000000ac41f
0xac407 T-----              83 fe 75            cmp    $0x75,%si
0xac40a T-----              74 2b               je     0x00000000000ac437
0xac40c T-----              81 fe ff 00         cmp    $0xff,%si
0xac410 T-----              74 25               je     0x00000000000ac437
0xac412 T-----              9a d8 02 2e b4      lcall  $0xb42e,$0x2d8 (0xb45b8) Copy of Cf memory 29719-29727 to 29674-29682; sets 29676=0x31, CF5=3=Emits/does not fire,
CF6=1=1/3 stops increment
0xac417 T-----              c6 06 a5 03 01      movb   $0x1,0x3a5
```

Disassembly

```
Set memory copied, 29676 unchanged:
0xac41c
0xac41c     ENTRY
0xac41c TE----          e9 99 00              jmp     0x00000000000ac4b8
0xac41f     TARGET
            XREFS 0xac405 0xac3f9 0xac3ef
0xac41f T--J--          80 3e a5 03 01        cmpb    $0x1,0x3a5
0xac424 T-----          75 11                 jne     0x00000000000ac437
0xac426 T-----          c6 06 a5 03 00        movb    $0x0,0x3a5
0xac42b T-----          6a 00                 push    $0x0
0xac42d T-----          6a 00                 push    $0x0
0xac42f T-----          9a 7b 02 2e b4        lcall   $0xb42e,$0x27b (0xb455b) Copy of Cf memory 29719-29727 to 29674-29682; no changes to 29676
0xac434 T-----          83 c4 04              add     $0x4,%sp
0xac437     TARGET
            XREFS 0xac424 0xac410 0xac40a
0xac437 T--J--          0e                    push    %cs
0xac438 T-----          68 0c 01              push    $0x10c
0xac43b T-----          9a 28 00 47 d1        lcall   $0xd147,$0x28 (0xd1498) _sprintf?
0xac440 T-----          83 c4 04              add     $0x4,%sp
0xac443 T-----          56                    push    %si
0xac444 T-----          9a 62 00 47 d1        lcall   $0xd147,$0x62 (0xd14d2) Print Hex String value, Use byte arg on stack as input
0xac449 T-----          59                    pop     %cx
```

Patches

poke 0x96cc2 0x90 Assign left key to 0xb0050, is AF Enabled/Disabled

poke 0x96cc3 0x3d

poke 0x96cc4 0x75 Assign right key to AF mode

poke 0x96cc5 0x3d

Other work

- CHDK
  - Canon Hack Development Kit
  - Canon point-and-shoots, many models
  - Replaces boot loader , takes over camera

- Magic Lantern
  - Canon DSLR's, many models
  - Replaces boot loader, takes over camera
  - Own GUI next to Canon
  - Calls native Canon API functions
  - Allows scripting in LUA
  - Dozens of still and video features

What happened to Wasia?