grand theft RECON

IoT


HELLO KITTY


Big Chill


PSYCHO
DISCO


the Streets

DAY 1 - PLAN

DAY 2 - REVERSE

DAY 3 - OWN

day one

the plan

LESTER

ASSIGNMENT

1. FIND A TARGET
2. FIND AN ATTACK ANGLE
3. FORM A CHAIN OF ATTACK

Name

LESTER CREST

Expertise

PLANNING

Favorite porn star

STEFAN ESSER

This is where all the loaded people come to have fun

THE HEIST

LESTER

CAFE ANASTASIA

Here's a photo from the inside. They store their cash in that POS

LESTER

POS TERMINAL

Time to refresh my memory on how these things are protected

LESTER

POS TERMINAL

POS: Ingredients

POS: Ingredients

- Terminal

  No direct access to cash

POS: Ingredients

- Card reader
  Heavily
  protected

POS: Ingredients

- Cashier
  Expensive
  to bribe

Is that it? Hit rewind, I'm sure we missed something.

LESTER

What's that steel box over there?

LESTER

POS: Ingredients

- Cash drawer

  Just a dumb box

  …or is it?

A Modern POS

…especially popular in bars and restaurants

APG NetPRO 488

- Most popular wireless model
- Connects over WiFi…
- To the INTERNET OF THINGS

Wait a minute... close up on that part

LESTER

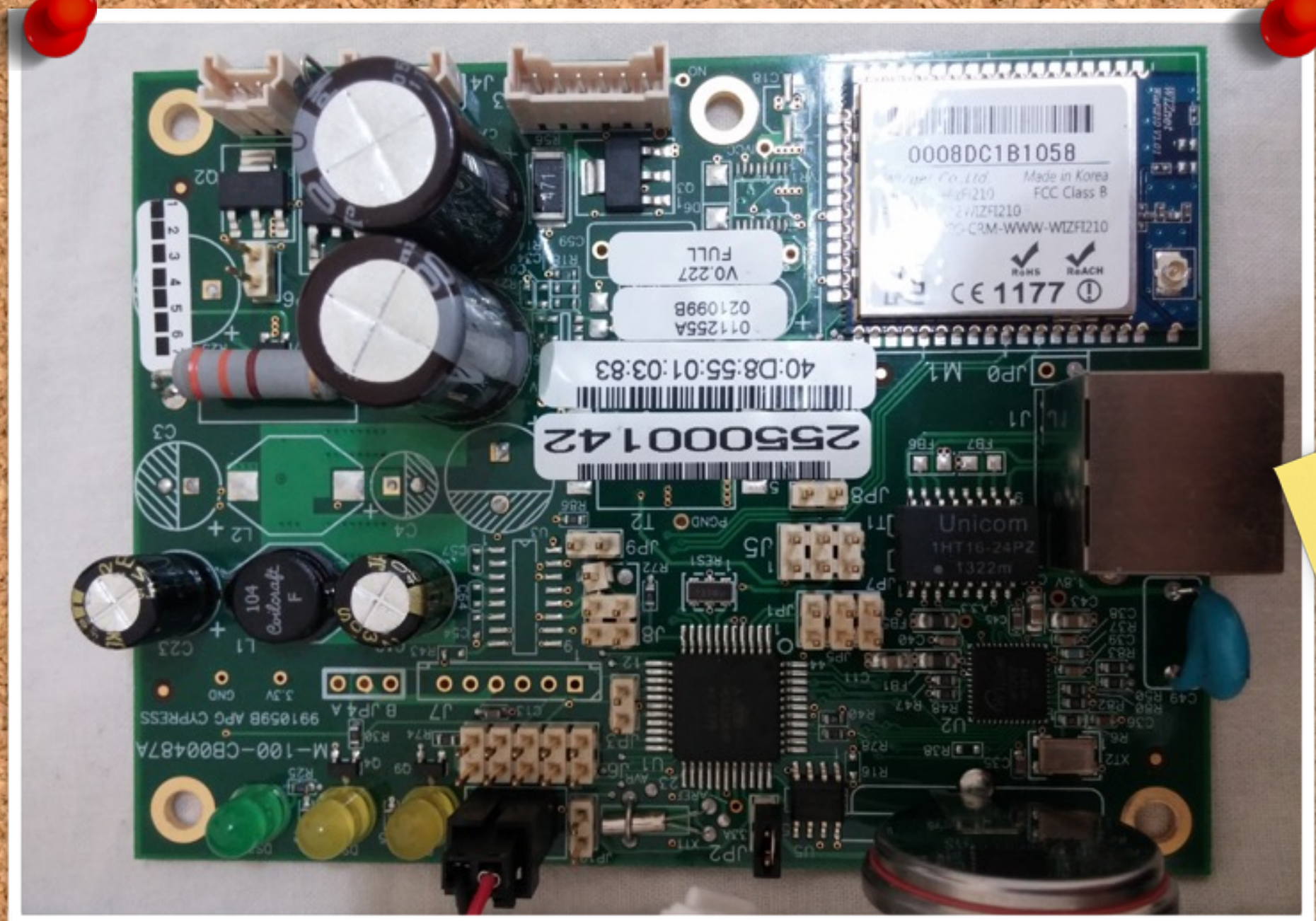
POS TERMINAL

Gentlemen... I believe we have a target

LESTER

POS TERMINAL

Let's get a device and crack it open

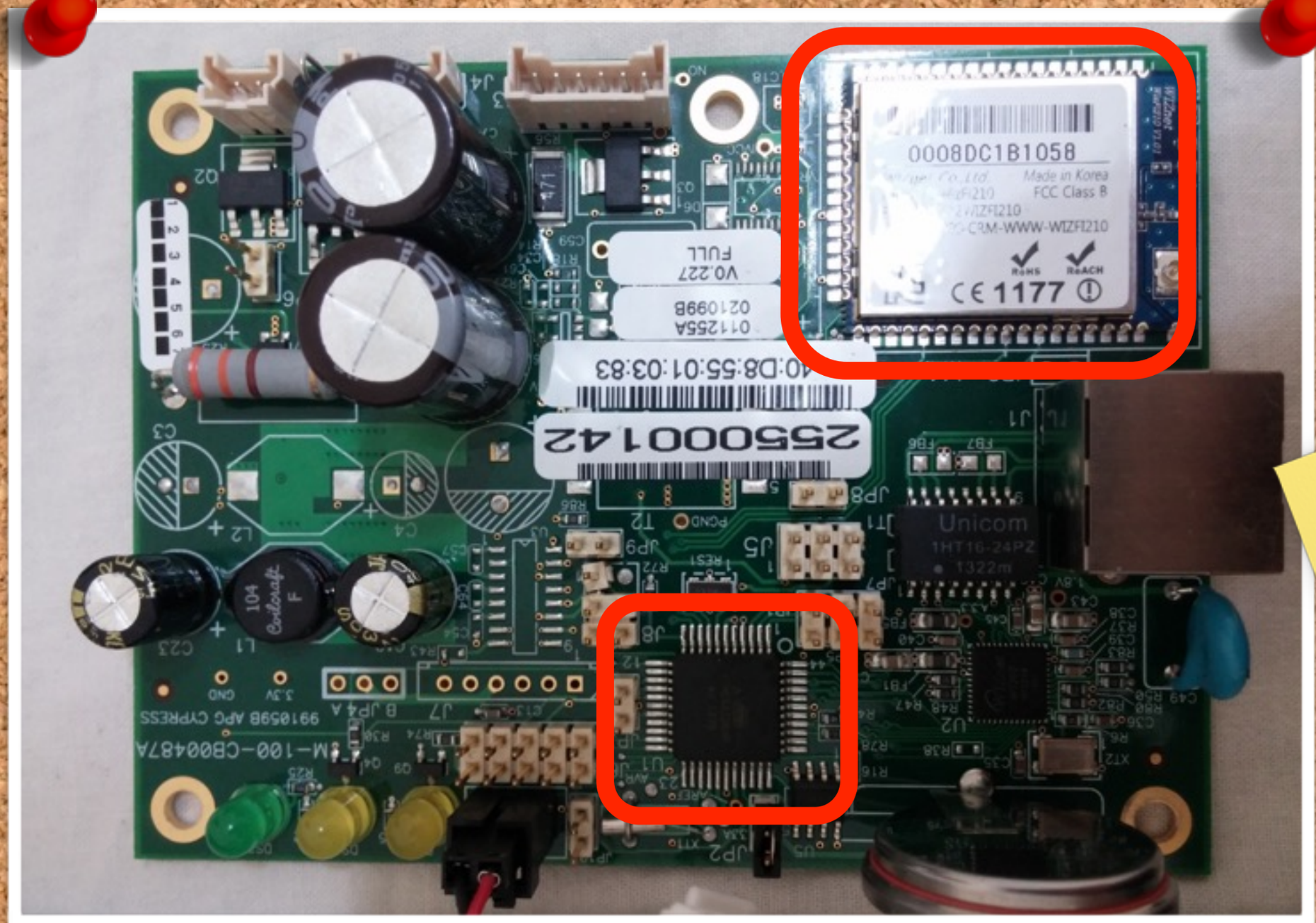


LESTER

APG 488

Give me a close up of those two chips

LESTER



APG 488

128K FLASH
16K RAM

ATMEL
ATMEGA
1284P

AT OVER
SERIAL

WIZNET
WIZFI210

Get the firmware - options

- No firmware online
- Play with UART?
- Extract from MCU?
  (AKA Suicide)
- Ask the manufacturer… nah!

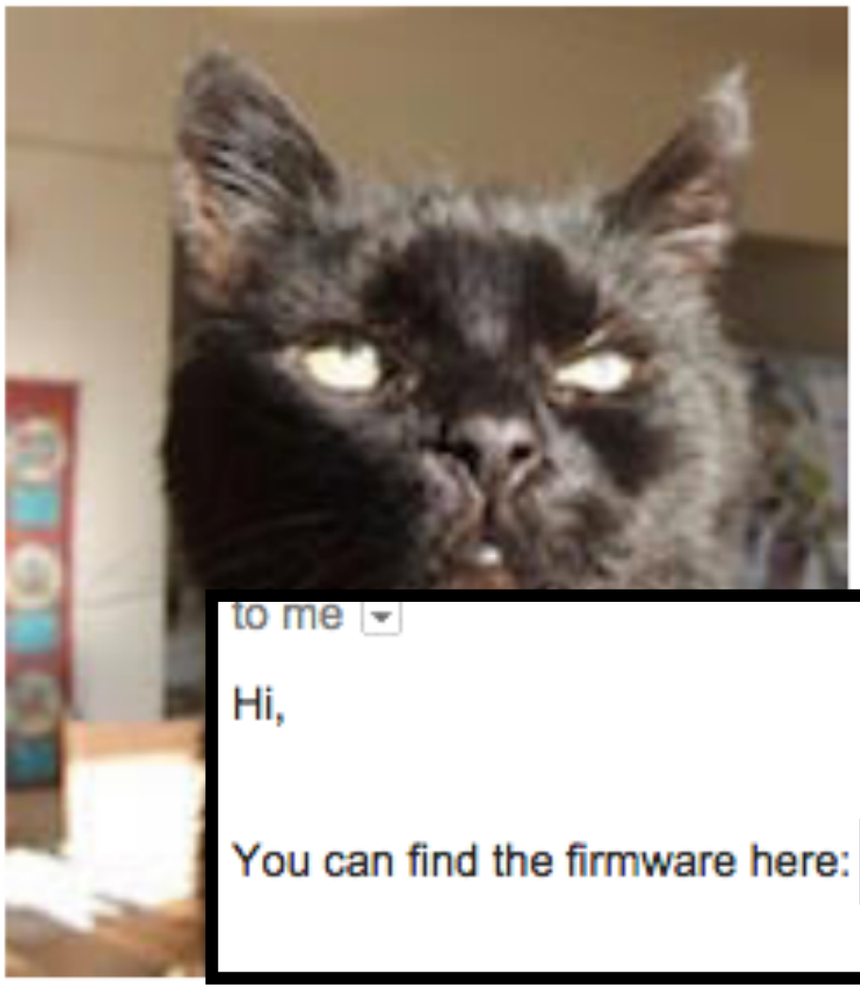
APG NetPRO 488 - Firmware

Inbox x

1337 Kat <l33tkat@gmail.com> 5:08 PM (1 hour ago)

to apgsupport

i can haz firmware?



to me

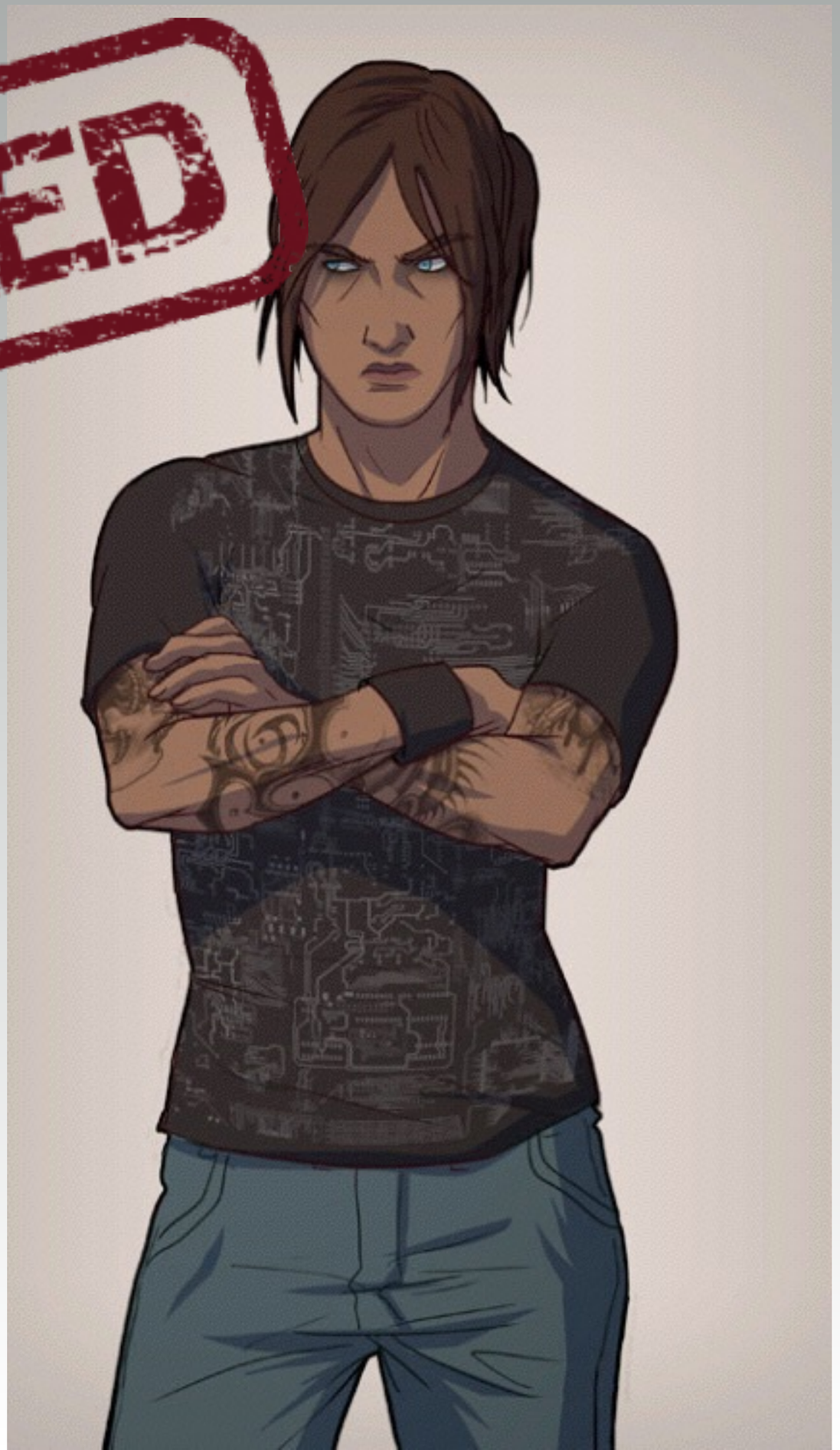
Hi,

You can find the firmware here: [redacted]

day two

reverse the firmware

HIRED



PAIGE

ASSIGNMENT

1. REVERSE THE BINARY

2. FIND A BUG

Name

PAIGE HARRIS

Expertise

Reverser

Favorite film

HOW I MET YOUR SKOCHINSKY

I heard that reversing Atmel code is a mindfuck because of these issues:

PAIGE

- Inconsistent register naming
- Creepy Harvard architecture
- Find xrefs to debug strings

Let's deal with this sucker first:

- Inconsistent register naming

```
$ rasm2 -d fw.bin

<..>
ldi r30, 0x15
ldi r31, 0xE
st r20, Z
<..>
```

ldi – load immediate into register
st – store register into byte at address

Dafuq did I just see?

What does Z stand for?

PAIGE

Zebra?

Zorg?

Zimbabwe?

```
                        loc_4D7A:
2F20                    mov     r18, r16
2B21                    or      r18, r17
F061                    breq    loc_4D89
```

```
                        loc_4D89:
E020                    ldi     r18, 0
932C                    st      X, r18
```

```
01F2                    movw    r30, r4
8120                    ld      r18, Z
01FD                    movw    r30, r26
8320                    st      Z, r18
E021                    ldi     r18, 1
0E42                    add     r4, r18
E020                    ldi     r18, 0
1E52                    adc     r5, r18
9611                    adiw    r26, 1
5001                    subi    r16, 1
4010                    sbci    r17, 0
CFF1                    rjmp    loc_4D7A
```

```
loc_4D8B:
ldi     r30, 6
jmp     FuncTerm_3
; End of function na_StrnCpySRAMToSRAM
```

AVR Programmer Manual:

In order to enable 16-bit addressing, the last six registers are paired to form registers X, Y and Z:

r26:27 - X
r28:29 - Y
r30:31 - Z

```
$ rasm2 -d fw.bin
<..>
ldi r30, 0x15
ldi r31, 0xE
// Z is now 0xE15
st r20, Z
<..>
```

ldi – load immediate into register
st – store register into byte at address

Nailed it! But where the hell are the strings?

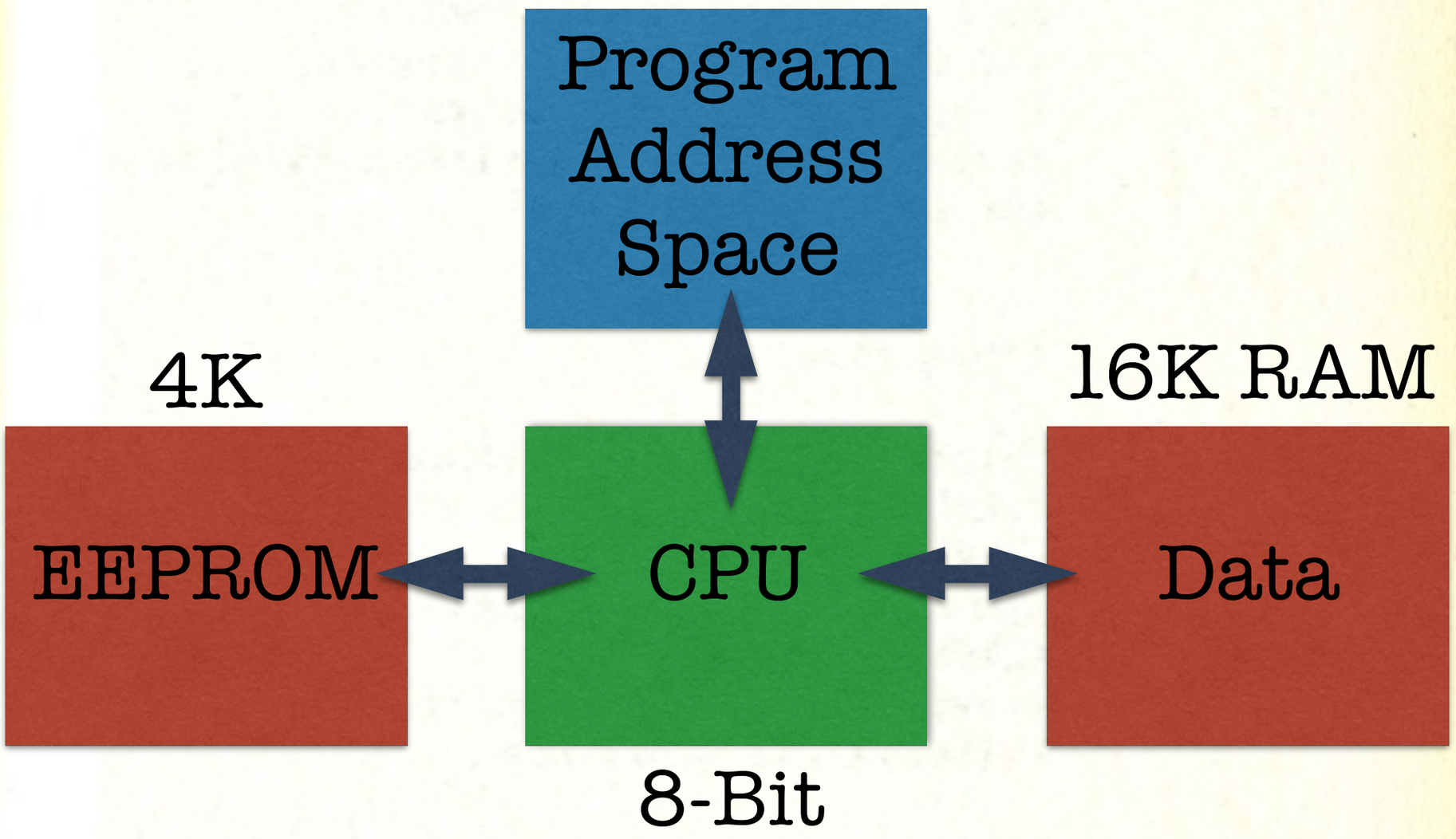
- ~~Inconsistent register naming~~
- Creepy Harvard architecture
- Find xrefs to debug strings

I have a hunch that solving the next challenge will help:

PAIGE

- Creepy Harvard architecture

AVR Programmer Manual:



Got it! Now I know how to find string refs.

- ~~Inconsistent register naming~~
- ~~Creepy Harvard architecture~~
- Find xrefs to debug strings

```
StrLen_PM:
<..>
adiw r30, 1
lpm r20, Z
tst r20
breq Return
<..>
```

```
StrLen_RAM:
<..>
adiw r30, 1
ld r20, Z
tst r20
breq Return
<..>
```

adiw – add immediate to register pair

lpm – load byte from program memory

```
                        loc_6778:
EA0E                    ldi     r16, 0xAE ; '«'
E01D                    ldi     r17, 0xD
940E 237A               call    StrStr
2B01                    or      r16, r17
F019                    breq    loc_6781
```

r16:17 == 0xdae

Word addressing: 0x6d7

```
ROM:06D7 6F65 6570 646E 6172+aEopendrawer:    .db "eopendrawer",0
```

Functions window

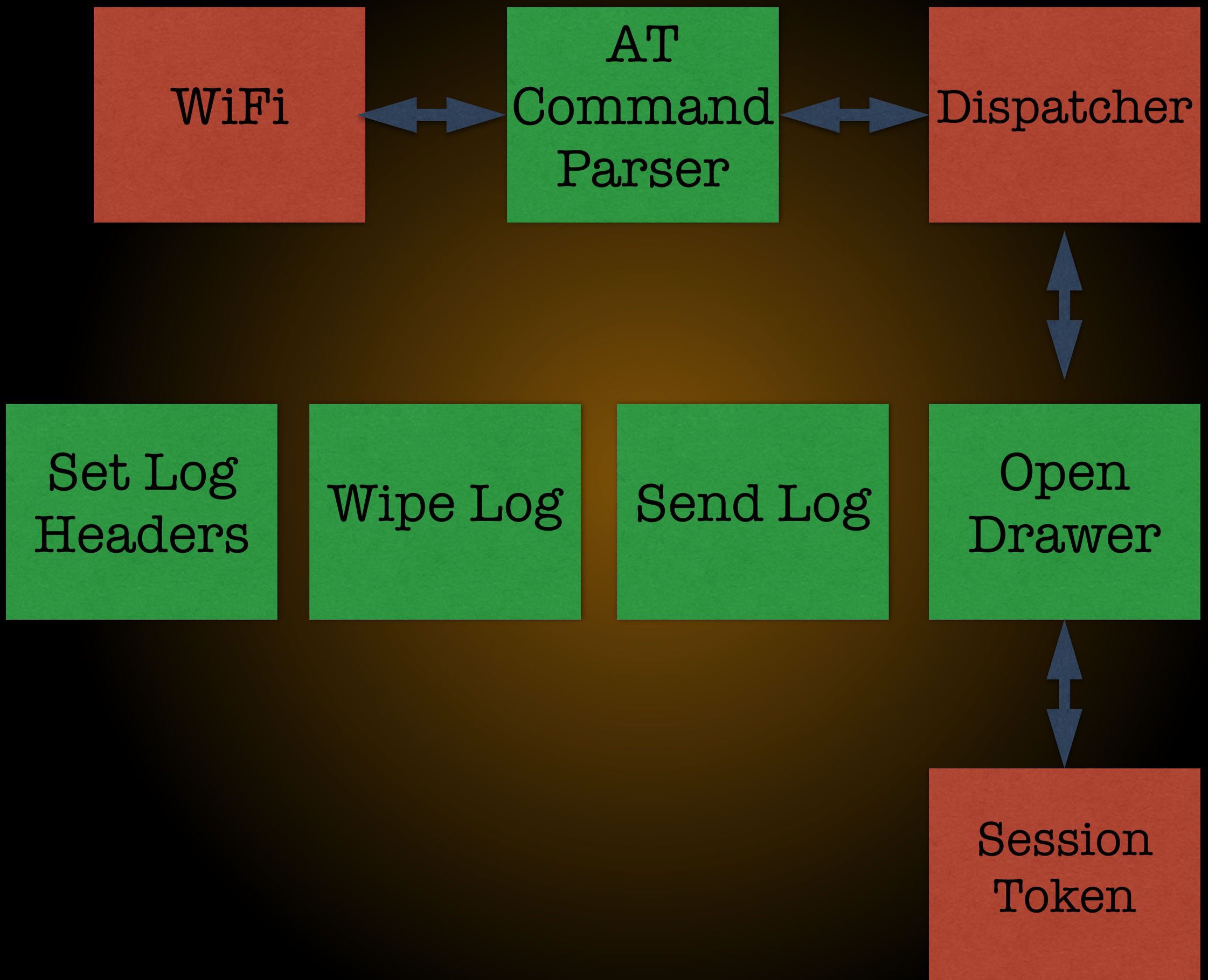
Function name	S
sub_FA5	R
sub_F69	R
sub_F2	R
sub_EA2	R
sub_E4D	R
sub_DF5	R
sub_D6B	R
sub_AF	R
sub_9B	R
sub_85	R
sub_73	R
sub_6CEE	R
sub_6CEA	R
sub_6CD3	R
sub_6CA3	R
sub_6CA0	R
sub_6B	R
sub_693A	R
sub_68E2	R
sub_65EC	R
sub_65B7	R
sub_646F	R
sub_6436	R
sub_64	R
sub_63F7	R
sub_63E6	R
sub_63CF	R

Functions window

Function name	S
na_HandleUSART_2	R
na_HandleUSART	R
na_GetDataOffset	R
na_GenerateClosingXMLTag	R
na_GenerateASCIIPortNumber	R
na_FreeLan	R
na_ForceFreeLAN	R
na_EventIP	R
na_EnableDHCP	R
na_DrawerOpenText	R
na_DrawerCloseText	R
na_DoRecoverAPIPA	R
na_DoNothing	R
na_DoFactoryDefault	R
na_DoConnectWrapper	R
na_DoConnect	R
na_DoCfgWPS	R
na_DoCfgDHCP	R
na_DoCfgBasic	R
na_DoCfgAssoc	R
na_DispatchMessageOrCallDisassociationEvent	R
na_Disconnect	R
na_DisassociationEvent	R
na_DisableDHCP	R
na_DecrementCounter	R
na_DecantRStack	R
na_CopyWizFiHeaderToResponseBuffer	R

Now that I got the debug strings, let's look at the attack surface

- ~~Inconsistent register naming~~
- ~~Creepy Harvard architecture~~
- ~~Find xrefs to debug strings~~


WiFi
AT Command Parser
Dispatcher
Set Log Headers
Wipe Log
Send Log
Open Drawer
Session Token

```php
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```

The attack surface is really tiny

day three

pwn all the things

HIRED

TREVOR

ASSIGNMENT

1. Find a bug
2. Exploit it
3. Get gold

Name

TREVOR PHILIPS

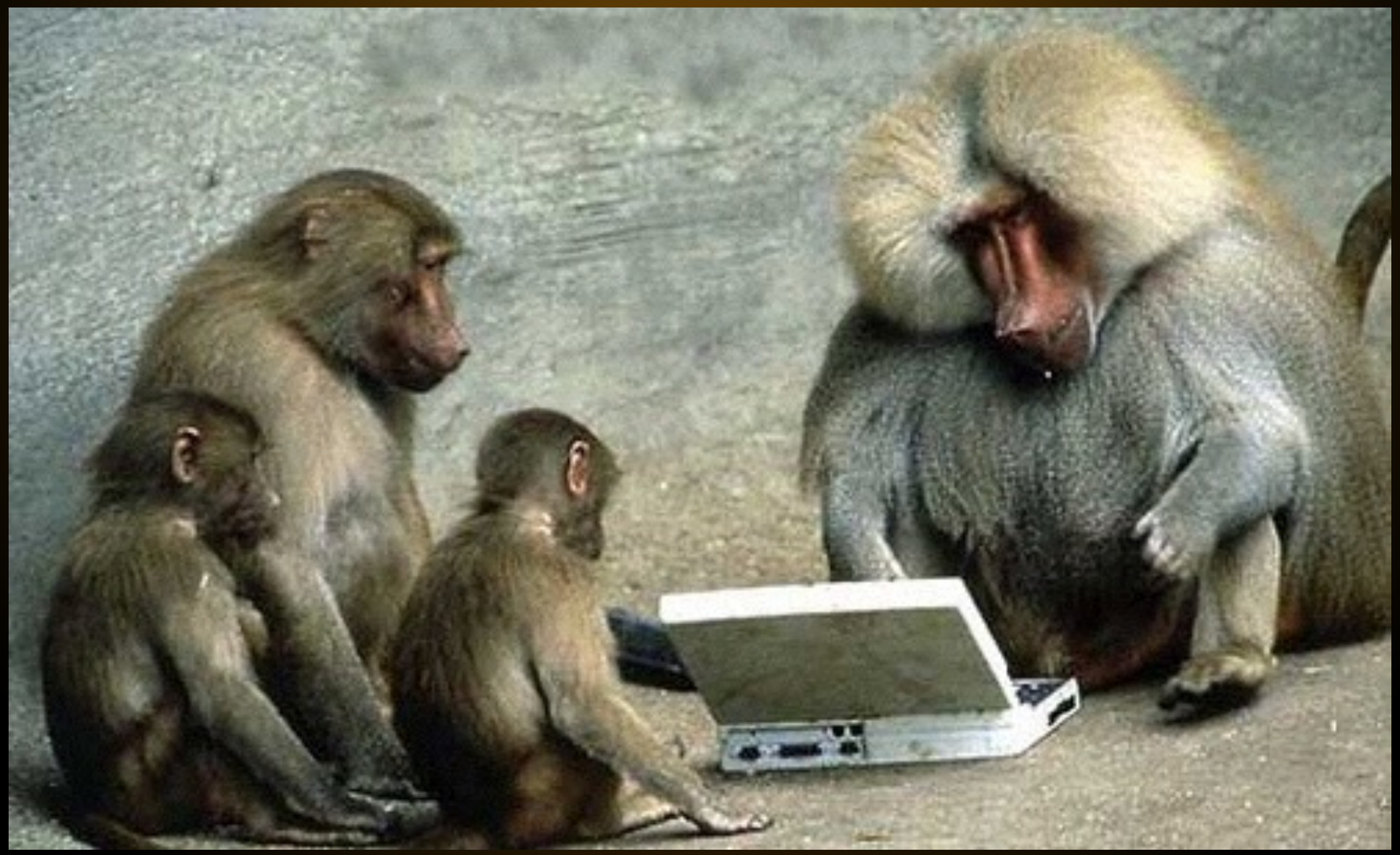
Expertise

MAYHEM

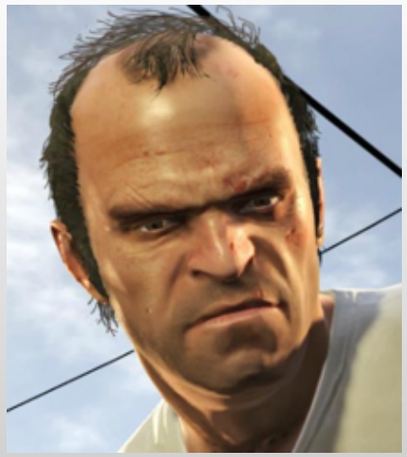
Favorite tool

DIE

Mystery: Who wrote their libc, and when?



strlen walks until a NULL is reached



TREVOR

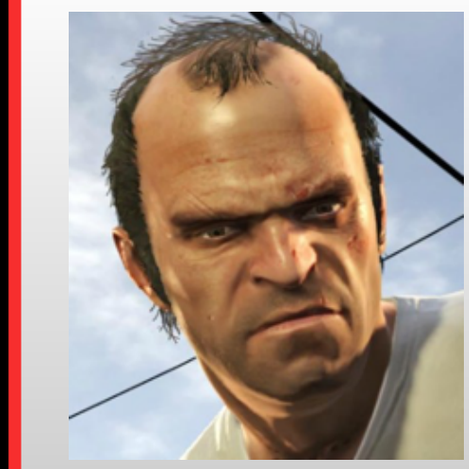
```
na_StrLen_SRAM:         ; CODE XREF: na_LDRHexRecDbgOut:loc_D7C↑p
                        ; na_LDRHexRecDbgOut:loc_D91↑p na_LDRHexRecDbgOut+37↑p
                        ; na_LDRHexRecDbgOut+52↑p na_LDRHexRecDbgOut+69↑p ...
movw    r18, r16
ldi     r16, 0
ldi     r17, 0
```

```
loc_4D4F:               ; CODE XREF: na_StrLen_SRAM+C↓j
movw    r30, r18
movw    r18, r30

subi    r18, -1
sbci    r19, -1

ld      r20, Z

tst     r20
breq    locret_4D59
```

```
subi    r16, -1
sbci    r17, -1
rjmp    loc_4D4F
```

```
locret_4D59:            ; CODE XREF: na_StrLen_SRAM+9↑j
ret
; End of function na_StrLen_SRAM
```

strcpy doesn't add a NULL byte to the end of the string




```
loc_218E:                               ; CODE XREF: i_bet_you_look_good_on_the_dance_floor+1C↓j
movw    r30, r26
ld      r16, Z
cpi     r16, 0xA
breq    bye
```

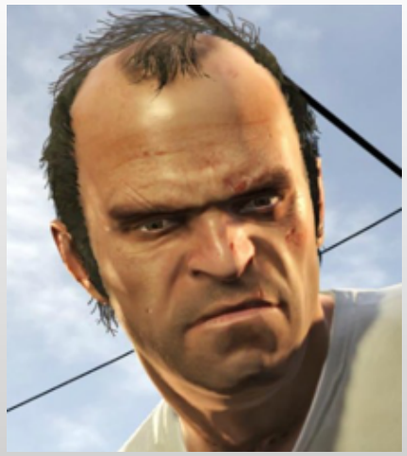
```
cpi     r24, 0x50 ; 'P'
ldi     r16, 0
cpc     r25, r16
brcc    bye
```

```
ld      r16, X
movw    r30, r24
subi    r30, 0x1A
sbci    r31, -0x14 ; '8'
st      Z, r16
adiw    r26, 1
adiw    r24, 1
rjmp    loc_218E
```

```
bye:                                    ; CODE XREF: i_bet_you_look_good_on_the_dance_floor+10↑j
                                        ; i_bet_you_look_good_on_the_dance_floor+14↑j
call    sub_1E35

ldi     r30, 4
jmp     gb_BL_epilogue_8
; End of function i_bet_you_look_good_on_the_dance_floor
```


wat

Using these two primitives we can get code execution



Where to write into?

Stack return address is stored at beginning of RAM

Three stage pwn

1

Log header buffer

Flags (set to non-0)

strlen returns wrong val

2

memcpy with value of strlen

Overwrite pointer

Trigger write to stack

3

Overwrite return address

Build ROP Chain

Trigger drawer open

Um, I think you've missed something

LESTER

The Money Function

```
                na_OpenCashDrawer:
939A            st      -Y, r25
938A            st      -Y, r24
B79F            in      r25, SREG       ; Status Register
94F8            cli
2F80            mov     r24, r16
9380 0E5A       sts     unk_100E5A, r24
9828            cbi     ADCH, 0
E302            ldi     r16, 0x32 ; '2'
E010            ldi     r17, 0
9300 0E58       sts     unk_100E58, r16
9310 0E59       sts     unk_100E59, r17
E200            ldi     r16, 0x20 ; ' '
EB1F            ldi     r17, 0xBF ; '+'
E022            ldi     r18, 2
E030            ldi     r19, 0
E5EF            ldi     r30, 0x5F ; '_'
E0FE            ldi     r31, 0xE
8300            st      Z, r16
8311            std     Z+1, r17
8322            std     Z+2, r18
8333            std     Z+3, r19
DFDA            rcall   na_SendSignalToEngine
E500            ldi     r16, 0x50 ; 'P'
E010            ldi     r17, 0
9300 0E5D       sts     unk_100E5D, r16
9310 0E5E       sts     unk_100E5E, r17
E001            ldi     r16, 1
9300 0E64       sts     unk_100E64, r16
BF9F            out     SREG, r25       ; Status Register
9189            ld      r24, Y+
9199            ld      r25, Y+
9508            ret
                ; End of function na_OpenCashDrawer
```

They forgot to check credentials!

the heist

Ready for the job of a lifetime? Here's the target

LESTER

CAFE
ANASTASIA

We have one gun on the spot to trigger the open

LESTER

CAFE ANASTASIA

And another gun to grab the cash when it's open

LESTER

CAFE ANASTASIA

...THIS IS IT! Go for it

LESTER

DEMO


SAMSUNG
questions?