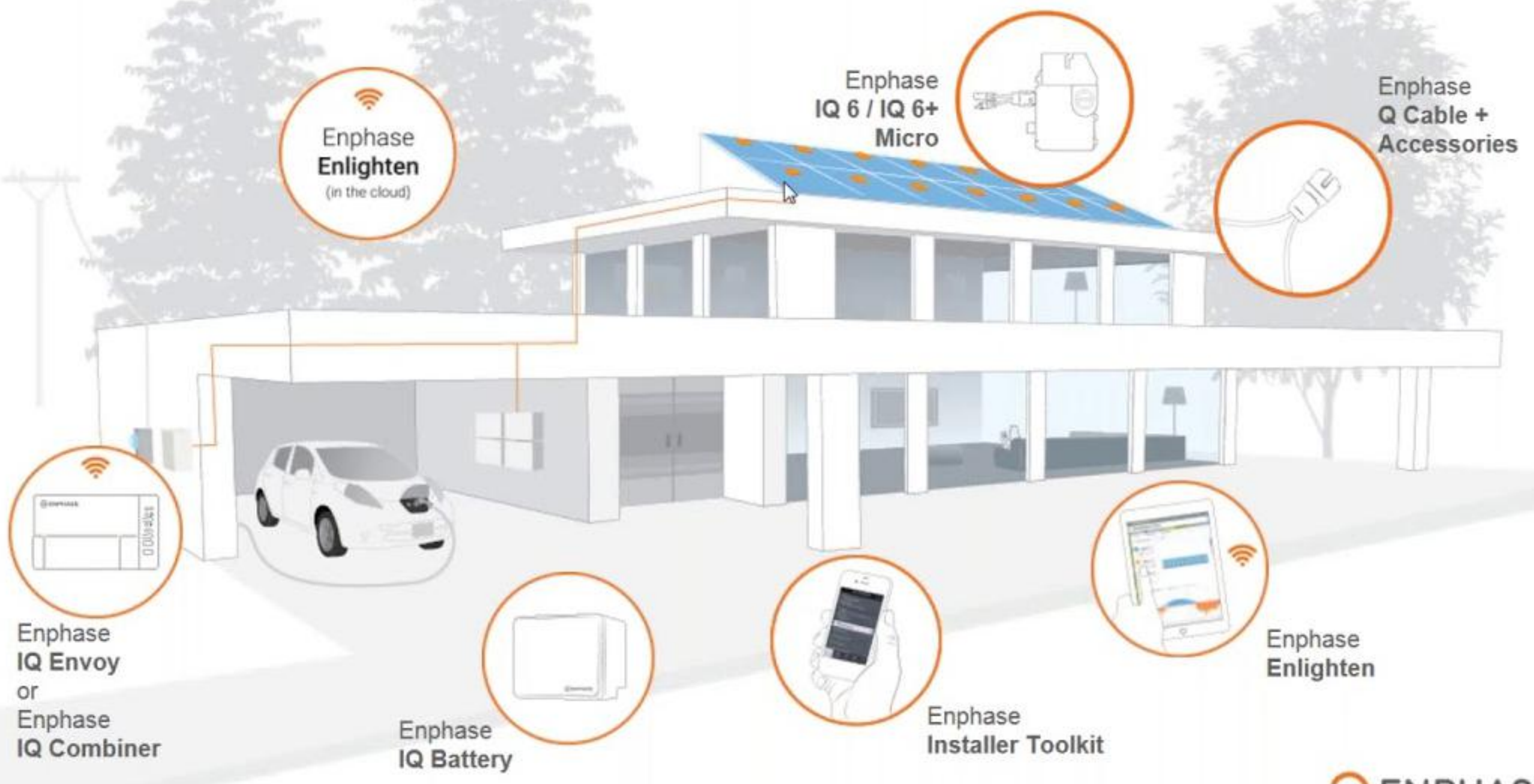
# 1.21 Gigawatts!

Vulnerabilities in solar panel controllers

# Background

# Phase 1:
# Opensource Research

# The Enphase Home Energy Solution



Enphase **Enlighten** (in the cloud)

Enphase **IQ 6 / IQ 6+ Micro**

Enphase **Q Cable + Accessories**

Enphase **IQ Envoy** or Enphase **IQ Combiner**

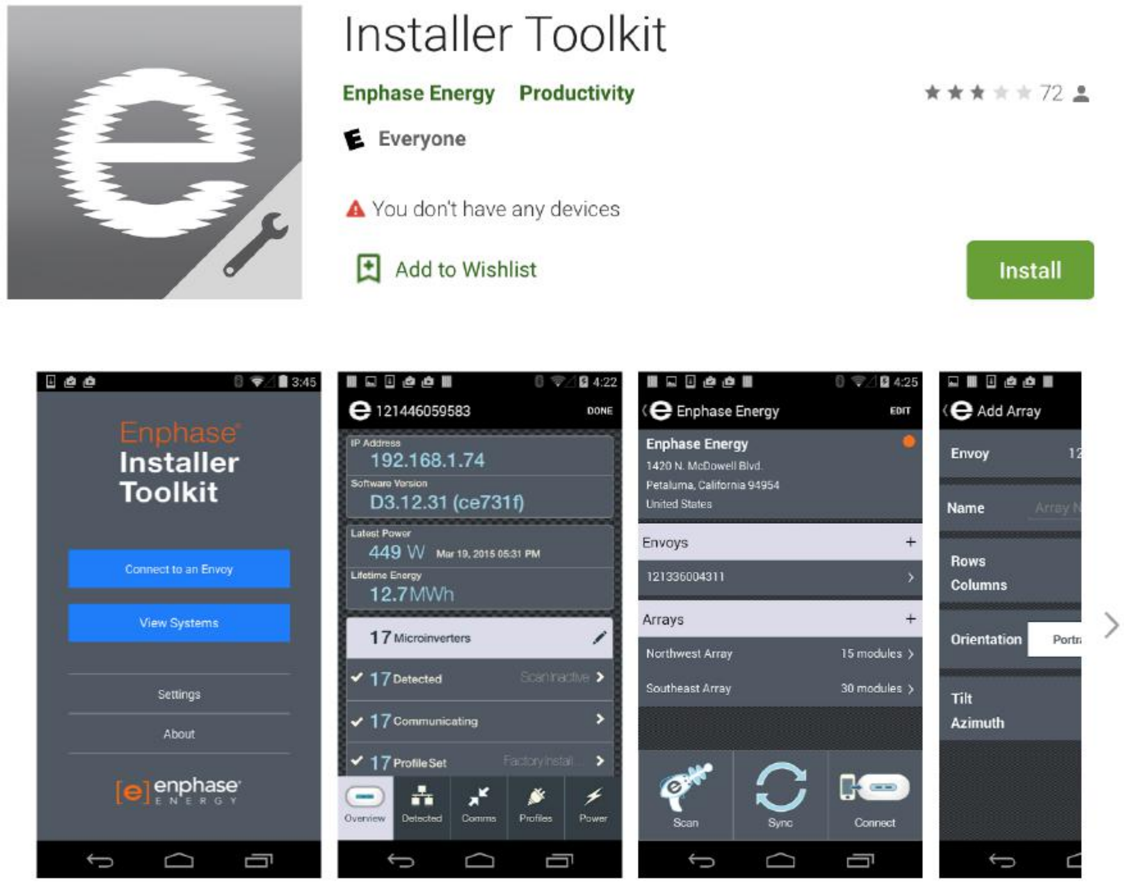
Enphase **IQ Battery**

Enphase **Installer Toolkit**

Enphase **Enlighten**

|

# Solar Management Apps



## Installer Toolkit

Enphase Energy Productivity

★★★☆☆ 72

Everyone

You don't have any devices

Add to Wishlist

Install

Enphase Installer Toolkit

121446059583

IP Address: 192.168.1.74

Software Version: D3.12.31 (ce731f)

Latest Power: 449 W Mar 19, 2015 05:31 PM

Lifetime Energy: 12.7MWh

17 Microinverters

✓ 17 Detected — Scan inactive

✓ 17 Communicating

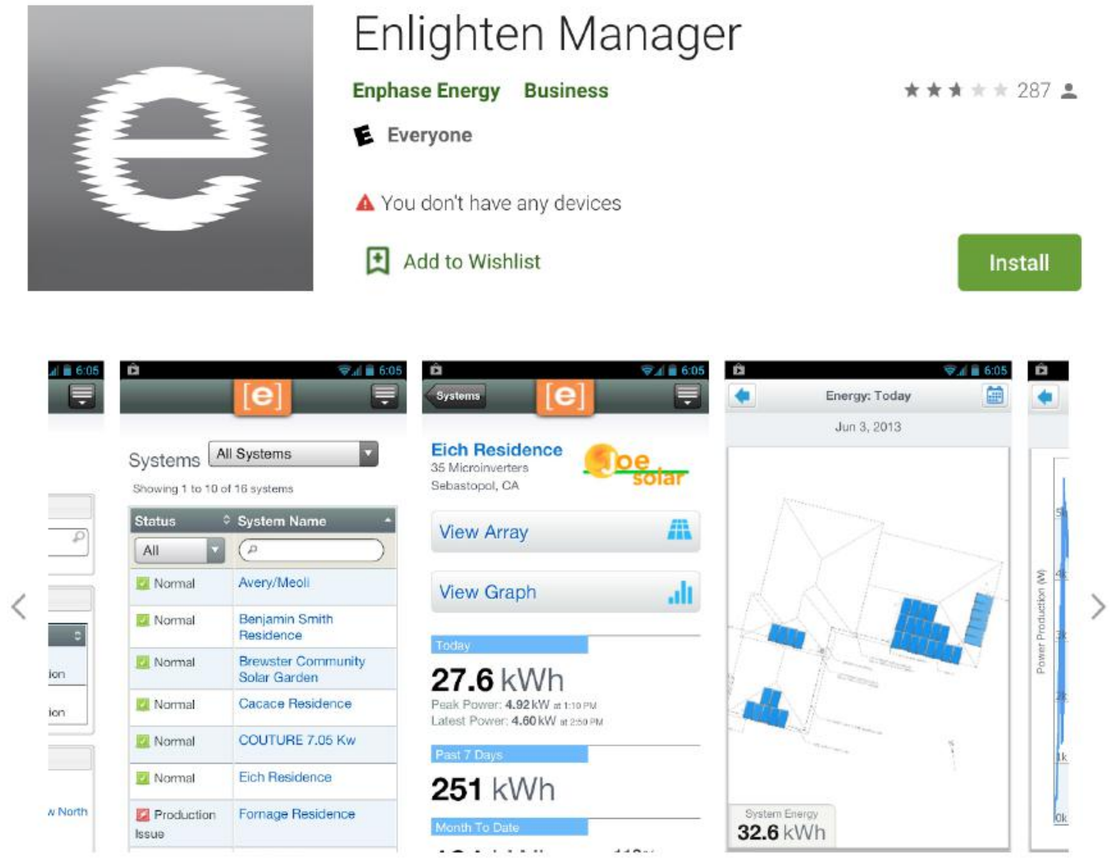
✓ 17 Profile Set — Factory install...

Enphase Energy
1420 N. McDowell Blvd.
Petaluma, California 94954
United States

Envoys: 121336004311

Arrays: Northwest Array — 15 modules; Southeast Array — 30 modules

Add Array: Envoy, Name, Rows, Columns, Orientation, Tilt, Azimuth



## Enlighten Manager

Enphase Energy Business

★★★☆☆ 287

Everyone

You don't have any devices

Add to Wishlist

Install

Systems: All Systems

Showing 1 to 10 of 16 systems

| Status | System Name |
|---|---|
| All | |
| Normal | Avery/Meoli |
| Normal | Benjamin Smith Residence |
| Normal | Brewster Community Solar Garden |
| Normal | Cacace Residence |
| Normal | COUTURE 7.05 Kw |
| Normal | Eich Residence |
| Production Issue | Fornage Residence |

Eich Residence
35 Microinverters
Sebastopol, CA

Today: 27.6 kWh
Peak Power: 4.92 kW at 1:10 PM
Latest Power: 4.60 kW at 2:50 PM

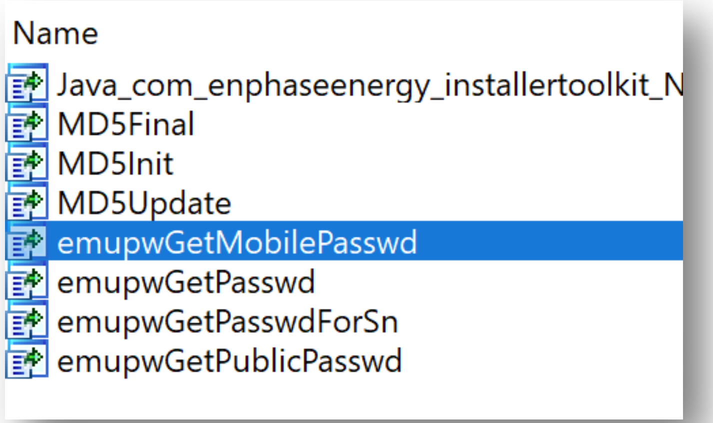
Past 7 Days: 251 kWh

Month To Date

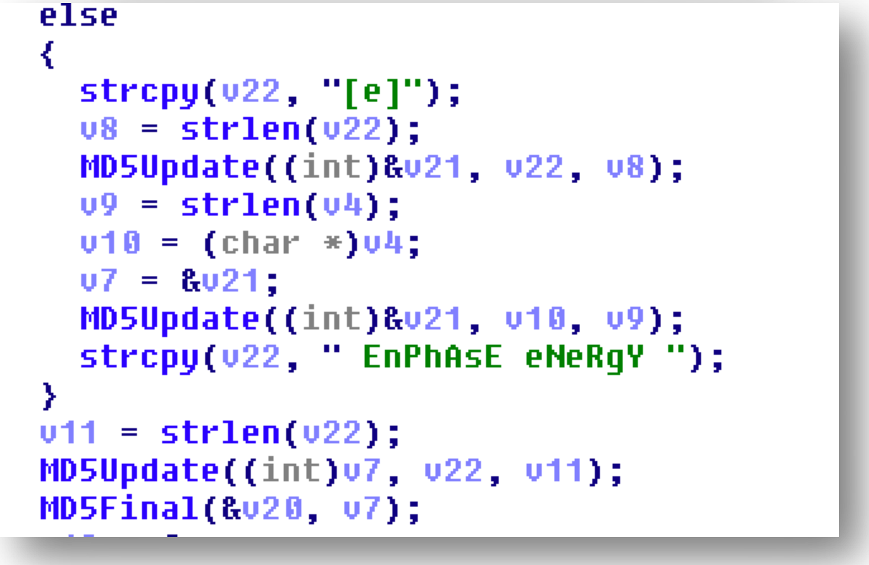
Energy: Today
Jun 3, 2013

System Energy 32.6 kWh

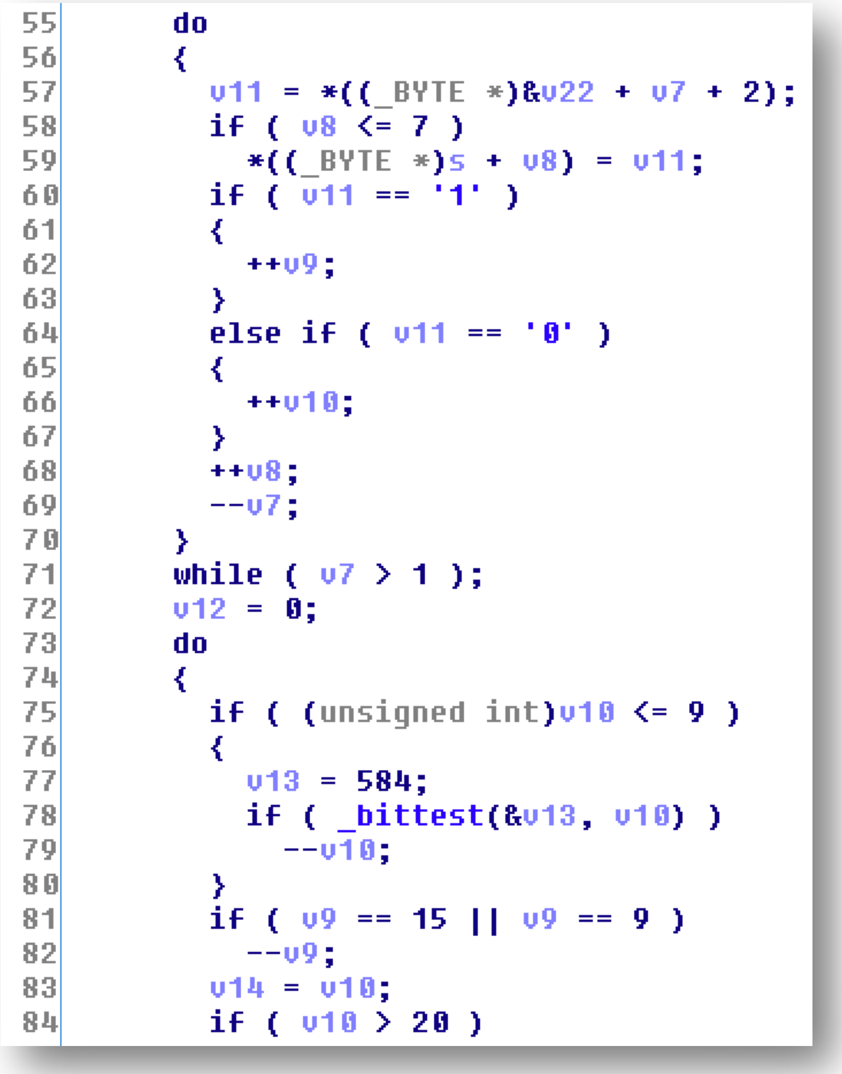
# Installer App Authentication



Name
- Java_com_enphaseenergy_installertoolkit_N
- MD5Final
- MD5Init
- MD5Update
- emupwGetMobilePasswd
- emupwGetPasswd
- emupwGetPasswdForSn
- emupwGetPublicPasswd

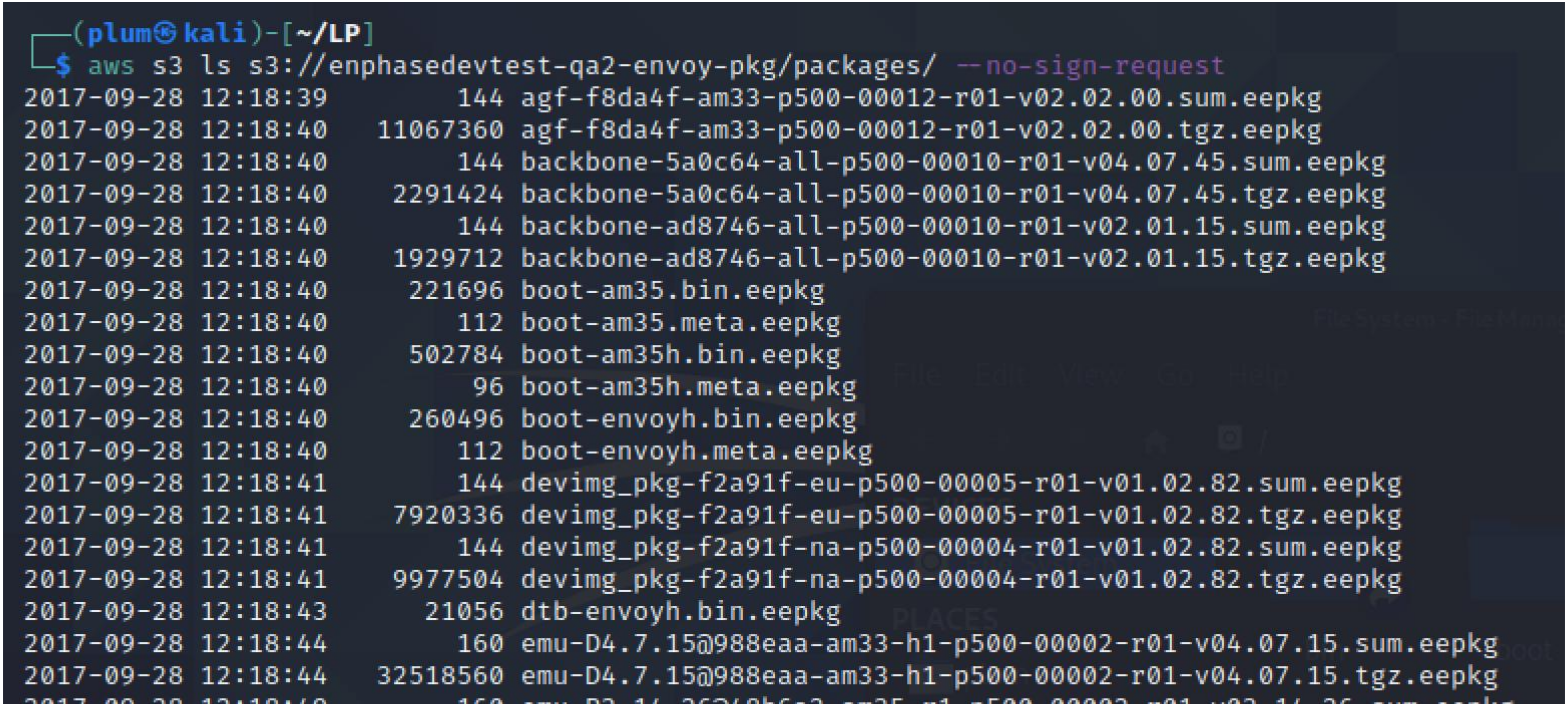


```
else
{
  strcpy(v22, "[e]");
  v8 = strlen(v22);
  MD5Update((int)&v21, v22, v8);
  v9 = strlen(v4);
  v10 = (char *)v4;
  v7 = &v21;
  MD5Update((int)&v21, v10, v9);
  strcpy(v22, " EnPhAsE eNeRgY ");
}
v11 = strlen(v22);
MD5Update((int)v7, v22, v11);
MD5Final(&v20, v7);
```



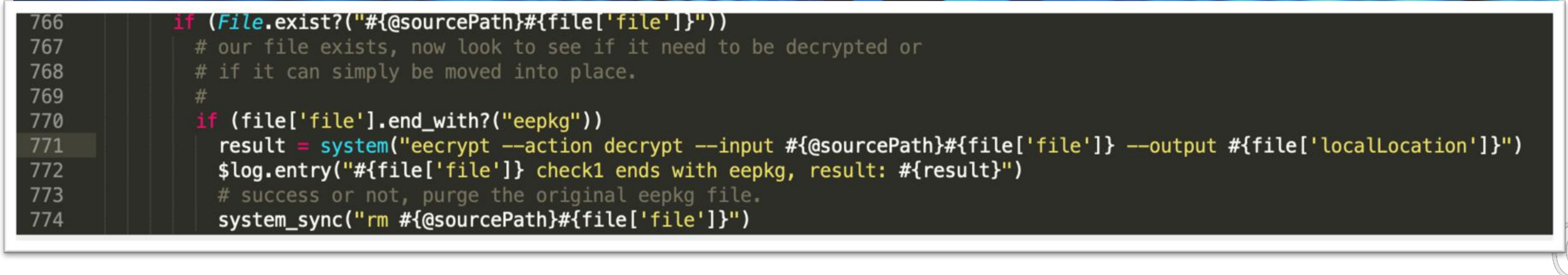
```
do
{
  v11 = *((_BYTE *)&v22 + v7 + 2);
  if ( v8 <= 7 )
    *((_BYTE *)s + v8) = v11;
  if ( v11 == '1' )
  {
    ++v9;
  }
  else if ( v11 == '0' )
  {
    ++v10;
  }
  ++v8;
  --v7;
}
while ( v7 > 1 );
v12 = 0;
do
{
  if ( (unsigned int)v10 <= 9 )
  {
    v13 = 584;
    if ( _bittest(&v13, v10) )
      --v10;
  }
  if ( v9 == 15 || v9 == 9 )
    --v9;
  v14 = v10;
  if ( v10 > 20 )
```

# Open S3 Bucket



```
┌──(plum㉿kali)-[~/LP]
└─$ aws s3 ls s3://enphasedevtest-qa2-envoy-pkg/packages/ --no-sign-request
2017-09-28 12:18:39        144 agf-f8da4f-am33-p500-00012-r01-v02.02.00.sum.eepkg
2017-09-28 12:18:40   11067360 agf-f8da4f-am33-p500-00012-r01-v02.02.00.tgz.eepkg
2017-09-28 12:18:40        144 backbone-5a0c64-all-p500-00010-r01-v04.07.45.sum.eepkg
2017-09-28 12:18:40    2291424 backbone-5a0c64-all-p500-00010-r01-v04.07.45.tgz.eepkg
2017-09-28 12:18:40        144 backbone-ad8746-all-p500-00010-r01-v02.01.15.sum.eepkg
2017-09-28 12:18:40    1929712 backbone-ad8746-all-p500-00010-r01-v02.01.15.tgz.eepkg
2017-09-28 12:18:40     221696 boot-am35.bin.eepkg
2017-09-28 12:18:40        112 boot-am35.meta.eepkg
2017-09-28 12:18:40     502784 boot-am35h.bin.eepkg
2017-09-28 12:18:40         96 boot-am35h.meta.eepkg
2017-09-28 12:18:40     260496 boot-envoyh.bin.eepkg
2017-09-28 12:18:40        112 boot-envoyh.meta.eepkg
2017-09-28 12:18:41        144 devimg_pkg-f2a91f-eu-p500-00005-r01-v01.02.82.sum.eepkg
2017-09-28 12:18:41    7920336 devimg_pkg-f2a91f-eu-p500-00005-r01-v01.02.82.tgz.eepkg
2017-09-28 12:18:41        144 devimg_pkg-f2a91f-na-p500-00004-r01-v01.02.82.sum.eepkg
2017-09-28 12:18:41    9977504 devimg_pkg-f2a91f-na-p500-00004-r01-v01.02.82.tgz.eepkg
2017-09-28 12:18:43      21056 dtb-envoyh.bin.eepkg
2017-09-28 12:18:44        160 emu-D4.7.15@988eaa-am33-h1-p500-00002-r01-v04.07.15.sum.eepkg
2017-09-28 12:18:44   32518560 emu-D4.7.15@988eaa-am33-h1-p500-00002-r01-v04.07.15.tgz.eepkg
```

# Encrypted Firmware Images



```
if (File.exist?("#{@sourcePath}#{file['file']}"))
  # our file exists, now look to see if it need to be decrypted or
  # if it can simply be moved into place.
  #
  if (file['file'].end_with?("eepkg"))
    result = system("eecrypt --action decrypt --input #{@sourcePath}#{file['file']} --output #{file['localLocation']}")
    $log.entry("#{file['file']} check1 ends with eepkg, result: #{result}")
    # success or not, purge the original eepkg file.
    system_sync("rm #{@sourcePath}#{file['file']}")
```

# Phase 2:
Hardware Attacks

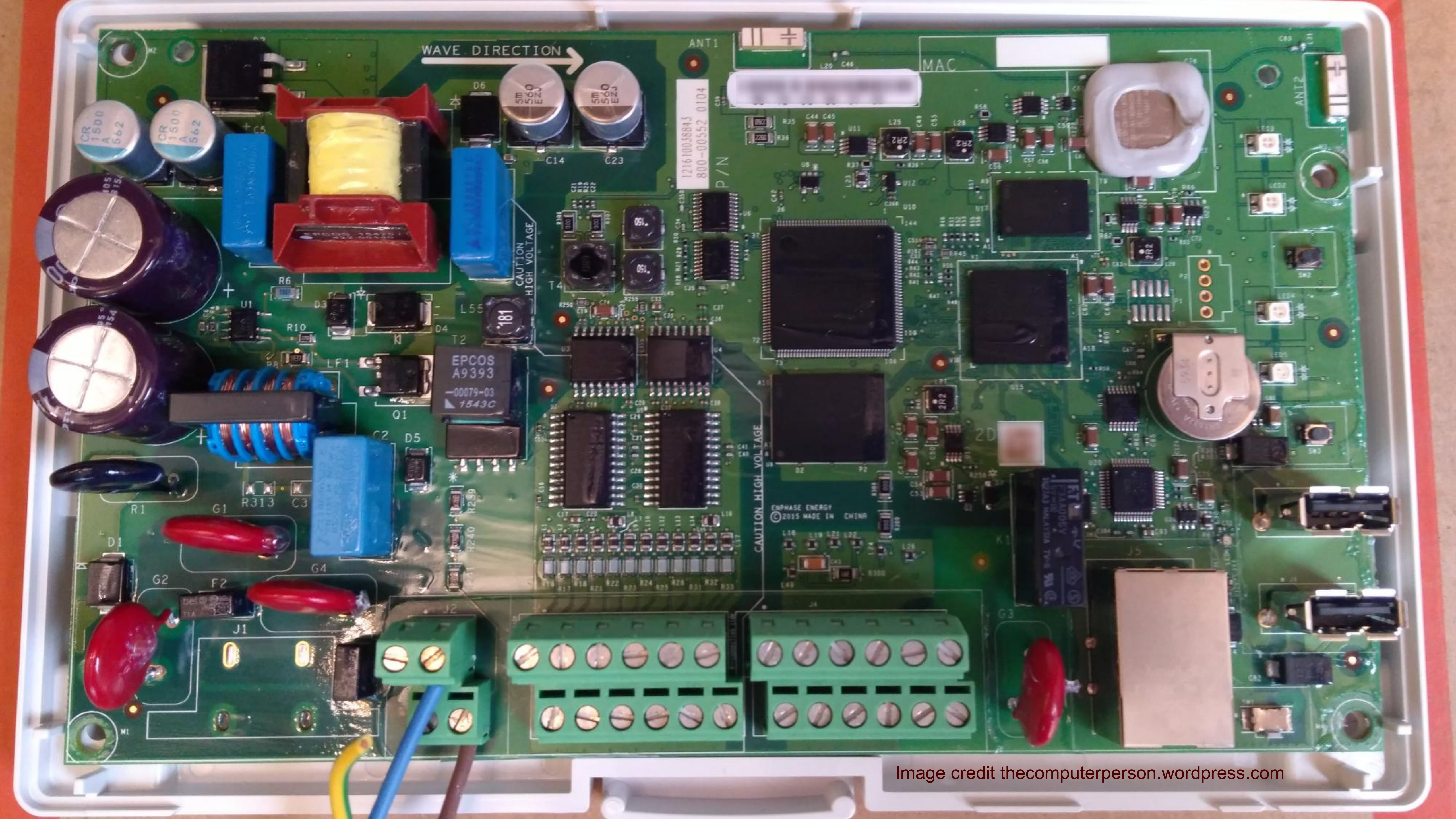
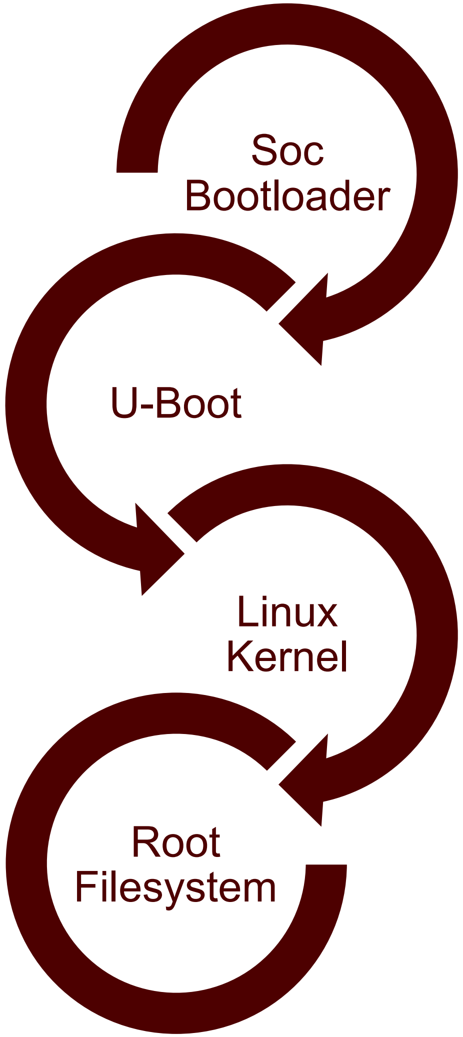


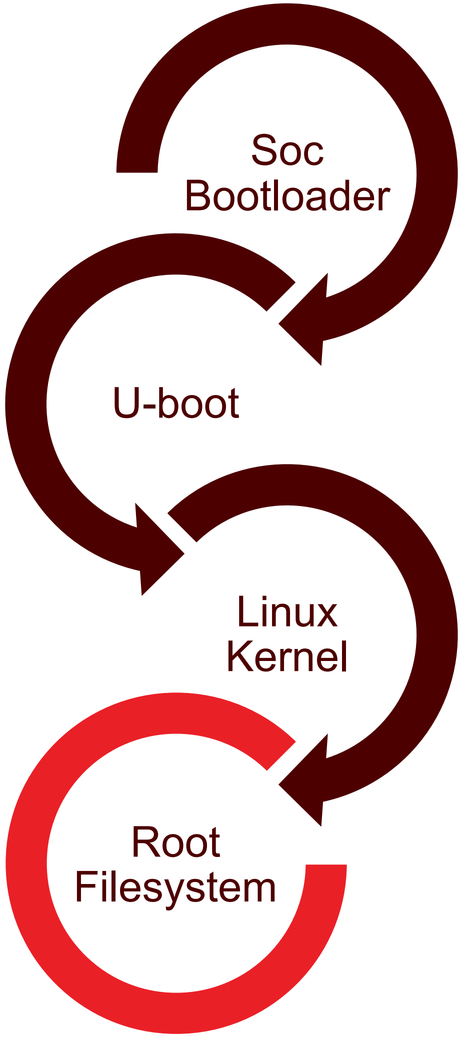
Image credit thecomputerperson.wordpress.com

# Typical Embedded Linux Boot Process



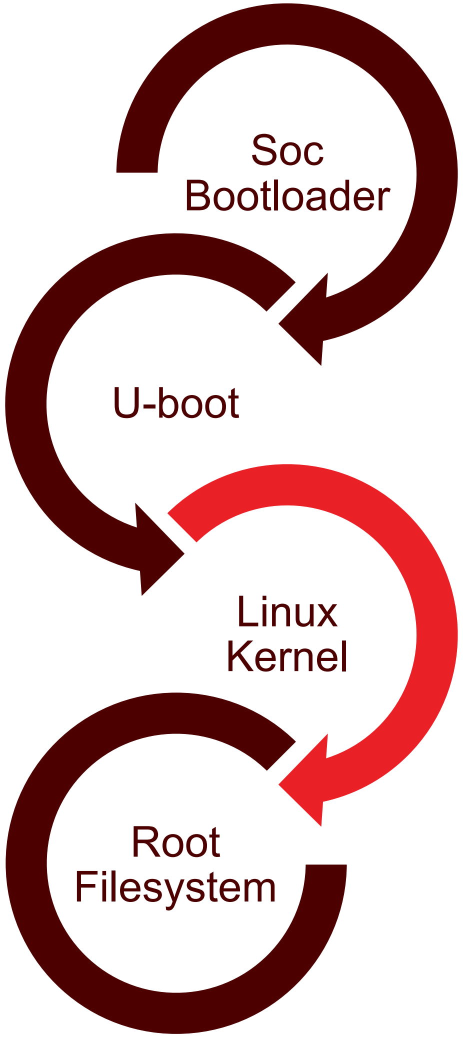
# Typical Embedded Linux Boot Process

- Root Filesystem mounted (run level 5)
  - Possible TTY console or login prompt
  - Running services
    - SSH
    - HTTP/S
    - Telnet
- Attacks at this stage
  - Debug ports
    - UART TTY
    - JTAG



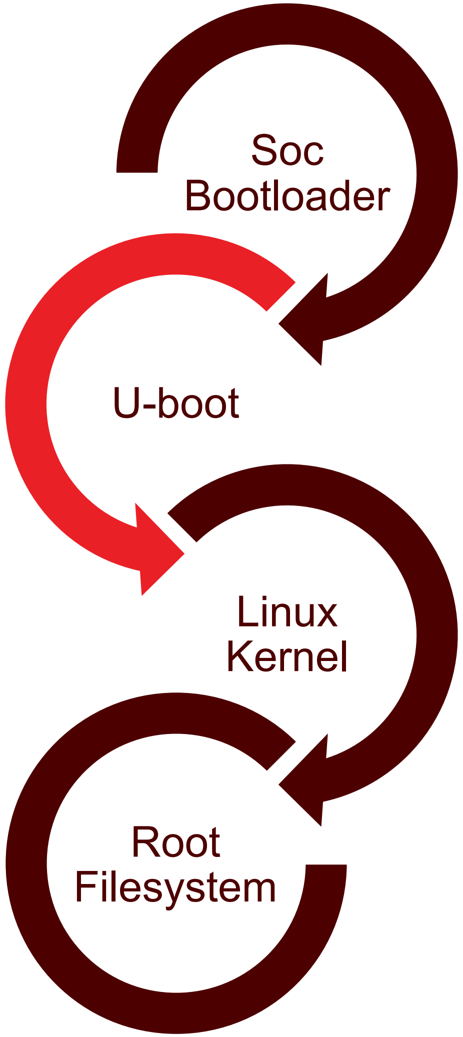
# Typical Embedded Linux Boot Process

- Linux Kernel startup
  - Linux Kernel loads itself
  - mounts file system(s)
  - Processes Init scripts
- Attacks at this stage
  - Boot arguments
    - Enable TTY console
    - Change run level
    - Single user mode
  - Glitching
    - Prevent kernel from correctly read EEMC



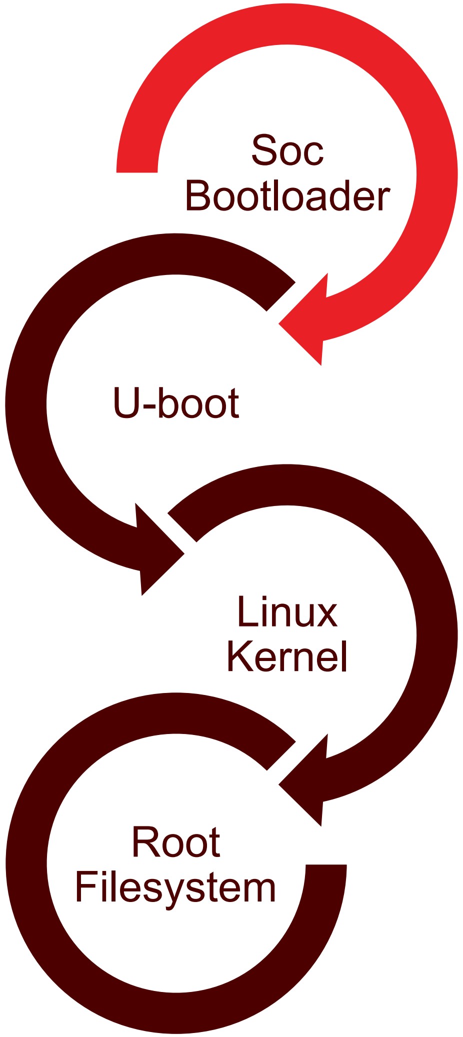
# Typical Embedded Linux Boot Process

- Second level bootloader (U-boot)
  - Loads U-boot environment variables
  - Finds root file system
  - Jumps to Linux Kernel
- Attacks at this stage
  - U-Boot user interrupt
  - Alter U-Boot Environment
  - Replace Kernel Image / Load alternate Kernel

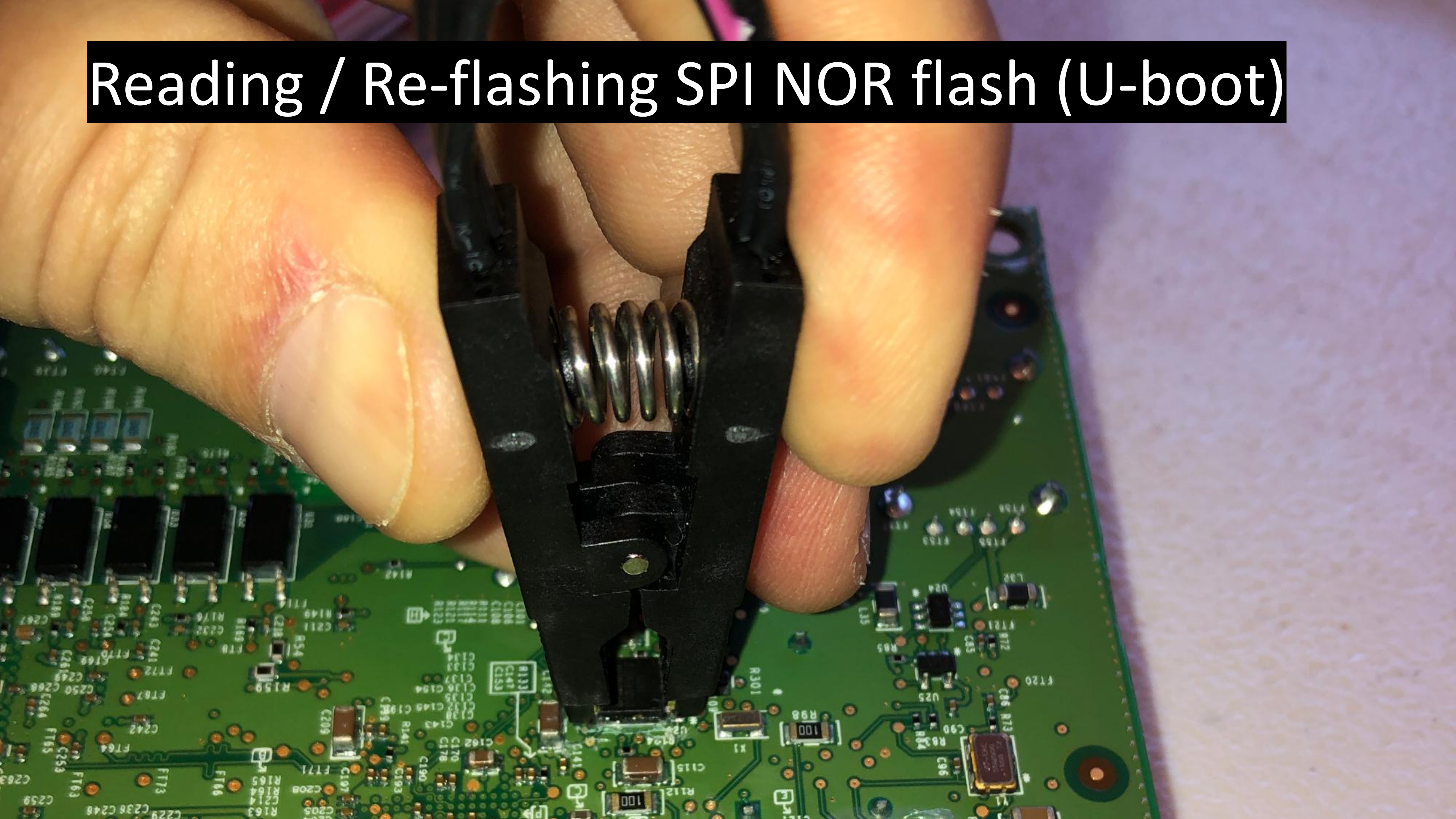


# Typical Embedded Linux Boot Process

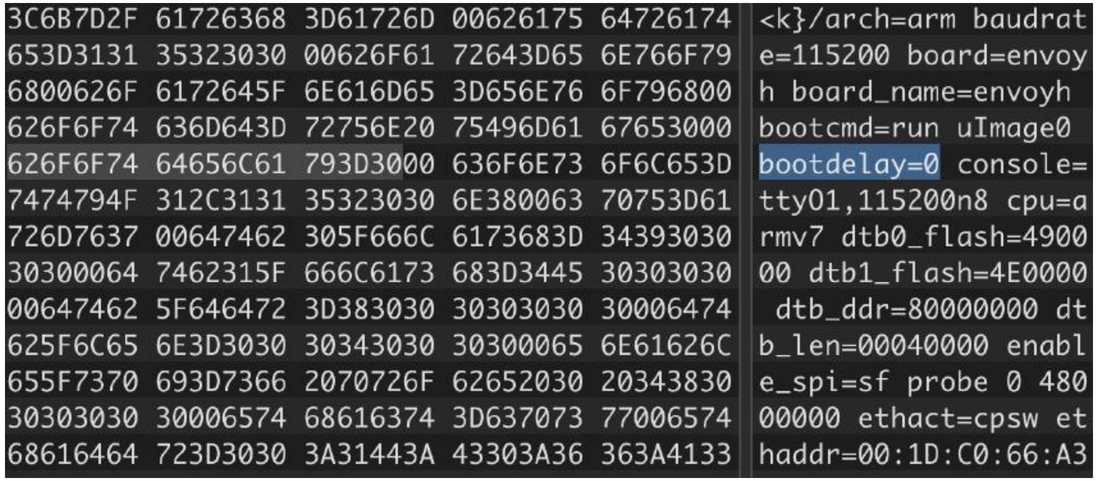
- First Stage bootloader
  - Soc ROM bootloader jumps to first stage bootloader
  - First stage bootloader locates U-boot image
    - Typically found on SPI flash chip
  - Passes execution to Second stage bootloader (U-boot)
- Attacks at this Stage
  - JTAG debugging
  - Replace second stage bootloader image (dangerous!)



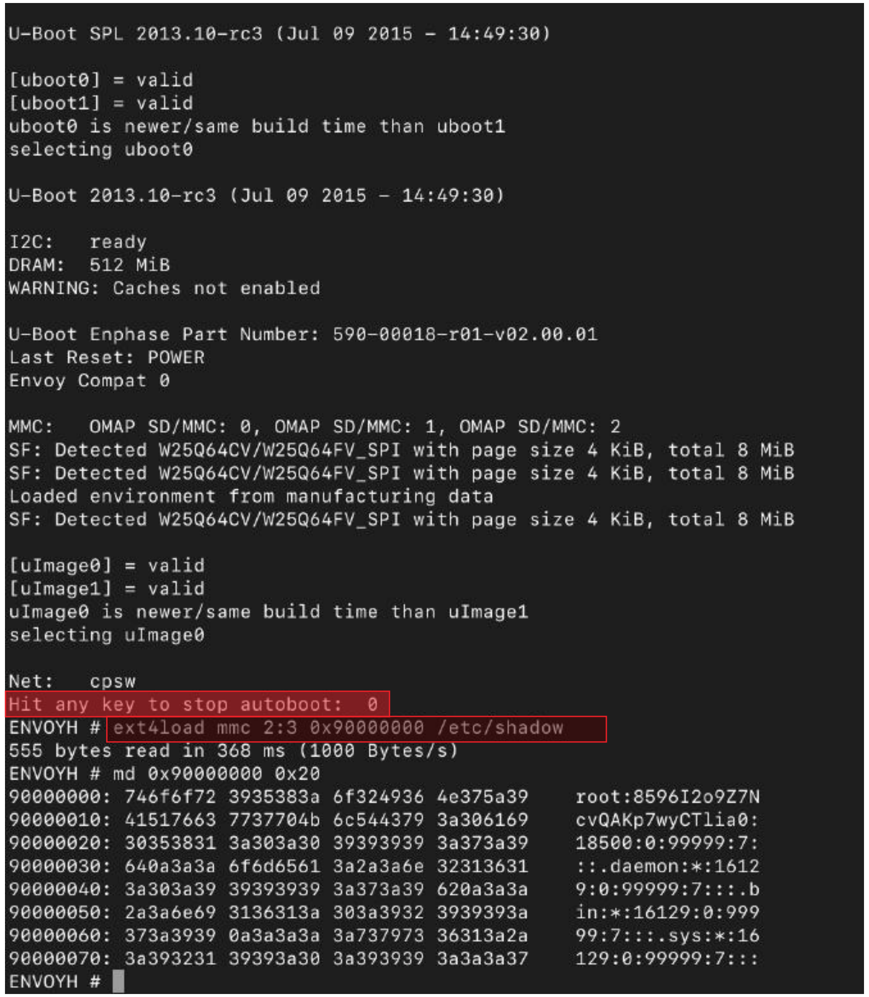
# Reading / Re-flashing SPI NOR flash (U-boot)



# Re-flashing bootloader

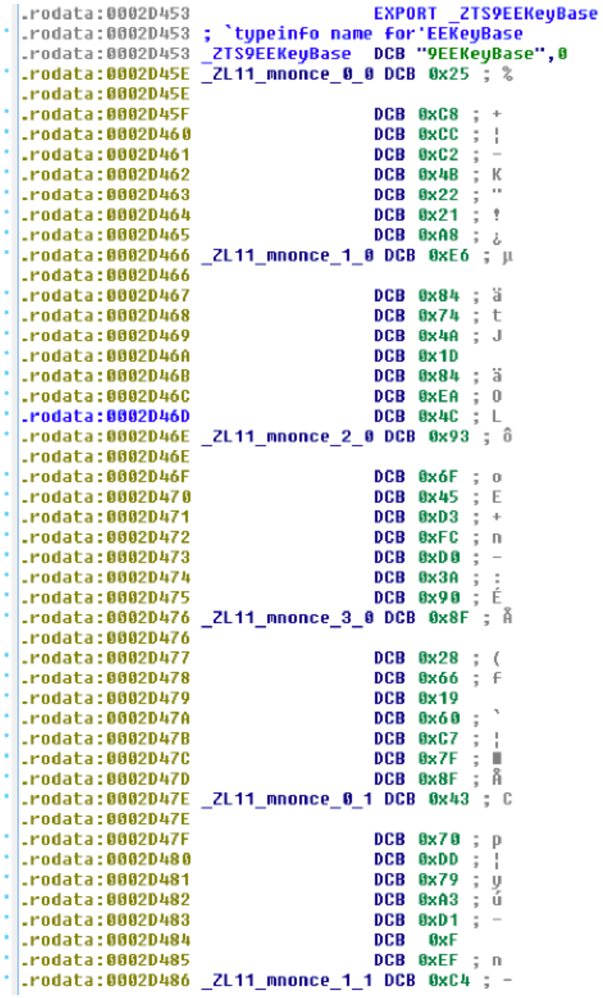


```
3C6B7D2F 61726368 3D61726D 00626175 64726174  <k}/arch=arm baudrat
653D3131 35323030 00626F61 72643D65 6E766F79  e=115200 board=envoy
6800626F 6172645F 6E616D65 3D656E76 6F796800  h board_name=envoyh
626F6F74 636D643D 72756E20 75496D61 67653000  bootcmd=run uImage0
626F6F74 64656C61 793D3000 636F6E73 6F6C653D  bootdelay=0 console=
7474794F 312C3131 35323030 6E380063 70753D61  tty01,115200n8 cpu=a
726D7637 00647462 305F666C 6173683D 34393030  rmv7 dtb0_flash=4900
30300064 7462315F 666C6173 683D3445 30303030  00 dtb1_flash=4E0000
00647462 5F646472 3D383030 30303030 30006474   dtb_ddr=80000000 dt
625F6C65 6E3D3030 30343030 30300065 6E61626C  b_len=00040000 enabl
655F7370 693D7366 2070726F 62652030 20343830  e_spi=sf probe 0 480
30303030 30006574 68616374 3D637073 77006574  00000 ethact=cpsw et
68616464 723D3030 3A31443A 43303A36 363A4133  haddr=00:1D:C0:66:A3
```

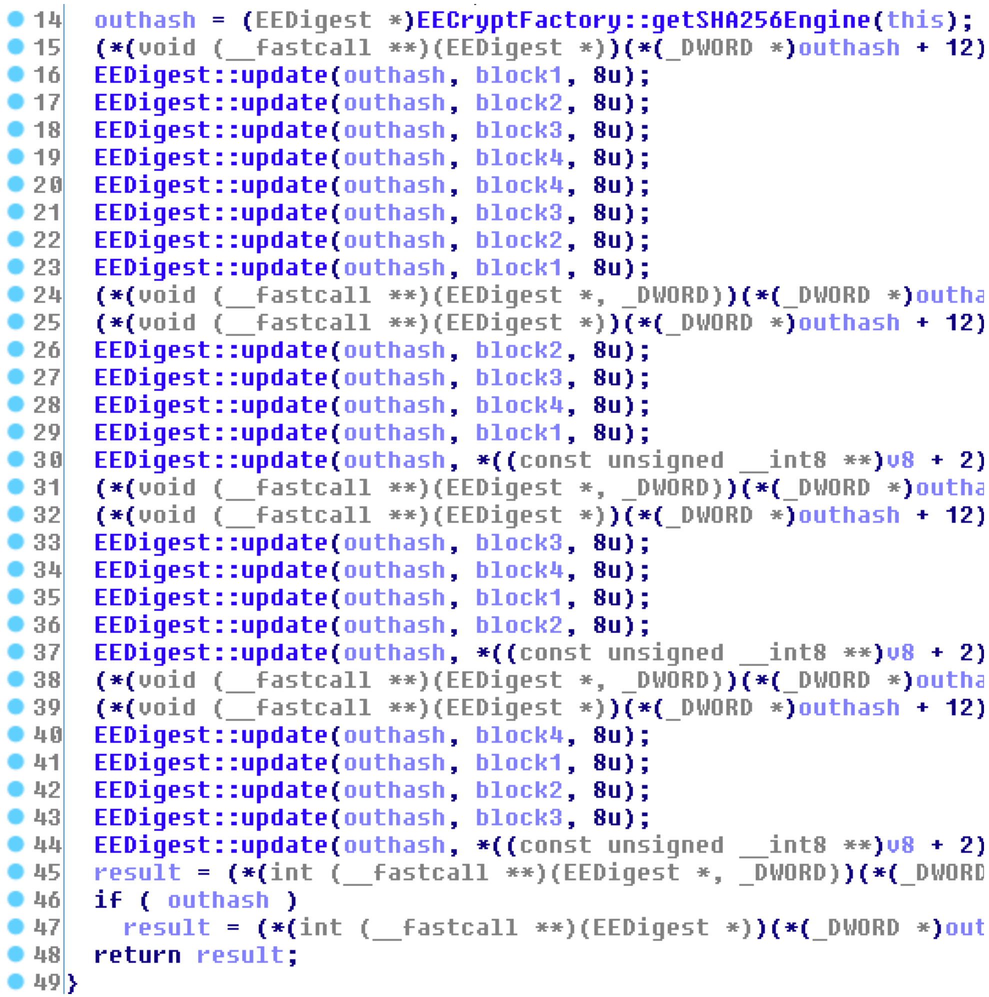


```
U-Boot SPL 2013.10-rc3 (Jul 09 2015 - 14:49:30)

[uboot0] = valid
[uboot1] = valid
uboot0 is newer/same build time than uboot1
selecting uboot0

U-Boot 2013.10-rc3 (Jul 09 2015 - 14:49:30)

I2C:   ready
DRAM:  512 MiB
WARNING: Caches not enabled

U-Boot Enphase Part Number: 590-00018-r01-v02.00.01
Last Reset: POWER
Envoy Compat 0

MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1, OMAP SD/MMC: 2
SF: Detected W25Q64CV/W25Q64FV_SPI with page size 4 KiB, total 8 MiB
SF: Detected W25Q64CV/W25Q64FV_SPI with page size 4 KiB, total 8 MiB
Loaded environment from manufacturing data
SF: Detected W25Q64CV/W25Q64FV_SPI with page size 4 KiB, total 8 MiB

[uImage0] = valid
[uImage1] = valid
uImage0 is newer/same build time than uImage1
selecting uImage0

Net:   cpsw
Hit any key to stop autoboot:  0
ENVOYH # ext4load mmc 2:3 0x90000000 /etc/shadow
555 bytes read in 368 ms (1000 Bytes/s)
ENVOYH # md 0x90000000 0x20
90000000: 746f6f72 3935383a 6f324936 4e375a39    root:8596I2o9Z7N
90000010: 41517663 7737704b 6c544379 3a306169    cvQAKp7wyCTlia0:
90000020: 30353831 3a303a30 39393939 3a373a39    18500:0:99999:7:
90000030: 640a3a3a 6f6d6561 3a2a3a6e 32313631    ::.daemon:*:1612
90000040: 3a303a39 39393939 3a373a39 620a3a3a    9:0:99999:7:::.b
90000050: 2a3a6e69 3136313a 303a3932 3939393a    in:*:16129:0:999
90000060: 373a3939 0a3a3a3a 3a737973 36313a2a    99:7:::.sys:*:16
90000070: 3a393231 39393a30 3a393939 3a3a3a37    129:0:99999:7:::
ENVOYH #
```

# eecrypt

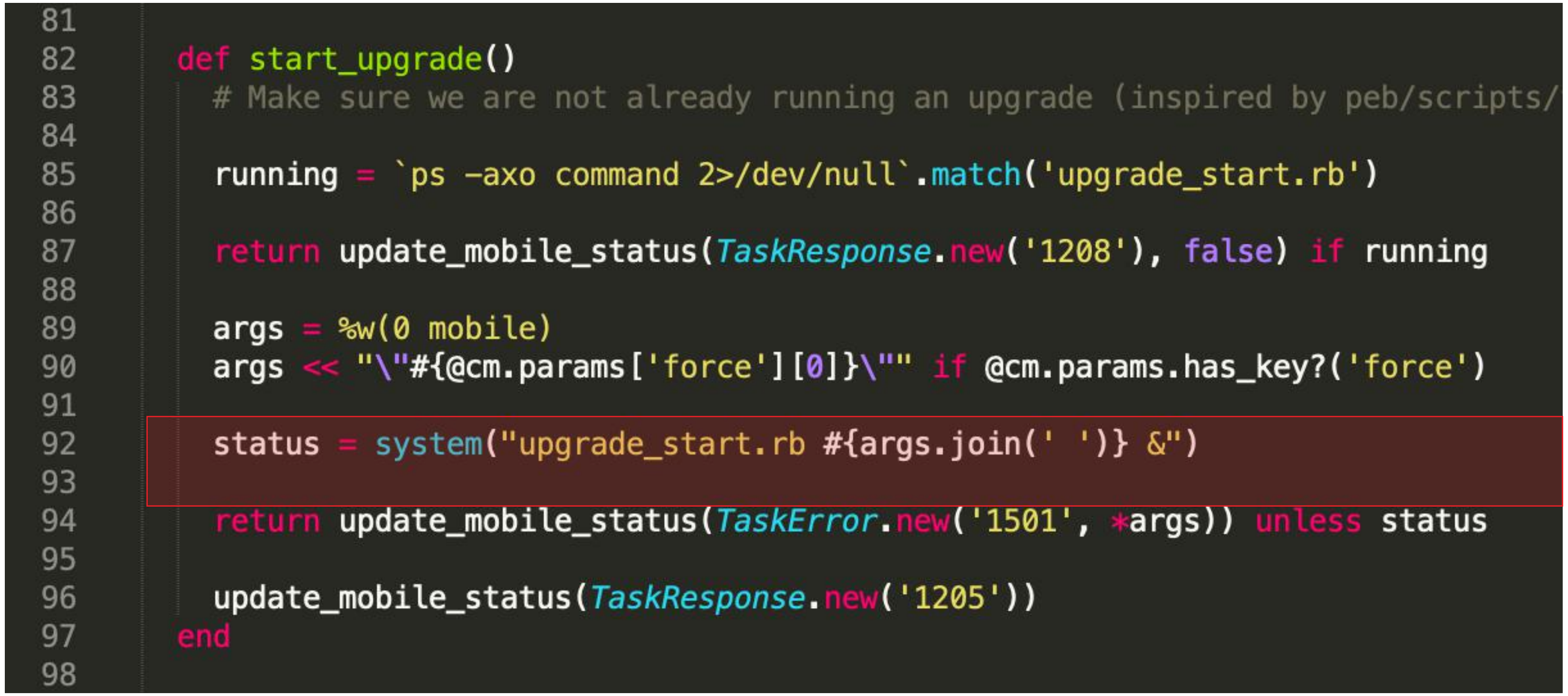


```
.rodata:0002D453                 EXPORT _ZTS9EEKeyBase
.rodata:0002D453 ; `typeinfo name for'EEKeyBase
.rodata:0002D453 _ZTS9EEKeyBase  DCB "9EEKeyBase",0
.rodata:0002D45E _ZL11_mnonce_0_0 DCB 0x25 ; %
.rodata:0002D45E
.rodata:0002D45F                 DCB 0xC8 ; +
.rodata:0002D460                 DCB 0xCC ; ¦
.rodata:0002D461                 DCB 0xC2 ; -
.rodata:0002D462                 DCB 0x4B ; K
.rodata:0002D463                 DCB 0x22 ; "
.rodata:0002D464                 DCB 0x21 ; !
.rodata:0002D465                 DCB 0xA8 ; ¿
.rodata:0002D466 _ZL11_mnonce_1_0 DCB 0xE6 ; µ
.rodata:0002D466
.rodata:0002D467                 DCB 0x84 ; ä
.rodata:0002D468                 DCB 0x74 ; t
.rodata:0002D469                 DCB 0x4A ; J
.rodata:0002D46A                 DCB 0x1D
.rodata:0002D46B                 DCB 0x84 ; ä
.rodata:0002D46C                 DCB 0xEA ; O
.rodata:0002D46D                 DCB 0x4C ; L
.rodata:0002D46E _ZL11_mnonce_2_0 DCB 0x93 ; ô
.rodata:0002D46E
.rodata:0002D46F                 DCB 0x6F ; o
.rodata:0002D470                 DCB 0x45 ; E
.rodata:0002D471                 DCB 0xD3 ; +
.rodata:0002D472                 DCB 0xFC ; n
.rodata:0002D473                 DCB 0xD0 ; -
.rodata:0002D474                 DCB 0x3A ; :
.rodata:0002D475                 DCB 0x90 ; É
.rodata:0002D476 _ZL11_mnonce_3_0 DCB 0x8F ; Å
.rodata:0002D476
.rodata:0002D477                 DCB 0x28 ; (
.rodata:0002D478                 DCB 0x66 ; f
.rodata:0002D479                 DCB 0x19
.rodata:0002D47A                 DCB 0x60 ; `
.rodata:0002D47B                 DCB 0xC7 ; ¦
.rodata:0002D47C                 DCB 0x7F ; ▮
.rodata:0002D47D                 DCB 0x8F ; Å
.rodata:0002D47E _ZL11_mnonce_0_1 DCB 0x43 ; C
.rodata:0002D47E
.rodata:0002D47F                 DCB 0x70 ; p
.rodata:0002D480                 DCB 0xDD ; ¦
.rodata:0002D481                 DCB 0x79 ; y
.rodata:0002D482                 DCB 0xA3 ; ú
.rodata:0002D483                 DCB 0xD1 ; -
.rodata:0002D484                 DCB  0xF
.rodata:0002D485                 DCB 0xEF ; n
.rodata:0002D486 _ZL11_mnonce_1_1 DCB 0xC4 ; -
```



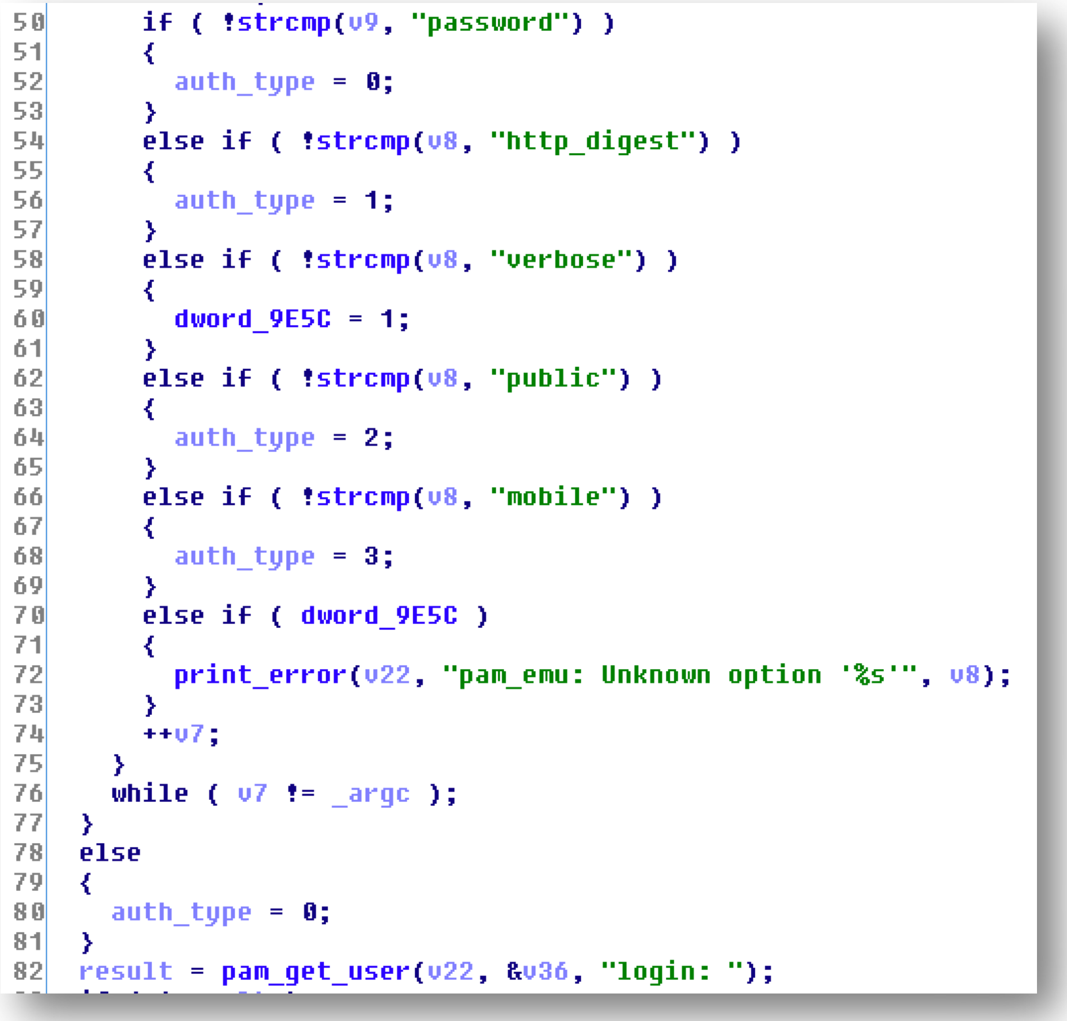
```
outhash = (EEDigest *)EECryptFactory::getSHA256Engine(this);
(*(void (__fastcall **)(EEDigest *))(*(_DWORD *)outhash + 12)
EEDigest::update(outhash, block1, 8u);
EEDigest::update(outhash, block2, 8u);
EEDigest::update(outhash, block3, 8u);
EEDigest::update(outhash, block4, 8u);
EEDigest::update(outhash, block4, 8u);
EEDigest::update(outhash, block3, 8u);
EEDigest::update(outhash, block2, 8u);
EEDigest::update(outhash, block1, 8u);
(*(void (__fastcall **)(EEDigest *, _DWORD))(*(_DWORD *)outha
(*(void (__fastcall **)(EEDigest *))(*(_DWORD *)outhash + 12)
EEDigest::update(outhash, block2, 8u);
EEDigest::update(outhash, block3, 8u);
EEDigest::update(outhash, block4, 8u);
EEDigest::update(outhash, block1, 8u);
EEDigest::update(outhash, *((const unsigned __int8 **)v8 + 2)
(*(void (__fastcall **)(EEDigest *, _DWORD))(*(_DWORD *)outha
(*(void (__fastcall **)(EEDigest *))(*(_DWORD *)outhash + 12)
EEDigest::update(outhash, block3, 8u);
EEDigest::update(outhash, block4, 8u);
EEDigest::update(outhash, block1, 8u);
EEDigest::update(outhash, block2, 8u);
EEDigest::update(outhash, *((const unsigned __int8 **)v8 + 2)
(*(void (__fastcall **)(EEDigest *, _DWORD))(*(_DWORD *)outha
(*(void (__fastcall **)(EEDigest *))(*(_DWORD *)outhash + 12)
EEDigest::update(outhash, block4, 8u);
EEDigest::update(outhash, block1, 8u);
EEDigest::update(outhash, block2, 8u);
EEDigest::update(outhash, block3, 8u);
EEDigest::update(outhash, *((const unsigned __int8 **)v8 + 2)
result = (*(int (__fastcall **)(EEDigest *, _DWORD))(*(_DWORD
if ( outhash )
  result = (*(int (__fastcall **)(EEDigest *))(*(_DWORD *)out
return result;
}
```

# CVE-2020-25755 (Un-sanitized user input RCE)

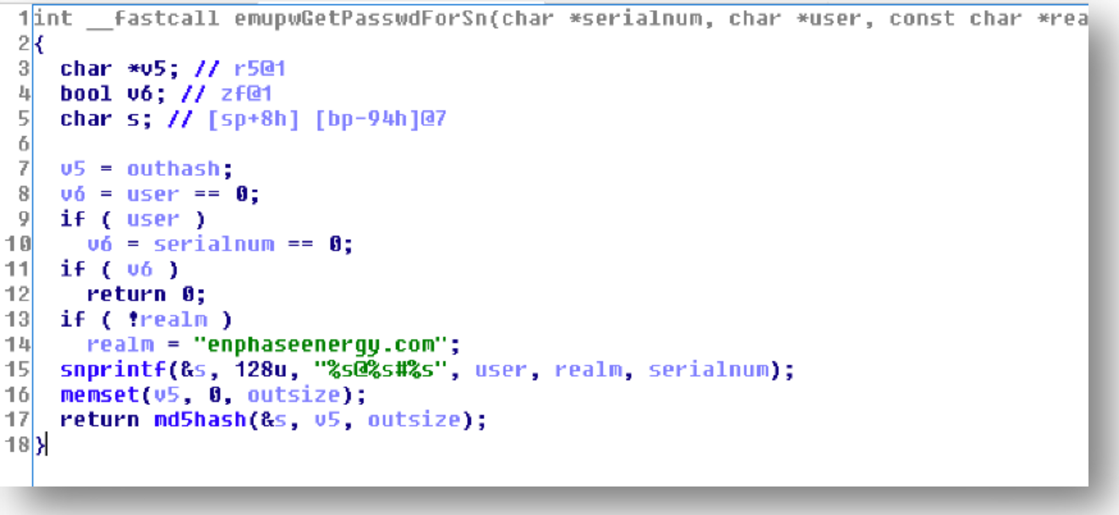


```
def start_upgrade()
  # Make sure we are not already running an upgrade (inspired by peb/scripts/

  running = `ps -axo command 2>/dev/null`.match('upgrade_start.rb')

  return update_mobile_status(TaskResponse.new('1208'), false) if running

  args = %w(0 mobile)
  args << "\"#{@cm.params['force'][0]}\"" if @cm.params.has_key?('force')

  status = system("upgrade_start.rb #{args.join(' ')} &")

  return update_mobile_status(TaskError.new('1501', *args)) unless status

  update_mobile_status(TaskResponse.new('1205'))
end

```

# Pam_emu.so



```
      if ( !strcmp(v9, "password") )
      {
        auth_type = 0;
      }
      else if ( !strcmp(v8, "http_digest") )
      {
        auth_type = 1;
      }
      else if ( !strcmp(v8, "verbose") )
      {
        dword_9E5C = 1;
      }
      else if ( !strcmp(v8, "public") )
      {
        auth_type = 2;
      }
      else if ( !strcmp(v8, "mobile") )
      {
        auth_type = 3;
      }
      else if ( dword_9E5C )
      {
        print_error(v22, "pam_emu: Unknown option '%s'", v8);
      }
      ++v7;
    }
    while ( v7 != _argc );
  }
  else
  {
    auth_type = 0;
  }
  result = pam_get_user(v22, &v36, "login: ");
```



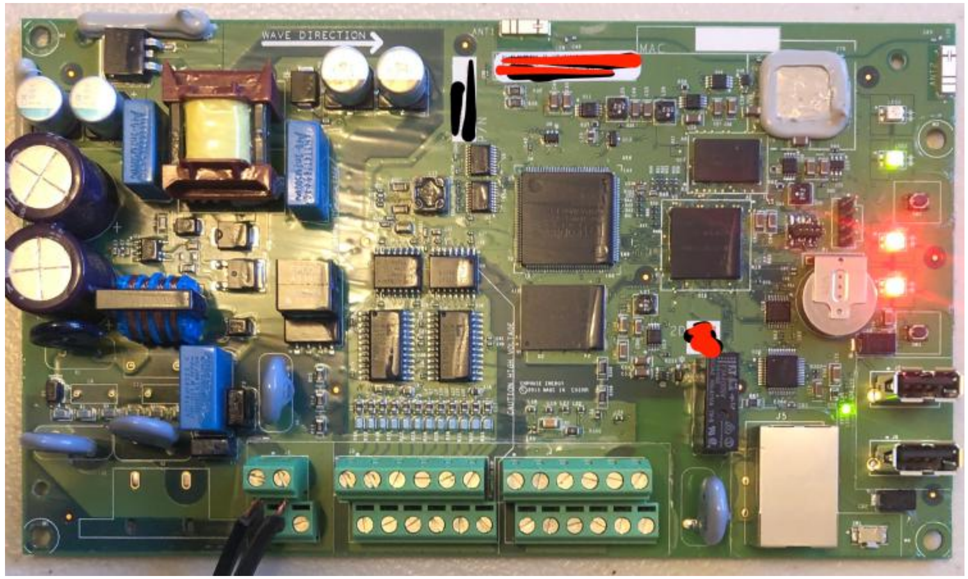
```
int __fastcall emupwGetPasswdForSn(char *serialnum, char *user, const char *rea
{
  char *v5; // r5@1
  bool v6; // zf@1
  char s; // [sp+8h] [bp-94h]@7

  v5 = outhash;
  v6 = user == 0;
  if ( user )
    v6 = serialnum == 0;
  if ( v6 )
    return 0;
  if ( !realm )
    realm = "enphaseenergy.com";
  snprintf(&s, 128u, "%s@%s#%s", user, realm, serialnum);
  memset(v5, 0, outsize);
  return md5hash(&s, v5, outsize);
}
```

# Multiple authentication issues from pam_emu.so

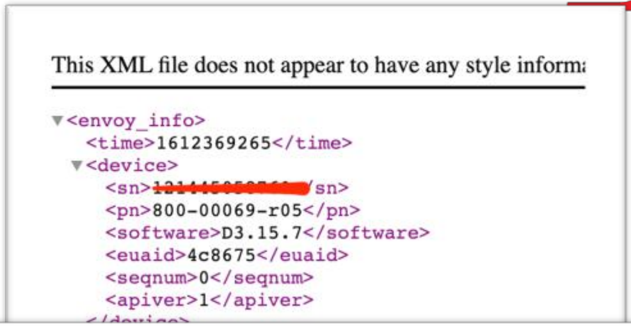
CVE-2020-25752, CVE-2020-25753, & CVE-2020-25754

- Built-in service accounts with weak passwords
- Hardcoded passwords to web administration page and SSH access
- Account passwords cannot be changed
- Account passwords are derived from username & serial number

# Secure Key Materials

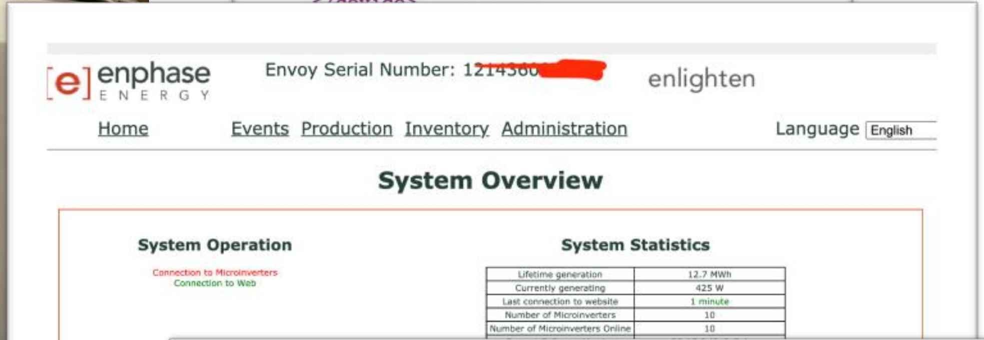


[redacted]
[e] enphase
ENERGY
ENPHASE COMBINER, 3 BRANCHES WITH ENVOY-S
CAT 5 CABLE (OPTIONAL)
ENVOY
120VAC RECEPTACLE FOR ENPHASE



This XML file does not appear to have any style informa

```
▼<envoy_info>
   <time>1612369265</time>
  ▼<device>
     <sn>[redacted]</sn>
     <pn>800-00069-r05</pn>
     <software>D3.15.7</software>
     <euaid>4c8675</euaid>
     <seqnum>0</seqnum>
     <apiver>1</apiver>
```



[e] enphase ENERGY  Envoy Serial Number: [redacted]  enlighten

Home  Events  Production  Inventory  Administration  Language English

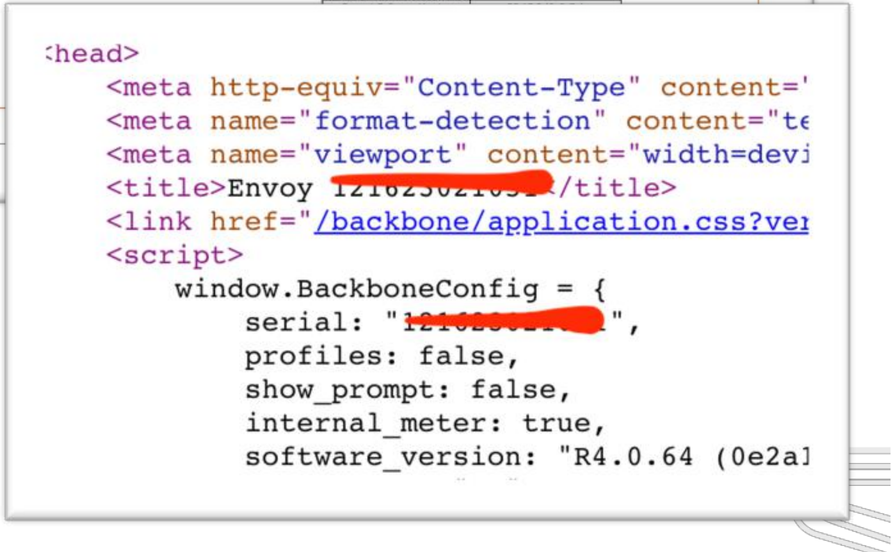
## System Overview

**System Operation**

Connection to Microinverters
Connection to Web

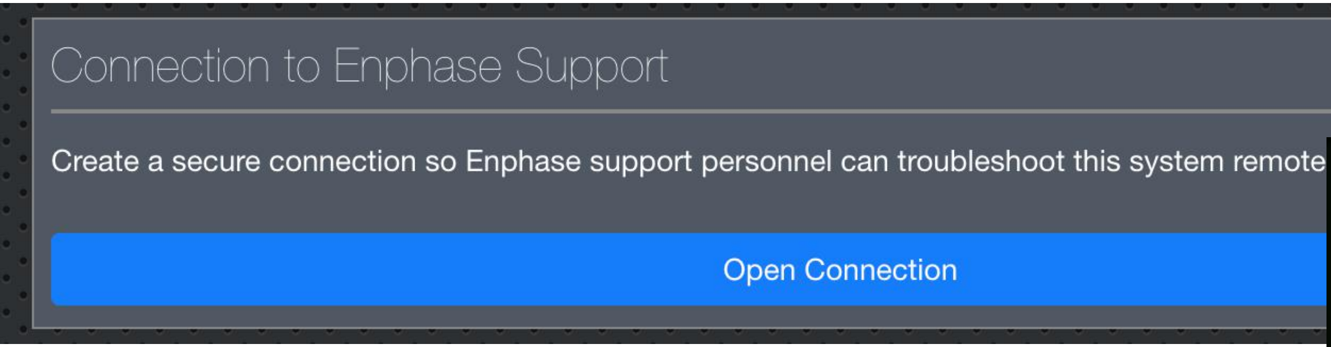
**System Statistics**

| | |
|---|---|
| Lifetime generation | 12.7 MWh |
| Currently generating | 425 W |
| Last connection to website | 1 minute |
| Number of Microinverters | 10 |
| Number of Microinverters Online | 10 |



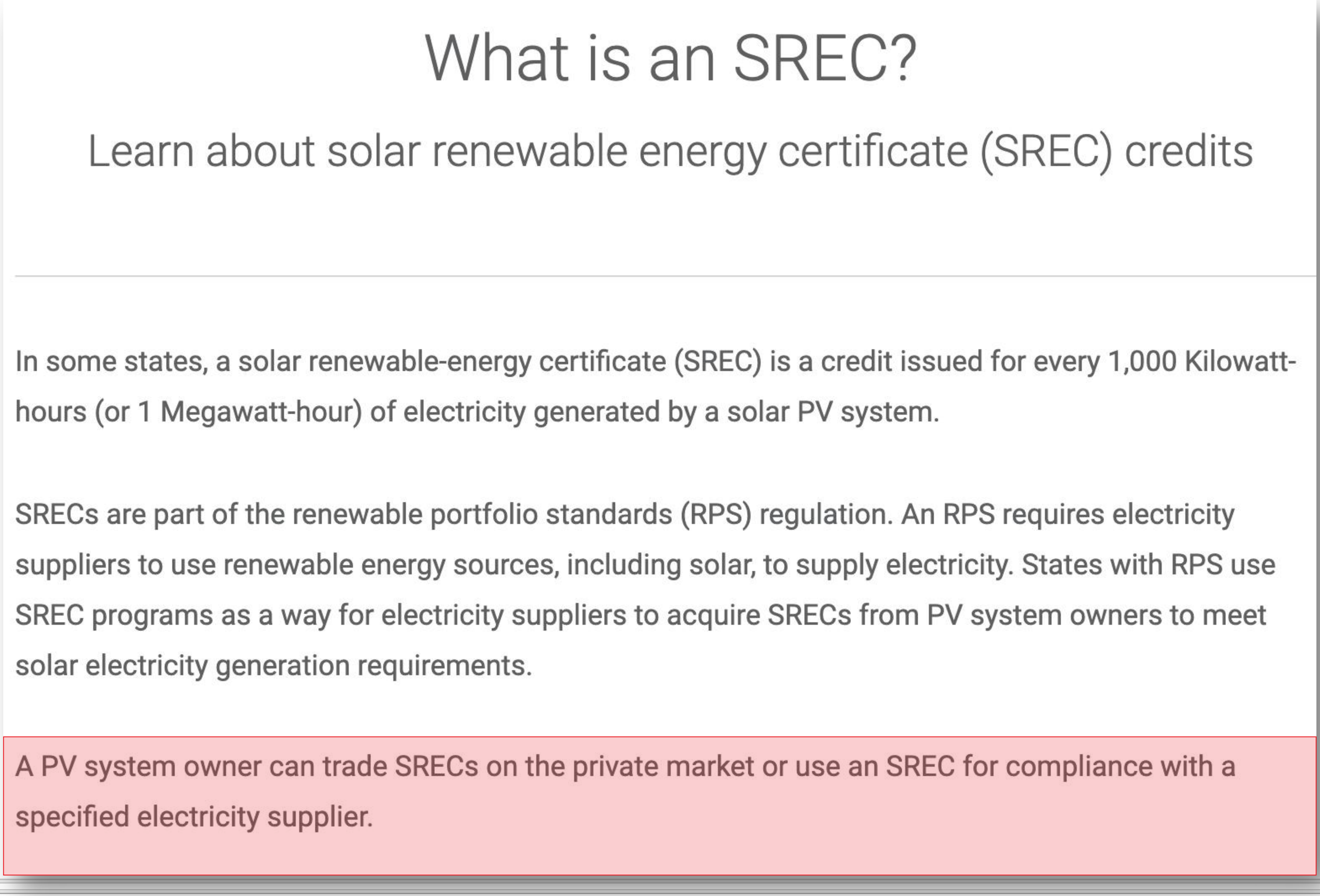
```
<head>
   <meta http-equiv="Content-Type" content='
   <meta name="format-detection" content="te
   <meta name="viewport" content="width=devi
   <title>Envoy [redacted]</title>
   <link href="/backbone/application.css?ver
   <script>
      window.BackboneConfig = {
         serial: "[redacted]",
         profiles: false,
         show_prompt: false,
         internal_meter: true,
         software_version: "R4.0.64 (0e2a1
```

# VPN tunnel to...



Connection to Enphase Support

Create a secure connection so Enphase support personnel can troubleshoot this system remote



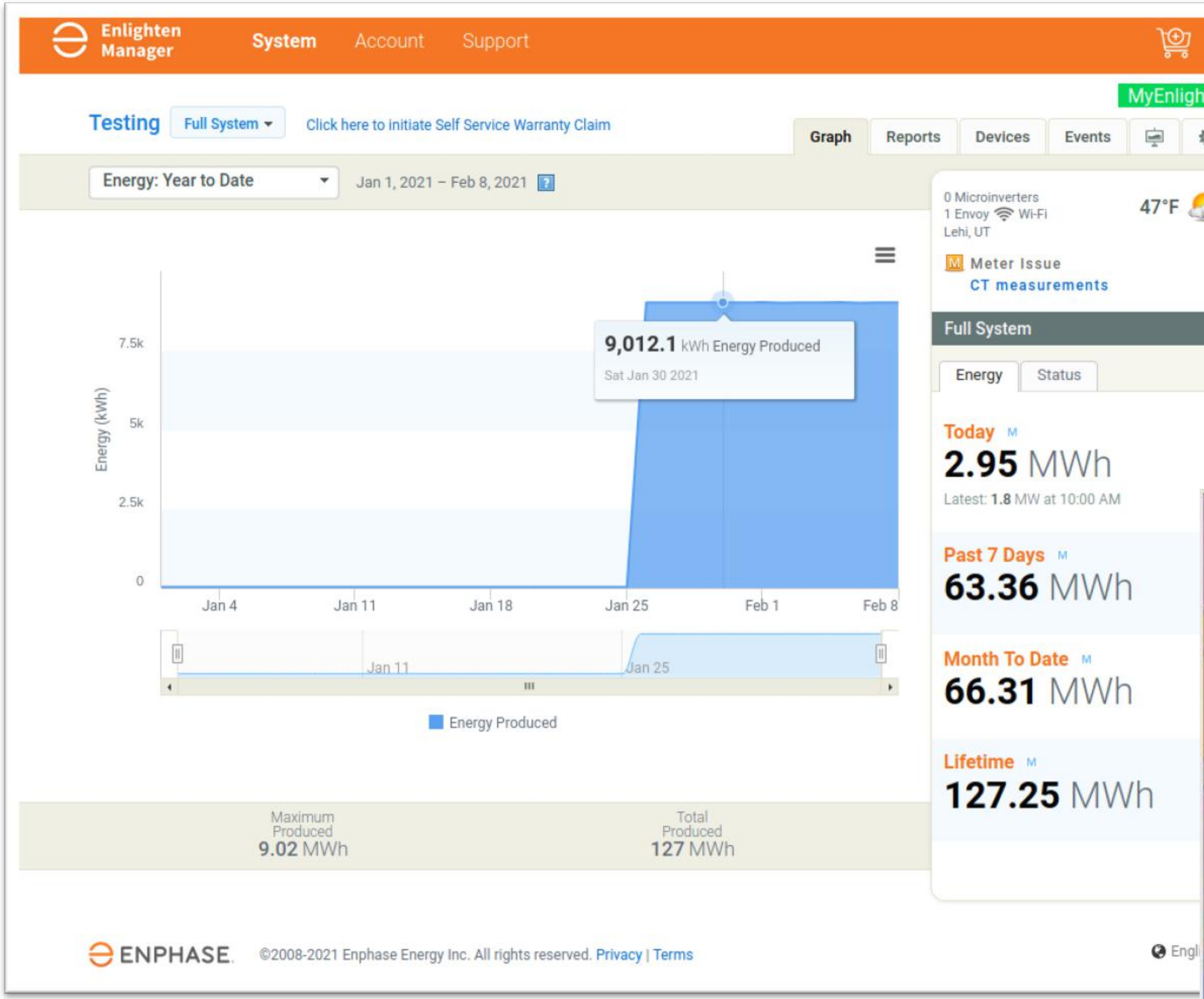
# What is an SREC?

## Learn about solar renewable energy certificate (SREC) credits

In some states, a solar renewable-energy certificate (SREC) is a credit issued for every 1,000 Kilowatt-hours (or 1 Megawatt-hour) of electricity generated by a solar PV system.

SRECs are part of the renewable portfolio standards (RPS) regulation. An RPS requires electricity suppliers to use renewable energy sources, including solar, to supply electricity. States with RPS use SREC programs as a way for electricity suppliers to acquire SRECs from PV system owners to meet solar electricity generation requirements.

A PV system owner can trade SRECs on the private market or use an SREC for compliance with a specified electricity supplier.



Enlighten Manager System Account Support

MyEnlighten View

Testing Full System ▾ Click here to initiate Self Service Warranty Claim

Graph Reports Devices Events

Energy: Year to Date ▾ Jan 1, 2021 – Feb 8, 2021 ?

Maximum Produced 9.02 MWh

Total Produced 127 MWh

0 Microinverters
1 Envoy Wi-Fi
Lehi, UT

47°F

M Meter Issue
CT measurements

Full System

Energy Status

Today M
2.95 MWh
Latest: 1.8 MW at 10:00 AM

Past 7 Days M
63.36 MWh

Month To Date M
66.31 MWh

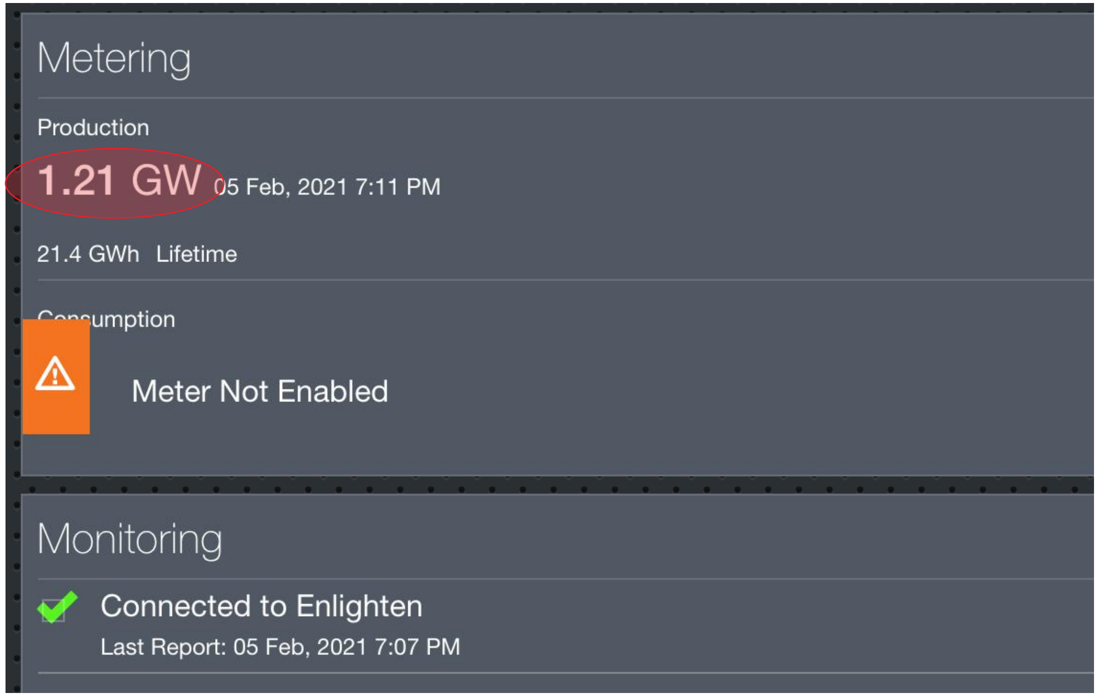
Lifetime M
127.25 MWh

Feedback

ENPHASE. ©2008-2021 Enphase Energy Inc. All rights reserved. Privacy | Terms

Engl

# Why stop there?



Metering

Production

1.21 GW 05 Feb, 2021 7:11 PM

21.4 GWh Lifetime

Consumption

⚠ Meter Not Enabled

Monitoring

✓ Connected to Enlighten

Last Report: 05 Feb, 2021 7:07 PM

# Apply

- Embedded hardware security is improving as an industry but still needs improvement
- Hardware attacks are becoming more difficult as more hardware includes options to disable debug access
- Never NEVER roll your own crypto
- Encrypted, signed, and limited access to firmware images goes a long ways

# Thank you!

Questions, Ideas, Quotes?

Waylon.Grange@stage2sec.com
@professor__plum
https://stage2sec.com