Global Information Assurance Certification Paper

DOCSIS Cable Modem Vulnerabilities and Countermeasures

A Service Provider Examines the Matthew S. Hallacy Exploit

Raymond J. Spreier
30 April 2002
GSEC certification, online course
v1.3 12 December 2001
Re-submission

Summary

In early 2002, a message was posted to two email discussion lists operated by SecurityFocus.com by Mr. Matthew Hallacy[1]. Mr. Hallacy, a software engineer living in Minnesota, was not satisfied with the level of service he was receiving from his local ISP, and decided to hack his own modem, ostensibly to prove to his provider that poor network management, and not his modem, was the bottleneck in the poor service he was receiving. The posted message provided detailed instructions as to how to exploit various vulnerabilities in a *DOCSIS*-compliant cable modem in order to develop and download your own configuration profile to the cable modem. The implication was clear that end users could bypass the configuration process as established by their broadband internet provider, and therefore enhance or enable features beyond those for which they were paying. This, of course, constitutes theft of service (which Mr. Hallacy explicitly acknowledged in his posting).

This paper will provide an overview of the vulnerabilities that were exploited by Mr. Hallacy, provide an overview of the hybrid fiber-coax (HFC) network and cable modem termination system (CMTS), the cable modem initialization process, and will outline some simple measures that can be implemented to help protect against tampering and unauthorized use. References are made throughout the paper to specific products, such as the Motorola *Surfboard* series of modems, the *Cisco Universal Broadband Router 7200VXR (uBR),* and the *Cisco Network Registrar*. The exploit described below is based on the modem configuration process as set forth in the *Data Over Cable Service Interface Specification* (DOCSIS)[2] v1.0 standard established by Cable Television Laboratories, which is the most widely implemented standard for providing high speed internet service over cable television networks. Therefore, references to key steps of the modem initialization process and references to most SNMP parameters should be applicable to most DOCSIS-compliant modems. Configuration parameters specific to the features of a given manufacturer or model will vary. This paper uses the Motorola Surfboard 4100[3] cable modem as an example.

Threat Detail

Mr. Hallacy outlined four basic steps required to exploit the modem configuration process:

[1] *DOCSIS Vulnerability*; Posting to the BugTraq board at SecurityFocus.Com; 11 March 2002
[2] http://www.cablemodem.com/specifications.html
[3] http://www.gi.com/noflash/sb4100.html

1. Use SNMP or the built-in HTTP resources of a legitimate, known-good modem on the provider network in order to determine the IP address of the provider's TFTP server.
2. Determine the name of the configuration file that the modem attempts to download as part of the initialization process.
3. Generate a modified DOCSIS configuration file, usually edited to allow for greater bandwidth allocation and an increased allowance in the number of client hosts (CPE's) that can be connected to the ethernet interface.
4. Use a spoofing technique to get the cable modem to download the altered configuration file from a local TFTP server.

Determining the Address of the TFTP Server and Configuration File Name

Mr. Hallacy pointed out that most manufacturers ship cable modems using default SNMP community strings of *public* and *private*. This allows any user to perform an SNMP *walk* to obtain a list of OID's and values in the device. Some products, such as the Motorola Surfboard, actually implement a web interface on the device at http://192.168.100.1, which makes it easy for users to view the configuration of the modem in order to obtain the DHCP server address and TFTP filename.

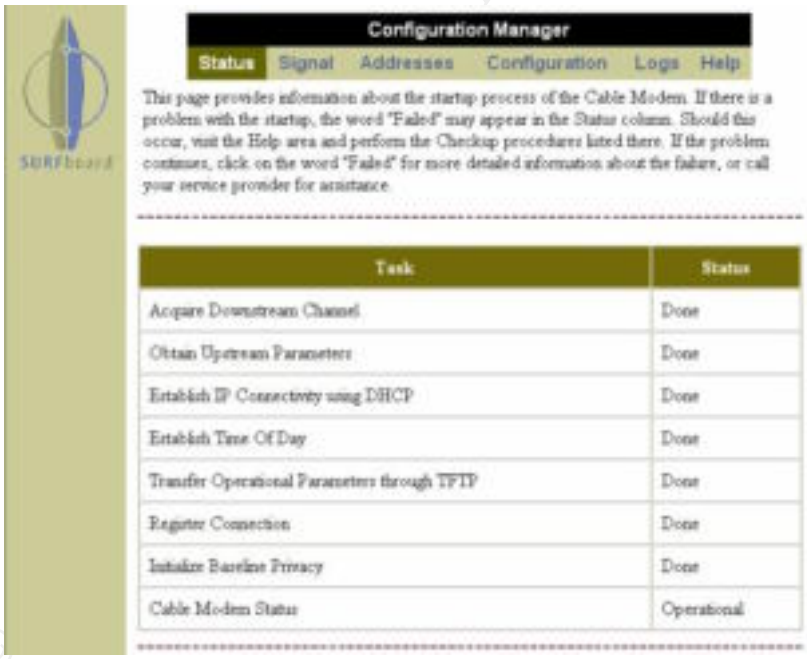


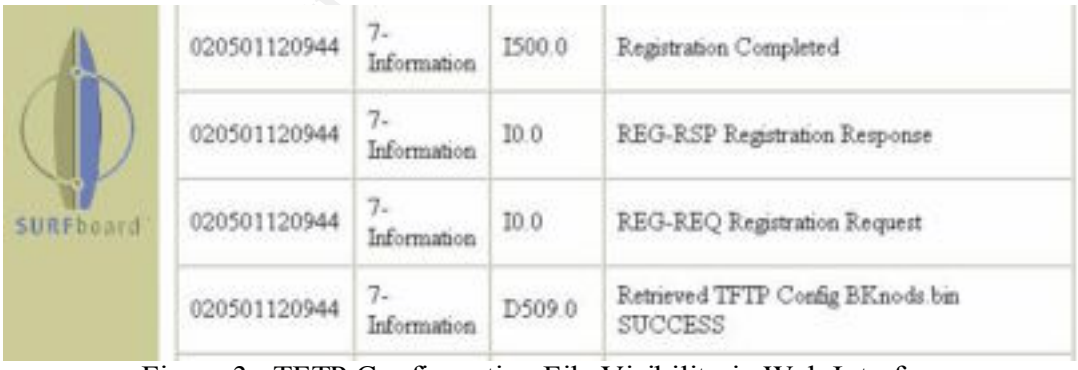
Figure 1 : Motorola Surfboard web interface at http://192.168.100.1

Clicking on the *Addresses* tab will bring up the window shown in figure 2 below. Note that the IP address of the DHCP server is plainly available.

Item	Value
Serial Number	1019011349561478020230000
HFC IP Address	10.5.0.13
HFC MAC Address	00:04:BD:C6:07:B2
Ethernet IP Address	192.168.100.1
Ethernet MAC Address	00:04:BD:C6:07:B3
CM USB IP Address	192.168.100.1
CM USB MAC Address	00:04:BD:C6:07:B3
CPE USB MAC Address	00:04:BD:D2:D1:49
DHCP Server Address	192.168.100.100
DHCP Information	Duration: 5286398 s Time: -28800

Figure 2 : DHCP Server Address Visibility in Web Interface

Finally, clicking on the *Logs* tab will allow the user to scroll through the log file and transcript of the initialization and registration process. In figure 3 below, note that the name of the downloaded TFTP configuration file is plainly available.



020501120944	7-Information	I500.0	Registration Completed
020501120944	7-Information	I0.0	REG-RSP Registration Response
020501120944	7-Information	I0.0	REG-REQ Registration Request
020501120944	7-Information	D509.0	Retrieved TFTP Config BKnods.bin SUCCESS

Figure 3 : TFTP Configuration File Visibility in Web Interface

Creating the DOCSIS Configuration File

Mr. Hallacy pointed out[4] that most providers use the DHCP server as the TFTP server. If not, it is usually a simple matter to examine the event log in the modem to determine the IP address from which the modem is attempting to download the configuration file.

Once the user has obtained the IP address of the TFTP server the modem searches for, and knowing the name of the TFTP file the modem will attempt to download, the next step is to create a DOCSIS modem configuration file with the altered values.

DOCSIS configuration file editors are readily available, such as the *DOCSIS CPE Configurator v 3.6* from Cisco (http://www.cisco.com) or the *DOCSIS Configuration File Creator, version 0.7.5* by Cornel Ciocirlan, available at http://docsis.sourceforge.net/.

The SourceForge DOCSIS Configuration File Creator is a set of simple, command line driven utilities that comes complete with example configuration files and SNMP management information base (MIB) files. With a rudimentary knowledge of the SNMP OIDs and a text editor, it is a simple matter to identify and modify key modem configuration parameters.

In the example below, taken from a generic DOCSIS configuration file template, the *ClassOfService* block contains the elements that govern the bandwidth allocated to the modem.

```
ClassOfService {
     ClassID          1;
     MaxRateDown      512000;
     MaxRateUp        64000;
     PriorityUp       3 ;
     GuaranteedUp     32000;
     MaxBurstUp       54314;
     PrivacyEnable    1;
}
```

By editing the *MaxRateDown*, *MaxRateUp*, and *GuaranteedUp* values (in bytes), a user can assign the modem a greater bandwidth allocation than usually permitted by the CMTS. One tricky aspect of altering the ClassOfService block is the ClassID field. This is a unique identifier by which the CMTS keeps track of the *quality of service* (QOS) profiles. Most CMTS systems have several pre-defined profiles that are used during the initialization and registration process. In addition, the provider will likely have defined one or more profiles that correspond to the classes of service offered to subscribers. An individual attempting to create a 'new' profile will need to select an unused ClassID.

The other commonly altered object is the *MaxCPE*. This is an integer value that governs the number of hosts (CPE's) allowed at the subscriber location on the ethernet side of the cable modem.

[4] *DOCSIS Vulnerability*; Posting to the BugTraq board at SecurityFocus.Com; 11 March 2002

An example of the successful creation of a QOS profile and how that is manifested on the CMTS is provided later in this paper under *QOS Restrictions*.

Once the configuration file has been edited, the next step is to compile it into a binary configuration file image using the *DOCSIS Configuration File Creator* utility, and then save it to a local system under the filename that will be requested during the TFTP phase of modem initialization. In our example, the filename would be the *BKnods.bin* as determined above in figure 3.

Downloading the Altered Configuration File to the Modem

The next step of the process outlined by Mr. Hallacy involves spoofing the modem into thinking that a local computer is the TFTP server, thereby sending the altered file to the modem during the registration and initialization process.

Mr. Hallacy first disconnected the RF network from the cable modem and configured a local computer with the IP address of the TFTP server that he had obtained by examining the cable modem. He then ran a local TFTP server daemon and had the altered configuration file ready to download under the filename that would be requested, also determined from examining the modem. He then pinged the modem repeatedly during the initialization process so that the modem ARP table would register his local MAC address with the IP address associated with the TFTP server. He then reconnected the RF network, whereupon the modem would begin the initialization process. When the modem reached the TFTP stage of the process (see below), chances were that the modem would connect to the local system and download the altered configuration file. Upon successful download of the altered file and completion of the modem initialization process, the modem would be ready to go with the altered parameters.

Summary of the Threat

Mr. Hallacy took advantage of several vulnerabilities in the DOCSIS cable modem, and particularly in his provider's implementation of DOCSIS service:

- Visibility of both the TFTP server IP address and the configuration filename via the modem's built in web interface.
- Ability to SNMP-walk the modem on the client ethernet side of the modem.
- The provider had deployed the cable modem without checking or altering the SNMP community strings.
- The provider had not hardened the modem or the CMTS to prevent unauthorized tampering with the modem or other boot helper resources (such as DHCP and TFTP).

Mr. Hallacy's actions were relatively benign in that he only modified parameters to enhance his own class of service. A more malicious intruder could alter configuration files to set the standard class of service policy to a minimal bandwidth, for example, and upload them back to the provider's TFTP server, potentially affecting every subscriber on the network.

The remainder of this paper provides a basic overview of a typical DOCSIS network and the modem initialization process, and then provides some examples of countermeasures that providers can take to protect their cable modems and network against unauthorized use such as that demonstrated by Mr. Hallacy.

Overview of the Hybrid Fiber-Coax (HFC) Network

In order to identify at what physical points the cable modem may be vulnerable to exploitation, it is important to understand the nature of the network (figure 4) to which the modem is connected.

Most commercial MSO's (multisystem operators) providing cable modem service do so over a hybrid network consisting of fiberoptics and coaxial cable (HFC). This same network infrastructure is used to deliver analog and digital video from a centralized broadcast location, called the *headend*, to the subscriber location.

At the subscriber location, a traditional CATV coaxial cable provides the connection between the network and the RF interface on the cable modem. DOCSIS (the *Data Over Cable Service Interface Specification*), an industry standard developed by Cable Television Laboratories (CableLabs), defines the requirements for the high speed transmission of data between the headend and the cable modem[5]. The most widely implemented version of DOCSIS at this time is DOCSIS 1.0. A traditional 10bT ethernet or USB connection provides the connection from the cable modem to the subscriber's equipment.

With respect to securing the physical HFC network, three issues are worth noting:

- In the case of a legitimate subscriber, the cable modem is usually provided by the MSO to the subscriber, and is either rented or owned by the end user. Nonetheless, the modem is still managed by the MSO, and depends on the CMTS for proper configuration every time it is booted. It is rare for an MSO to encourage the use of user-provided cable modems, not only due to the business aspects involved, but from a standpoint of having to manage the resulting hodge-podge of configuration parameters. At the very least, the provider has to contend with the situation of provider-owned or provider-managed equipment placed out at the subscriber's site,. Accordingly, the cable modem and its associated interfaces are the obvious points of physical vulnerability for potential tampering or theft of service. Managing access to the ethernet and RF interfaces of the cable modem will prove to be a key element in implementing a sound security policy.
- Key aggregation points, such as coaxial splitters, amplifiers, and media converters, are distributed throughout the entire service area, and must be secured against unauthorized access. This has traditionally been addressed by the use of locked pedestals and equipment cages.
- In a traditional HFC architecture, all IP communication in a given node must pass through the CMTS. Commercially available CMTS systems, such as the Cisco uBR series or the Motorola Cable Router, are, in fact, traditional routers with additional

[5] http://www.cablemodem.com/specifications.html

broadband interface capabilities, and thus can be configured using access control lists and logic such as would be applied in traditional router configurations.

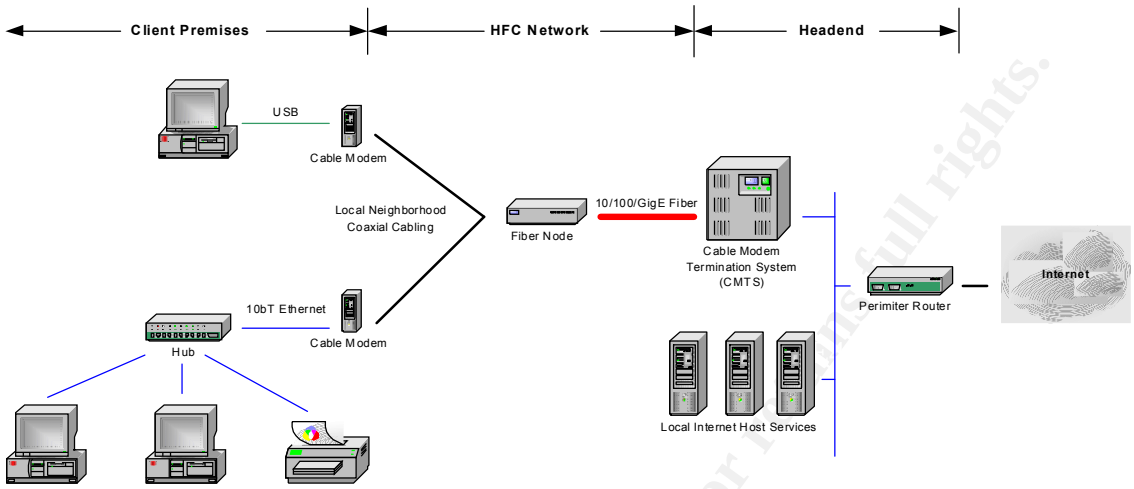


Figure 4 : Overview of the Broadband HFC Network

Overview of the Cable Modem Initialization Process

Having identified the physical points of vulnerability, it is then necessary to look at the cable modem initialization process. Examining this process will help us to understand precisely what parameters are being sent to the cable modem at what point during the initialization process in order to enable or restrict certain features. We can then take steps to secure the process itself and the associated data.

DOCSIS cable modem initialization on the network consists of two major processes:

- Establishing Communication on the RF network
- TCP/IP Device Configuration.

Establishing Communication on the RF Network

Understanding the process by which the cable modem is bootstrapped on the RF network calls for the IT professional to step 'out of the box' and into the world of the CATV RF engineer. The outline below provides a simplified list of the steps involved. A complete description of the DOCSIS initialization process can be found at the CableLabs website[6].

Downstream Acquisition : When the cable modem is first powered up, it searches for a downstream data channel – i.e., a specific frequency range from the overall CATV spectrum that has been allocated by the provider for modem downstream communication.

[6] http://www.cablemodem.com/specifications.html

SYNC : The modem then looks for a *SYNC* message, which is transmitted periodically by the CMTS. The CMTS communicates with all cable modems on the network using a token approach in which each modem is allocated shared access and bandwidth based upon a *'mini-slot'* of time. The SYNC message contains a time stamp that the cable modem uses to synchronize its own internal clock so that upstream transmissions will correspond with the mini-slots, which are dynamic and run between 5 and 15 ms in length.

UCD Configuration : Having SYNC'd, the cable modem then obtains the parameters for upstream communication, called upstream channel descriptors, or *UCD*'s.

Initial Ranging : The cable modem now begins the process of fine-tuning the RF connection. The CMTS send out a message describing a broadcast interval during which the modem can connect. The modem responds with a *ranging request*. The CMTS, in turn, sends back a *ranging response*. During this phase, the CMTS assigns a unique *system identifier* (SID), allocates bandwidth (time slots) to the SID, and auto-adjusts the power level in the modem, the timing offset, and the frequency. Downstream and upstream channels are also set at this point.

Admission : The CMTS puts the cable modem into the forwarding tables and sends out another broadcast to the SID indicating an opportunity for maintenance. The cable modem responds by ranging once again with the new settings. If successful, the modem SID is registered in the CMTS.

TCP/IP and Device Configuration

Once the cable modem has established RF configuration, it then begins the process of configuring the IP communication parameters.

DHCP Request : The cable modem sends out a DHCP request to the DHCP server via the CMTS. The IP address of the DHCP server (boot helper) has been defined by the provider in the CMTS configuration file. The DHCP server returns:

- IP address and subnet mask
- Cable modem configuration filename and IP address of the TFTP server from which the configuration file will be downloaded. This address is specified in the SIADDR field of the DHCP response.
- Local time offset from UT.
- IP address of the time of day (TOD) server.

TOD : The cable modem then sends out a request to the specified TOD server, which responds with the GMT.

TFTP : The cable modem downloads the configuration file from the specified TFTP server and applies the parameters contained in the configuration file. As we have seen, it is vital to secure the configuration file, in that it sets basic network access and configuration

settings, the class of service and resulting bandwidth, and the number of CPE's allowed on the ethernet side of the modem.

Registration : The cable modem then sends a registration request once again to the CMTS, using the parameters supplied by the configuration file. The CMTS then checks the modem MAC address, assigns a SID, and allocates the bandwidth to the SID for the requested class of service. The CMTS then modifies the forwarding tables to allow access to the network. At this point, the modem can pass IP data.

Countermeasures

Motorola recommends[7] that providers implement the following DOCSIS protection mechanisms:

- Use a combination of access control lists (ACL's) in the CMTS and cable modem IP filters to restrict access to the network TFTP server.
- Implement use of *shared-secret* password to ensure that only provider authorized configuration files are implemented in the modem.
- Change the SNMP read-only and read-write community strings, and implement specific SNMP access table entries in order to restrict SNMP access to the cable modem itself.

In addition, providers using the Cisco uBR product in the CMTS can implement restrictions to prevent the cable modem from attempting to create its own *quality of service* (QOS) profile, and thus alter the bandwidth allocation associated with a particular class of service.

Restricting Access to the TFTP Server

The first step in protecting the TFTP server is to prevent the subscriber from directly accessing the server to download a configuration file. By using an access control list in the CMTS, the provider can deny access to access the TFTP server from the address range normally assigned to subscriber hosts.

In the following example, the assumptions are as follows:

- Provider TFTP server is located at 10.1.1.100
- Subscriber CPEs are issued addresses from 192.168.100.0 / 24 pool.
- The network interface on the CMTS system is FastEthernet0/0.
- "Outfilters" is the name of the ACL you wish to create.

With this information, the provider could implement the following ACL on the CMTS router in the privileged configuration mode:

```
ip access-list extended outfilters
deny udp 192.168.100.0 0.0.0.255 host 10.1.1.100 eq tftp
```

[7] *DOCSIS Cable Modem Security - Motorola Surfboard Technical Bulletin STB# 02-007*; 19 March 2002; Motorola Broadband Communications Sector.

```
permit ip any any
```

The first line defines the ACL under the name "outfilters".
The second line explicitly denies access from the subscriber network to the TFTP server.
The third line explicitly allows IP traffic out of the HFC cloud to the network.

You would then apply this ACL to the inside network interface on the CMTS :

```
interface Fastethernet0/0
ip access-group outfilters out
```

The commands above are those used on the Cisco uBR product. Similar commands are available on most manufacturers' CMTS.

The second step in protecting the TFTP server is to disable TFTP traffic from passing through the cable modem. This will protect against the user who would attempt to assign a CPE address from the same pool that is normally assigned to modems in an attempt to bypass the ACL outlined above.

Using a DOCSIS modem configuration editing tool, enter or edit the following SNMP OIDs and values[8]:

```
docsDevFilterIpStatus.1     (1.3.6.1.2.69.1.6.4.1.2.1)  Integer 4  (create and go)
docsDevFilterIpControl.1    (1.3.6.1.2.69.1.6.4.1.3.1)  Integer 1  (discard)
docsDevFilterIpIfIndex.1    (1.3.6.1.2.69.1.6.4.1.4.1)  Integer 0  (on all interface)
docsDevFilterIpDirection.1  (1.3.6.1.2.69.1.6.4.1.5.1)  Integer 3  (bidirectional)
docsDevFilterIpBroadcast.1  (1.3.6.1.2.69.1.6.4.1.6.1)  Integer 2  (false)
docsDevFilterIpSaddr.1      (1.3.6.1.2.69.1.6.4.1.7.1)  IP Address 0.0.0.0 (any src)
docsDevFilterIpSmask.1      (1.3.6.1.2.69.1.6.4.1.8.1)  IP Address 0.0.0.0
docsDevFilterIpDaddr.1      (1.3.6.1.2.69.1.6.4.1.9.1)  IP Address 10.1.1.100 (tftp)
docsDevFilterIpDmask.1      (1.3.6.1.2.69.1.6.4.1.10.1) IP Address 0.0.0.0
docsDevFilterIpProtocol.1   (1.3.6.1.2.69.1.6.4.1.11.1) Integer 256  (all protocols)
docsDevFilterIpSourcePortLow.1  (1.3.6.1.2.69.1.6.4.1.12.1) Integer 0
docsDevFilterIpSourcePortHigh.1 (1.3.6.1.2.69.1.6.4.1.13.1) Integer 65535
docsDevFilterIpDestPortLow.1    (1.3.6.1.2.69.1.6.4.1.14.1) Integer 69  (tftp port)
docsDevFilterIpDestPortHigh.1   (1.3.6.1.2.69.1.6.4.1.15.1) Integer 69
```

The above example uses a table index of "1". The above commands set the cable modem to discard TFTP packets (port 69), originating from any source port on any address, from passing through the modem to the TFTP server at 10.1.1.100. It is important to note that these filters are applied to the modem *after* the configuration file has already been downloaded from the TFTP server during the registration process.

[8] *DOCSIS Cable Modem Security - Motorola Surfboard Technical Bulletin STB# 02-007*; 19 March 2002; Motorola Broadband Communications Sector.

Use of Shared-Secret

Another method to protect the TFTP configuration file is to implement the use of the *shared-secret* string. The shared-secret string is simply a password that is used to restrict access to and editing of the configuration file. When the file is saved, the user is prompted to enter the shared-secret string. The string is not stored within the configuration file, but is used in calculating the file's MD5 checksum. Thus, even if an individual obtains a copy of the configuration file, the password cannot be determined from the contents of the file itself.

By attaching a shared-secret string to the configuration file, and by assigning the same shared-secret string to the HFC interface on the CMTS, only modems that have downloaded the "official" configuration file from the provider's TFTP system can fully register on the network.

In the Cisco uBR CMTS, the command to implement the shared-secret string is:

```
cable shared-secret <string>
```

Restricting SNMP Access to the Cable Modem

In order to prevent unauthorized SNMP examination and tampering with the modem on the network, the provider can (a) change the default SNMP community strings, and (b) implement an access list to restrict the trusted hosts that may use SNMP to access the modem.

In the example below, the following assumptions are used:

- We wish to use *alpha* as the community string for read and get functions.
- We wish to use *omega* as the community string for write operations.
- We wish to restrict SNMP to the RF interface only.
- We wish to only allow SNMP from the trusted hosts in the 10.1.1.0 network.

Motorola recommends[9] that the following edits be made to the modem configuration files using a DOCSIS configuration file editing utility.

```
docsDevNmAccessIp.1         (1.3.6.1.2.1.69.1.2.1.2.1)  Ip Address 10.1.1.0
docsDevNmAccessIpMask.1     (1.3.6.1.2.1.69.1.2.1.3.1)  Ip Address 255.255.255.0
docsDevNmAccessCommunity.1  (1.3.6.1.2.1.69.1.2.1.4.1)  Octet String alpha
docsDevNmAccessControl.1    (1.3.6.1.2.1.69.1.2.1.5.1)  Integer 2 (read and get)
docsDevNmAccessInterfaces.1 (1.3.6.1.2.1.69.1.2.1.6.1)  Octets 0x40 (RF IF only)
docsDevNmAccessStatus.1     (1.3.6.1.2.1.69.1.2.1.7.1)  Integer 4 (create and go)

docsDevNmAccessIp.2         (1.3.6.1.2.1.69.1.2.1.2.2)  Ip Address 10.1.1.0
docsDevNmAccessIpMask.2     (1.3.6.1.2.1.69.1.2.1.3.2)  Ip Address 255.255.255.0
docsDevNmAccessCommunity.2  (1.3.6.1.2.1.69.1.2.1.4.2)  Octet String omega
docsDevNmAccessControl.2    (1.3.6.1.2.1.69.1.2.1.5.2)  Integer 3 (writes and sets)
docsDevNmAccessInterfaces.2 (1.3.6.1.2.1.69.1.2.1.6.2)  Octets 0x40 (RF IF only)
docsDevNmAccessStatus.2     (1.3.6.1.2.1.69.1.2.1.7.2)  Integer 4 (create and go)
```

[9] *DOCSIS Cable Modem Security - Motorola Surfboard Technical Bulletin STB# 02-007*; 19 March 2002; Motorola Broadband Communications Sector.

The above example uses table indexes of "1" and "2". The first set of six OIDs sets up the restrictions on *read* and *get* operations (public community). Lines 1 and 2 setup the IP access restriction to allow access only from the trusted 10.1.1.0 network. Line 3 establishes the new community string of *alpha*. Line 4 identifies the community to which this string applies (read and get). Line 5 restricts SNMP access to the RF port only. Line 6 writes and implements the changes.

The second set of OIDs implements the same restrictions, only using the community string *omega*, and applying this to the private (write and set) community.

QOS Restrictions

In the Cisco uBR product, the *quality of service* (QOS) profile determines the class of service, and thus the bandwidth, available to a given modem. Specific profiles corresponding to a class of service are usually defined in the CMTS. However, as described above, in the registration process following TFTP, the modem can attempt to register on the network with a specific class of service.

The active QOS profiles on a Cisco UBR router can be viewed by issuing the command in the privileged mode:

```
show cable qos profile
```

The listing below shows the QOS profiles on a Cisco uBR 7200 series router. Note that profiles 1 through 4 are defined in the default CMTS configuration, and are required for the modem initialization process. Profiles 5 and 6 are defined by the provider and correspond to the classes of service available to the subscriber. Profile 5 denotes a class of service with 2Mbps downstream and 256Kbps upstream. Profile 6 denotes a class of service with 2Mbps downstream and 768 Kbps upstream.

Profiles 7 and 8, however show successful attempts by a user to register a cable modem on the network, creating a new QOS with a non-standard bandwidth allocation. The user creating profile 7 has set the modem to a 10 Mbps upstream bandwidth and a downstream bandwidth of 52 Mbs. The user in profile 8 has set a symmetric profile for 10 Mbps upstream / downstream bandwidth. It should be noted that there is a physical limitation to the bandwidth available to the modem, based on the modem physical limitations (38 Mbs in the case of the Motorola Surfboard 4100[10]) and the time slot availability of the CMTS. Setting a QOS value outside of these limits simply results in a maximum available / best effort speed.

[10] http://www.gi.com/modem/SB4100.pdf

```
ubr1#sho cable qos pr
ID  Prio Max       Guarantee Max        Max   TOS  TOS   Create  B     IP prec.
         upstream  upstream  downstream tx    mask value by      priv  rate
         bandwidth bandwidth bandwidth  burst                     enab  enab
1   0    0         0         0          0     0x0  0x0   cmts(r) no    no
2   0    64000     0         1000000    0     0x0  0x0   cmts(r) no    no
3   7    31200     31200     0          0     0x0  0x0   cmts    yes   no
4   7    87200     87200     0          0     0x0  0x0   cmts    yes   no
5   0    256000    0         2000000    0     0x0  0x0   mgmt    yes   no
6   0    768000    0         2000000    0     0x0  0x0   mgmt    yes   no
7   0    10000000  10000000  52000000   0     0x0  0x0   cm      no    no
8   0    10000000  0         10000000   0     0x0  0x0   cm      yes   no
```

The provider can isolate the user by entering the following command at the uBR command prompt in the privileged mode:

```
show cable modem | inc x   y
```

where x is the QOS profile ID and y is the number of active CPE's running behind the modem. There are three spaces between x and y. The resulting output will list modems matching the above criteria. (You may also receive false listings where there just happens to be three spaces between the two numerals you entered, so parse the list with caution).

```
Cable3/0/U4 75  online(pt) 3359  0.75  7   1   10.30.9.248  0020.40ab.9828
```

The fields listed present the following information:

Field 1 – The blade and HFC interface in the uBR router through which the modem is communicating.
Field 2 – The unique SID assigned to the modem.
Field 3 – The status of the modem. In this case, online with baseline privacy activated.
Field 4 – The distance from the uBR to the modem, measured using an internal time-tick metric.
Field 5 – The power of the modem signal, in dB.
Field 6 – The QOS profile under which the modem was registered.
Field 7 – The number of CPEs online behind the modem.
Field 8 – The IP address assigned to the modem.
Field 9 – The MAC address of the modem.

More importantly, the provider can restrict the ability of any cable modem to create a QOS profile at registration by entering the following command into the uBR configuration file:

```
no cable qos permission modems
```

The final step is then to then reset the renegade modems and force them to re-initialize, drawing an authorized QOS profile. This can be accomplished by issuing the following command at the uBR command prompt in the privileged mode:

```
clear cable modem <MAC Address> reset
```

Summary

This paper sought to address issues specifically associated with protecting the DOCSIS network from exploits such as those demonstrated by Mr. Hallacy. There are certainly other issues associated with protecting the subscriber and headend networks that an MSO should address as part of a defense-in-depth philosophy. Among these are:

- Not allowing unknown / unregistered modems to obtain a DHCP address on the network nor access the TFTP server. Most commercial boot-helper products, such as *Cisco Network Registrar*, allow the provider to register the MAC addresses of devices authorized to obtain DHCP and TFTP services. This will help in protecting against individuals who stand up "rogue" modems on the network.
- Implementation of DOCSIS baseline privacy so that CMTS / modem traffic is encrypted.
- Implementing proper perimeter security and protection of MSO hosts providing internet services to subscribers.
- A clear acceptable use policy that is part of the subscriber agreement.
- Active monitoring and periodic audits. With tools such as network query language (NQL) scripting, a provider can readily develop monitoring tools that proactively check for events such as the existence of modems running non-authorized QOS policies.

The release of DOCSIS 1.1[11] will provide additional features to enhance network security[12]. Chief among these are:

- The implementation of digital certificates at the modem level.
- Digital signatures attached to vendor-authorized code downloaded to the modem.
- Standardized event logging and a common event messaging format.
- Implementation of SNMP v3 privacy, authentication, and authorization (documented in RFC 2571-2576).

However, providers have made a major investment in deploying thousands of DOCSIS 1.0 compliant devices to their subscribers. Even though some devices are firmware-upgradeable, it will take a substantial commitment for any MSO to embark on a deliberate migration to a standardized DOCSIS 1.1 compliant feature set. Hence, it stands to be some time before the industry sees a widespread implementation of the security features provided via DOCSIS 1.1.

The DOCSIS modem vulnerability outlined in this paper is all too typical in today's networking environment. OEM cable modems are often deployed "as-shipped" from the factory in an effort to simply keep up with the increasing demand for higher speed broadband internet service. Many smaller MSO providers actually do not maintain their own internet services, but outsource them to larger entities, and simply connect their headend to the vendor's NOC located elsewhere

[11] http://www.cablemodem.com/specifications.html
[12] *DOCSIS Cable Modem Connection Process* – PowerPoint presentation from SCTE, Northwest Chapter meeting February 2002. Prepared by Motorola Broadband Communications Sector.

via a dedicated circuit. Those MSOs that do maintain their own provisioning systems often are hard pressed to allocate the time and the resources necessary to perform a thorough review of security practices. Finally, the geographic size of service areas and the sheer number of subscribers presents a daunting task to the IT manager attempting to secure the client network. The network examined in this paper covered an area of 65x40 miles, and consisted of over 7500 subscriber premise nodes. Nonetheless, the certain aspects of the HFC network architecture and the nature of the DOCSIS initialization process make it feasible for providers to implement some effective measures to secure the subscriber network. The Hallacy incident has hopefully raised provider awareness so that such steps will be undertaken. You can rest assured that savvy subscribers have taken notice.

References

Hallacy, Matthew S.; *DOCSIS Vulnerability*; Posting to the BugTraq board at SecurityFocus.Com; 11 March 2002; http://online.securityfocus.com/archive/82/261454

McWilliams, Brian; *Cable Modem Hacking Tips Uncapped Online*; (c) 2002 The Washington Post Company; http://www.newsbytes.com/news/02/175201.html

Mauro, Douglas R. and Schmidt, Kevin J,; *Essential SNMP*; O'Reilly & Associates; Sebastapol, California; (c) 2001

St. Johns, M. (editor); *RFC2669 - DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination*; The Internet Society; (c) August 1999; http://www.ietf.org/rfc/rfc2669.txt

St. Johns, M. (editor); *RFC2670 - Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS Compliant RF Interfaces*; The Internet Society; (c) August 1999; http://www.ietf.org/rfc/rfc2670.txt

Woundy, R; *RFC3083 - Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems*; The Internet Society; (c) March 2001; http://www.ietf.org/rfc/rfc3083.txt

DOCSIS Cable Modem Connection Process – PowerPoint presentation from SCTE, Northwest Chapter meeting February 2002. (c) Motorola Broadband Communications Sector.

DOCSIS Cable Modem Security - Motorola Surfboard Technical Bulletin STB# 02-007; 19 March 2002; (c) 2002 Motorola Broadband Communications Sector.

Security in DOCSIS-based Cable Modem Systems; CableLabs White Paper; (c) 2000-2002, Cable Television Laboratories. http://www.cablemodem.com/downloads/Security_in_DOCSIS.pdf

TFTP Configuration File Authentication- Motorola Surfboard Technical Bulletin STB# 02-004; 22 March 2002; Motorola Broadband Communications Sector.

Last Updated: July 28th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401**	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Indianapolis SEC401	Indianapolis, IN	Oct 09, 2017 - Oct 14, 2017	Community SANS