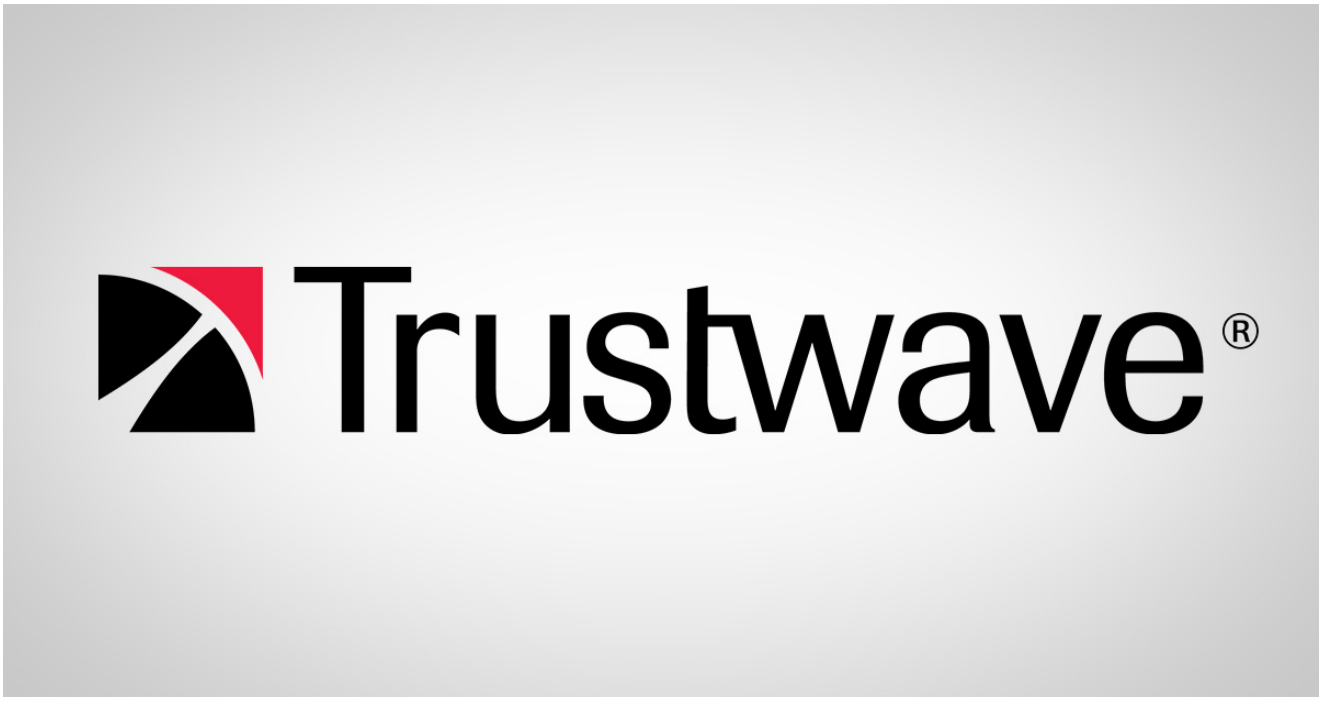
BrickerBot mod_plaintext Analysis



Blogs & Stories

SpiderLabs Blog

Attracting more than a half-million annual readers, this is the security community's go-to destination for technical breakdowns of the latest threats, critical vulnerability disclosures and cutting-edge research.

A week ago, the author of BrickerBot claimed that they retired and published their manifesto along with some source code of their bot.

In the manifesto, they wrote: "Take a look at the number of payloads, 0-days and techniques and let the reality sink in for a moment."

So I decided to take a look at the code and find those 0-days.

Breaking it down

The code "as is" can't be run, however we can see the payloads and techniques, so it is very likely those might be integrated into Mirai or other botnets. If there are indeed 0-days in the code, we could see a massive infection wave.

However, after analyzing the code, I was very disappointed to find only 1 "real" 0-day. Since it is Authenticated Remote Code Execution (RCE) vulnerability it shouldn't be that severe if you change default passwords.

The bot consists of 5 attack vectors:

- SSH crawler that the author didn't publish, because, and I quote "My ssh crawler is too dangerous to publish"
- Telnet crawler
- HTTP module that use some Exploits as well as automated authenticated requests with default passwords
- HNAP module that contains 1 exploit and Authenticated requests with default passwords
- SOAP module that exploits 3 vulnerabilities

Detailed analysis of the code

The main purpose of the bot is to brick devices so they won't be usable. "Bricking" is the process of changing the code of the device such that the hardware can no longer be used, thereby turning the device essentially into a brick.

Why make a bot that simply wrecks devices? According to the author every unsecure device is a potential target to be hacked and integrated into a botnet. This project was meant to illustrate that point and remove these potential targets from the reach of IoT botnets such as Mirai. The author never seemed to care that the unfortunate side effect of this process is that the device is also rendered useless for its owner.

The author achieves this goal of bricking devices by gaining shell access or remote command execution using exploits. After gaining access commands are run that write garbage into the partitions of the hacked device and overwrite the file system, leaving the device without operating system, in "bricked" state.

```
iiiIII = 'cat /proc/mounts\ncat /dev/urandom | mtd_write mtd0 - 0 32768\ncat /dev/urandom | mtd_write mtd1 - 0 32768\n'
iiiIII += 'busybox cat /dev/urandom >/dev/mtd0 &\nbusybox cat /dev/urandom >/dev/sda &\nbusybox cat /dev/urandom >/dev/mtd1 &\nbusybox cat
iiiIII += 'busybox route del default\ncat /dev/urandom >/dev/mtdblock0 &\ncat /dev/urandom >/dev/mtdblock1 &\ncat /dev/urandom >/dev/mtdbl
iiiIII += 'route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &\niptables -F;iptables -t nat -F;iptables -A
```

Fig 1: Code snippet of the bot overwriting the file system

The code is very device-specific and each crawler scans for specific targets that the author knows are vulnerable. For each known vulnerable device, they wrote a specific command set to brick the device, as each device shell and file system is different. This targeted process is also meant to reduce "noise" and avoid attacking honeypots.

Attack Vector: Telnet

Although the ssh crawler code wasn't published, it is most likely very similar to the telnet crawler. The telnet crawler looks for specific telnet banners and tries to login with default credentials. The list of devices that the telnet crawler attacks include the following devices and manufactures:

Ingenic devices
3Com Access Points
a5-v11
ADBGlobal
Alcatel OmniSwitch
Artila
Avaya
Aver DVR
Axerra
Bintec-elmeg
Broadcom based products
BusyBox based products
CalAmp Fusion LTE
Cell-technology Janus
Cisco
Comtrend
Dahua DVR
Dasan Networks
DASAN ZHONE SOLUTIONS
Davolink
Digitel NetRouter
D-Link
Elsist
EV ZLX Two-way Speaker
Extremenetworks
Fortigate
Freescale Semiconductor
Hikvision
HiLinux cam
HiSilicon
HooToo TripMate
HP Printers
HUAWEI
Idirect
Integrated Dell Remote Access
Intellisyn Intelliserver

ip-com
itwatchdogs
Juniper Networks
KingType Modem/Router
KYlink SIP
Maipu
Maxon Intelimax
Meritlilin
Microhardcorp Bullet-LTE
Mikrotik
Multiqb
Nateks
Netbox
NetScreen Technologies
Netvanta
Netween CCTV & cameras
Nomadix
OpenWRT
Oxygenbroadband
Phyhome
Polycom
Protei
Q-See DVR
Qtech
Quagga
Ricoh
Robustel
Ruckus
Sagemcom
Samsung Ubigate
Shanghai Telecoms E8
Sixpon
TrendChip Technologies
uClinux
UTTGlobal ReOS
Vigor
VXworks
Welotec
Westermo
Windows CE Telnet Service

Xiongmai DVR
ZTE
ZyXEL

Attack Vector: HTTP

The HTTP module includes exploits & techniques that are described by the author in their previous manifest. If they are unable to brick the device, they will restore it to factory default settings, shut it down, or change some configuration option to force the device offline.

One method of gaining access to perform these attacks is to simply try default passwords. BrickerBot uses this technique against the following HTTP devices:

Observa Telecom Devices
Hikvision
Sifytechnologies Devices
Zyxel p66
Realtron Cameras
Supernet ADSL Modems
PLDThome DSL/Fiber Devices
FosCam
Aztech
Mediatek Devices
Grandstream Devices

Other HTTP devices are attacked with RCE exploits to gain access:

AVTECH Multiple Vulnerabilities
WIFICAM Multiple vulnerabilities
Dahua Backdoor
ZTE ZXDSL 831
EnGenius RCE
CrossWeb DVR RCE
Hanbanggaoke IP Camera
WIFICAM cameras
D-Link DIR-600 / DIR-300 (Rev B) - Multiple Vulnerabilities
D-Link dsl-2750u ISP "backdoor" account
D-Link 850L Multiple Vulnerabilities
Unauthenticated command execution on Netgear DGN devices
NETGEAR R7000 / R6400 - 'cgi-bin' Command Injection
Vacron NVR Remote Command Execution
"JAWS" unbranded DVR
Ubiquiti AirOS 6.x - Arbitrary File Upload
Huawei B593 Authenticated RCE

The last one for Huawei routers is the only one you could consider a 0-day. Even though the vulnerability dates back to 2013, exploiting it on Huawei HG532 & HG532a model routers is a previously unknown attack vector. That said the attacker must be authenticated to exploit this vulnerability and, as we've already seen, authenticated access to the device is often all you need to brick it anyway.

Unfortunately, it is quite common in IoT devices to patch only a specific model against a known vulnerability while missing the exactly same vulnerability in other products. I myself stumbled upon this situation while finding a new vulnerability in NETGEAR routers.

Attack Vector: HNAP

The HNAP module uses an exploit for D-Link devices to achieve RCE.

It also tries to change network configuration to make the device unusable via authenticated requests with default passwords.

Attack Vector: SOAP

The SOAP module uses 3 exploits:

Eircom D-1000
Huawei CVE-2017-17215
Realtek CVE-2014-8361

Was this effective?

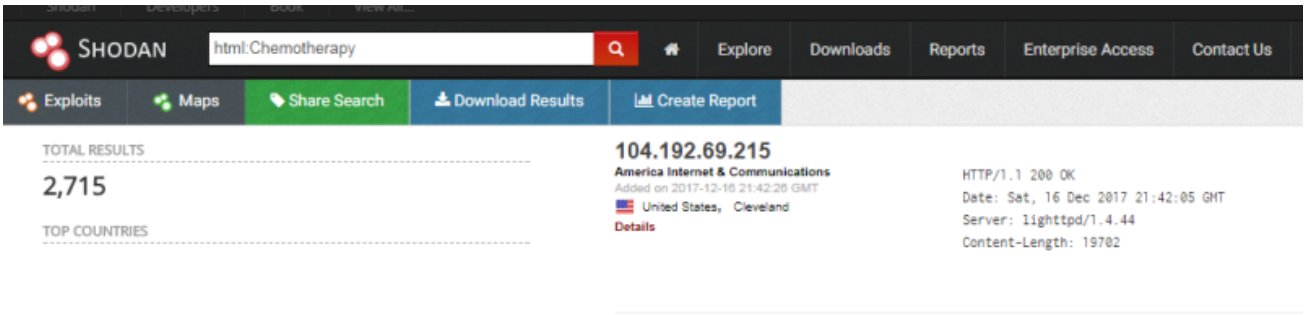
The author of BrickerBot claims they have bricked 10 million devices, but I believe this is an exaggeration.

Many of the devices they attacked can't be truly bricked and the author just restored them to default configuration, shut them down, rebooted them or changed configuration. This is more in line with a Denial of Service attack rather than a real "bricking".

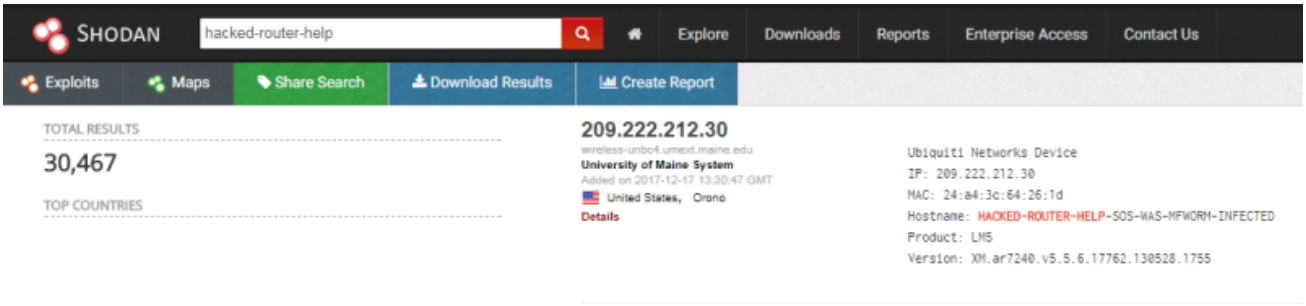
Once the owners of those devices restore them to working order, the bot probably repeats the same attack on the same target. So while the bot may have successfully reached 10 million hits, many of those hits were likely duplicates and didn't really brick those devices.

Even if we take a look at Shodan we can see that BrickerBot managed to attack quite a few devices successfully, but the numbers are far lower.

You can find that the author left their manifest on few thousand devices:



And there are also quite a lot Ubiquiti devices successfully breached:



What we can learn from this?

Whether or not you support this bot's approach to raising awareness in regards to IoT security, the issue is real and there are takeaways here for both manufacturers and consumers of IoT devices.

If you are a manufacturer of devices that are connected to the Internet: don't use default passwords, don't leave "backdoor" accounts, and don't run everything under root. If you are not sure about the security of the products you offer for sale, you might want to open a bug bounty program. Many hardware manufacturers have discovered that the payouts for disclosure via these types of programs are lower than the cost of the implications if your devices are being compromised by a 0-day.

If you are an end user who has one of the devices that are on the BrickerBot list, make sure to install the latest firmware if there is an exploit for your device and make sure you didn't leave any default passwords intact.

Another good, basic security process is to disable remote administration unless you specifically need it. If you do require remote administration I recommend the following:

If you are a more advanced user, set it up so that you connect to your local network via a more secure channel first such as VPN or an SSH tunnel and then connect to your device.

If the above is still too advanced or too much work, you should at least use a strong password and keep the device patched and up to date, You may also look at changing the settings to run the service on a random port so it would be harder to find. It's not the greatest of solutions, but it helps.

IoCs

Look for the following files in your device's filesystem:

/tmp/system/update/sentinel.reload
/tmp/system/control.cfg