Automating attacks against the Google Home assistant

Rodney Beede

BSides SATX 2020

Short Abstract

Research on the attack surface during boot/provision for Google Home assistant devices. Not dropping a 0-day but research study of ways to trick the home assistant to return to provision mode allowing an attacker to put the device on a wireless network they control.

Google Home, Alexa, other home assistants are making their way into the homes of people without the realization of how an attacker could abuse them. IoT security and privacy are areas of much needed improvement. This talk will cover research into behavior of the Google Home device and ways it can be forced remotely into provisioning mode for attempted manipulation via Wi-Fi. This isn’t a 0-day drop but will provide the attendee with knowledge of tested attack surfaces and the defenses for them.

Why this talk?

Power Outage at home

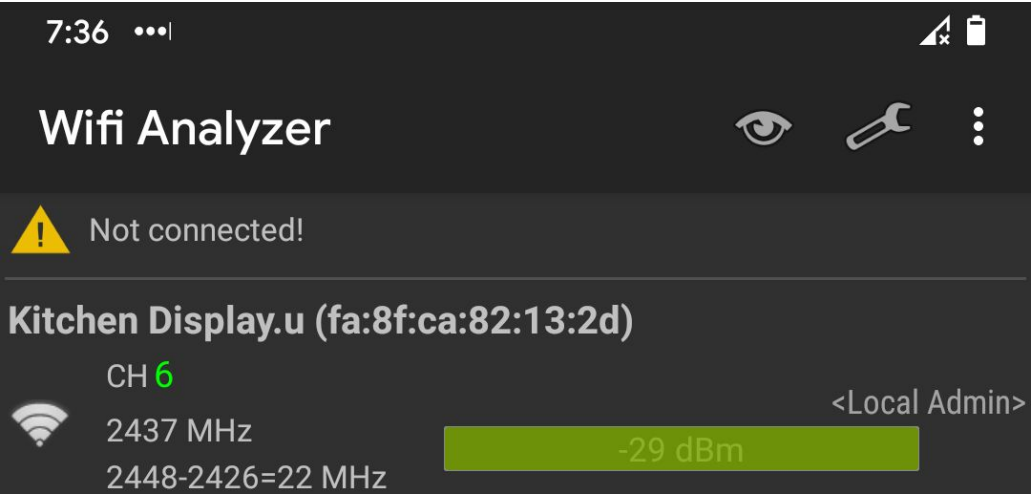
- Wi-Fi was on UPS
- Google Home Hub was not

Power comes back on few hours later

- UPS depleted and did not auto-switch wireless network back on
- Google Home Hub booted on up

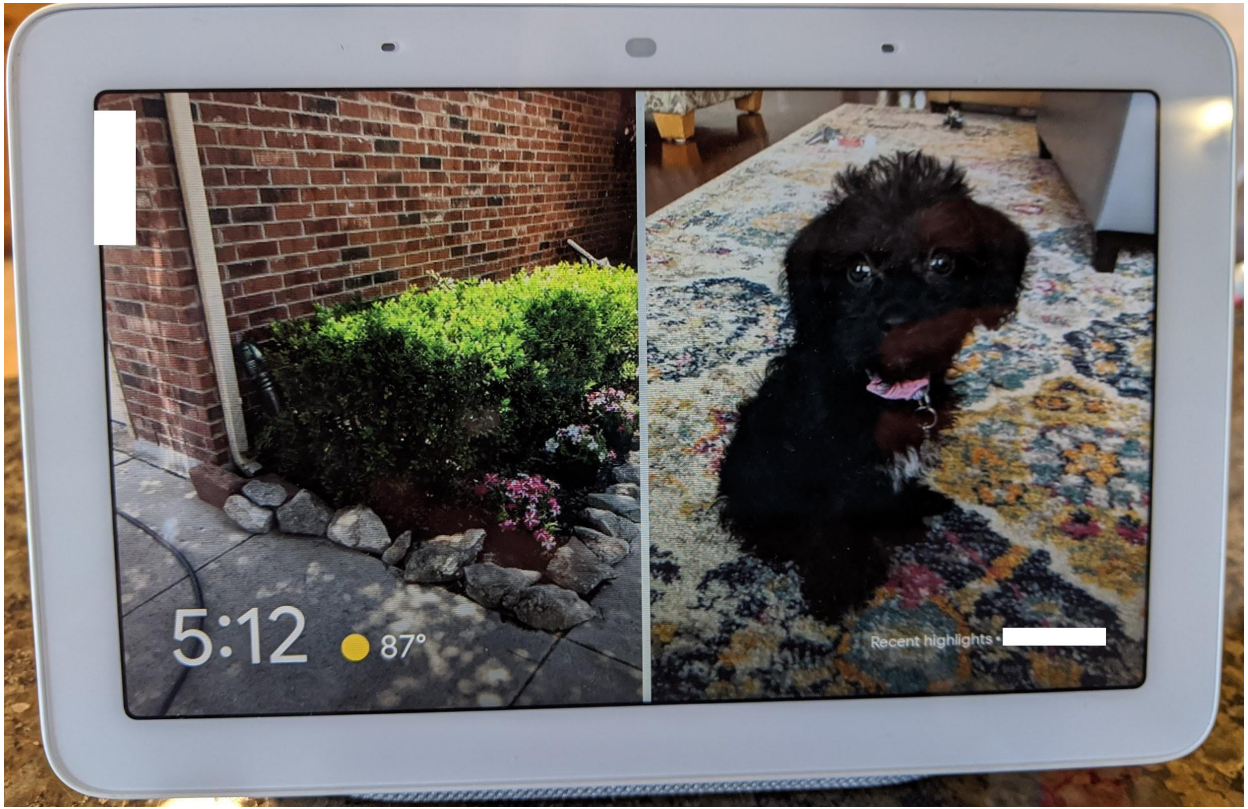
Noticed this behavior

No Wi-Fi



This value seems to be static

History of Google (aka Nest) Hub

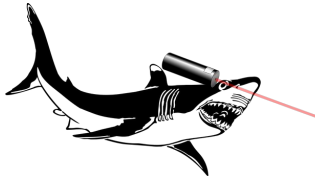


Previous Vulns

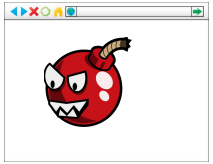
Smart Spies: Google Home Eavesdropping
https://www.youtube.com/watch?v=X2gdddqD1wUI&feature=youtu.be
https://srlabs.de/bites/smart-spies/

Phishing

Researchers hack Siri, Alexa, and Google Home by shining lasers at them
https://arstechnica.com/information-technology/2019/11/researchers-hack-siri-alexa-and-google-home-by-shining-lasers-at-them/



DEF CON 2019: Researchers Demo Hacking Google Home for RCE
https://www.defcon.org/html/defcon-27/dc-27-speakers.html#Qian



Google Questions Assertion That Google Home Hub Is Vulnerable To Remote Hacking
https://hothardware.com/news/google-home-hub-insecure

The APIs mentioned in this claim are used by mobile apps to configure the device and are only accessible when those apps and the Google Home device are on the **same Wi-Fi network**." November 2018
Also: https://www.androidauthority.com/google-home-hub-security-920291/

Filters

Can we observe traffic to determine client is Google Home device?

MAC PREFIX (**1C:F2:9A**:37:0E:4B)

SSID FOR SETUP

Kitchen **Display.u** (different mac of **fa:8f:ca:**82:13:2d)

Assuming encrypted Wi-Fi so can’t see any IP traffic, just Wi-Fi packets

Do the Wi-Fi packets give away any patterns?

ENTER MAC ADDRESS OR OUI (FIRST 6 DIGITS)

1CF29A370E4B lookup MA

SELECT LOOKUP TYPE: LOOKUP MAC LOOK

example: 00:0B:14

This database was last updated on 29 April 2020

Results for MAC address 1CF29A370E4B

Found 1 results.

MAC Address/OUI	Vendor {Company}
1CF29A370E4B	Google, Inc.

MAC Address and OUI Lookup

This program displays the name of the co

ENTER MAC ADDRESS OR OUI (FIRST 6 DIGITS)

FA8FCA82132D lookup M

SELECT LOOKUP TYPE: LOOKUP MAC LOO

example: 00:0B:14

This database was last updated on 29 April 2020

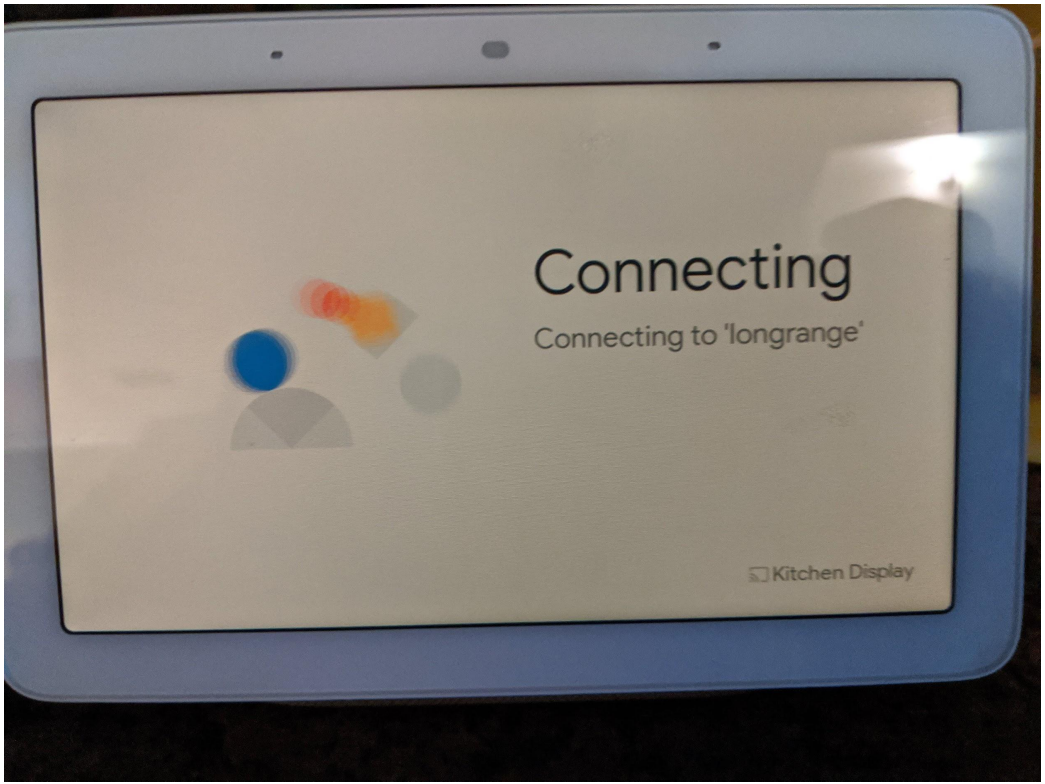
Results for MAC address FA8FCA82132D

Found 1 results.

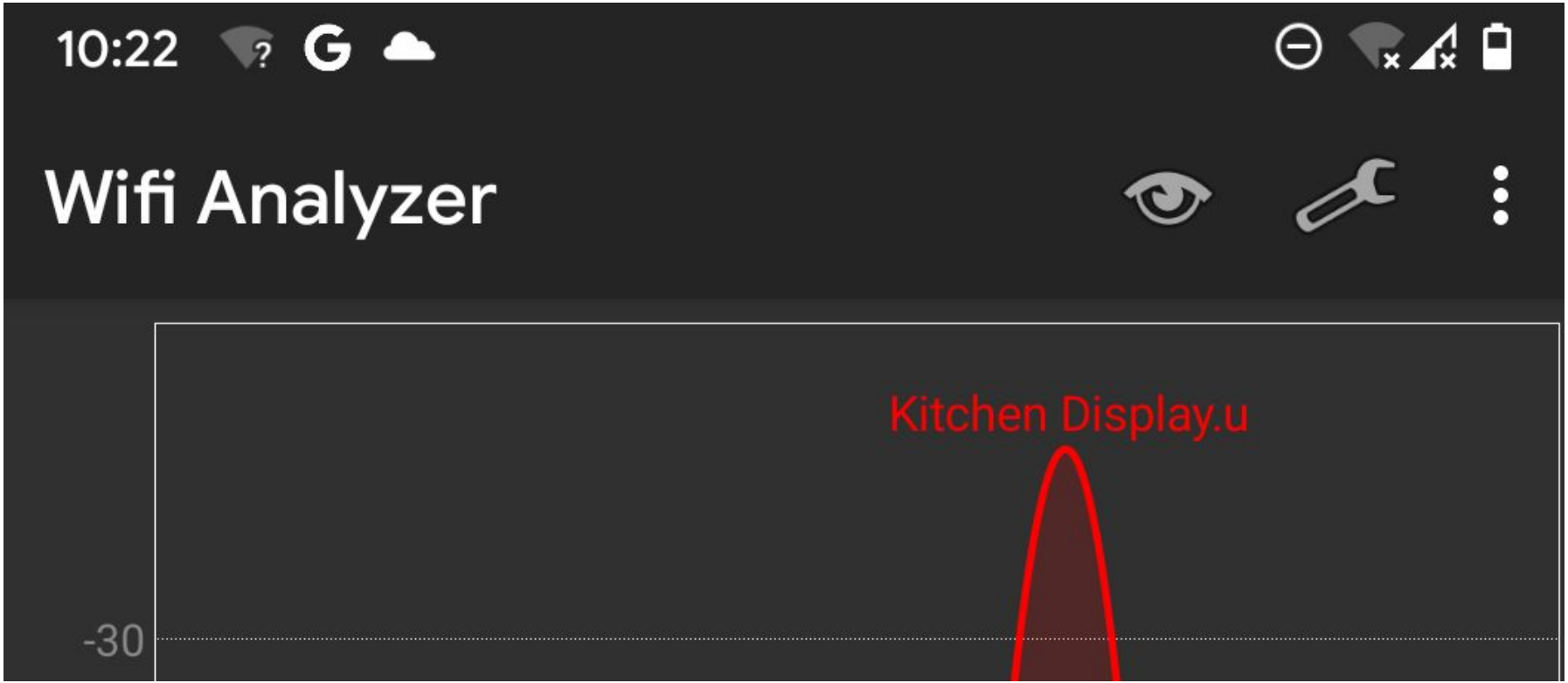
MAC Address/OUI	Vendor {Company}
FA8FCA82132D	Google Chromecast

DeAuth a device

```
.
Sent 1 packets.
Send deauth #158...
.
Sent 1 packets.
.
Sent 1 packets.
Send deauth #159...
```



Screenshot from Wi-Fi scanner



Got a Wi-Fi network, Now what?

What services are available in this mode?

nmap of normal operation (pre-deauth)

```
root@rbeede-virtual-machine:~# nmap -n -oA google_nest_home_hub-nmap-2020-06-13 192.168.0.3
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-14 19:14 CDT
Nmap scan report for 192.168.0.3
Host is up (0.018s latency).
Not shown: 735 filtered ports, 260 closed ports
PORT      STATE SERVICE
8008/tcp  open  http
8009/tcp  open  ajp13
8443/tcp  open  https-alt
9000/tcp  open  cslistener
10001/tcp open  scp-config

Nmap done: 1 IP address (1 host up) scanned in 26.28 seconds
```

nmap of provision Wi-Fi network

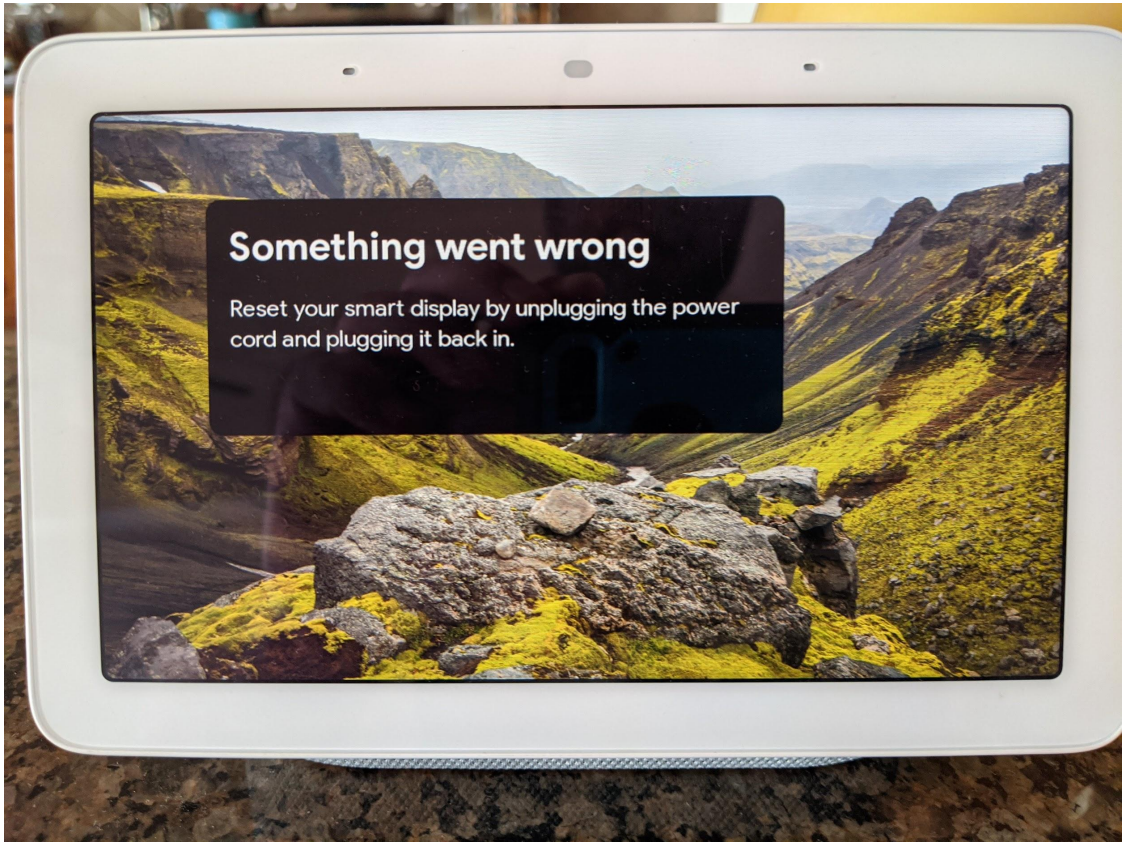
```
rbeede@rbeede-virtual-machine:~/Downloads$ cat route.txt
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.255.249 0.0.0.0         UG    20600  0        0 wlan0
192.168.255.248 0.0.0.0         255.255.255.248 U     600    0        0 wlan0
rbeede@rbeede-virtual-machine:~/Downloads$ cat nmap.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 21:16 UTC
Nmap scan report for 192.168.255.249
Host is up (0.021s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
8008/tcp  open  http
8009/tcp  open  ajp13
8443/tcp  open  https-alt
9000/tcp  open  cslistener
10001/tcp open  scp-config
MAC Address: FA:8F:CA:82:13:2D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 5.07 seconds
```

Interesting Observation

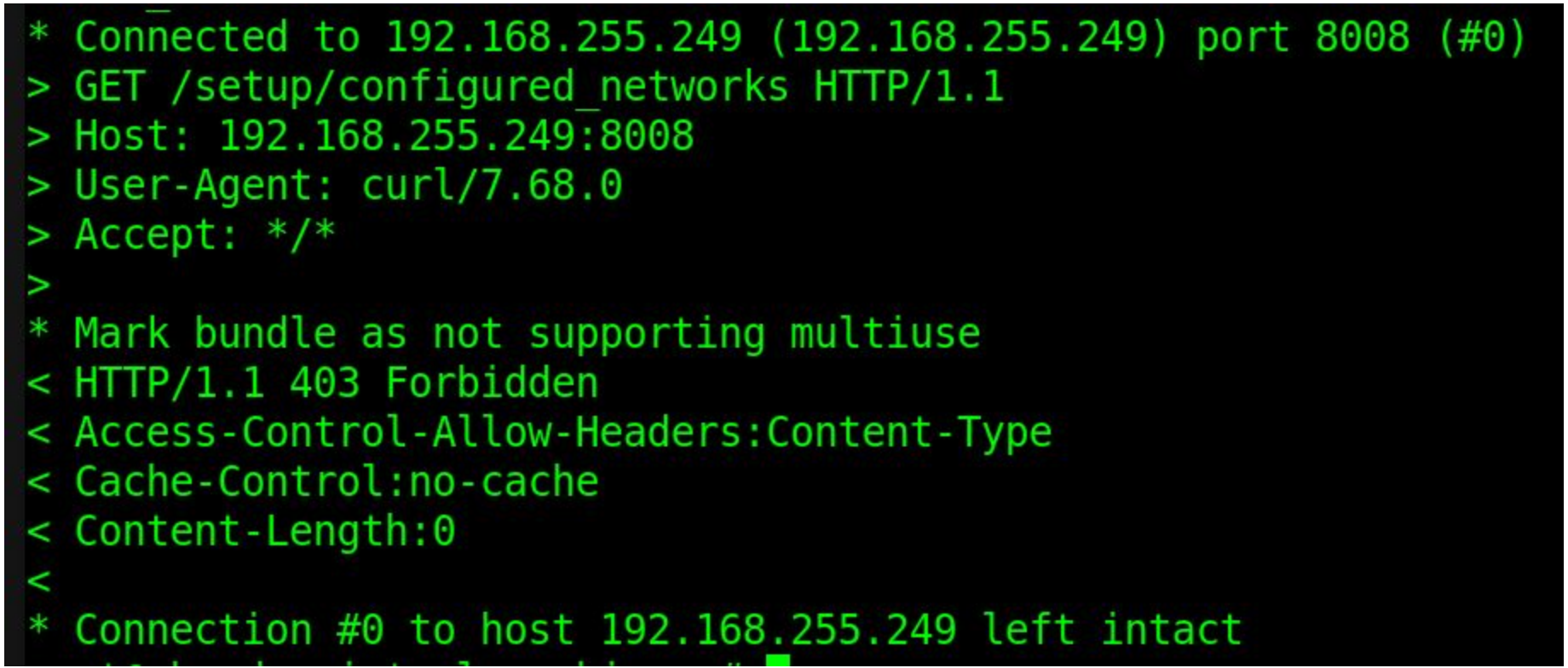
After running deauths for 1 hour the Google Home gave up on the legit Wi-Fi.

1. Stopped running deauths
2. Unplugged power
    a. Device Rebooted
3. Google Home wanted the mobile app to connect
4. Refused legit Wi-Fi network even though it was there

Still broadcasting Kitchen Display.u network



First access 403 Forbidden



https://www.androidpolice.com/2018/01/16/google-home-hidden-api-local-devices-can-use/

Thank you API reverse engineers!

```
root@rbeede-virtual-machine:~# curl http://192.168.255.249:8008/setup/eureka_in
fo; echo
{"bssid":"","build_version":"203796","cast_build_revision":"1.46.203796","close
d_caption":{},"connected":false,"ethernet_connected":false,"has_update":false,"
hotspot_bssid":"FA:8F:CA:82:13:2D","locale":"en-US","location":{"country_code":
"US","latitude":255.0,"longitude":255.0},"mac_address":"1C:F2:9A:37:0E:4B","nam
e":"Kitchen Display","opencast_pin_code":"","opt_in":{"crash":true,"opencast":t
rue,"stats":true},"public_key":"MIIBCgKCAQEAkxmlgYM8pqAsyl7xrJ1f4eH420mnZx05ijr
HTDAH+5k3NOL5kwnlFQEVS0K1HNGizfDZ1dBPX/Z7DfTSySBIPLVCVgjDCeKNHD/SgKRDp94z3F7UvW
hDdLWdDa/5esC8tbXJtIrSL0as+vsJHXbEDkRn9NgJxm9KdB9uOC/3FC5kEU8i+LO75jL/rsOzEe6+H
GtBfasxhYkomwZhhVu4LyQLrZH0YZCBsDFSBCOWWm4oWfRAV5OdrQeP2vYz3XSlheW2cKJVRw+J+r1k
NGhTBVZ6YrkxsmkSMMcPGyVGHJ2buAYNzmPF4/SF958oMQDB1fy6jYMlYj3VTbZiy3mKfwIDAQAB","
release_track":"stable-channel","setup_state":31,"setup_stats":{"historically_s
ucceeded":true,"num_check_connectivity":0,"num_connect_wifi":0,"num_connected_w
ifi_not_saved":0,"num_initial_eureka_info":0,"num_obtain_ip":0},"ssdp_udn":"d2d
c46ff-2d54-e2d6-a959-e9a153b5362f","ssid":"longrange","time_format":1,"timezone
":"America/Chicago","tos_accepted":true,"uma_client_id":"1ddf7ecf-e076-4d40-8f0
3-7631a89ed35a","uptime":135760.960366,"version":10,"wpa_configured":true,"wpa_
id":2,"wpa_state":4}
```

CREDIT https://rithvikvibhu.github.io/GHLocalApi/

Current State

- Hub will not connect to Internet
  - Without manual user intervention to fix
- Broadcasting setup/provision wireless network SSID
  - Wants user to connect mobile app to fix Internet problem
- Port 8008 is more locked down
  - June 2019 firmware update
    - Restricts unauth functionality
  - Before update could reboot, reset, etc. on device
    - Was pwnable

Video Demo

https://github.com/rbeede/BSidesSATX2020/tree/master/silent-demo-video

- Shows tool being run
- Result of device disconnecting
- Attacker connecting to provision network and querying API

Denied easy network hijacking change

```
root@rbeede-virtual-machine:~# curl -v -H 'Content-Type: application/json' -d
{"ssid": "hackersWifi"}' http://192.168.255.249:8008/setup/connect_wifi
*   Trying 192.168.255.249:8008...
* TCP_NODELAY set
* Connected to 192.168.255.249 (192.168.255.249) port 8008 (#0)
> POST /setup/connect_wifi HTTP/1.1
> Host: 192.168.255.249:8008
> User-Agent: curl/7.68.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 23
>
* upload completely sent off: 23 out of 23 bytes
* Mark bundle as not supporting multiuse
< HTTP/1.1 403 Forbidden
< Access-Control-Allow-Headers:Content-Type
< Cache-Control:no-cache
< Content-Length:0
<
* Connection #0 to host 192.168.255.249 left intact
```

Can’t even reboot without some token

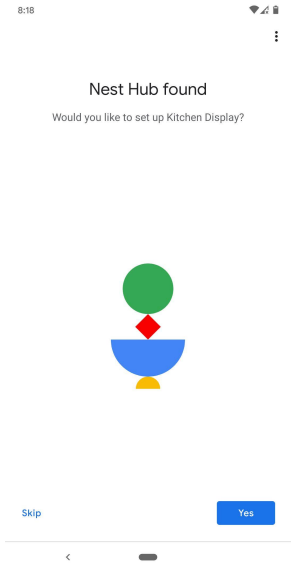
```
root@rbeede-virtual-machine:~# curl -v -H 'Content-Type: application/json' -d '
{"params": "now"}' http://192.168.255.249:8008/setup/reboot
*   Trying 192.168.255.249:8008...
* TCP_NODELAY set
* Connected to 192.168.255.249 (192.168.255.249) port 8008 (#0)
> POST /setup/reboot HTTP/1.1
> Host: 192.168.255.249:8008
> User-Agent: curl/7.68.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 17
>
* upload completely sent off: 17 out of 17 bytes
* Mark bundle as not supporting multiuse
< HTTP/1.1 403 Forbidden
< Access-Control-Allow-Headers:Content-Type
< Cache-Control:no-cache
< Content-Length:0
<
* Connection #0 to host 192.168.255.249 left intact
```

So Security Status

- Hardening by Google
  - An already setup device requires a token from the mobile app
- Prior
  - API was available over port 8008 with many functions
- This is a **good** thing for IoT security!
  - Falling into a provision state to fix Wi-Fi is only safe if attacker can’t reset device to factory settings or mess with it too
- What if the user lost their token (i.e. phone)
  - Physical access to reset device must be required for a factory reset

But is it really safe?

```
Sent 1 packets.
Found a matching provisioning mode AP
{'CHANNEL': 1, 'SSID': 'Kitchen display.u'}
Send deauth #140...
```



Remember that code?

Well even if attacker can't see it just say YES

Did you pwn it?

Found an attack surface

- Attacker using “Home” app can start provision process
  - Used a different google account that is NOT the legit users
- Device did accept using attacker’s Wi-Fi network
  - Since it could not connect to legit owner’s Wi-Fi anymore due to deauth
  - Meaning: I forced the device to connect to my Wi-Fi network with a different SSID
    - No evil twin required, able to precision target just the Home hub device
- But Home hub kept legit owners account, photos, etc.
  - Kicked my attacker “Home” app out and did not use my evil Google account
- Danger zone
  - Device is on Wi-Fi network controlled by attacker
  - Mitigating Control: the API (since June 2019) restricts calls
    - Only http://192.168.0.3:8008/setup/eureka_info is usable
  - Other APIs use TLS encrypted channels
  - But if future vuln/bug in API allows unauth attack?

Future Work

- https://github.com/rithvikvibhu/GHLocalApi/issues/39
  - Can an attacker generate a token without user's creds?
    - Not likely but script to create token with known creds exists
  - Any alternative ways to reset the device without the token?
- Smarter device searching
  - Just a MAC prefix filter also matches Chromecast and other devices
  - Need better fingerprinting before the deauth
  - Perhaps profiles of the encrypted Wi-Fi traffic?
  - How does the app do this?
- Does this work against other brands of smart home devices?
  - Do they protect the provisioning API better?
- What about the 1st time provision out of the box
  - Automate race to grab the device first?
- Update the scapy code to use two wireless devices
  - One for deauth, Other for provision network tampering
- Do any third party connections/apps make insecure requests?

References

- https://github.com/rbeede/BSidesSATX2020
- https://docs.google.com/presentation/d/1TC-WErW9naXuXSyRaIBLUCJYGtm69KuRLapNtox71CE
- Background photos care of
  - https://www.pxfuel.com/en/free-photo-epfpc
  - https://pixabay.com/vectors/attack-death-ray-evil-laser-menace-1294254/