Security analysis of Residential Gateways and ISPs – Global network domination is (sneakily) possible

Ta-Lun Yen
Senior Vulnerability Researcher,
TXOne Research

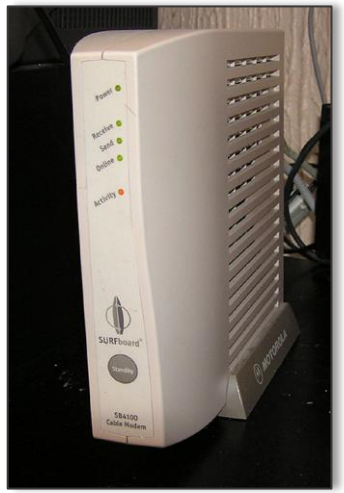
whoami

- Ta-Lun Yen (@logonfail)
- Vulnerability Researcher, TXOne Networks
  - Break Everything™

    (software & hardware, reverse engineering, embedded systems)
  - Various International InfoSec Conferences
  - Taiwanese hacker group "UCCU Hacker"

What is a Residential Gateway?

- Bridges premises to Internet
- Definition –
  - Modem **modulates** {fiber, coaxial, phone line} to/from Ethernet
  - **Residential Gateway** performs modem + computing
    - e.g. NAT, Firewall, Routing, DHCP
- Refers to many devices; **focusing on ones from ISP**



Why is RG important and worth studying into?

- 79% of household(*) has access to fixed internet (=has a RG)
- Gateway devices are lucrative targets for adversaries;
  - not yet RGs (ones by ISPs)

CHINA-LINKED APT GROUP SALT TYPHOON COMPROMISED SOME U.S. INTERNET SERVICE PROVIDERS (ISPS)

Pierluigi Paganini September 26, 2024

Washington, D.C.
FBI National Press Office
(202) 324-3691

X.com Facebook Ema

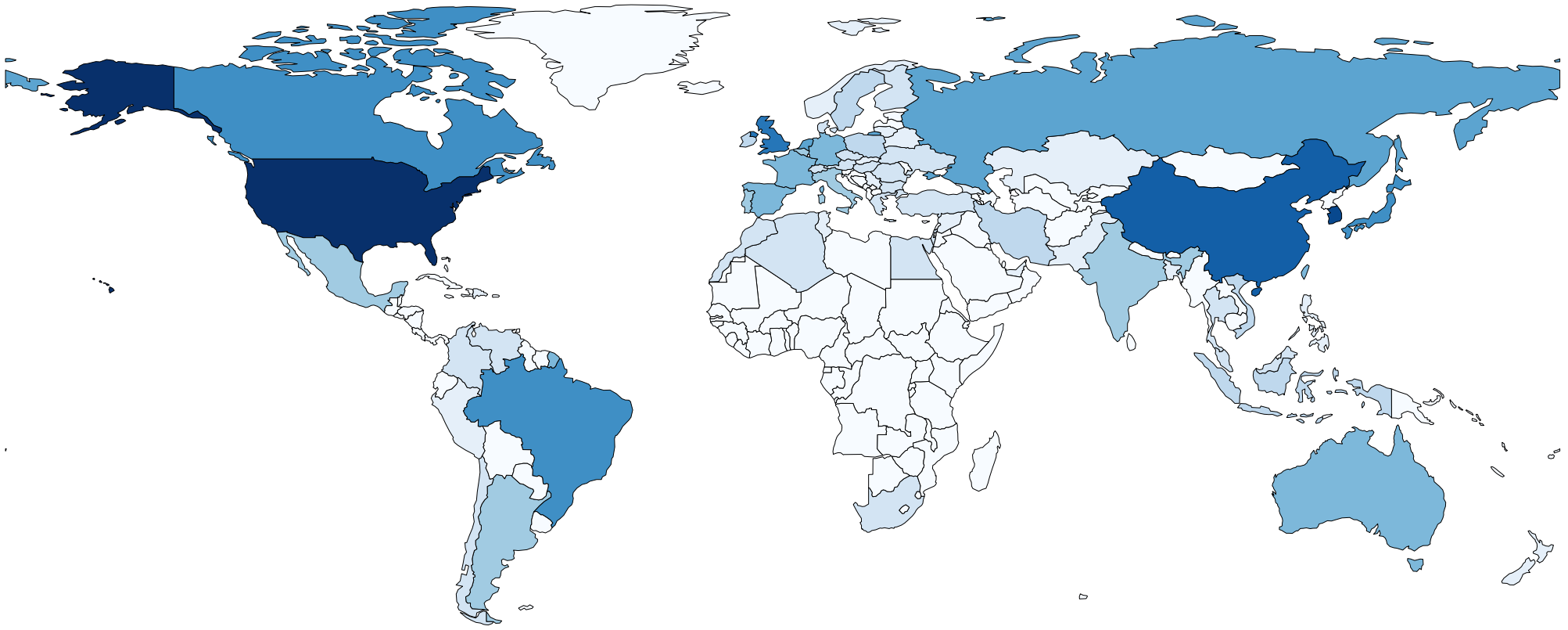
November 13, 2024

Joint Statement from FBI and CISA on the People's Republic of China Targeting of Commercial Telecommunications Infrastructure

(*) OECD ICT Access and Usage by Households and Individuals Database,
Household with fixed broadband Internet access at home
https://oe.cd/dx/ict-access-usage

Q: How many Residential Gateways (RGs) on Earth?

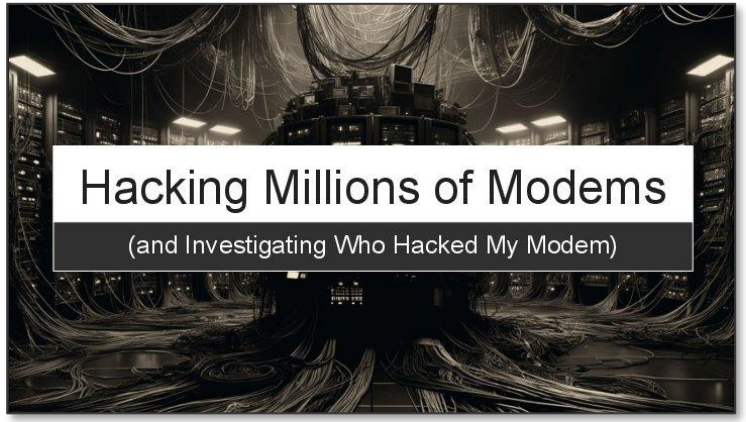
Answer: Could be at least 153 million (*)

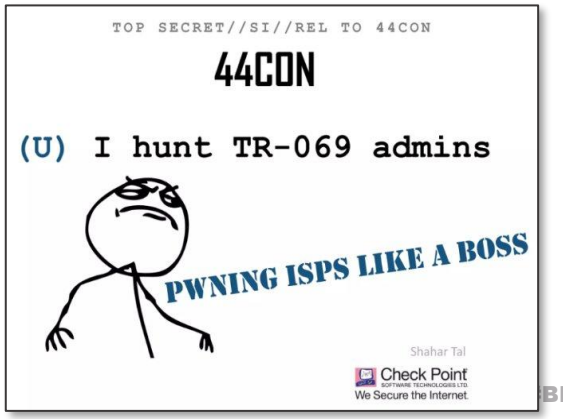


(*) Based on “Broadcom SDK Un-stealthy Stealth Mode”. Not accurate depiction.

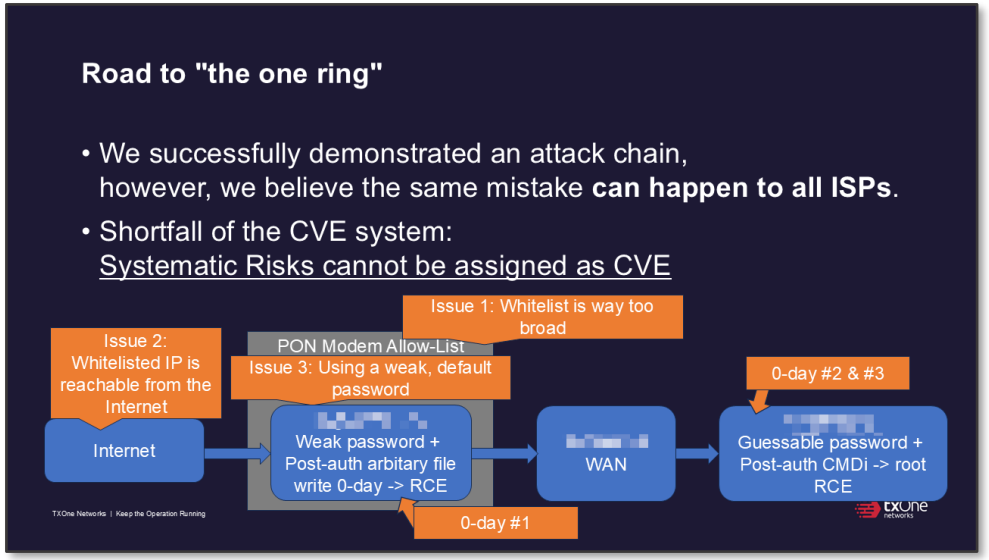
Past cases of finding bugs against ISP management/RGs

- Shahar Tal, 44CON (2014)
  - ISP-side remote management takeover from exposed infrastructure
- Peter Geissler & Steven Ketelaar, HITB AMS (2013)
  - Buffer overflow leading to RCE on exposed TR-069 daemon on RG's WAN
- Sam Curry (2024)
  - Authentication bypass on ISP-side remote management infrastructure
  - Execute commands on RG via command injection through management

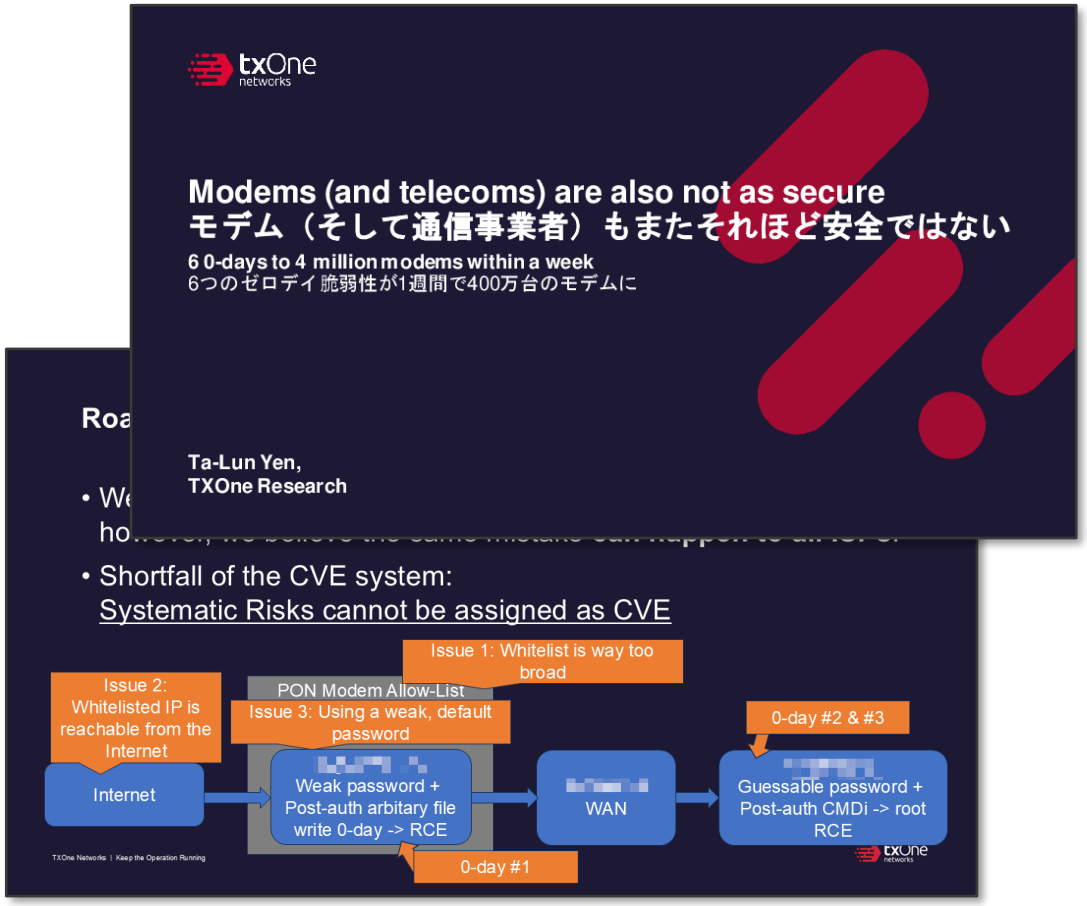


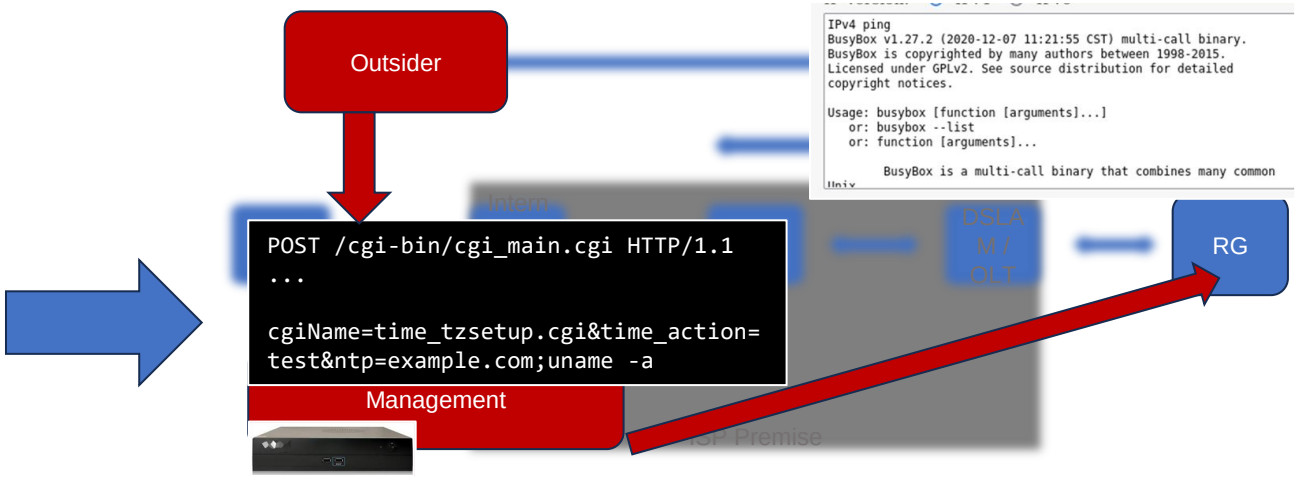


Inspiration of research / Brief Conclusion



Inspiration of research / Brief Conclusion

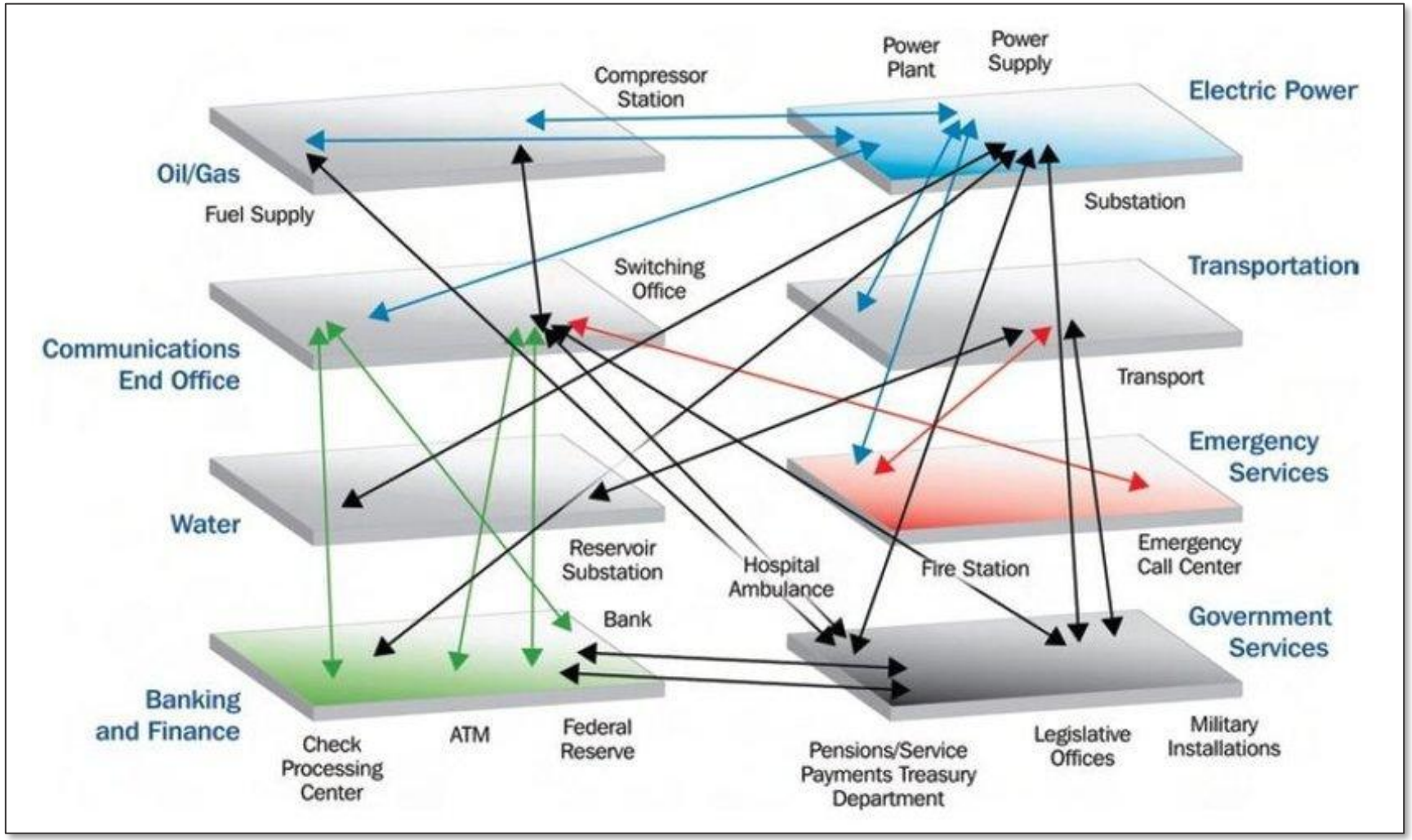




Inspiration of research / Brief Conclusion

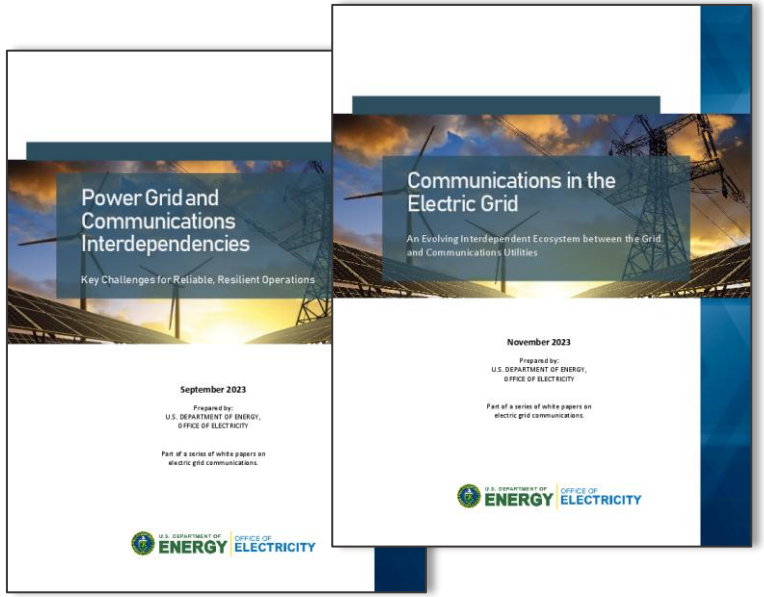
- 14 RGs, 11 ISPs, 9 countries
- RGs are not very safe,
  neither the ISPs
- Demonstration –
  - How to study your RG –
    From board to ISP and many RGs
  - Bypassing OEM's implementation of
    Broadcom TrustZone – Misuse of SDK
  - Among a popular SoC –
    Detecting all RGs on the Internet

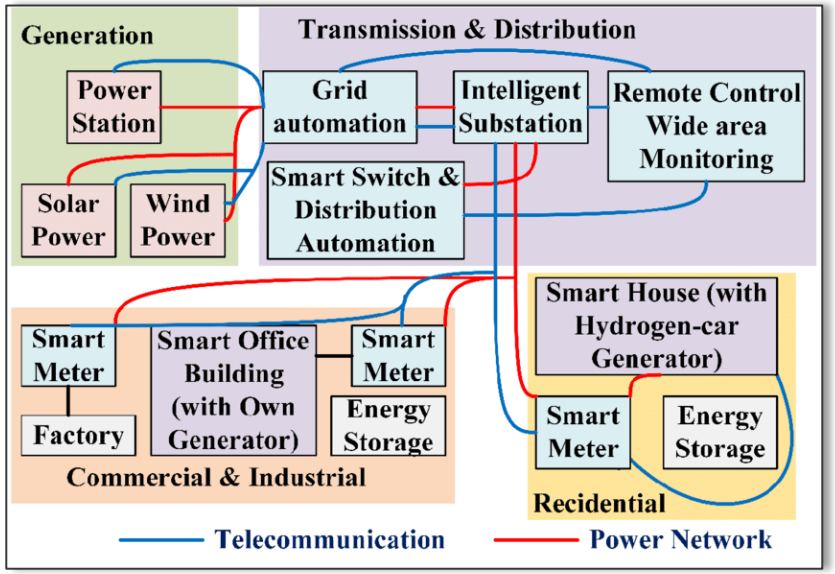
Modern Infrastructure: Everything needs telecommunications



Ehlen, Mark & Vargas, Vanessa. (2013). Multi-hazard, multi-infrastructure, economic scenario analysis. Environment Systems & Decisions. 33. 10.1007/s10669-013-9432-y.

Modern Infrastructure: Everything needs telecommunications

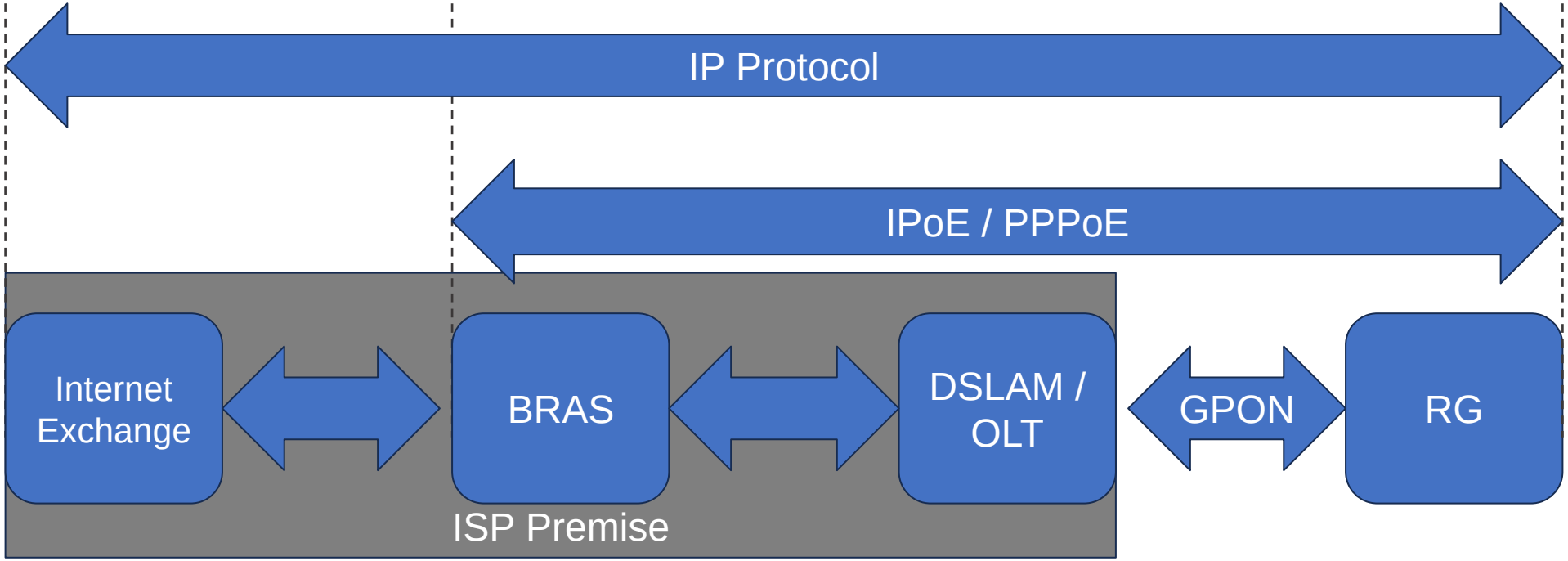




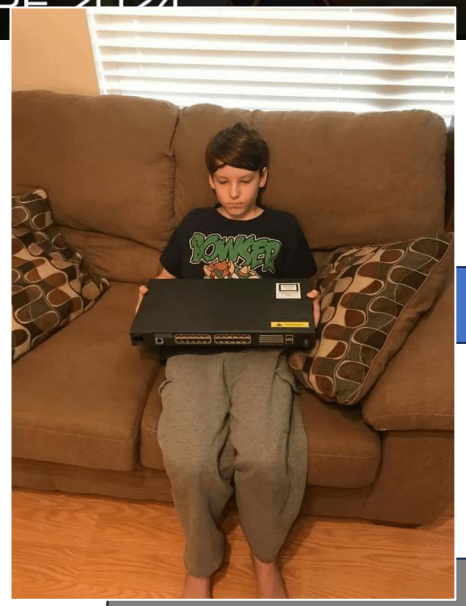
Let's analyze RGs and providers behind them

- Layer-1 Protocols:
  - DOCSIS (cable)
  - xPON (fiber)
  - VDSL (phone line)
- Network Protocols:
  - PPPoE,
  - IPoE,
  - or just IP4/6
- Management Protocols:
  - TR-069 (CWMP)
  - SNMP
  - SSH/HTTP
- SoC makers
  - Broadcom, Intel, Lantiq, Realtek, Huawei & ZTE

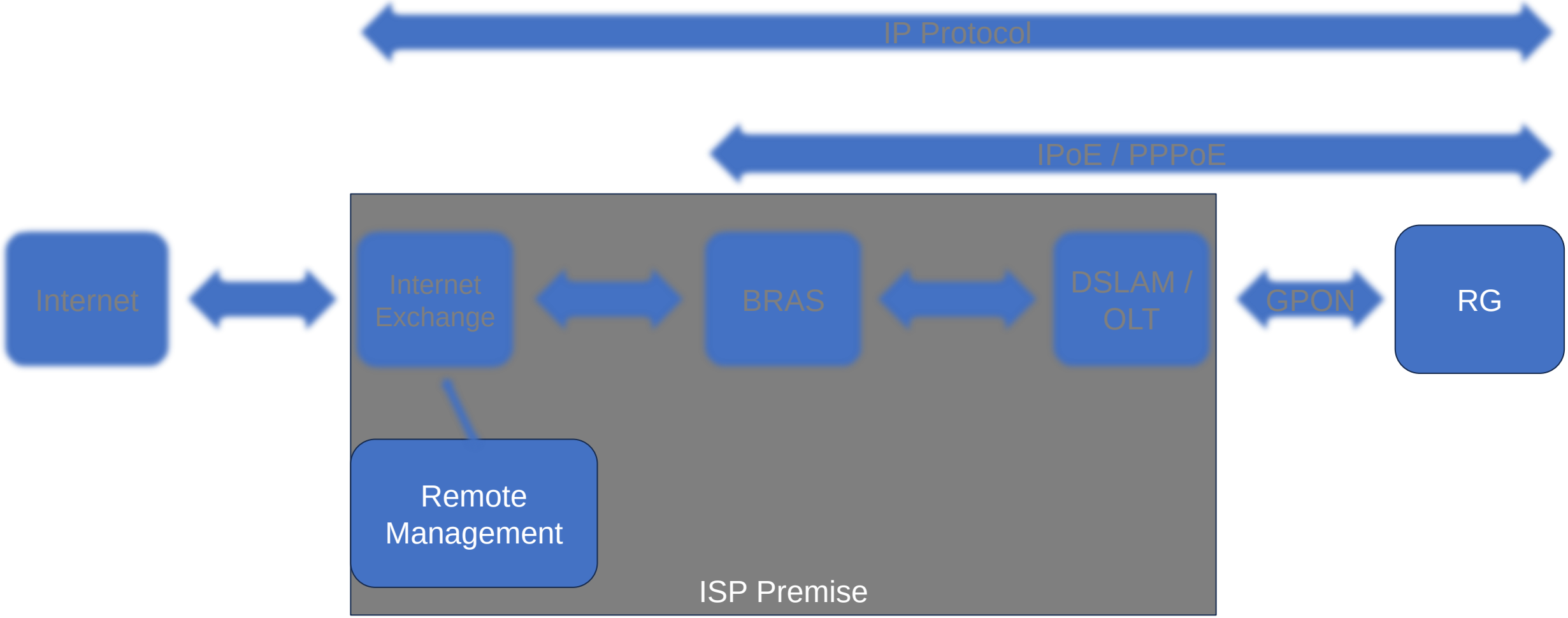
From the provider to your premise



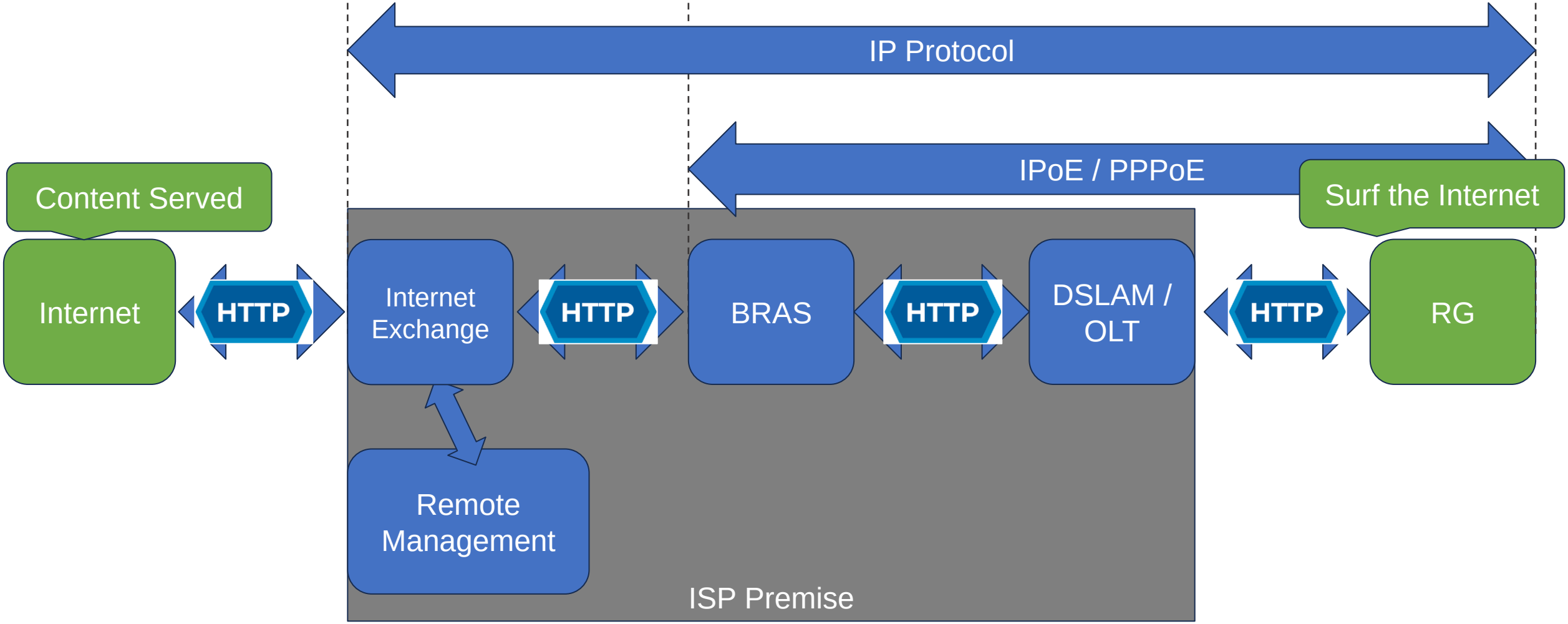
From the provider to your premise



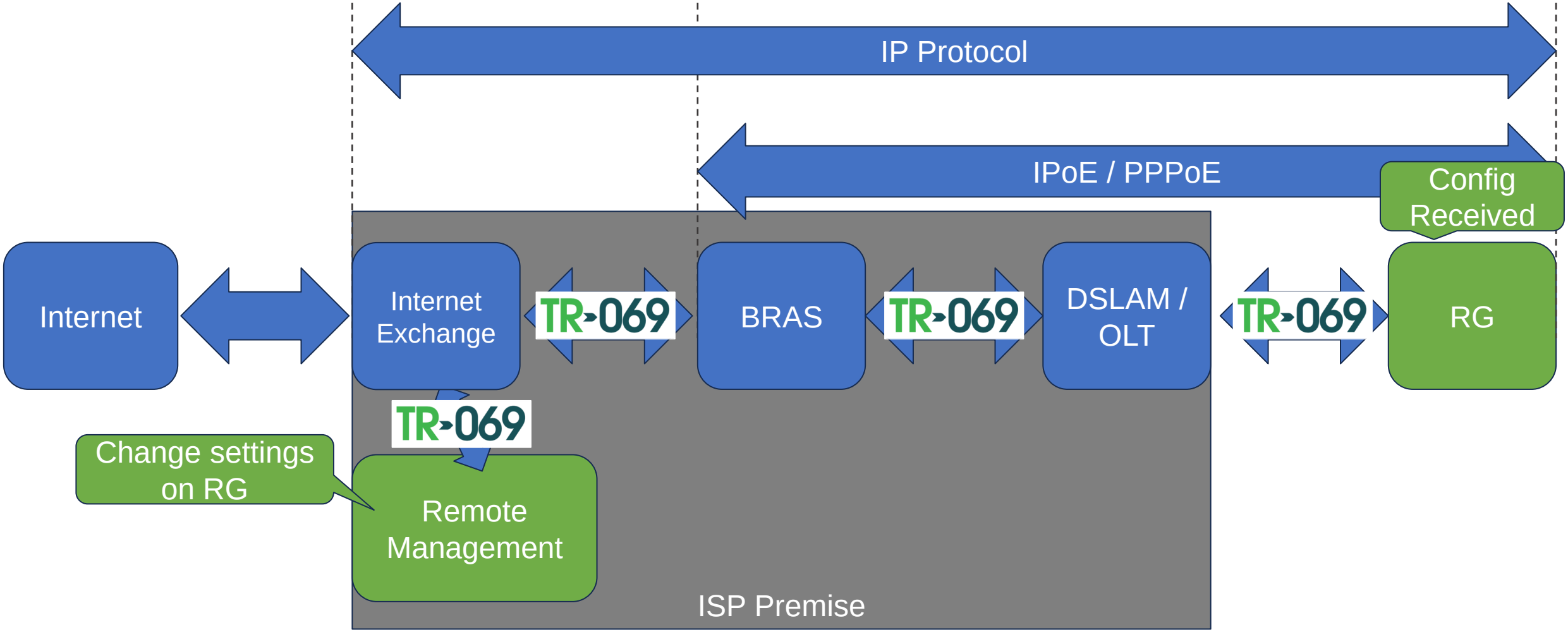
From the provider to your premise



From the provider to your premise

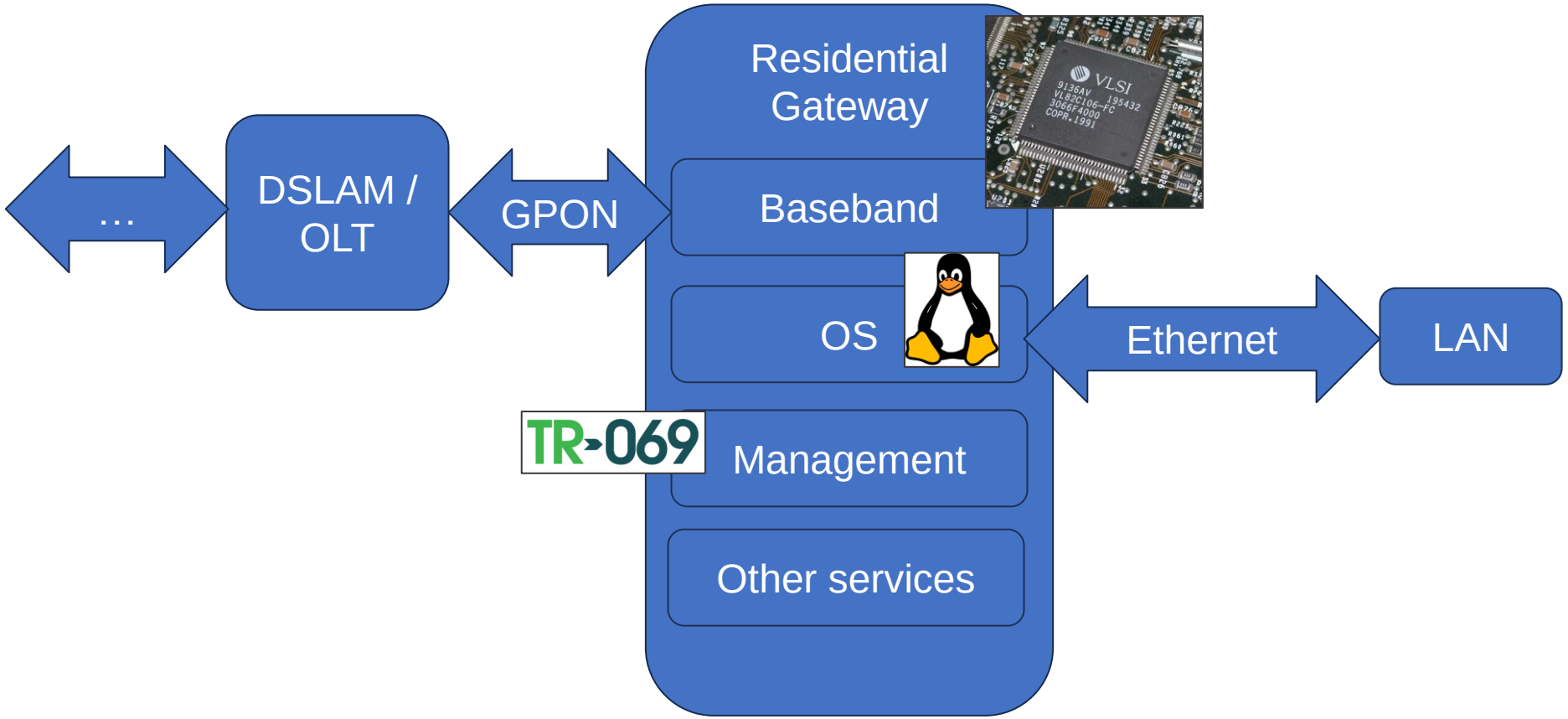


From the provider to your premise



Testing Methodology

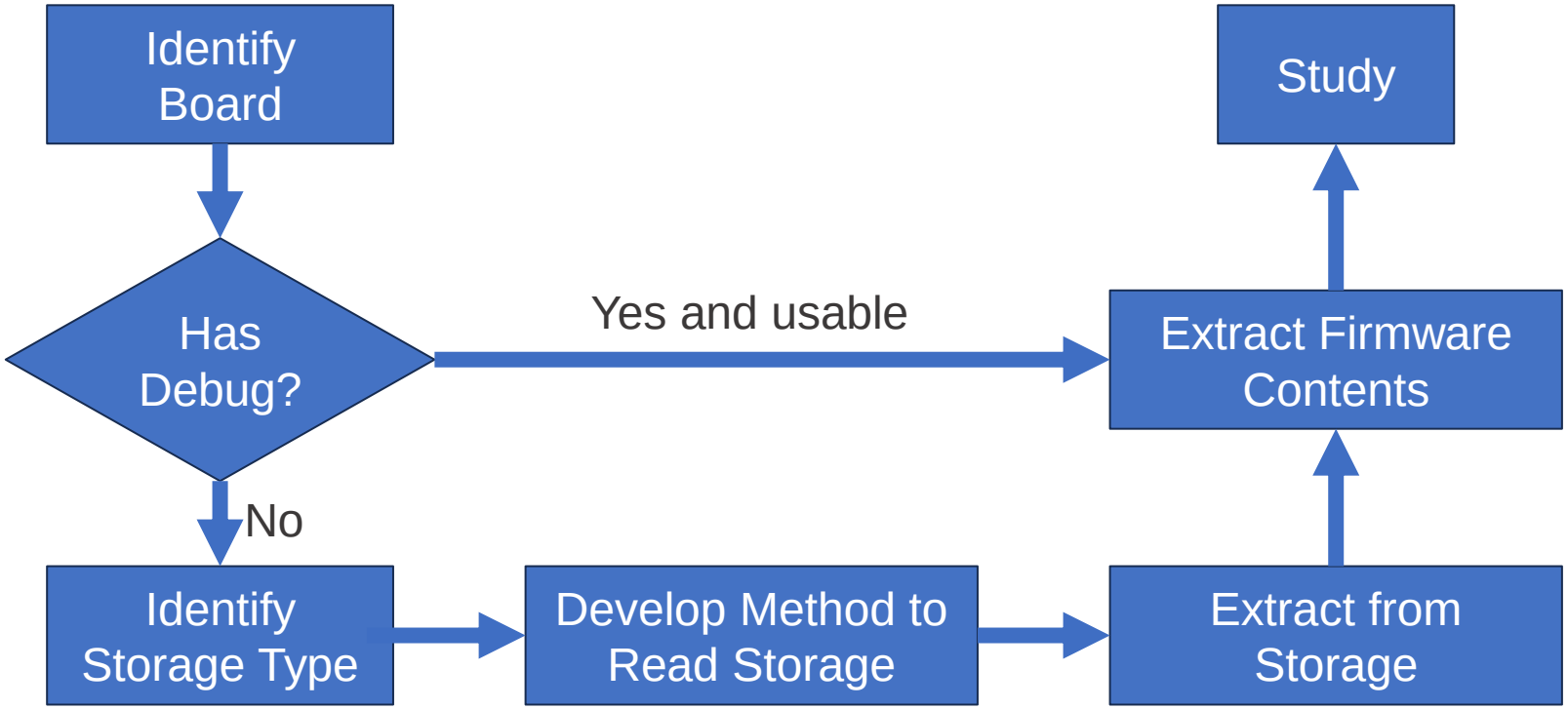
- Focus on **RG**
  - Well connected, many attack surfaces
  - From hardware, software & networking stack to ISP & **Remote management**



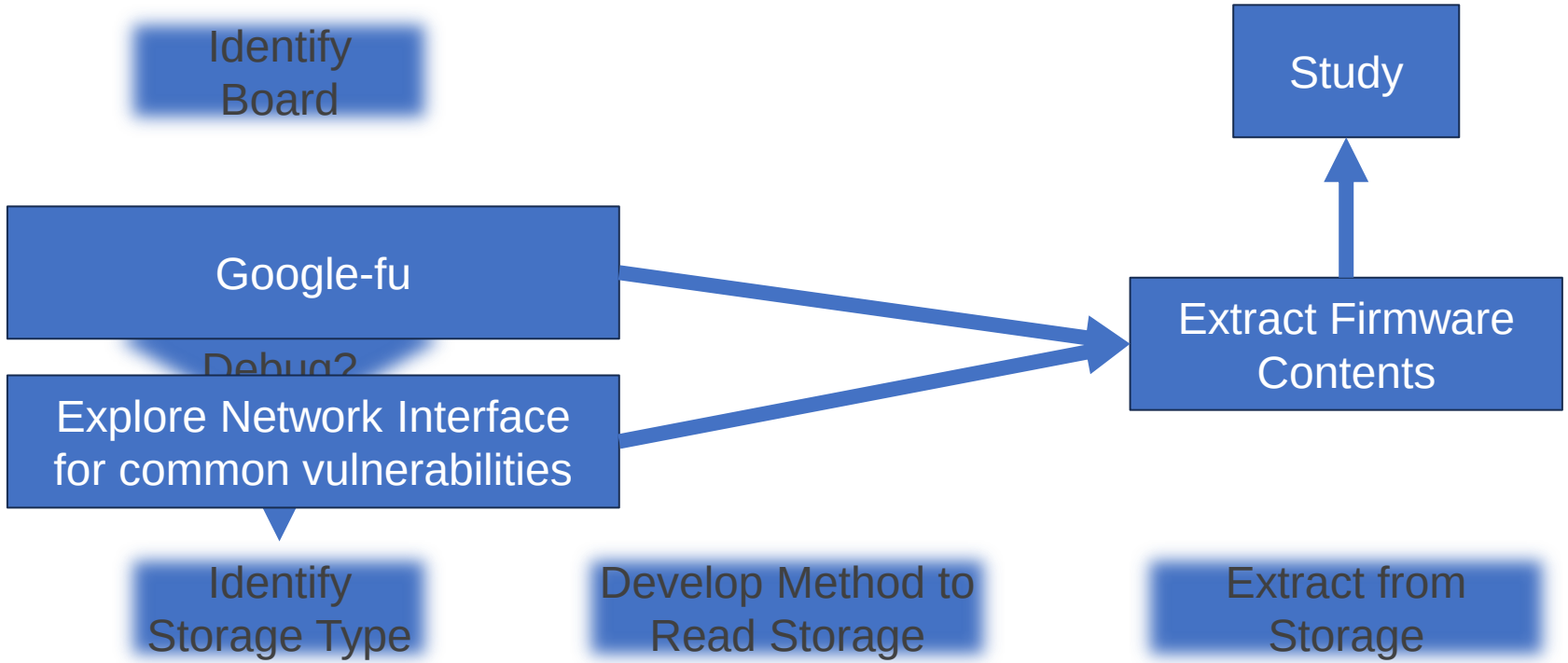
Brief conclusion of analysis

- 14 RGs, 11 ISPs, 9 countries
- RGs -
  - Common:
    - Lack of modern practices
    - "Solder-UART-to-root“
    - Command Injections
- ISPs -
  - Exposed management is common
  - Huawei & ZTE still in Europe

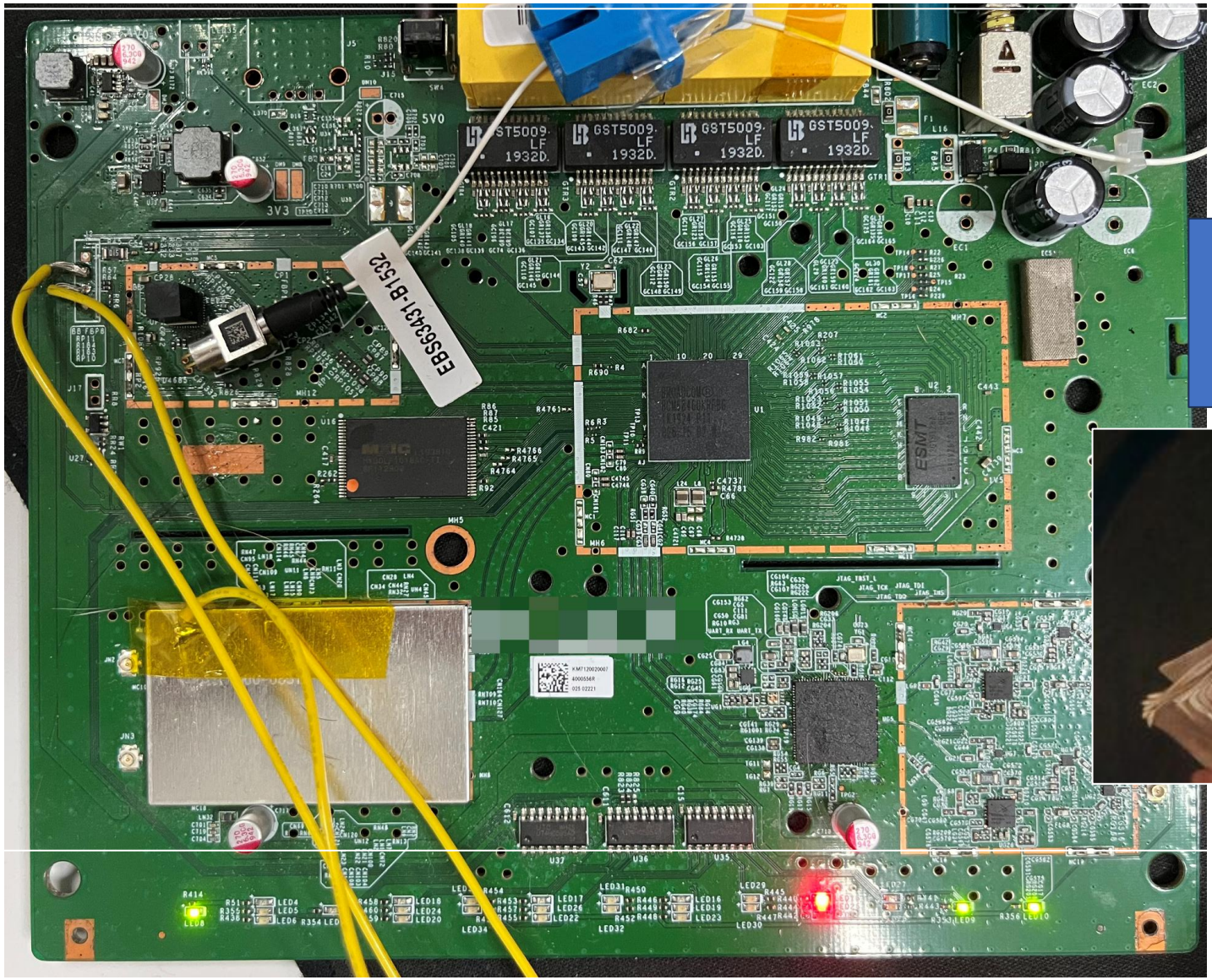
Assessing entry methods



Assessing entry methods, many cases

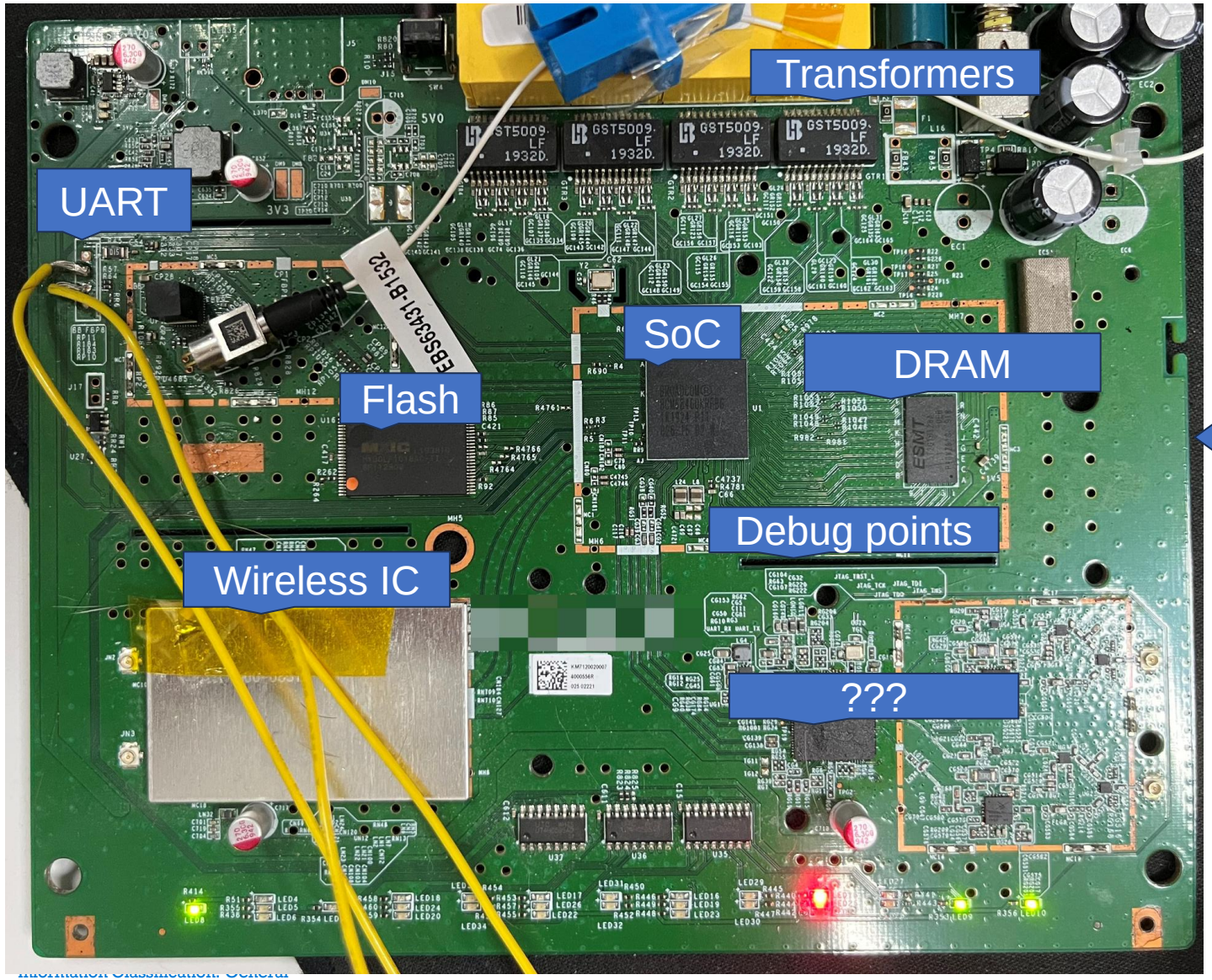


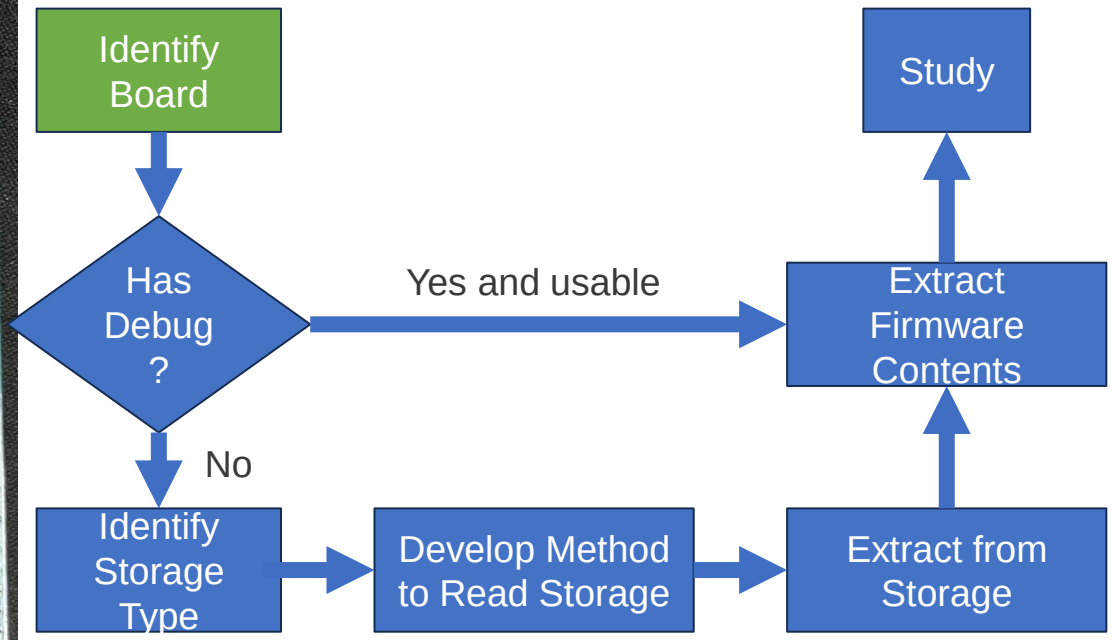
Board component identification & assessing entry methods



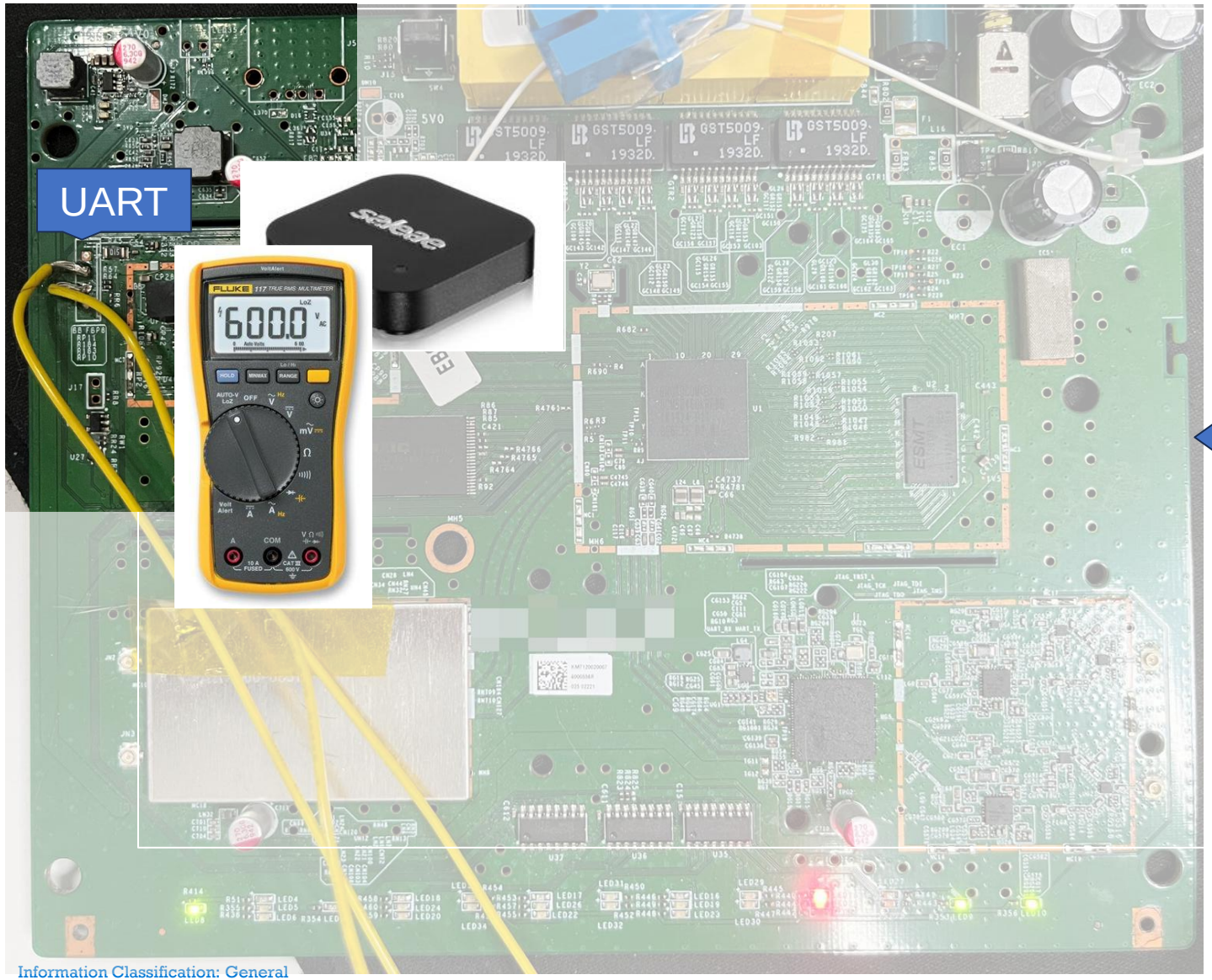
How to interact with the board?

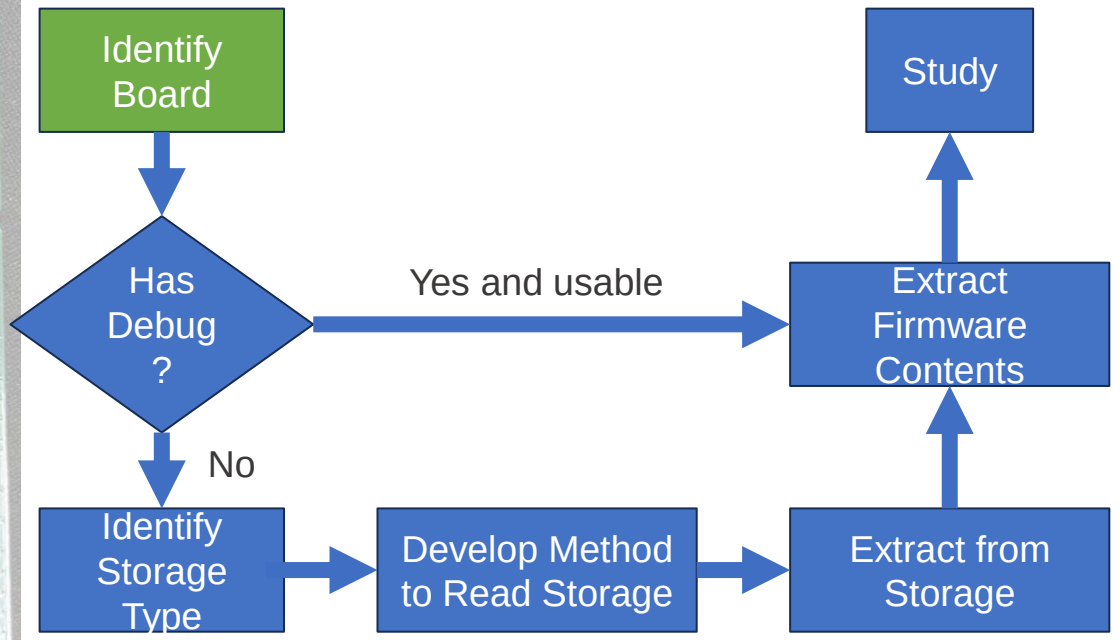
Board component identification & assessing entry methods



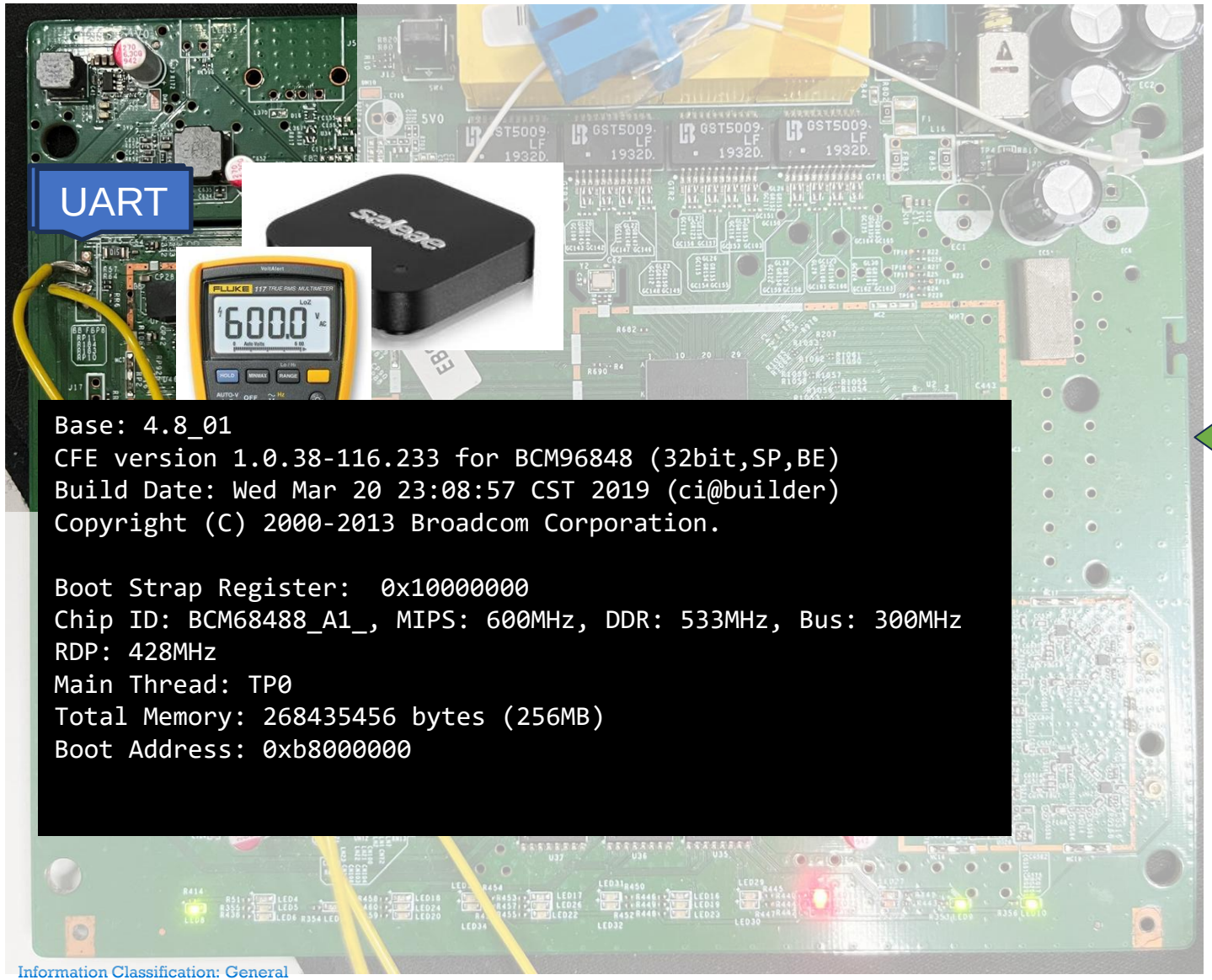


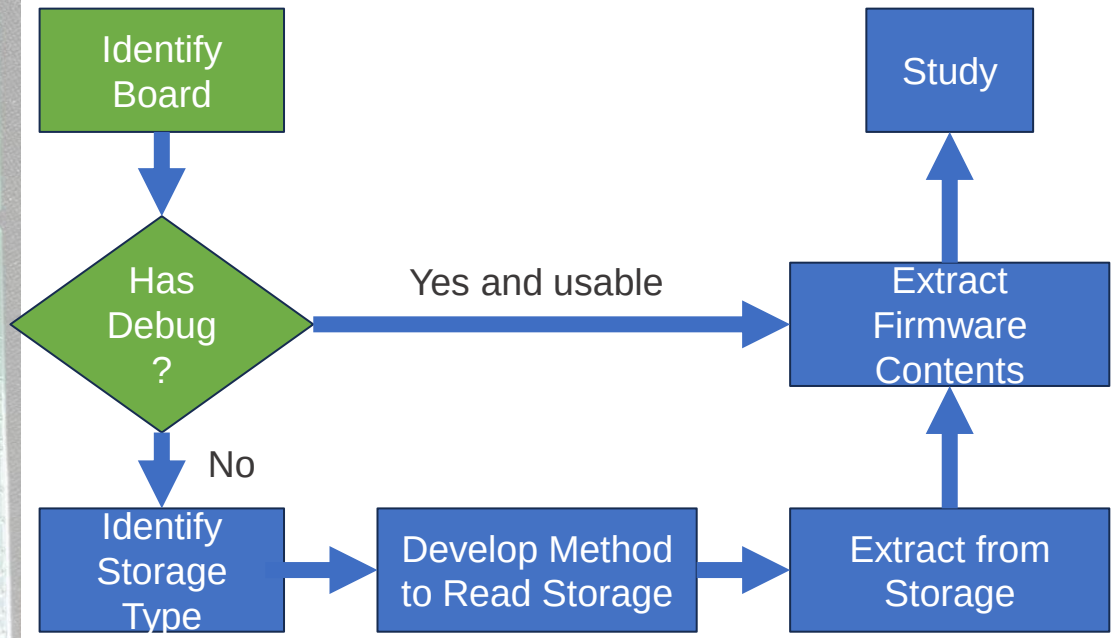
Board component identification & assessing entry methods



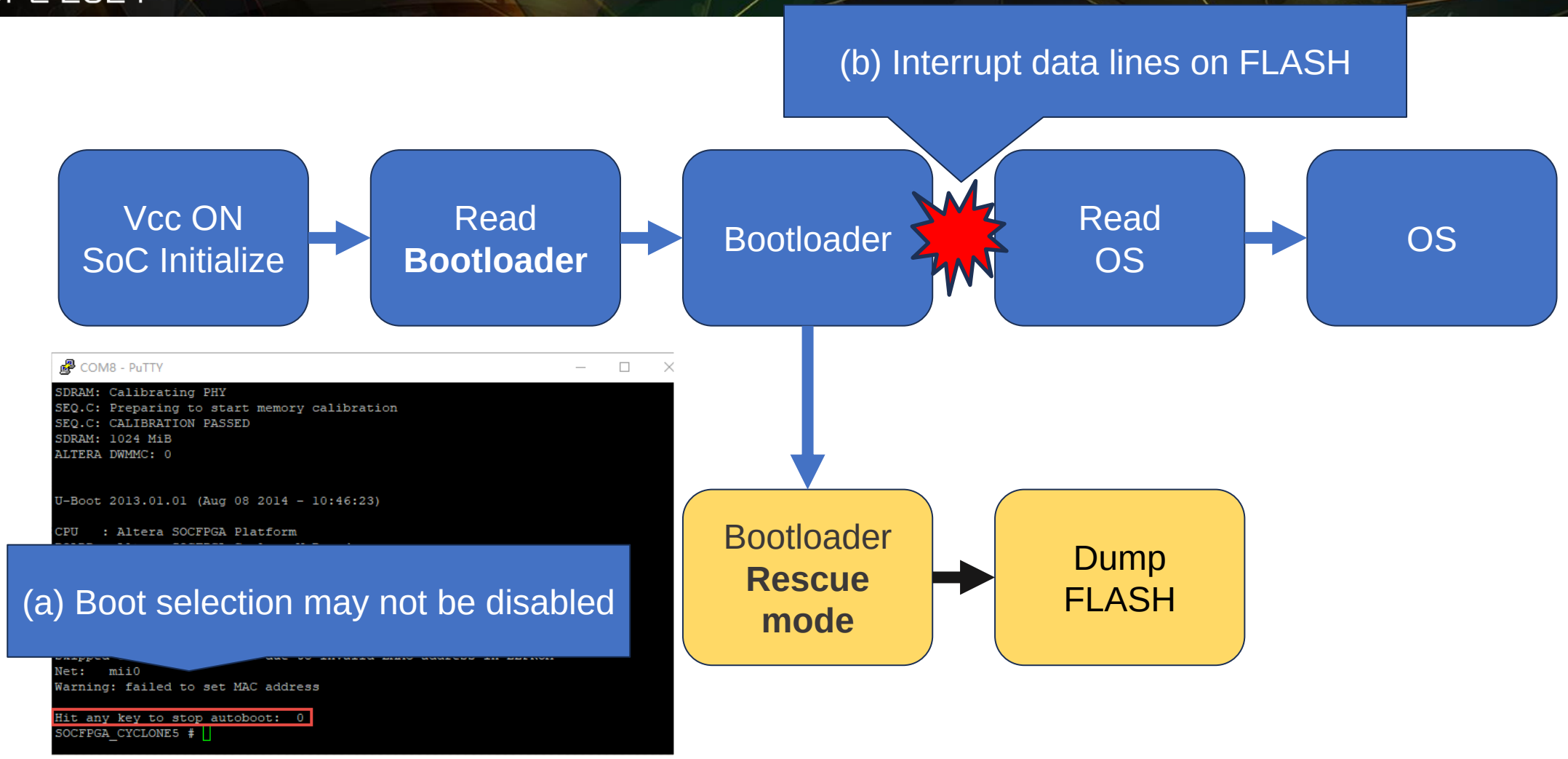


Board component identification & assessing entry methods



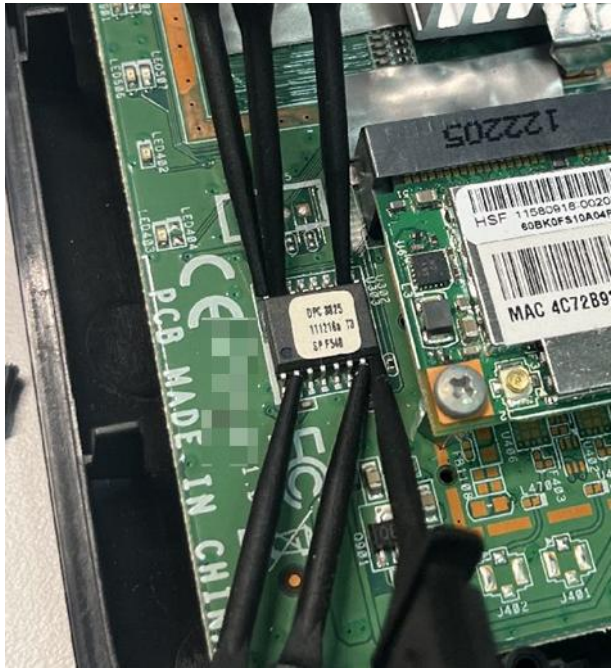


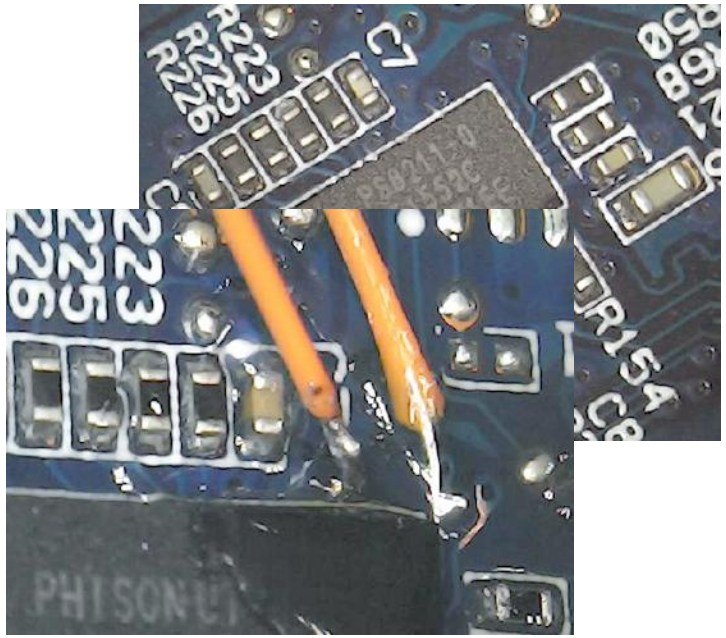
Flash extraction via Pre-boot environment



Reading firmware/configuration files from board

- For targets like RG, de-soldering may not be best
  - Risks breaking the board, depending on experience level

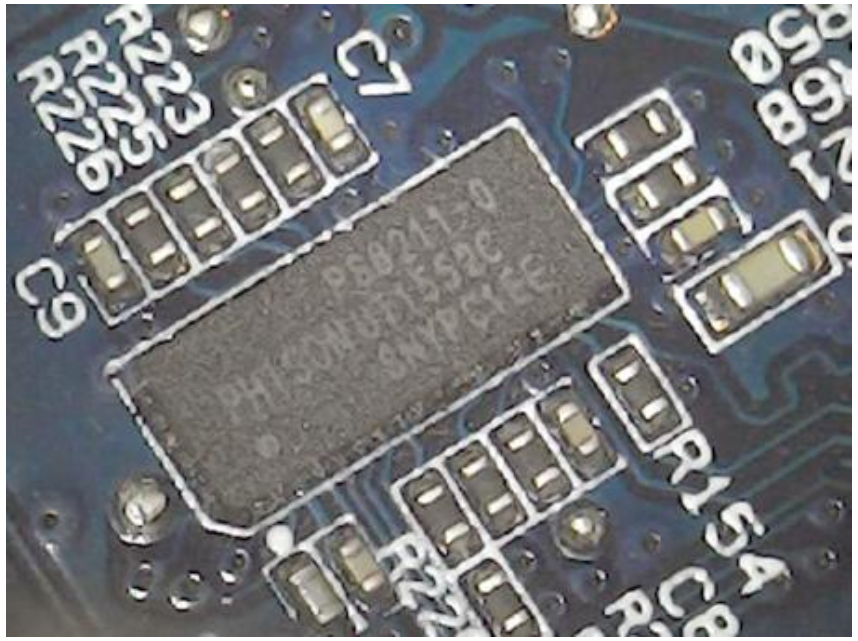


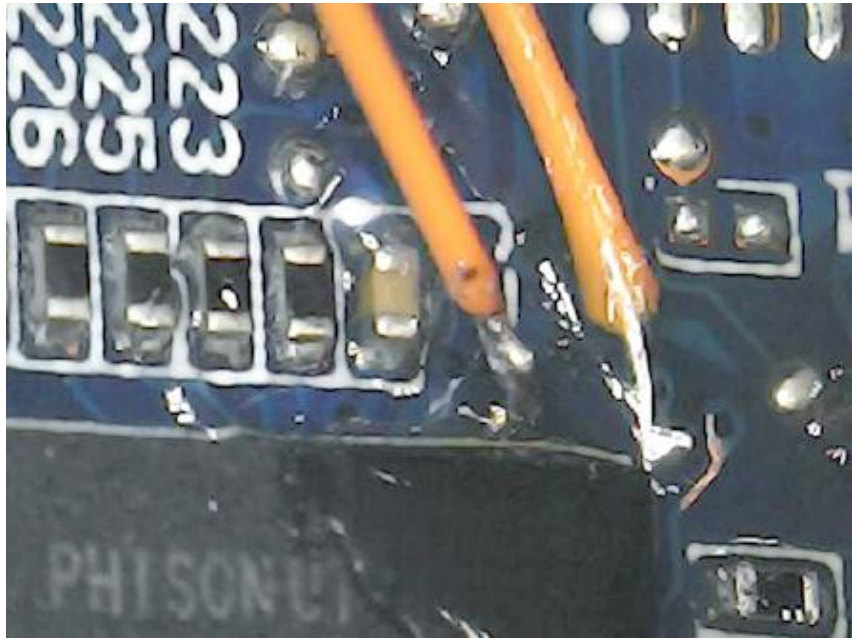


://<string>

Reading firmware/configuration files from board

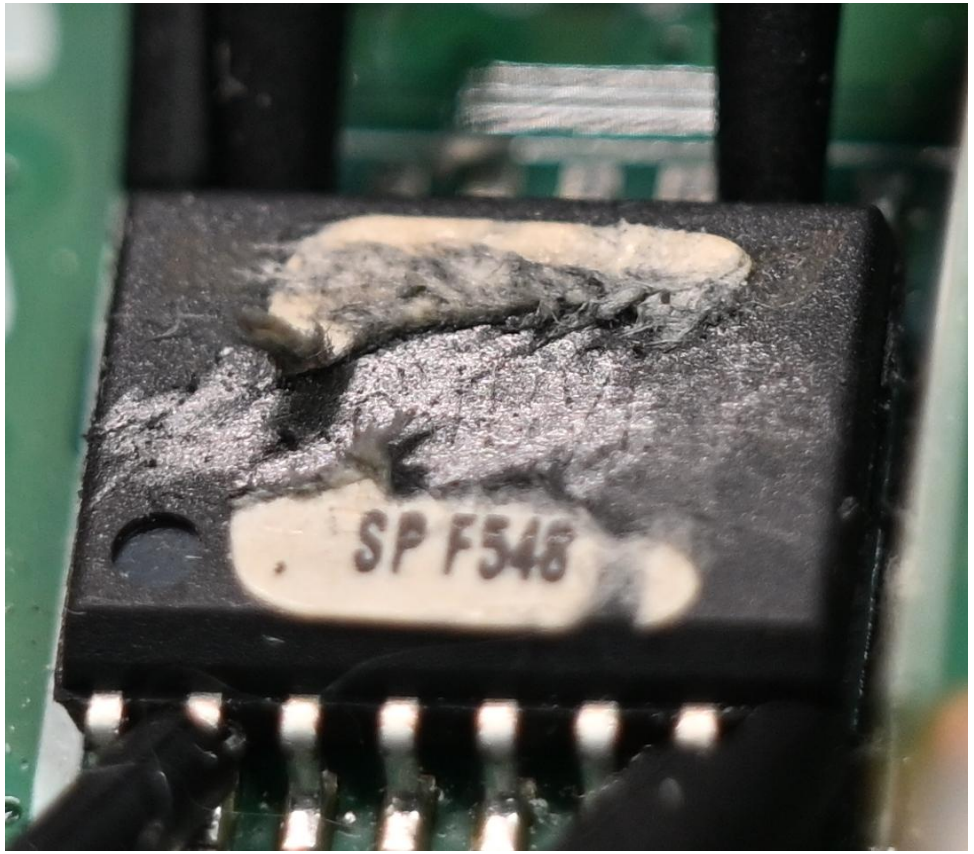
- Some boards requires “MITM method” (scraping)
  - Sometimes having to soldering to BGA solder joints beneath the chip
- Wire length and impendence matching matters
- Use proper breakout boards





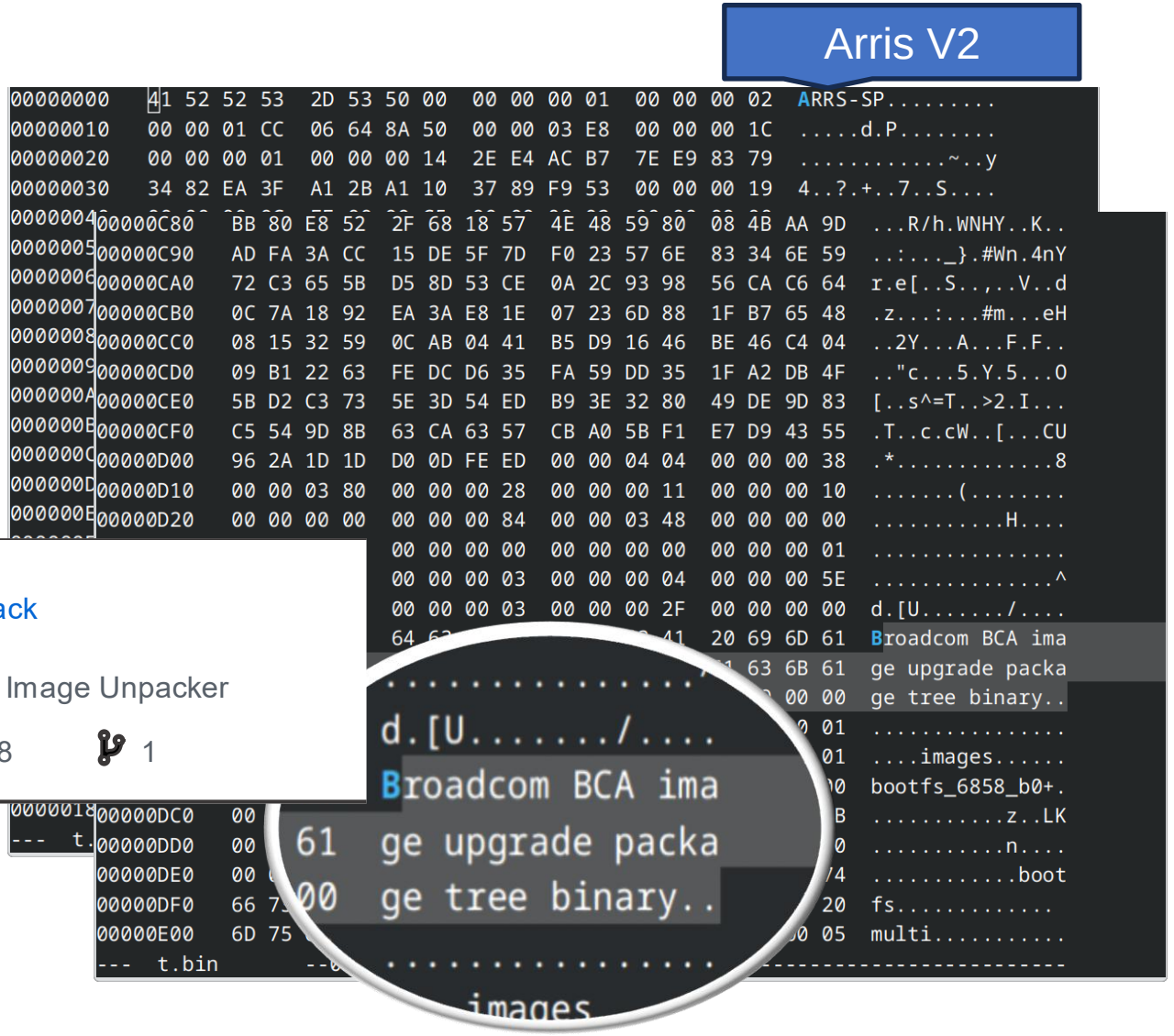
No Chip Marking? No Problem!

- What if markings on chip got erased away/eroded away?
- Markings are etched via laser = grooves on packaging
- I recover them with a pencil



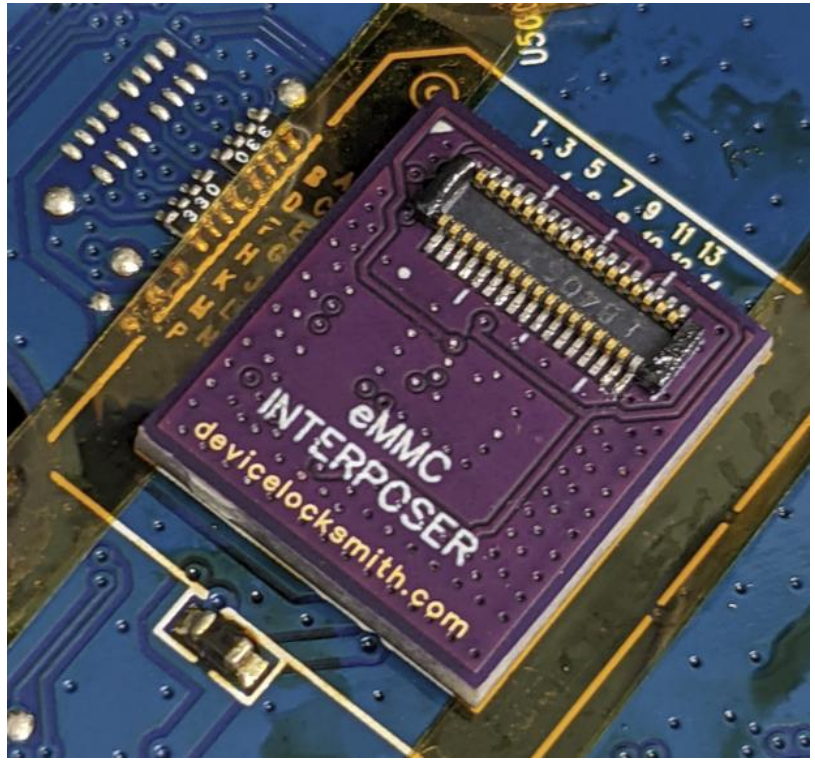
Reading firmware/configuration files from board

- If `binwalk` doesn't work:
  - Try `hexedit` or `strings`
  - Try finding magic
    - `5d 00 00 10 00`,
      `5d 00 00 01 00` (LZMA)…
  - Look for regularities
- Try  ://<string>



Actual Study, Case 1

- Broadcom Gen 3
  - Secure Boot & Root-of-Trust
  - FDE
- 802.1x to authenticate with ISP
- Difficult to desolder/scrape traces
  - BGA56
  - Tight Clearance



Case 1, procrastinate on soldering

- Found discussion of Case 1 in China:

- Found firmware distribution page
- Site offline
  - AWS S3 -- Wayback Machine
  - Retrieved another model's firmware by same vendor
  - Unencrypted

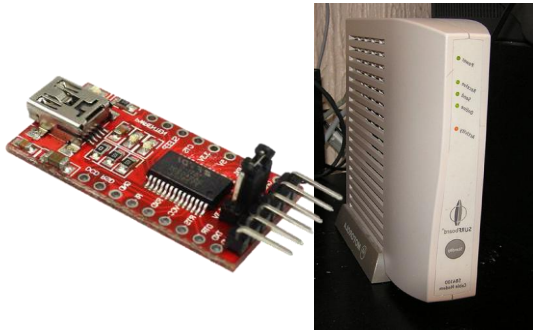
Case 1, procrastinate on soldering

- Needs a primitive to dump firmware & code exec
  - LAN management looks safe
  - Not much on WAN
  - Don’t want to desolder
- Looked at unencrypted dump from another device

```
$ ff '*.rules*'
./etc/udev/rules.d/85-SerialPort.rules
./etc/udev/rules.d/50-config.rules
```

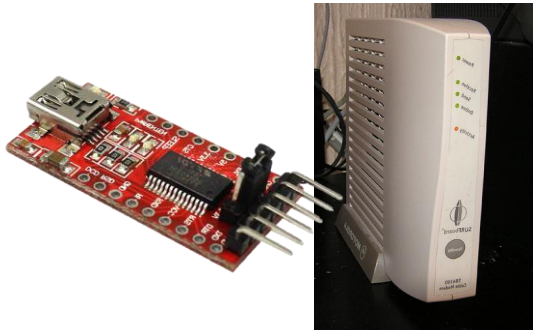
```
$ cat ./etc/udev/rules.d/85-SerialPort.rules
ACTION=="add", KERNEL=="ttyUSB[0-9]*", SUBSYSTEM=="tty", \
ATTRS{idVendor}==“0403", ATTRS{idProduct}==“6001", \
RUN+="/bin/sh /bin/start_debug"
```

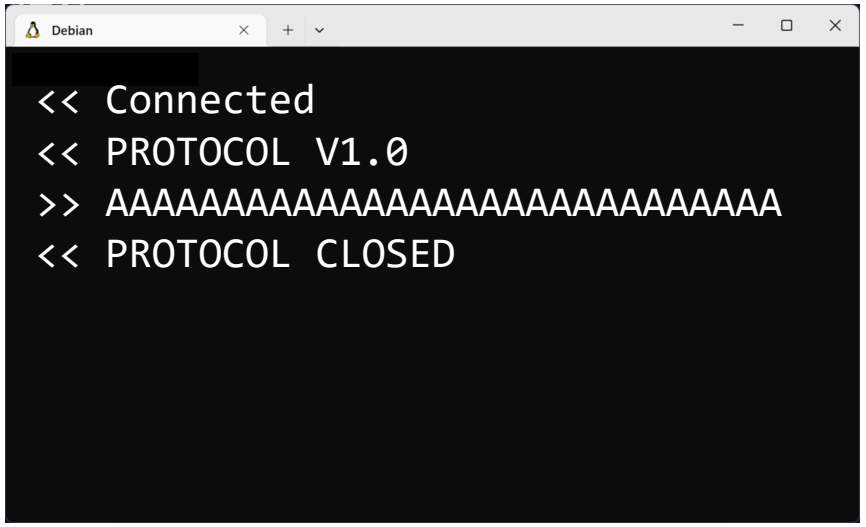
Case 1, /bin/start_debug



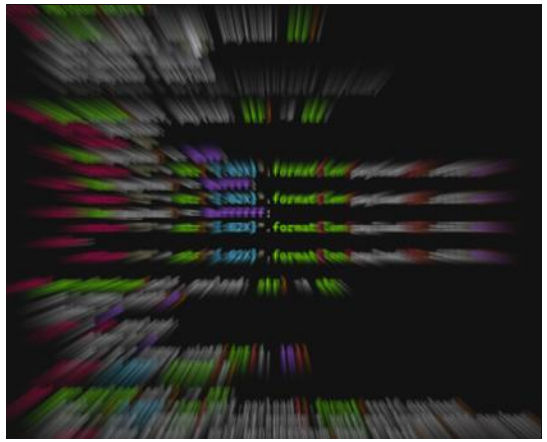
```
Debian
<< Connected
<< PROTOCOL V1.0
>> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<< PROTOCOL CLOSED
```

Case 1, /bin/start_debug

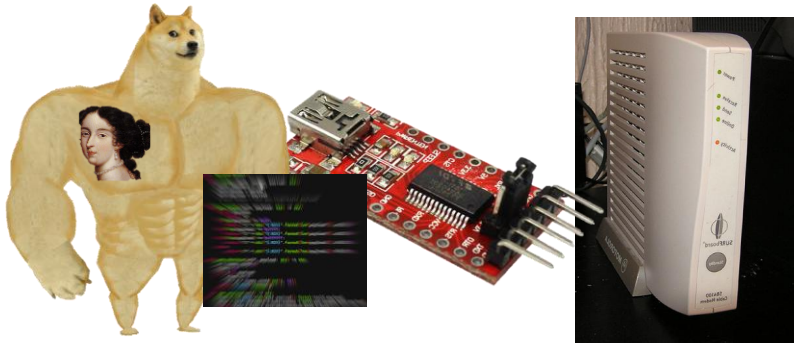




- Proprietary protocol
- OpCode-based
  - 8001 for reboot, 8002 for update…
- Requires fixed password
- Wrote dissector



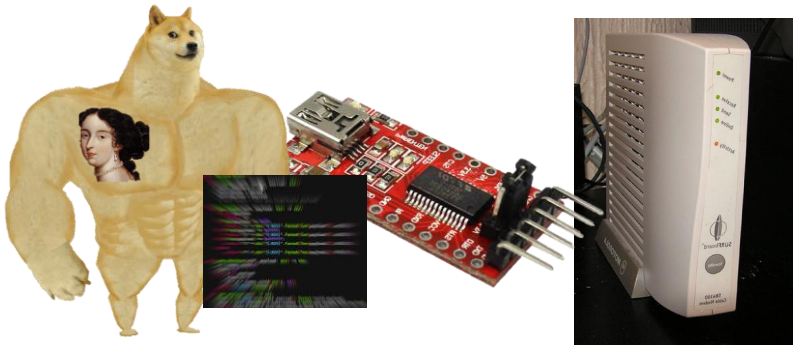
Case 1, /bin/start_debug



Debian

```
<< Connected
<< PROTOCOL V1.0
>> PASSWORD
<< PROTOCOL STARTED
>> EA8002/payload;id;uname
<< root; Linux 2.6..
```

Case 1, /bin/start_debug




```
Debian
<< Connected
<< PROTOCOL V1.0
>> PASSWORD
<< PROTOCOL STARTED
>> EA8002/payload;id;uname
<< root; Linux 2.6..
```

- Bypassed security guarantees
  - ***L***CE as root on device
  - “Bypass TrustZone”
  - Keys in root-of-trust and decrypted via TrustZone
  - Not secured when data in use:
    - Extracted 802.1x keys
    - Extracted FDE keys

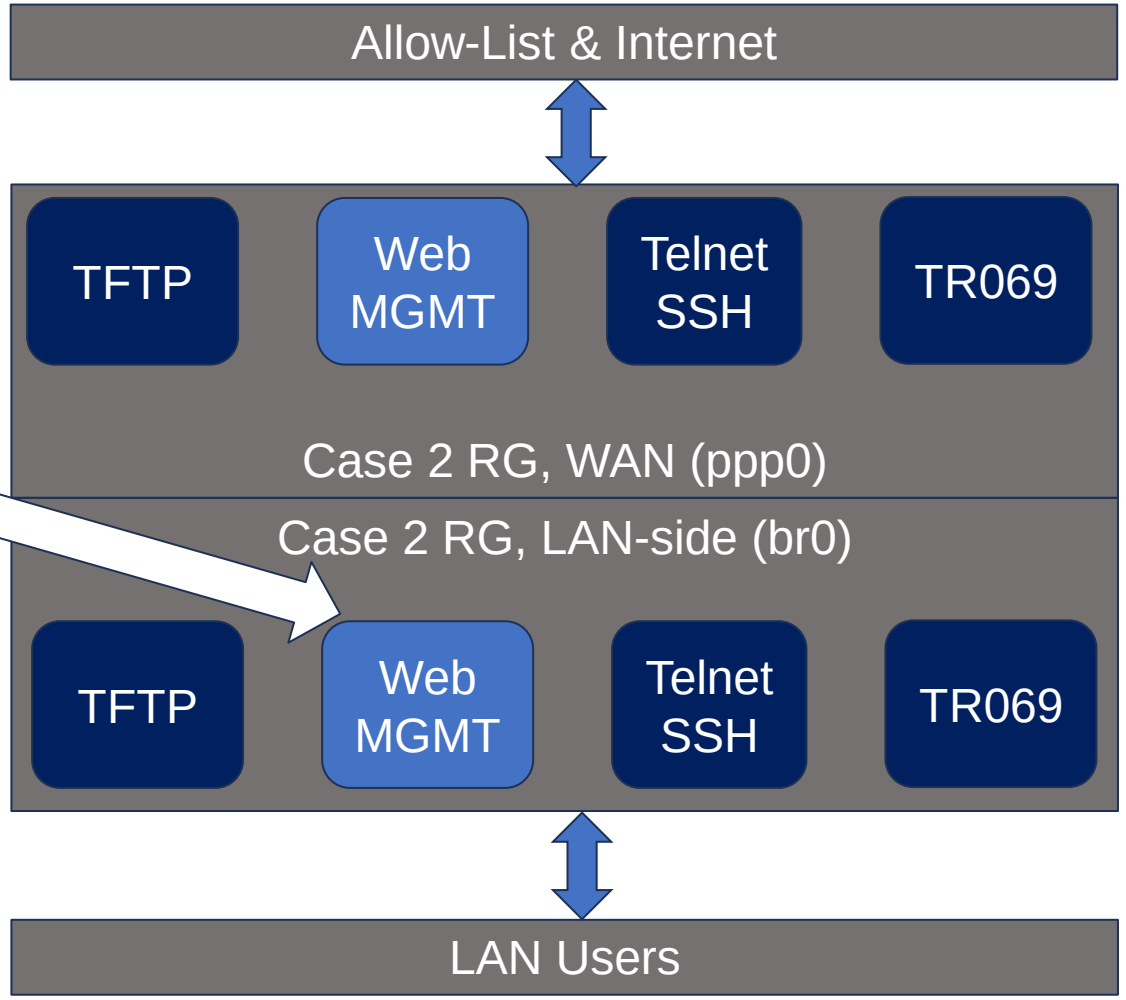
Actual Study, Case 2

Common problem with RG software

- Lack of hardening is common (e.g. no NX, no canary)
- But this is found everywhere:

```
all_args[0] = "sh";
all_args[1] = "-c";
all_args[2] = input;
all_args[3] = 0;
sub_82EC("/bin/sh", all_args);
```

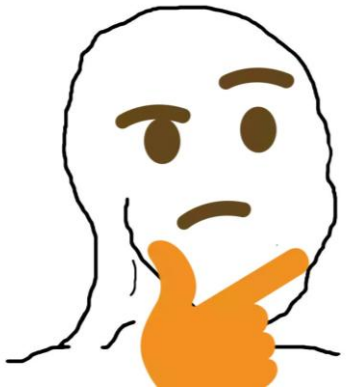
- Bugs are LAN-side (yet)

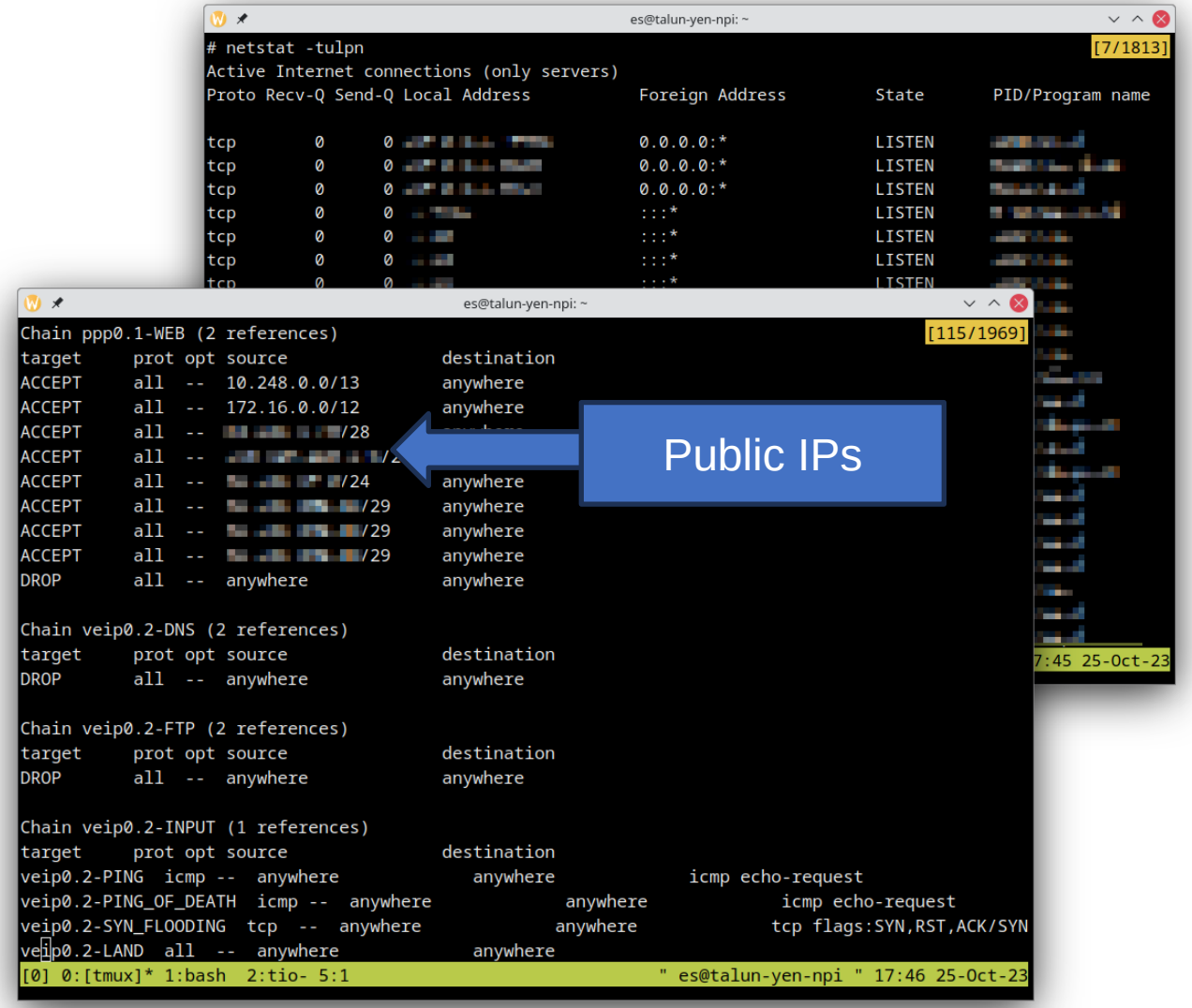


How to expand our attack primitive?

Case 2 - Cross-referencing iptables & services

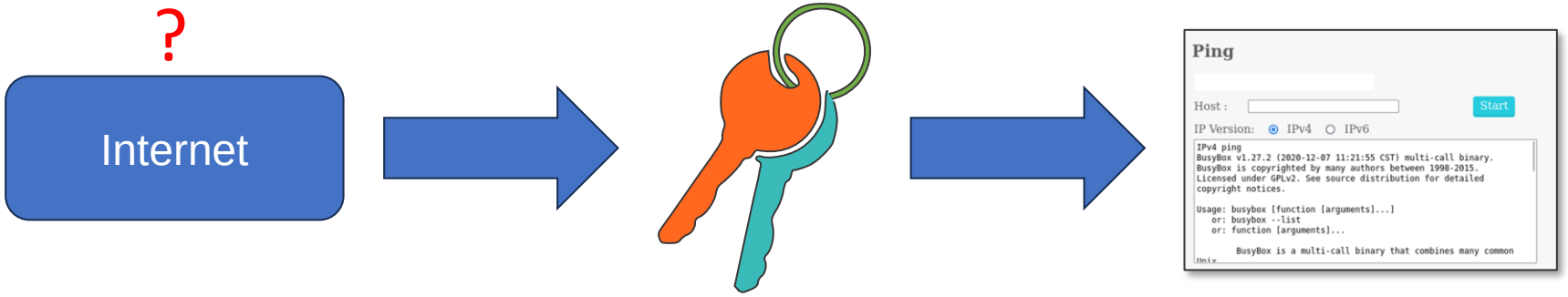
- Certain IP ranges can reach management via WAN
- Only blocks ICMP Request (not other types)



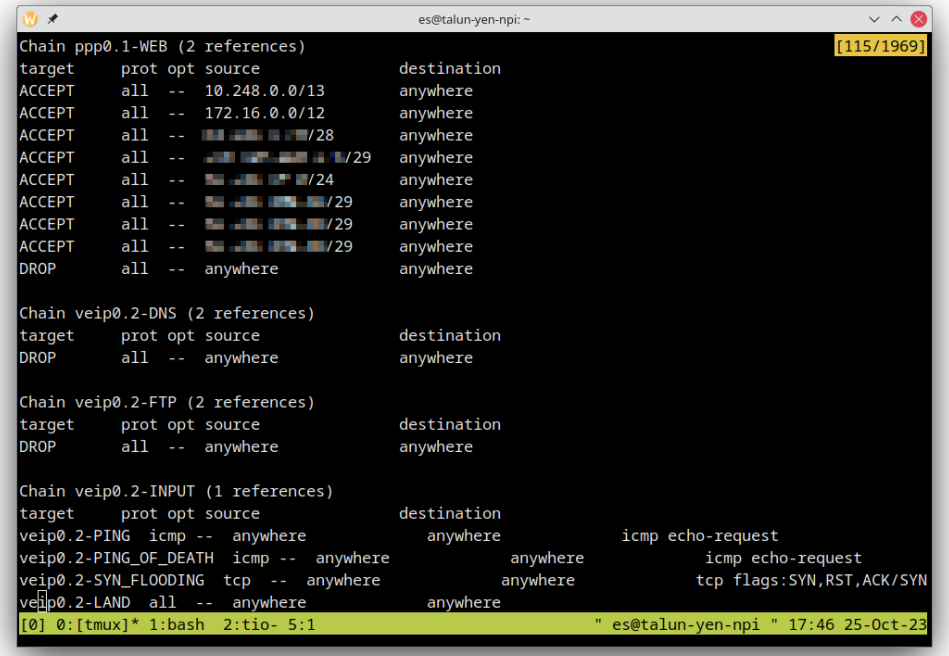


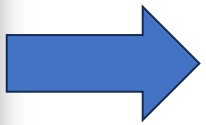
How to get inside everyone's RG?

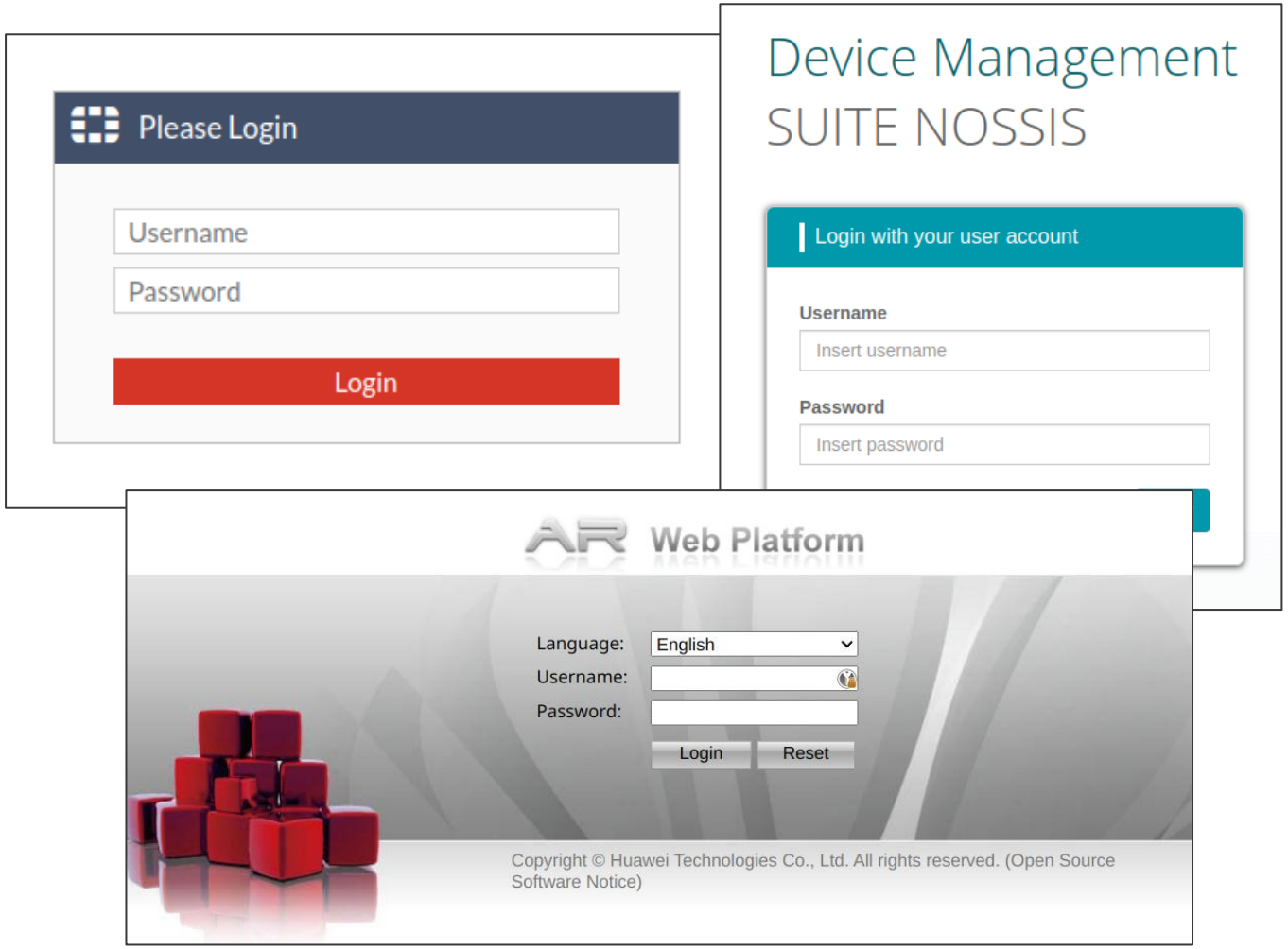
- Has post-auth RCE on management interface
- Needs to escalate RCE bug to pre-auth
  - High Privileged Account –
    - Fixed username, Password tied to **ETH0_MAC_ADDR[-4:]**
    - Shared "Guest" account allows reading **ETH0_MAC_ADDR**
- Needs to reach management interface's WAN side



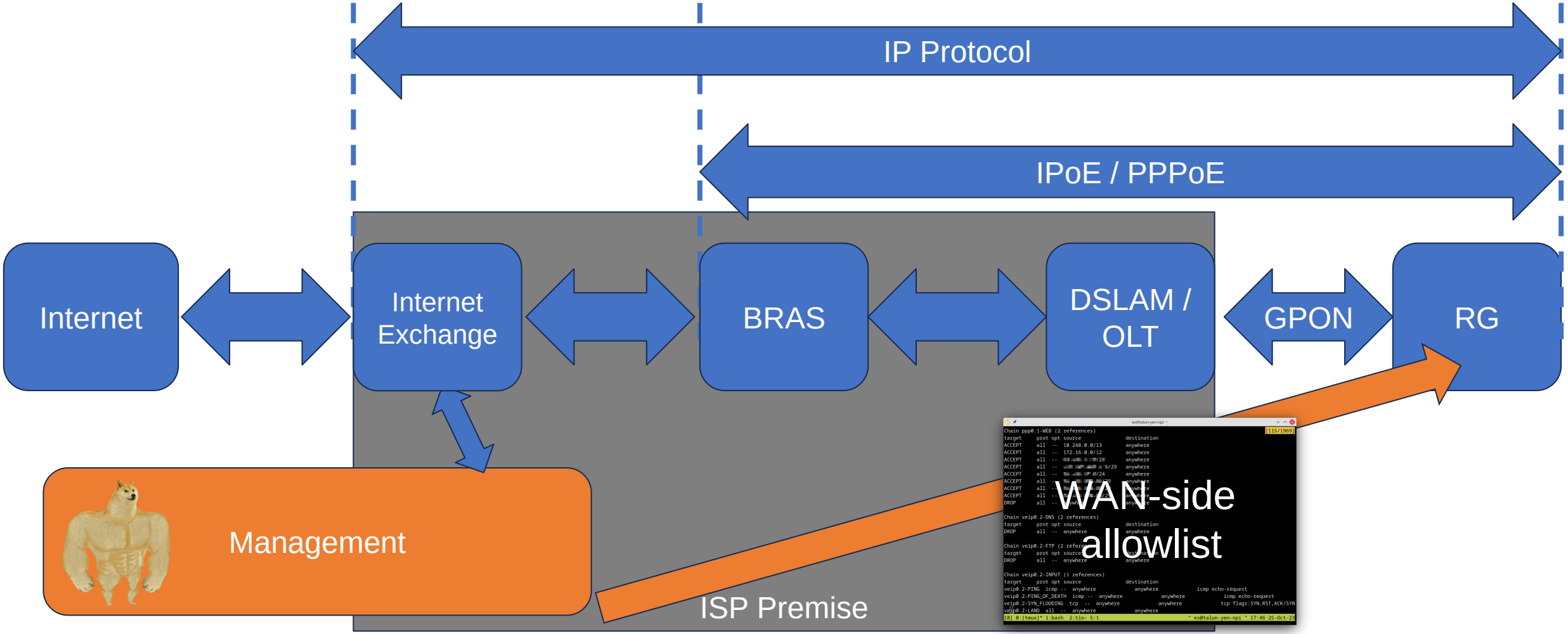
Finding Management Infrastructure on Internet, Example



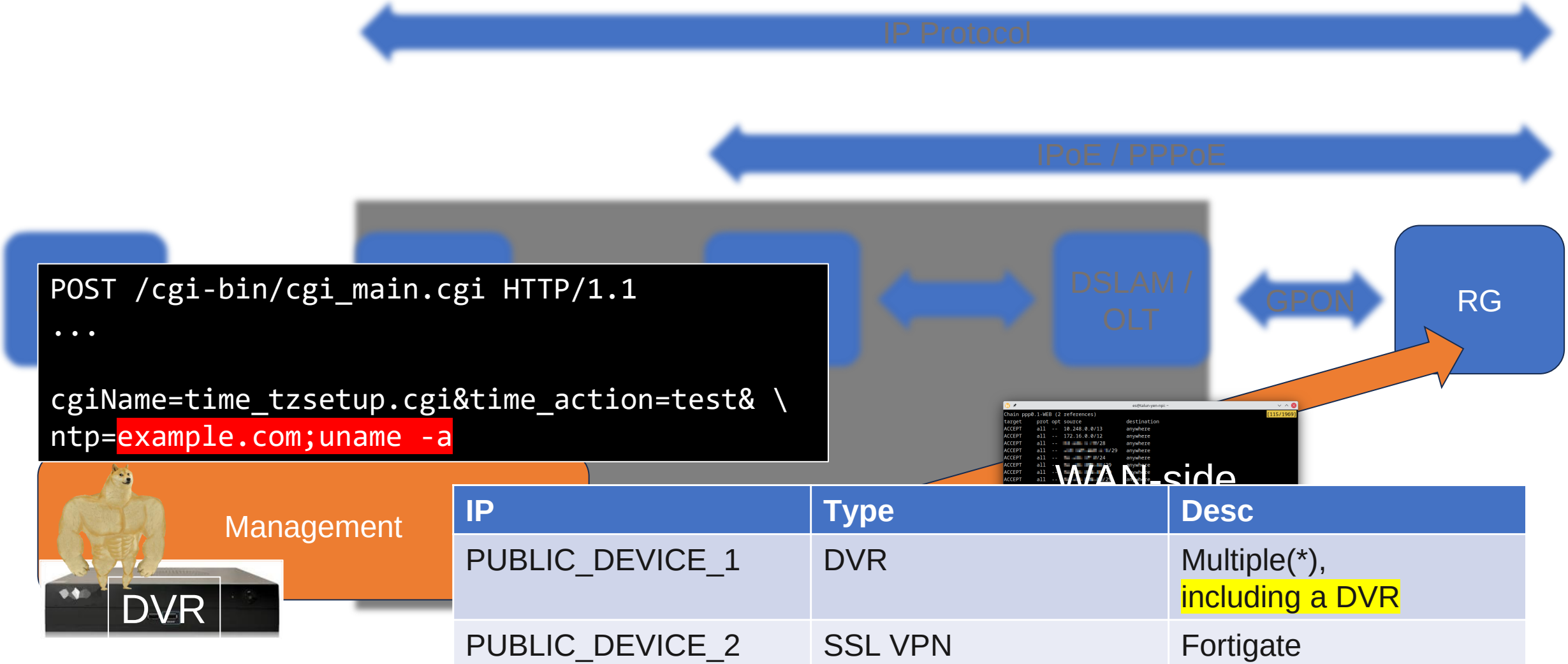




From the “provider” to your premise



From the “provider” to your premise



IP	Type	Desc
PUBLIC_DEVICE_1	DVR	Multiple(*), including a DVR
PUBLIC_DEVICE_2	SSL VPN	Fortigate

From the “provider” to your premise

Internet

```
IPv4 ping
BusyBox v1.27.2 (2020-12-07 11:21:55 CST) multi-call binary.
BusyBox is copyrighted by many authors between 1998-2015.
Licensed under GPLv2. See source distribution for detailed
copyright notices.

Usage: busybox [function [arguments]...]
   or: busybox --list
   or: function [arguments]...

        BusyBox is a multi-call binary that combines many common
Unix
```

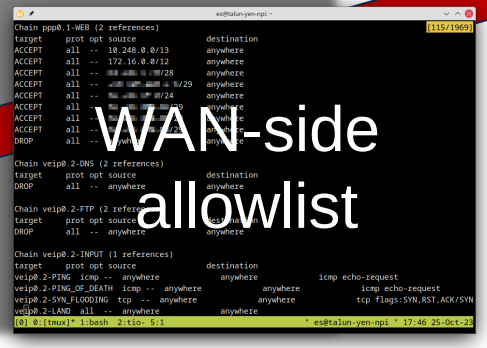
```
POST /cgi-bin/cgi_main.cgi HTTP/1.1
...

cgiName=time_tzsetup.cgi&time_action=test& \
ntp=example.com;uname -a
```

Management

DSLAM / OLT

GPON

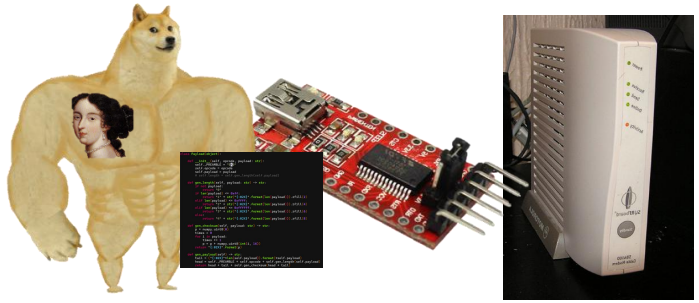
RG

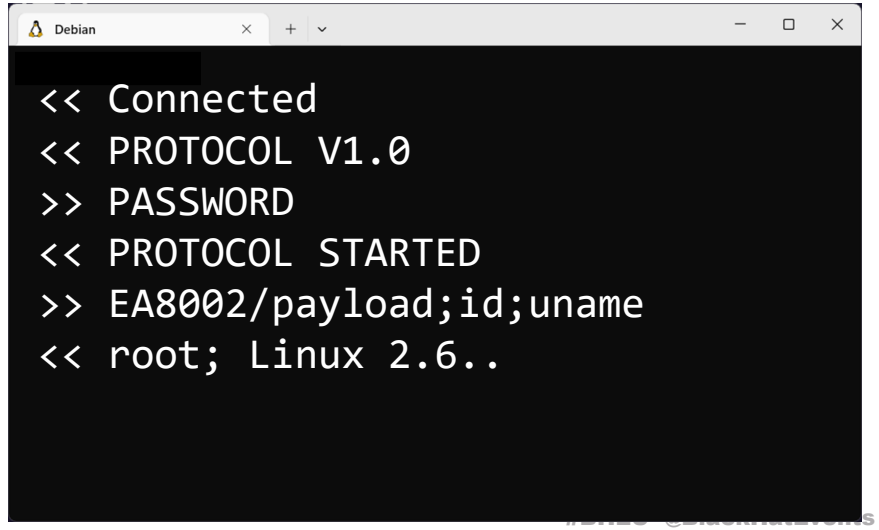


WAN-side allowlist

From the “provider” to your premise

- Case 2 – 4M affected
  - RCE on all devices
  - Device removed immediately
  - Fixed in two weeks
- Case 1 –
  - Bypassed TrustZone
  - Extracted FDE key & 802.1X credentials
  - Can “Bring your own GPON”





Shared bug from SDK (ICMP, CMDi)

://bcmdrivers

48.4k files (401 ms)

Save ···

...c-rt-5.04axhnd.675x/bcmdrivers/Makefile · Makefile · master

```
  1  # File: bcmdrivers/Makefile
  2  #
  3  # Makefile for the Linux kernel modules.

121  #    whether or not the driver will be compiled
122  #  DIRECTORY is the directory (relative to bcmdrivers)
     where all the implX subdirectories
123  #    reside
```

SoC Vendor SDK: Un-stealthy Stealth Mode

```
$ strings libcms_core.so |grep -i icmp-type
-p icmp -m icmp --icmp-type 8
iptables -A INPUT -i %s -p icmp --icmp-type 8 -j DROP 2>/dev/null
iptables -A OUTPUT -o %s -p icmp --icmp-type 3/3 -j DROP 2>/dev/null
iptables -A OUTPUT -o %s -p icmp --icmp-type 11 -j DROP 2>/dev/null
iptables -D INPUT -i %s -p icmp --icmp-type 8 -j DROP 2>/dev/null
iptables -D OUTPUT -o %s -p icmp --icmp-type 3/3 -j DROP 2>/dev/null
iptables -D OUTPUT -o %s -p icmp --icmp-type 11 -j DROP 2>/dev/null
```

SoC Vendor SDK: Un-stealthy Stealth Mode

- `libcms_core` responsible for parsing config
- RFC 792 ICMP has multiple message types
- If blocking 8 (Echo):
  - Timestamp (13)
  - Redirect (5)
- Uncovers device if blocking 8 and not 13

```
$ strings libcms_core.so |grep -i icmp-type
-p icmp -m icmp --icmp-type 8
iptables -A INPUT -i %s -p icmp --icmp-type 8 -j DROP 2>/dev/null
iptables -A OUTPUT -o %s -p icmp --icmp-type 3/3 -j DROP 2>/dev/null
iptables -A OUTPUT -o %s -p icmp --icmp-type 11 -j DROP 2>/dev/null
iptables -D INPUT -i %s -p icmp --icmp-type 8 -j DROP 2>/dev/null
iptables -D OUTPUT -o %s -p icmp --icmp-type 3/3 -j DROP 2>/dev/null
iptables -D OUTPUT -o %s -p icmp --icmp-type 11 -j DROP 2>/dev/null
```

- Code Fields
  - Type 0 — Echo Reply
  - Type 1 — Unassigned
  - Type 2 — Unassigned
  - Type 3 — Destination Unreachable
  - Type 4 — Source Quench (Deprecated)
  - Type 5 — Redirect
  - Type 6 — Alternate Host Address (Deprecated)
  - Type 7 — Unassigned
  - Type 8 — Echo
  - Type 9 — Router Advertisement
  - Type 10 — Router Selection
  - Type 11 — Time Exceeded
  - Type 12 — Parameter Problem
  - Type 13 — Timestamp
  - Type 14 — Timestamp Reply
  - Type 15 — Information Request (Deprecated)
  - Type 16 — Information Reply (Deprecated)
  - Type 17 — Address Mask Request (Deprecated)
  - Type 18 — Address Mask Reply (Deprecated)
  - Type 19 — Reserved (for Security)
  - Types 20-29 — Reserved (for Robustness Experiment)
  - Type 30 — Traceroute (Deprecated)
  - Type 31 — Datagram Conversion Error (Deprecated)
  - Type 32 — Mobile Host Redirect (Deprecated)
  - Type 33 — IPv6 Where-Are-You (Deprecated)
  - Type 34 — IPv6 I-Am-Here (Deprecated)
  - Type 35 — Mobile Registration Request (Deprecated)
  - Type 36 — Mobile Registration Reply (Deprecated)
  - Type 37 — Domain Name Request (Deprecated)
  - Type 38 — Domain Name Reply (Deprecated)
  - Type 39 — SKIP (Deprecated)
  - Type 40 — Photuris
  - Type 41 — ICMP messages utilized by experimental mobili
  - Type 42 — Extended Echo Request
  - Type 43 — Extended Echo Reply
  - Types 44-252 — Unassigned
  - Type 253 — RFC3692-style Experiment 1
  - Type 254 — RFC3692-style Experiment 2

SoC Vendor SDK: Command Injection in CMS CLI

- Function intended for CLI; Used by vendor with volition

```
__pid_t __fastcall real_runCommandInShell(char *input)
{
  __pid_t v2; // r0
  __pid_t v3; // r4
  int i; // r4
  int v5; // r0
  char *all_args[8]; // [sp+0h] [bp-20h] BYREF

  v2 = fork();
  v3 = v2;
  if ( v2 == -1 )
  {
    sub_870C(3, "runCommandI
  }
  else if ( !v2 )
  {
    for ( i = 3; i != 51; ++
    {
      v5 = i;
      close(v5);
    }
    all_args[0] = "sh";
    all_args[1] = "-c";
    all_args[2] = input;
    all_args[3] = 0;
    sub_82EC("/bin/sh", all_args);
    sub_870C(3, "runCommandInShell", 116, "Should not have reached here!");
    exit(127);
  }
  return v3;
}
```

```
all_args[0] = "sh";
all_args[1] = "-c";
all_args[2] = input;
all_args[3] = 0;
sub_82EC("/bin/sh", all_args);
sub_870C(3, "runCommandInShell",
exit(127);
```

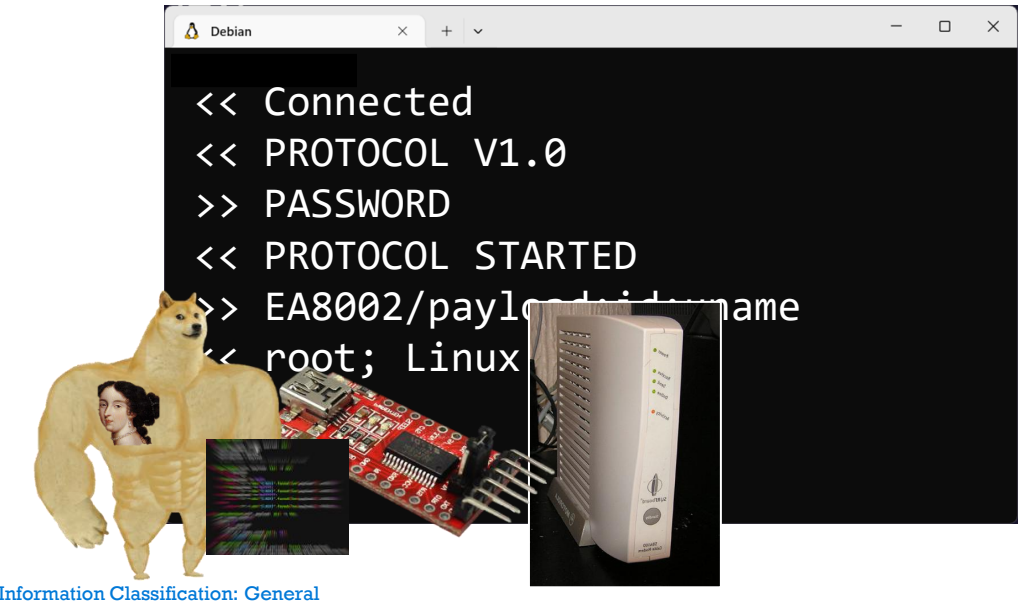
Shared bug from SoC Vendor SDK (ICMP, CMDi)

- Case 2 bugs were actually inside SoC vendor SDK:
  - ICMP –
    Allows discovery of device over Internet
  - Command Injection –
    Shared among all boards
- Reported, **fixed in 22 days**

Time	
2023-10-24	Consulted vendor about vulnerability program's scope
2023-10-27	Vendor is willing to take reports
2023-11-30	Vulnerability reported to vendor
2023-12-01	Vendor validated the reports and is working on a fix
2023-12-22	Vendor published private advisory with fix
2024-11-11	Informed vendor of intent to public disclosure

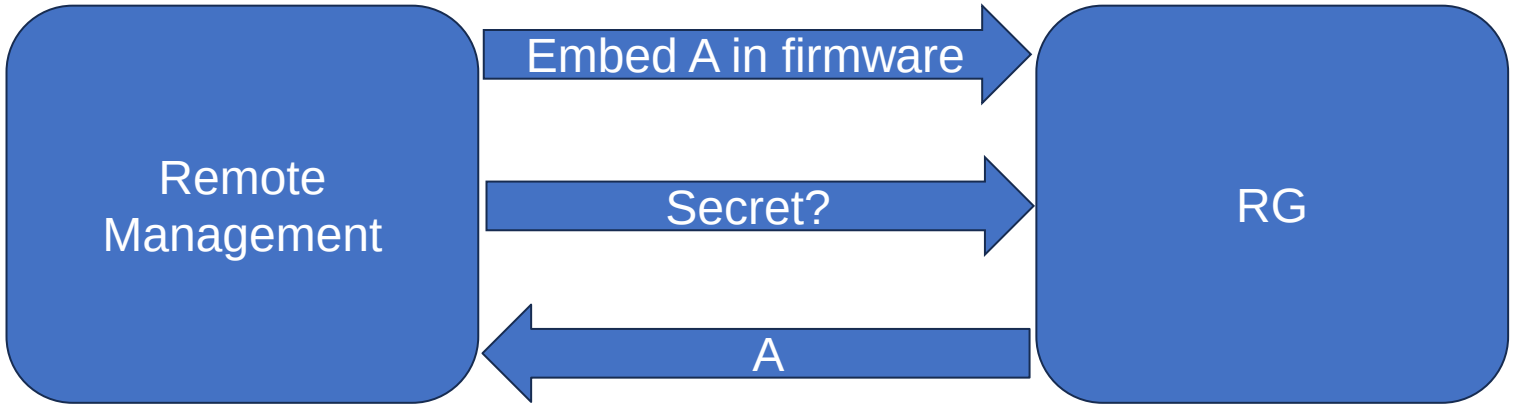
Summary

- Presented actual cases –
  - Case 1 – From the board to the ISP
  - Case 2 – TrustZone bypass leading to key extraction
  - SDK –
    - ICMP stealth mode allows discovery; shared command injection bug



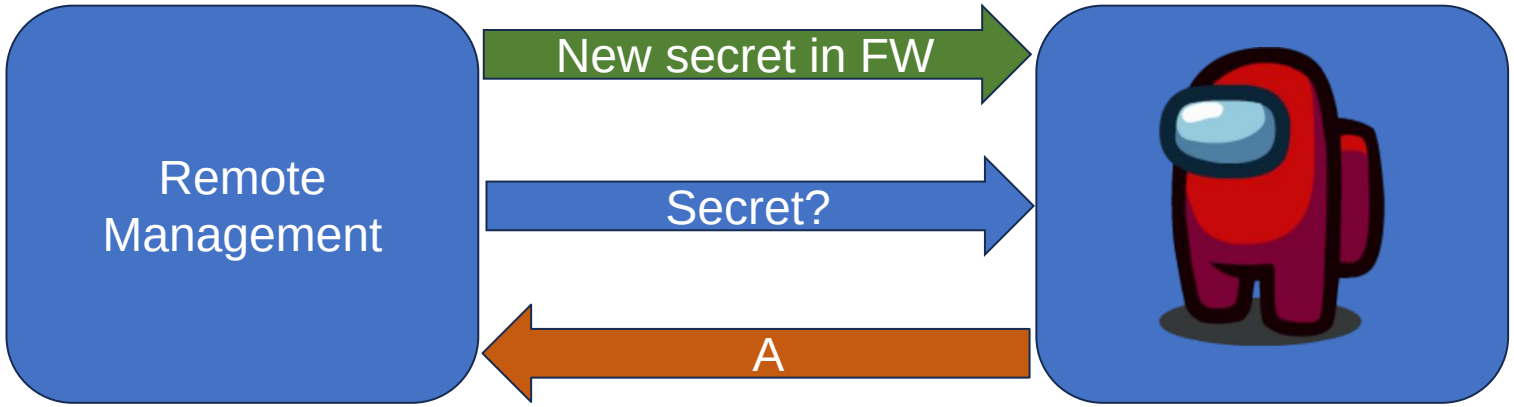
Residential Gateway Security Recommendations for End-users, Telecommunication Providers

- Prime Question:
  - How to detect compromise of RGs?
    - Adversaries could update RG with rootkits
    - No TrustZone/Secure Boot to validate running firmware
  - Integrity Check Canary



Residential Gateway Security Recommendations for End-users, Telecommunication Providers

- Prime Question:
  - How to detect compromise of RGs?
    - Adversaries could update RG with rootkits
    - No TrustZone/Secure Boot to validate running firmware
  - Integrity Check Canary



Residential Gateway Security Recommendations for End-users

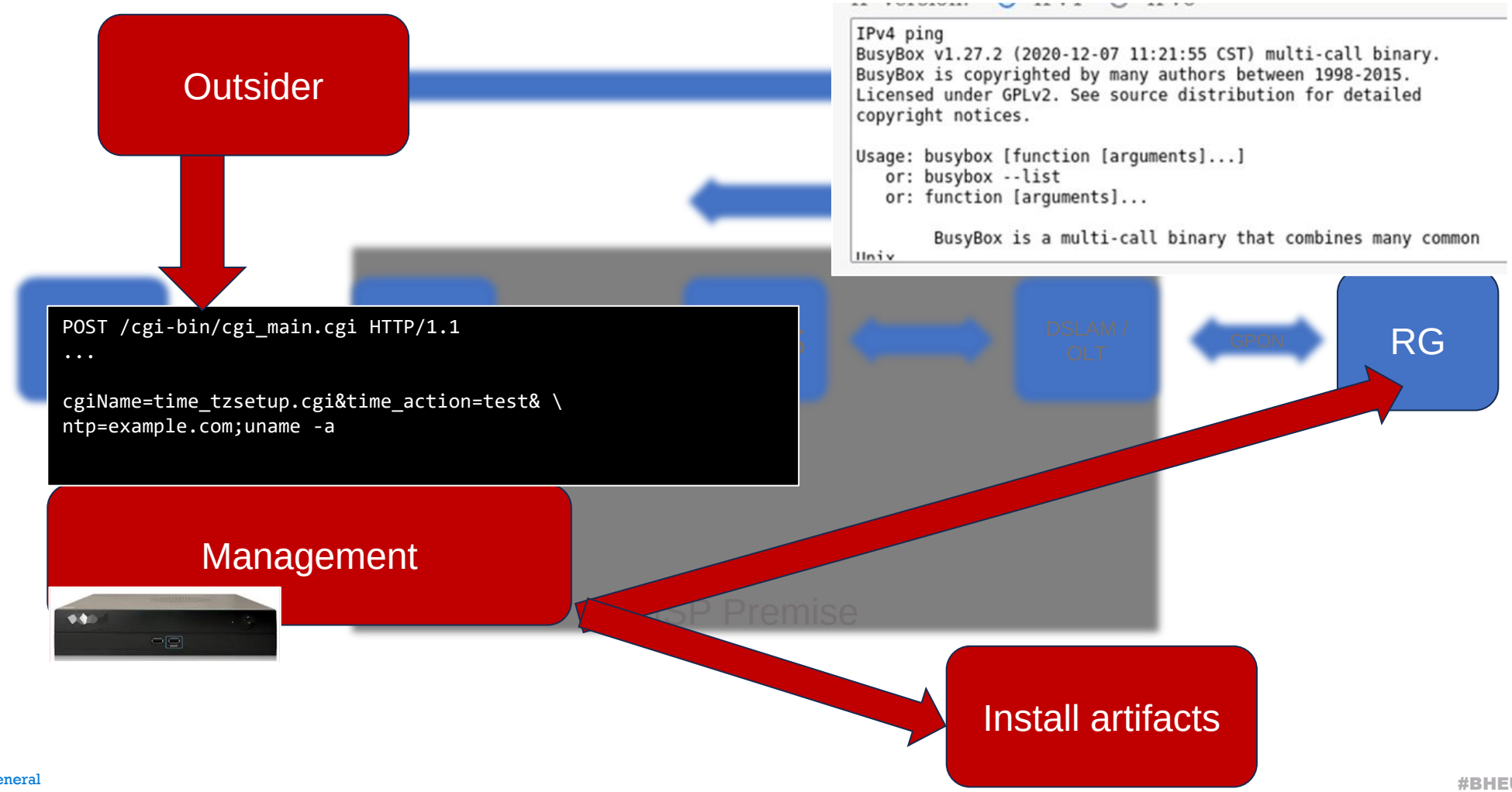
- Solution – End-users
  - Employ a gateway/firewall behind RG
  - Block private address range on incoming firewall
  - Configure RG as “modem mode” (disable routing)

Residential Gateway Security Recommendations for OEMs/Telecommunication Providers

- Detect abnormal network behavior in control plane

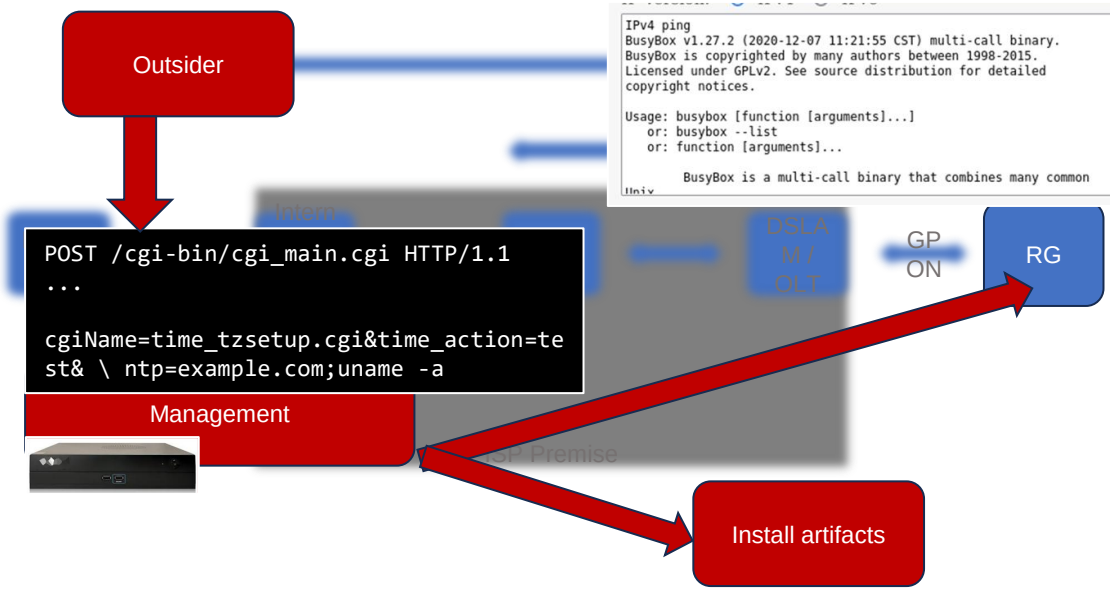
Residential Gateway Security Recommendations for OEMs/Telecommunication Providers

- Detect abnormal network behavior in control plane



Residential Gateway Security Recommendations for OEMs/Telecommunication Providers

- Detect abnormal network behavior in control plane
- Mandate baselines & standards –
  - Hardware-backed secure boot, proper use of TrustZone
  - FIPS 140-2, ISO/IEC 62443 4-2 Level >=2
  - EN 303 645
- Apply secure coding practices



Residential Gateway Security Recommendations for Upstream vendors

- Solution – Upstream vendors (SoC makers)
- OEMs may utilize SDK with volition
  - Employ secure coding practices
  - Employ defensive programming & ensure program robustness
- Employ SoCs with Secure Boot/TrustZone
- Demonstrate usage correctly in SDK
  - Encrypting flash with LUKS plain-text key is NOT proper encryption
  - Utilize TrustZone for critical cryptographic materials

Black Hat Sound Bytes / Takeaways

RGs and providers are unsafe

- RGs are lucrative targets, sheer in numbers, yet behind in terms of security.

Treat end-devices as hostile

- End-user device may be studied extensively by anyone. Risk assessment and modern defense options are important.

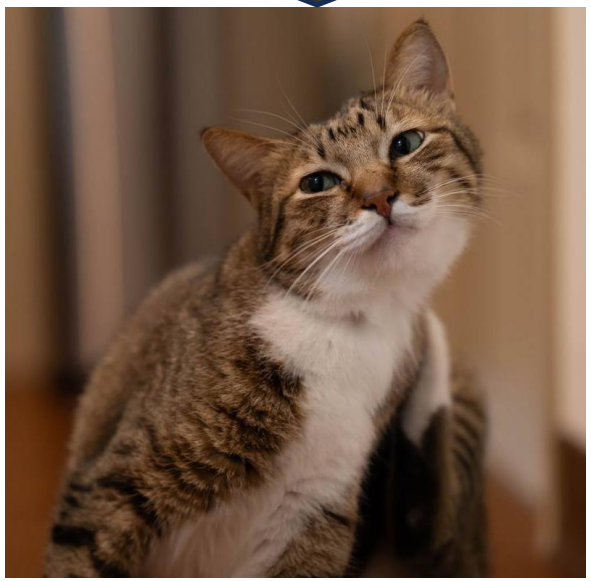
Supply chain security is difficult

- Supply chain security requires effort
- SoC vendors needs to prevent misuse & build better documentation

Questions?

logonfail

talun_yen@txone.com



Special thanks

Canaan Kao, TXOne Networks

Federico Maggi, Black Hat SCP

Whitepaper, disclosure and write-up coming soon – txone.com/blog