PwdLess:
Exploitation Tales from RouterLand

Cristofaro Mune
c.mune@pulse-sec.com
@pulsoid

Me

- Cristofaro Mune
  - Product Security Consultant
  - Security trainer
  - Research:
    - Fault injection
    - TEEs
    - White-box Cryptography
    - Device exploitation

Goals

- Discuss EOL devices:
  - Case study with actual data
- Challenge perceived relevance
  - Are we assessing it correctly?
- Publish findings and related vulnerabilities
- Share some tips, approach and methodology:
  - Hopefully useful for many young researchers at Nullcon!

LET ME INTRODUCE YOU TO …

D-Link DSL-2640B

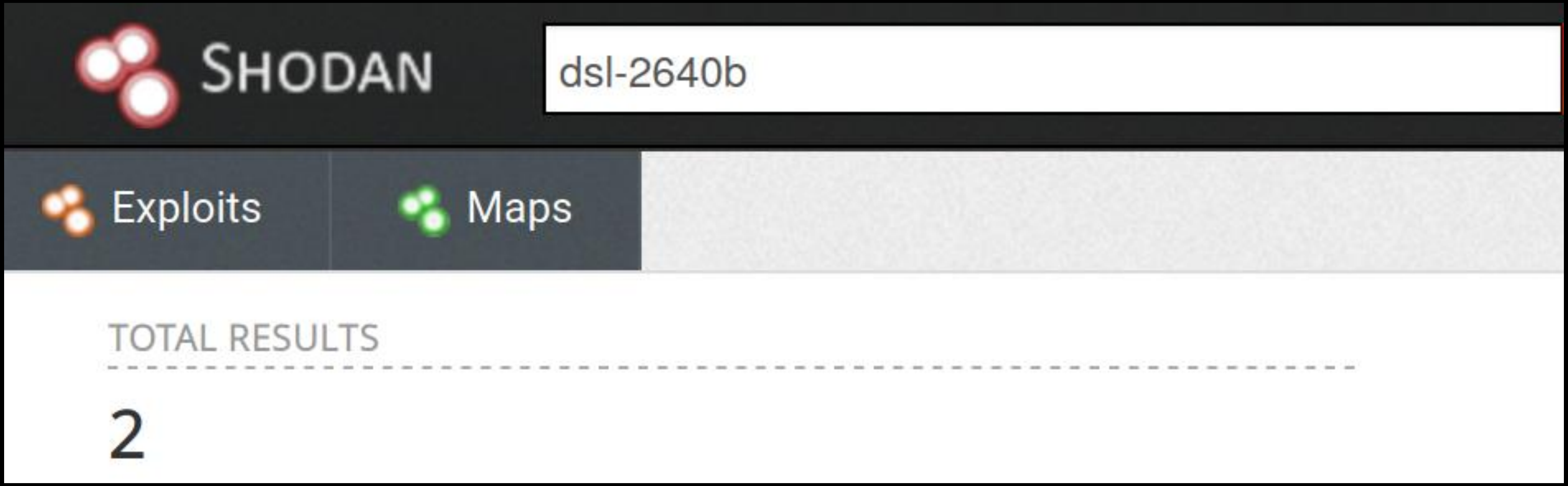
- D-Link ADSL Gateway (EU Version)
- HW:
  - Broadcom SoC
  - MIPS @256MHz (Big endian)
  - DDR: 4 Mbytes, Flash: 16 Mbytes
  - Max Upstream Data rate: 3.5 Mbps
- SW:
  - Version: EU_4.01B
  - Source code: Previous version available
  - Firmware image: available

Defaults
IP: **192.168.1.1**
User: **admin**
Password: **admin**

A brief overview

- Released: 2007
- Country-based firmware customization
  - Differences can be significant
- End-of-Life (EOL) since May 2013
- Only 1 CVE:
  - CVE-2012-1308: XSS in redpass.cgi
- No exploit mitigation in place

Shodan says…



dsl-2640b

TOTAL RESULTS

2

Virtually disappeared

IS THIS INTERESTING?

You may think…

- A 13 years old router
- 7 years in EOL
- Only 1 minor impact CVE
- Almost disappeared
- No exploitation fun

- Did I say OLD?
- Manufacturer: not interested
- Attackers: not interested
- Users: not interested
- Researchers: not interested

WHY ARE YOU EVEN HERE?

BECAUSE WE ARE GETTING IT WRONG

INTERESTINGLY

WRONG

2018-2019: Malware Campaign on routers

- Research/Advisory: “Ongoing DNSChanger campaign targeting consumer routers”
  - Detected by Bad Packets honeypots

- DNSChanger malware modifying router settings: 7 “waves”
  - Last wave detected on April 2019

- Also targeting DSL-2640B
  - With which vulnerability?

DNSChanger campaigns

- 2016: targeting D-Link DSL-2740R
  - EU version
- 2018: Malware extended to include DSL-2640B:
  - Exploited vulnerability seems to affect only specific country releases (Malaysia)

Target intentionally included in 2018

The vulnerability

- Unauthenticated configuration of DNS settings:
  - CGI module: redpass.cgi
- Exploit:
  - Released: 2017
  - No CVE assigned
  - SW version: GE_1.07

RESEARCH actually…exists.

D-Link (MANUFACTURER)

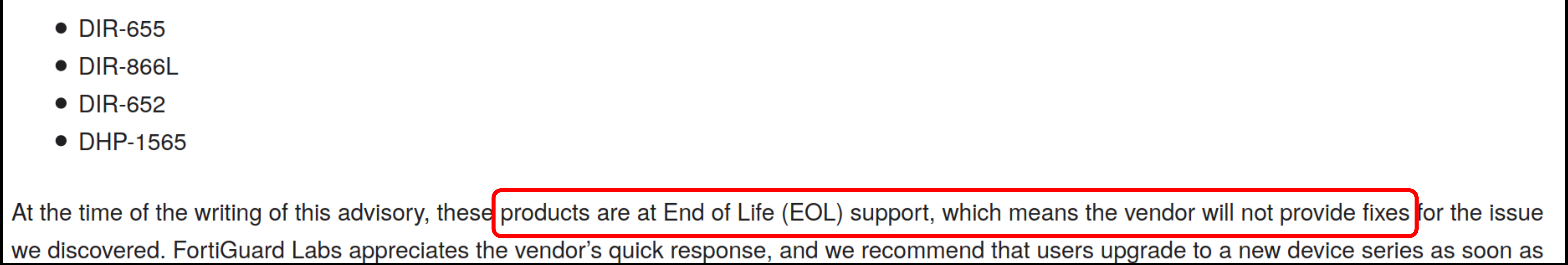
- 2016: Security advisory released
  - Along with a security fix for DSL-2740R

- 2019: security advisory update to include DSL-2640B
  - No security fix for DSL-2640B

2020: Still vulnerable

[10/2019]: Fortinet D-Link Routers RCE

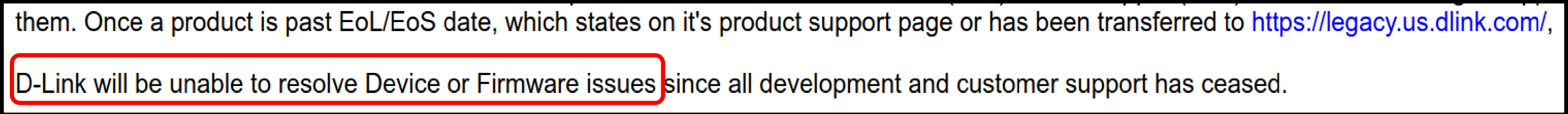
Fortinet Security advisory



- DIR-655
- DIR-866L
- DIR-652
- DHP-1565

At the time of the writing of this advisory, these products are at End of Life (EOL) support, which means the vendor will not provide fixes for the issue we discovered. FortiGuard Labs appreciates the vendor's quick response, and we recommend that users upgrade to a new device series as soon as

D-Link Support Announcement



them. Once a product is past EoL/EoS date, which states on it's product support page or has been transferred to https://legacy.us.dlink.com/,

D-Link will be unable to resolve Device or Firmware issues since all development and customer support has ceased.

EoL Policy in effect.

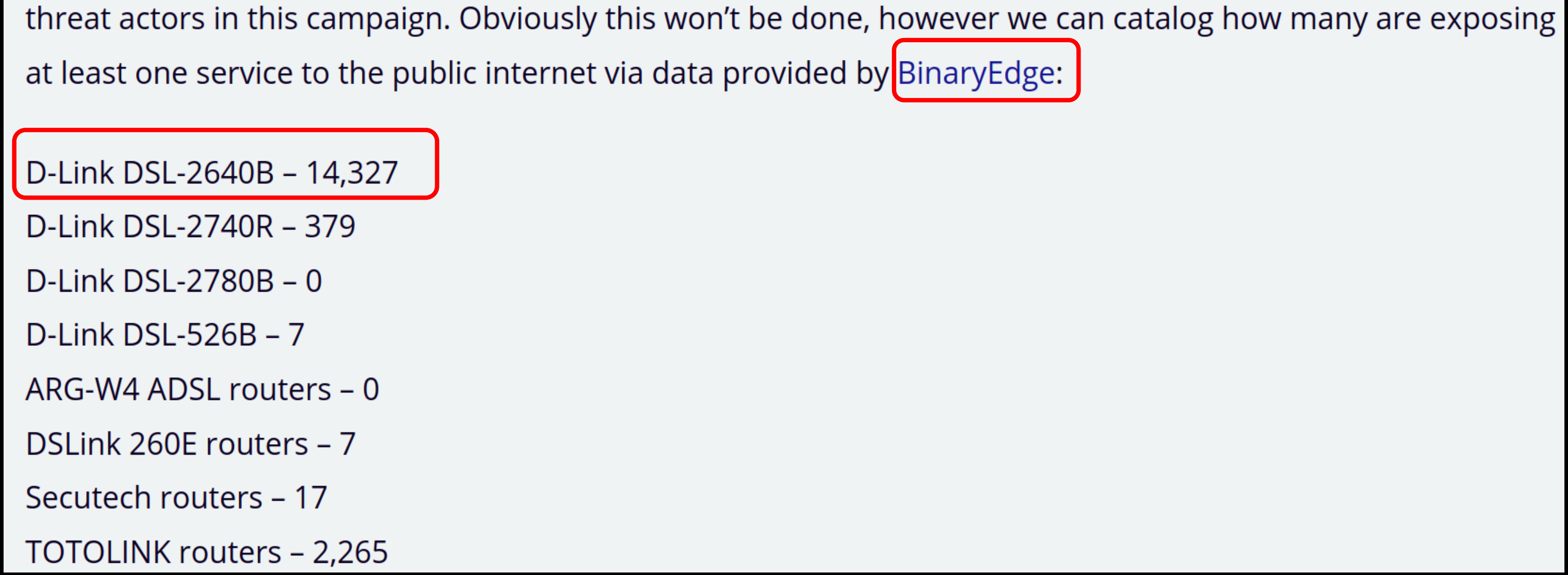
ATTACKERS?

- Exploits with a guaranteed infinite lifetime
  - How do we call them? NO-Days?

- Impact depends on number of connected devices.
  - Only 2 DSL-2640B (Shodan)
- Does not compute

Why an Attacker would even care to extend a malware?

Are we counting them wrong?



threat actors in this campaign. Obviously this won’t be done, however we can catalog how many are exposing at least one service to the public internet via data provided by BinaryEdge:

D-Link DSL-2640B – 14,327

D-Link DSL-2740R – 379

D-Link DSL-2780B – 0

D-Link DSL-526B – 7

ARG-W4 ADSL routers – 0

DSLink 260E routers – 7

Secutech routers – 17

TOTOLINK routers – 2,265

BinaryEdge is also “mapping” the Internet…

2019: BinaryEdge

- 14k+ DSL-2640B reachable over the Internet, AFTER 6 years EOL
  - Only devices with services exposed to Internet
  - Actual population may be larger

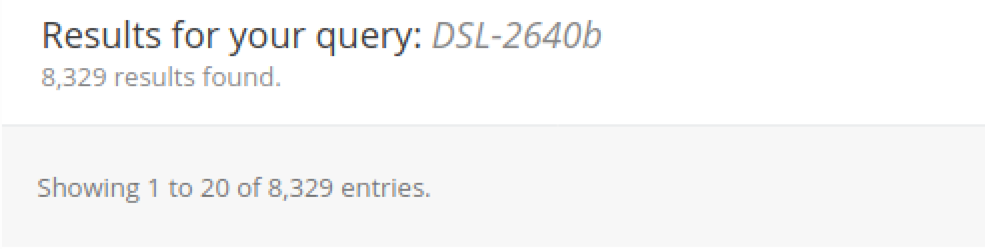
- Aggregated upstream bandwidth: ~49Gbps:
  - DDoS anyone?

Unexpected numbers

Now: 2020

- 8k+ DSL-2640B reachable over the Internet, AFTER 7 years EOL

- Aggregated upstream bandwidth: ~29Gbps



Results for your query: *DSL-2640b*

8,329 results found.

Showing 1 to 20 of 8,329 entries.

Numbers

- Very different results scale: 2 (Shodan) vs 14k (BinaryEdge)
  - A 10^4 factor!
- Completely change the perspective upon:
  - Attacker interest
  - Attack impact
  - Affected userbase
  - Ecosystem threats (DDoS)
  - Research impact
  - Exploits value

RELEVANCE

Some provoking thoughts…

- EOL?
- Not actively researched
- Low impact
- 14k devices (after 6yrs EOL!)
- No exploitation fun?

- Attackers: infinite lifetime vulns
- Is CVE counting a good metric?
- Are we even counting correctly?
- Users: Large userbase affected
- Exploits could be still valuable

INTERESTING

!=

RELEVANT

Summary

- Old router
- Expected to be virtually disappeared
- Still largely alive after 7 years without support
- Actively exploited by attackers
- Potential for scaled attacks
- Cannot be “removed” from the Internet
- We cannot count its population reliably
- We have no idea how vulnerable it can be…

…may apply to many EoL device out there...

HOW BAD CAN IT GET?

RESEARCH

OF LOST PASSWORD AND EXPLOITS…

-

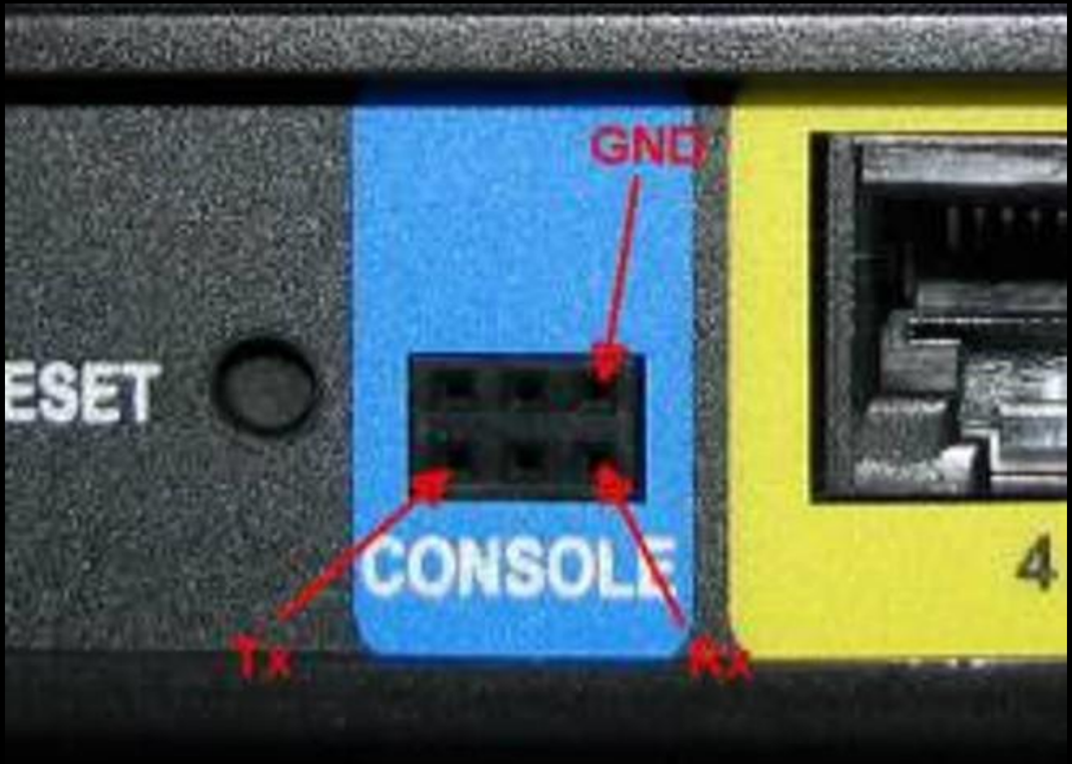
A ROUTERLAND TALE

PwdLess: how it started

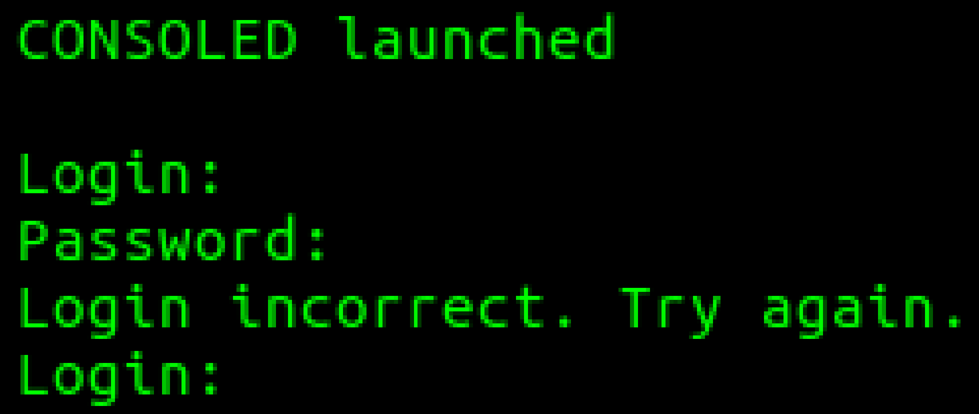
- Lost a DSL-2640B password
- Password needed
- Configuration reset to be avoided:
  - Device not under my control
  - No config backup available
- Had some notes from a previous reconnaissance:
  - I always perform one when new device arrives

Step 0: Serial console…of course.

- Conveniently available
- No surprises:
  - 3.3V TTL
  - 115200
  - 8-N-1
- Just get a FTDI USB-TTL 3.3V cable…



No way in.

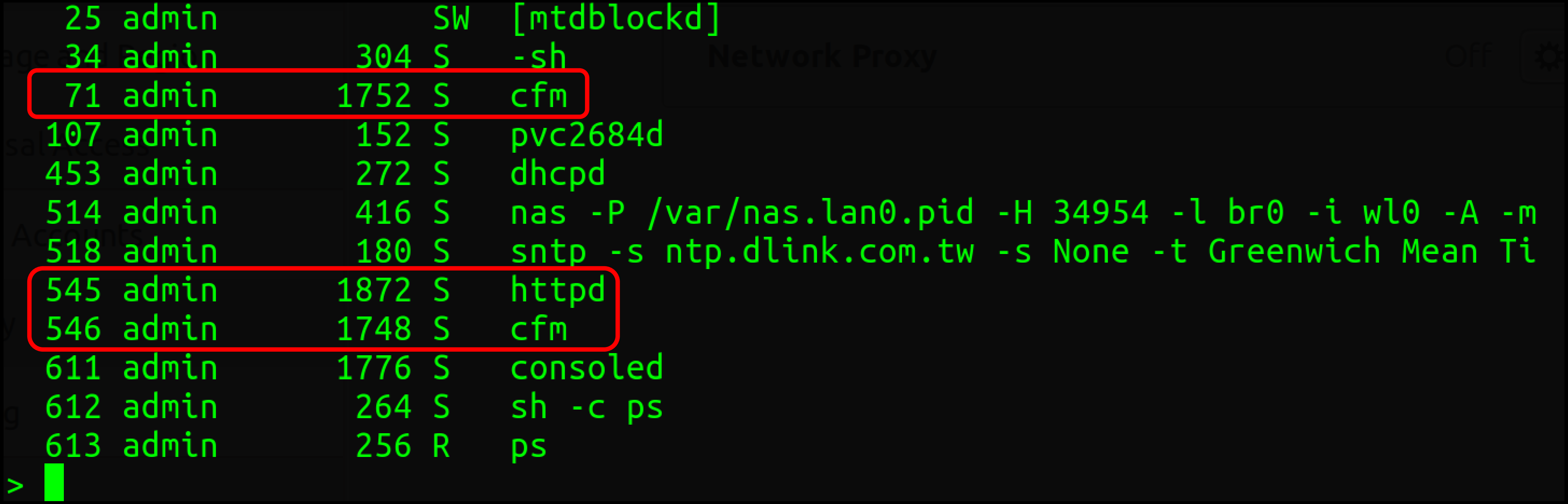


```
CONSOLED launched

Login:
Password:
Login incorrect. Try again.
Login:
```

We need a vulnerability.

Notes: Processes



```
   25 admin              SW   [mtdblockd]
   34 admin          304 S    -sh
   71 admin         1752 S    cfm
  107 admin          152 S    pvc2684d
  453 admin          272 S    dhcpd
  514 admin          416 S    nas -P /var/nas.lan0.pid -H 34954 -l br0 -i wl0 -A -m
  518 admin          180 S    sntp -s ntp.dlink.com.tw -s None -t Greenwich Mean Ti
  545 admin         1872 S    httpd
  546 admin         1748 S    cfm
  611 admin         1776 S    consoled
  612 admin          264 S    sh -c ps
  613 admin          256 R    ps
>
```

ps

SW overview

- Very stripped down console
  - Missing: ls, netstat, wget, curl, ftp, bash, find, stat,…
  - Minimal shell via busybox

- cfm: started at boot
  - Implements all the relevant router services
- Relevant services
  - http
  - device configuration
  - …more

TIP: Listing files without ls

- echo:
  - echo *: Lists current directory
  - echo bin/*: lists ./bin content

- Other useful commands (not available on DSL-2640B)
  - find –maxdepth 1
  - vim .
- A few more here

Notes: Available services

```
>cat /proc/net/udp
  sl  local_address rem_address   st tx_queue rx_queue tr tm->when
retrnsmt   uid  timeout inode
  69: 00000000:0045 00000000:0000 07 00000000:00000000 00:00000000
00000000     0         0 1316
 106: 00000000:FDEA 00000000                                000000
00000000     0         0 1297
 107: 00000000:13EB 00000000:0000 07 00000000:00000000 00:00000000
00000000     0         0 1352 2 8060a900
 108: 00000000:13EC 00000000:0000 07 00000000:00000000 00:00000000
00000000     0         0 1351 2 805a2060
```

Port UDP/65002

(No netstat)

First approach

- Analysis: UDP port used for device configuration
  - Proprietary protocols
  - Likely prone to vulnerabilities
  - Already exploited a few in the past

- Started VERY dumb fuzzing:
  - cat /dev/urandom | | nc -u 192.168.1.1 65002
  - …while downloading firmware

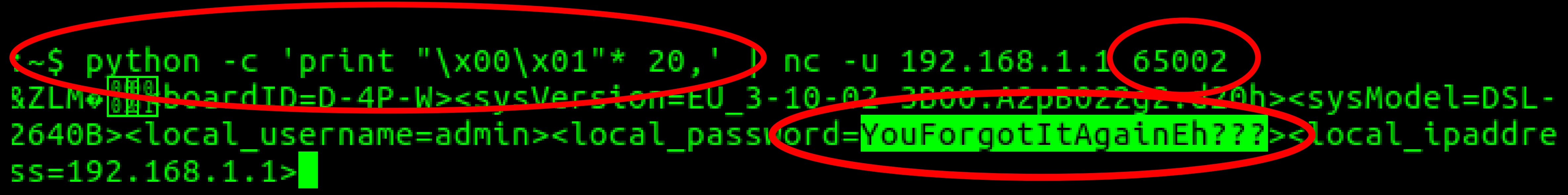
KISS: Cheap and easy go first!

Exceeding expectations

- Expected a crash:
  - Device reset visible on console
- Got MUCH more…
  - Password printed on console
- Unexpected: Did this REALLY work???
  - Had no traffic recording on.
- Restarted fuzzing with tcpdump (…and tons of disbelief)
  - Repeatable!

CVE-2020-9275

Want a pass?



```
:~$ python -c 'print "\x00\x01"* 20,' | nc -u 192.168.1.1 65002
&ZLM◆boardID=D-4P-W><sysVersion=EU_3-10-02-3B00.A2pB022g2.d20h><sysModel=DSL-
2640B><local_username=admin><local_password=YouForgotItAgainEh???><local_ipaddre
ss=192.168.1.1>
```

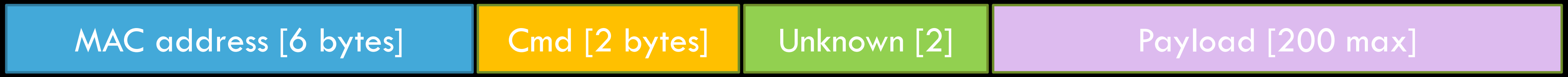
…just ask politely

Device configuration

- Service implemented by cfm
  - pcApplication function
- Allows configuration settings read/write
  - E.g. user and password
- No authentication:
  - Device MAC address (???) required for most commands

Remote Credentials Exfiltration

Request format



MAC address [6 bytes]

Cmd [2 bytes]

Unknown [2]

Payload [200 max]

- Cmd: “\x00\x01”
  - Unauthenticated retrieval of system info
  - Admin user and password
- Everything else is ignored

DEMO

Analysis

- Administrative credentials can be obtained
  - Full device control via web GUI
  - Device re-flashing possible. Malicious firmware upload

- Very likely exploitable from LAN/WiFi interfaces only

- Unsuitable for ‘browser pivoting’ :
  - UDP
  - Credentials in response payload (Cross-origin request)

NEXT STAGE

CURIOSITY

Research questions

- Was anything remote possible?
  - WAN interface
  - Browser pivoting
- Is a password needed?
- Potential for cross-device vulnerabilities?
  - Shared codebases
- Everybody loves RCE and shells…

FTWR

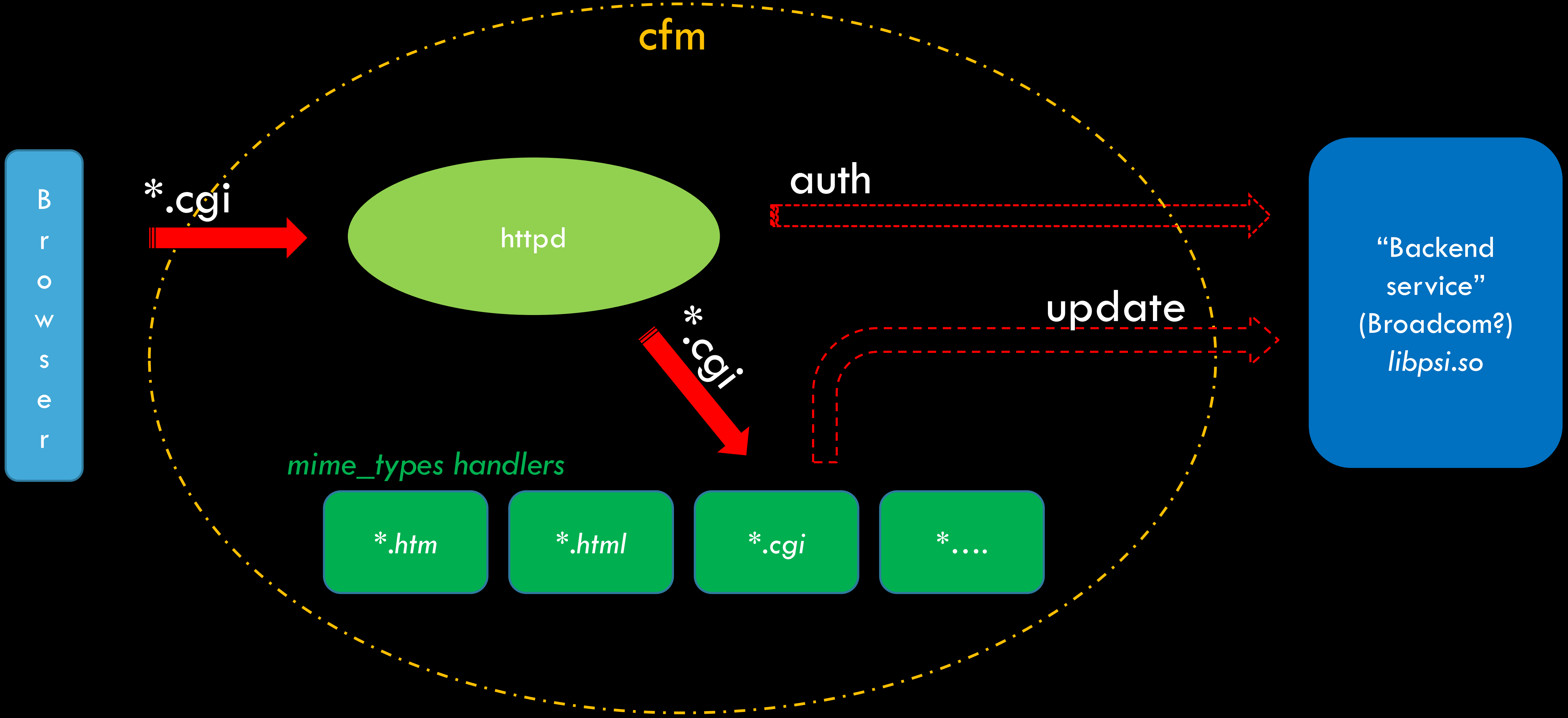
Firmware analysis

- Firmware: 3.1 Mbytes compressed
- Typical structure
  - CFE
  - Kernel
  - SquashFS filesystem (*lzma*)
- Extraction:
  - Binwalk: OK for bootloader and kernel. Yields empty files for filesystem
  - Sasquatch: works out of the box for the filesystem

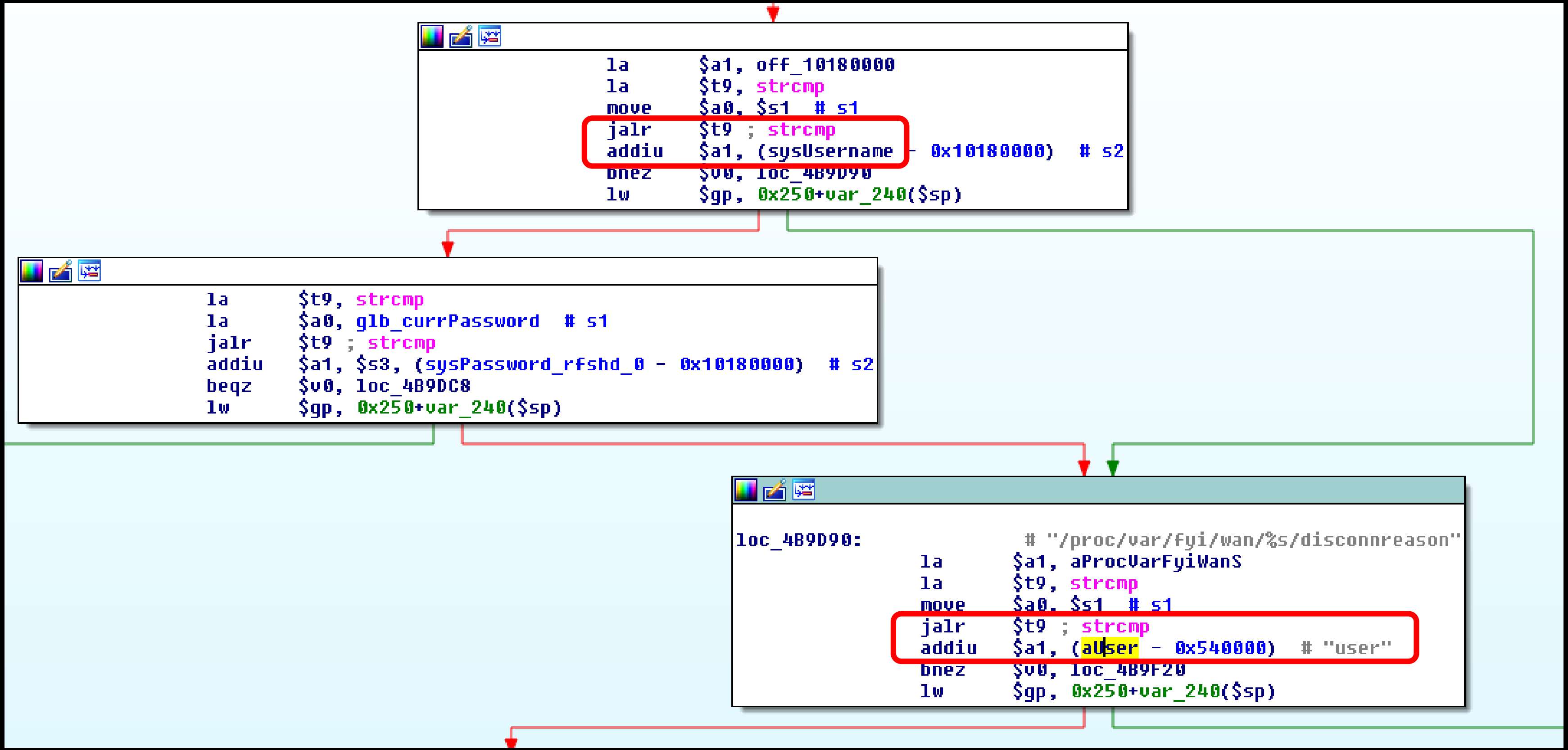
Filesystem Exploration: cfm

- One large binary for all services: cfm
  - 3.1 Mbytes uncompressed, stripped
- Only available in binary form:
  - Not present in GPL source code
- Implements web server:
  - Modified *micro_httpd*
- Authentication via an external library
  - *libpsi.so* (Broadcom?)

Web services: pwd update (example)



Ghost in the shell…



```
                la      $a1, off_10180000
                la      $t9, strcmp
                move    $a0, $s1  # s1
                jalr    $t9 ; strcmp
                addiu   $a1, (sysUsername - 0x10180000)  # s2
                bnez    $v0, loc_4B9D90
                lw      $gp, 0x250+var_240($sp)
```

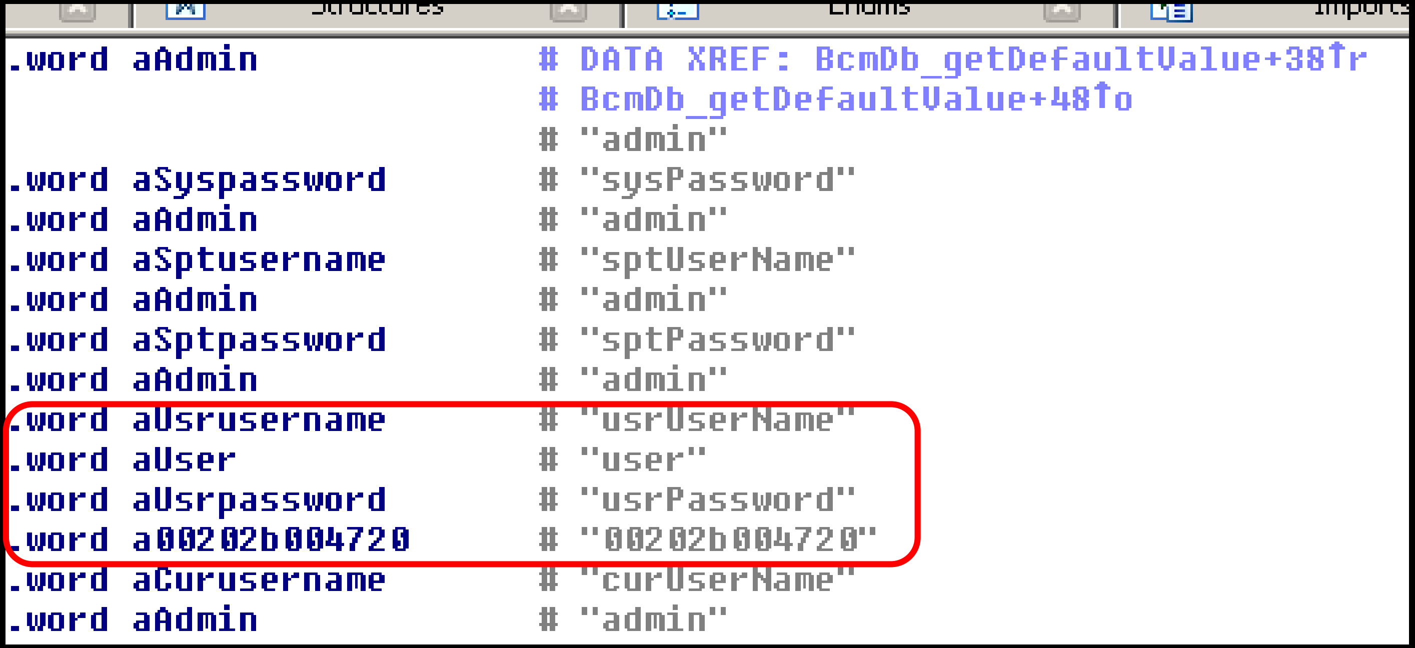
```
                la      $t9, strcmp
                la      $a0, glb_currPassword  # s1
                jalr    $t9 ; strcmp
                addiu   $a1, $s3, (sysPassword_rfshd_0 - 0x10180000)  # s2
                beqz    $v0, loc_4B9DC8
                lw      $gp, 0x250+var_240($sp)
```

```
loc_4B9D90:                  # "/proc/var/fyi/wan/%s/disconnreason"
                la      $a1, aProcVarFyiWanS
                la      $t9, strcmp
                move    $a0, $s1  # s1
                jalr    $t9 ; strcmp
                addiu   $a1, (aUser - 0x540000)  # "user"
                bnez    $v0, loc_4B9F20
                lw      $gp, 0x250+var_240($sp)
```

Auth is possible also for user “user”

CVE-2020-9279

Hard-coded privileged account



```
.word aAdmin                  # DATA XREF: BcmDb_getDefaultValue+38↑r
                              # BcmDb_getDefaultValue+48↑o
                              # "admin"
.word aSyspassword            # "sysPassword"
.word aAdmin                  # "admin"
.word aSptusername            # "sptUserName"
.word aAdmin                  # "admin"
.word aSptpassword            # "sptPassword"
.word aAdmin                  # "admin"
.word aUsrusername            # "usrUserName"
.word aUser                   # "user"
.word aUsrpassword            # "usrPassword"
.word a00202b004720           # "00202b004720"
.word aCurusername            # "curUserName"
.word aAdmin                  # "admin"
```

- libpsi.so provides system defaults to authentication objects
  - "User" password default value: 00202b004720

DEMO

Analysis

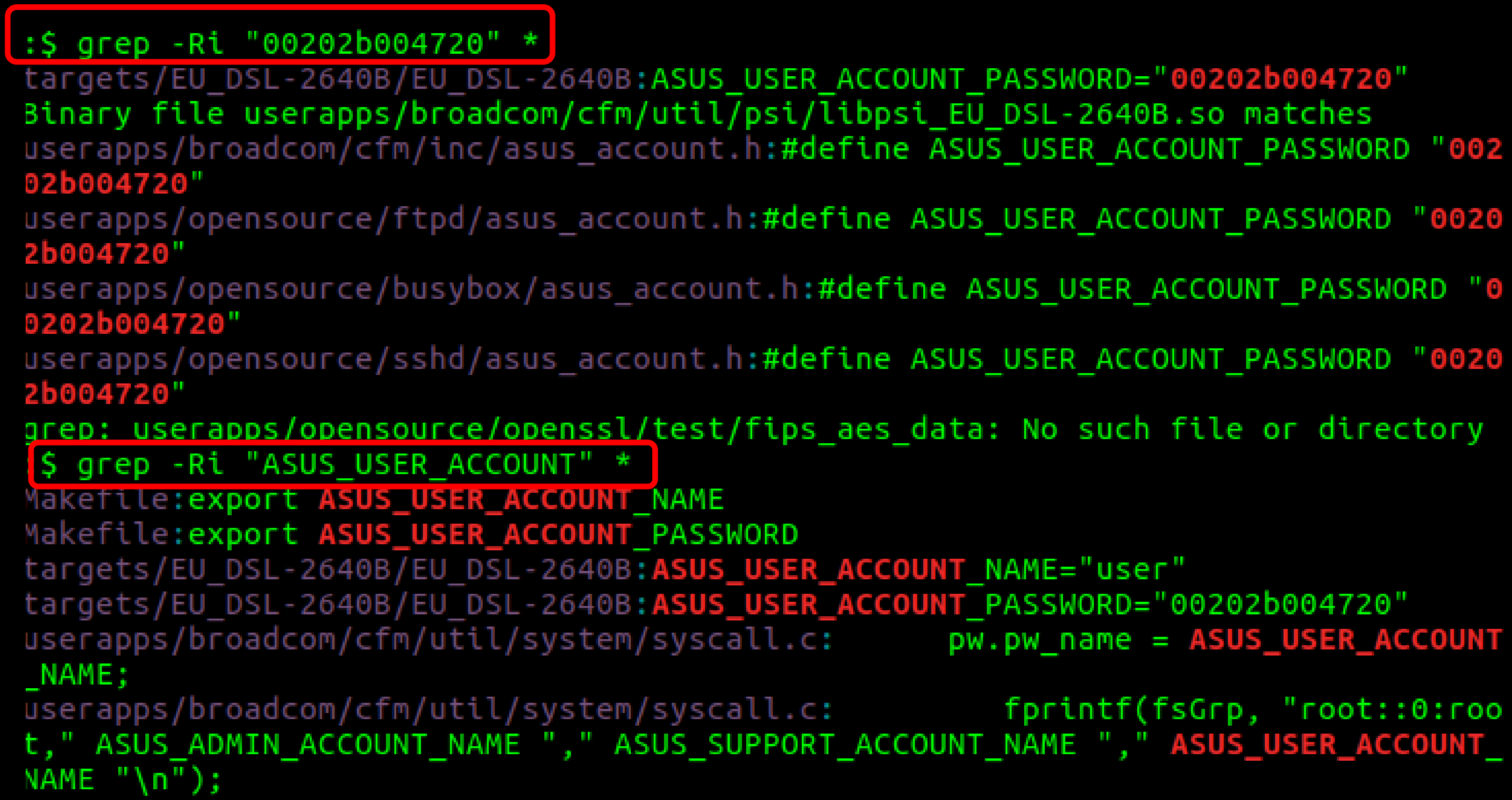
- User basically has admin privileges:
  - No privilege management
- Account hard-coded in library
- Password cannot be easily changed:
  - Authentication objects defaults COULD be updated
  - Not possible from Web GUI
- Maybe possible via direct calls to:
  - CGI modules (HTTP request)?
  - Object methods? (runtime exec required)

Impact

- Credentials scope:
  - LAN/WIFI: Yes (HTTP) . WAN: Likely not
- Attractive vuln:
  - Resilient: almost unchangeable
  - Can be used in browser pivoting attacks
- Also valid for ftp, telnet, ssh
- Maybe applicable to:
  - all DSL-2640B?
  - More recent models? (e.g. DSL-2641B)

We can sleep well in case of password loss ;-)

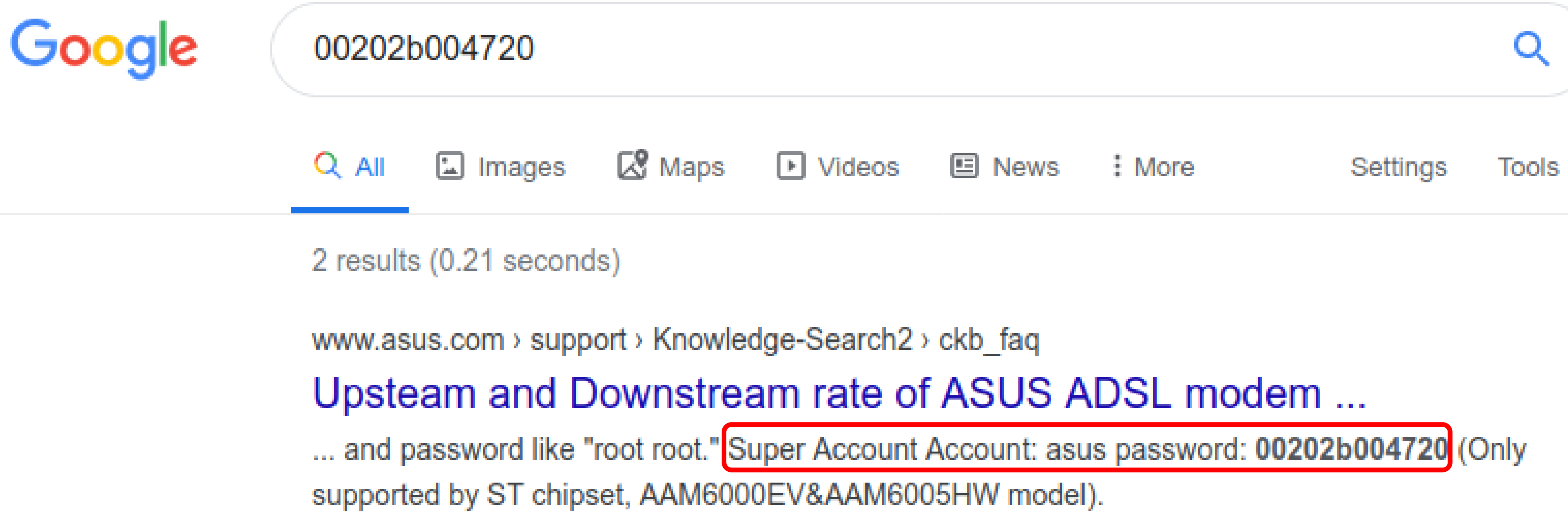
Observations: Code



```
:$ grep -Ri "00202b004720" *
targets/EU_DSL-2640B/EU_DSL-2640B:ASUS_USER_ACCOUNT_PASSWORD="00202b004720"
Binary file userapps/broadcom/cfm/util/psi/libpsi_EU_DSL-2640B.so matches
userapps/broadcom/cfm/inc/asus_account.h:#define ASUS_USER_ACCOUNT_PASSWORD "002
02b004720"
userapps/opensource/ftpd/asus_account.h:#define ASUS_USER_ACCOUNT_PASSWORD "0020
2b004720"
userapps/opensource/busybox/asus_account.h:#define ASUS_USER_ACCOUNT_PASSWORD "0
0202b004720"
userapps/opensource/sshd/asus_account.h:#define ASUS_USER_ACCOUNT_PASSWORD "0020
2b004720"
grep: userapps/opensource/openssl/test/fips_aes_data: No such file or directory
$ grep -Ri "ASUS_USER_ACCOUNT" *
Makefile:export ASUS_USER_ACCOUNT_NAME
Makefile:export ASUS_USER_ACCOUNT_PASSWORD
targets/EU_DSL-2640B/EU_DSL-2640B:ASUS_USER_ACCOUNT_NAME="user"
targets/EU_DSL-2640B/EU_DSL-2640B:ASUS_USER_ACCOUNT_PASSWORD="00202b004720"
userapps/broadcom/cfm/util/system/syscall.c:      pw.pw_name = ASUS_USER_ACCOUNT
_NAME;
userapps/broadcom/cfm/util/system/syscall.c:         fprintf(fsGrp, "root::0:roo
t," ASUS_ADMIN_ACCOUNT_NAME "," ASUS_SUPPORT_ACCOUNT_NAME "," ASUS_USER_ACCOUNT_
NAME "\n");
```

Present in source code…

Observations: Internet



00202b004720

2 results (0.21 seconds)

www.asus.com › support › Knowledge-Search2 › ckb_faq

Upsteam and Downstream rate of ASUS ADSL modem ...

... and password like "root root." Super Account Account: asus password: **00202b004720** (Only supported by ST chipset, AAM6000EV&AAM6005HW model).

It's a feature.

Some questions…

- Why an "ASUS SuperUser account" (?) is present on a D-Link router?

- Supply chain magic?
- code reuse?
- Malicious intent? (Unlikely, IMHO)
  - Account was visible in plain sight in source code.
  - No visible effort for hiding a powerful "backdoor"

NEXT STAGE

PASSWORD RESTORE?

CVE-2020-9278

Passwords are overrated

- *rebootinfo.cgi*
  - Reboot router
- *ppppasswordinfo.cgi*
  - save PPP password and reboot
- *qosqueue.cmd?action=savReboot*
  - Guess...
- *logout.html*
  - Troll Mode: Logout ANYbody logged in. From ANY IP.

No authentication required

Unauthenticated Configuration Reset

- *restoreinfo.cgi*
  - Full router configuration reset

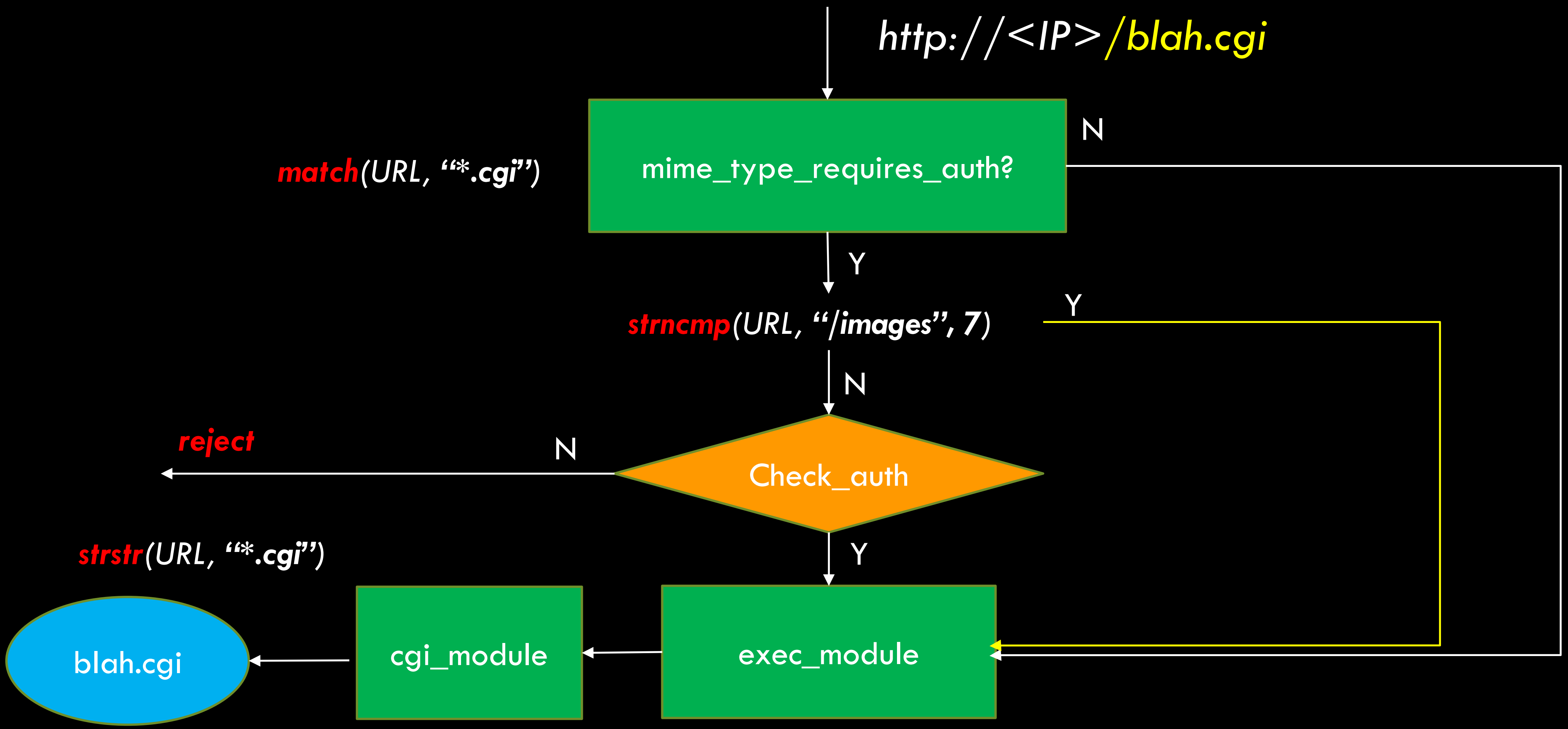
- Admin password is restored to initial value: admin
- Useful if target is on default IP address:
  - Still reachable after the exploit

DEMO

NEXT STAGE

PASSWORD NEEDED?

Authentication flow



CVE-2020-9277

CGI Authentication bypass

- *match(URL, "*.cgi")* → *".cgi"* must be at the end
- *strncmp(URL, "/images", 7)* → *"/images"* must be at the start
  - No null byte match.
  - String can continue
- *strstr(URL, "module_name")* → *"module_name"* can be anywhere

URL: /images/makemeasandwich.cgi → No auth

DEMO

Who needs password anyway?

- Inconsistent logic in URL checks
- Any cgi module can be executed
  - No auth required. Just prepend "*/images*"
- Complete Pwnage:
  - Change Admin Passwords
  - Firmware upload?
  - Be creative.
- Suitable for browser pivoting attacks

NEXT STAGE

RCE

do_cgi buffer overflow

- *do_cgi* module has a trivial stack overflow
  - Buffer for module name: *0x420 bytes,* but…
  - HTTP Request can be up to 0x2710 bytes long
  - Post-authentication vulnerability

- Can be reached unathenticated via CVE-2020-9277
- No exploit mitigations:
  - ASLR, NX, Stack cookies,…

Exploitation strategy

- Overflow module name in URL:
  - Overwrite saved $ra on the stack
- Shellcode in *Host* header
  - In URL it gets mangled by sanity checks (../, /.. , /../)
  - Hardcoded buffer values
    - No aim to portability here.
  - Reverse TCP Connect Shell
- No cache-incoherency
- Shell is limited
  - Better payload by calling internal APIs.

DEMO

Browser pivoting?

- Suitable for browser pivoting

- Not so trivial to achieve:
  - Return address must be in URL
  - Browser mangles non-printable chars in request (URL-encoding)
  - No mapping at printable addresses

- Still a few ideas to test…

NEXT STAGE

CONCLUSION

Ecosystem

- EoL devices pose an ecosystem problem:
  - Bound to increase every year
- No established way to address the problem
- Perception of relevance is bound to numbers
- No unambiguous way for counting
  - Relevance may be underestimated
  - Impact underestimated

Research

- Disclosed a few vulnerabilities
- Some tips for IoT (black-box) security testing:
  - Quick attack surface exploration
  - Vuln identification & exploitation
- An old target can still:
  - Provide food for thoughts
  - Yield unpatchable vulnerabilities
  - Be useful for educational purposes

NEXT STAGE

CHANGE ROUTER

PULSE

Thank you!

Cristofaro Mune
c.mune@pulse-sec.com
@pulsoid