



black hat[®]
EUROPE 2024
DECEMBER 11-12, 2024
BRIEFINGS

When (Remote) Shells Fall Into The Same Hole: Rooting DrayTek Routers Before Attackers Can Do It Again

Stanislav Dashevskyi, Francesco La Spina

The researchers



Stanislav Dashevskiy



Vulnerability Research

- ▶ Focus on vulnerabilities against managed and unmanaged devices (IT/IoT/IoMT/OT)
- ▶ 200+ vulnerabilities discovered in last 5 years

VEDERE LABS

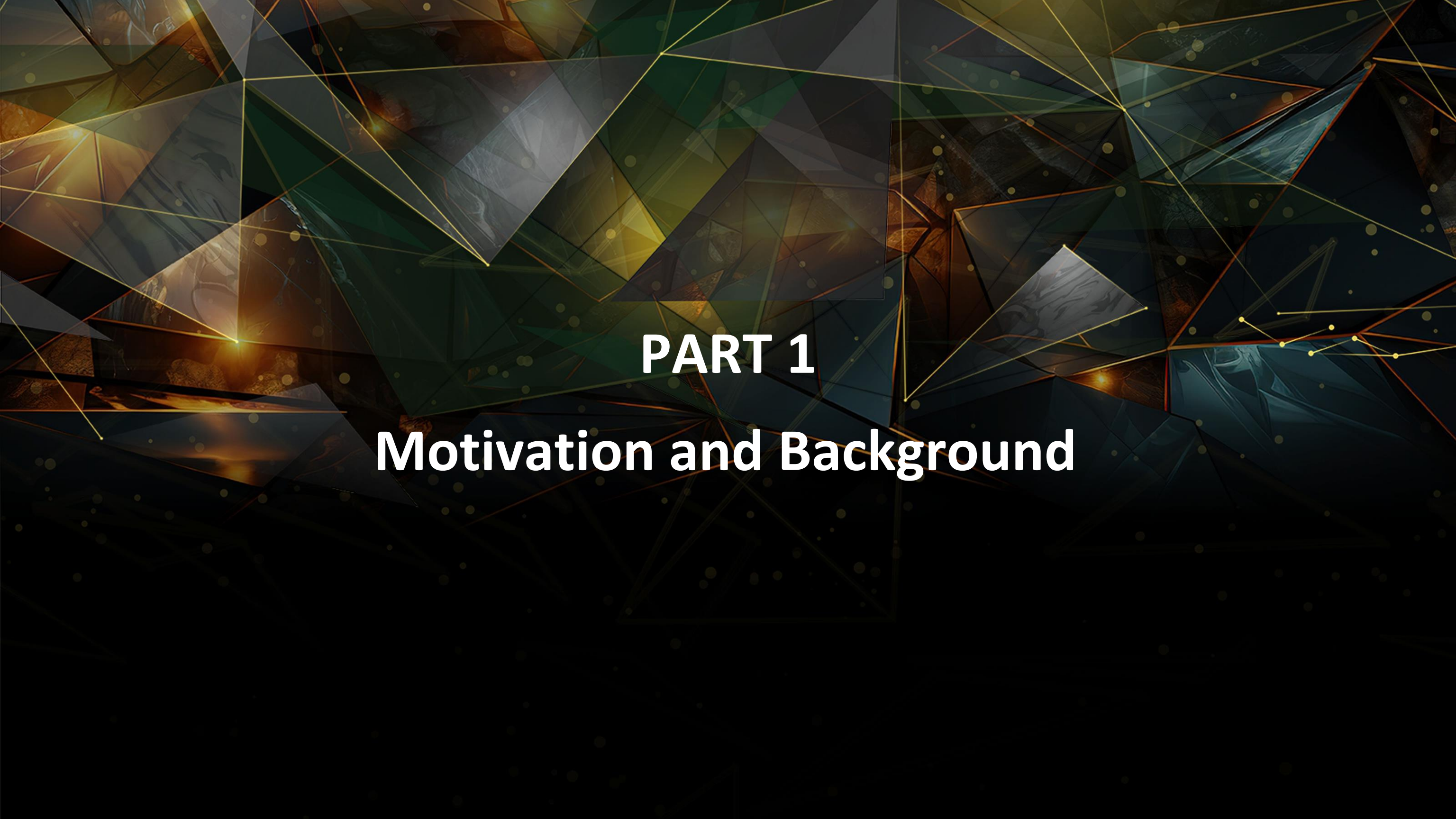


Threat Reports

- ▶ Manual and automatic analysis of malware samples collected via customer telemetry and other sources



Francesco La Spina

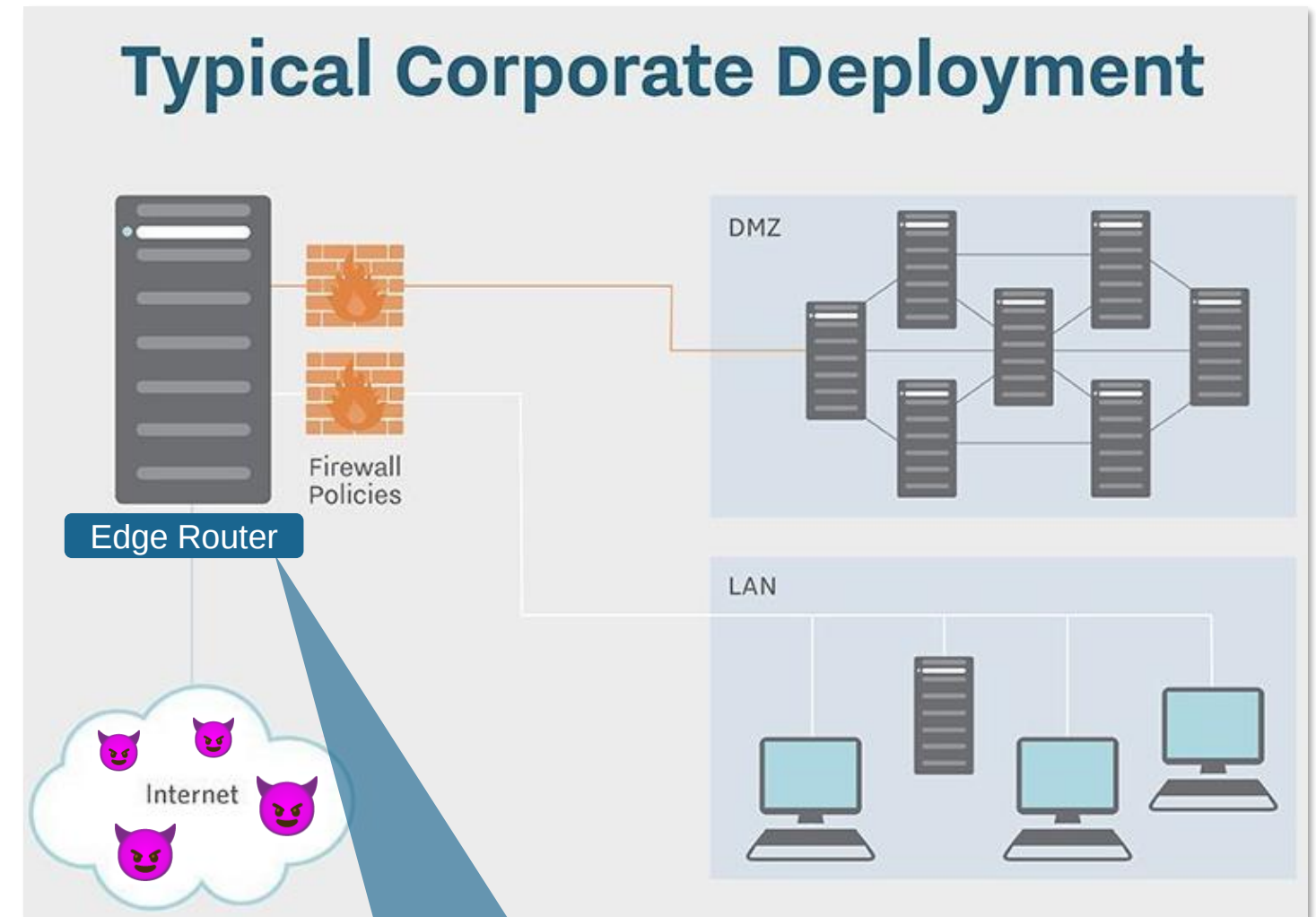


PART 1

Motivation and Background

It's rough around the edges

- Last year we did research on Sierra Wireless gateways and found critical vulnerabilities
- We also looked at firmware of five different IoT/OT edge routers and it did not look good...
- ...**lack of binary hardening, outdated software components, known vulnerabilities, “custom” security patches, default credentials...**
- **Edge devices serve the threat actors as perfect entry points into businesses**

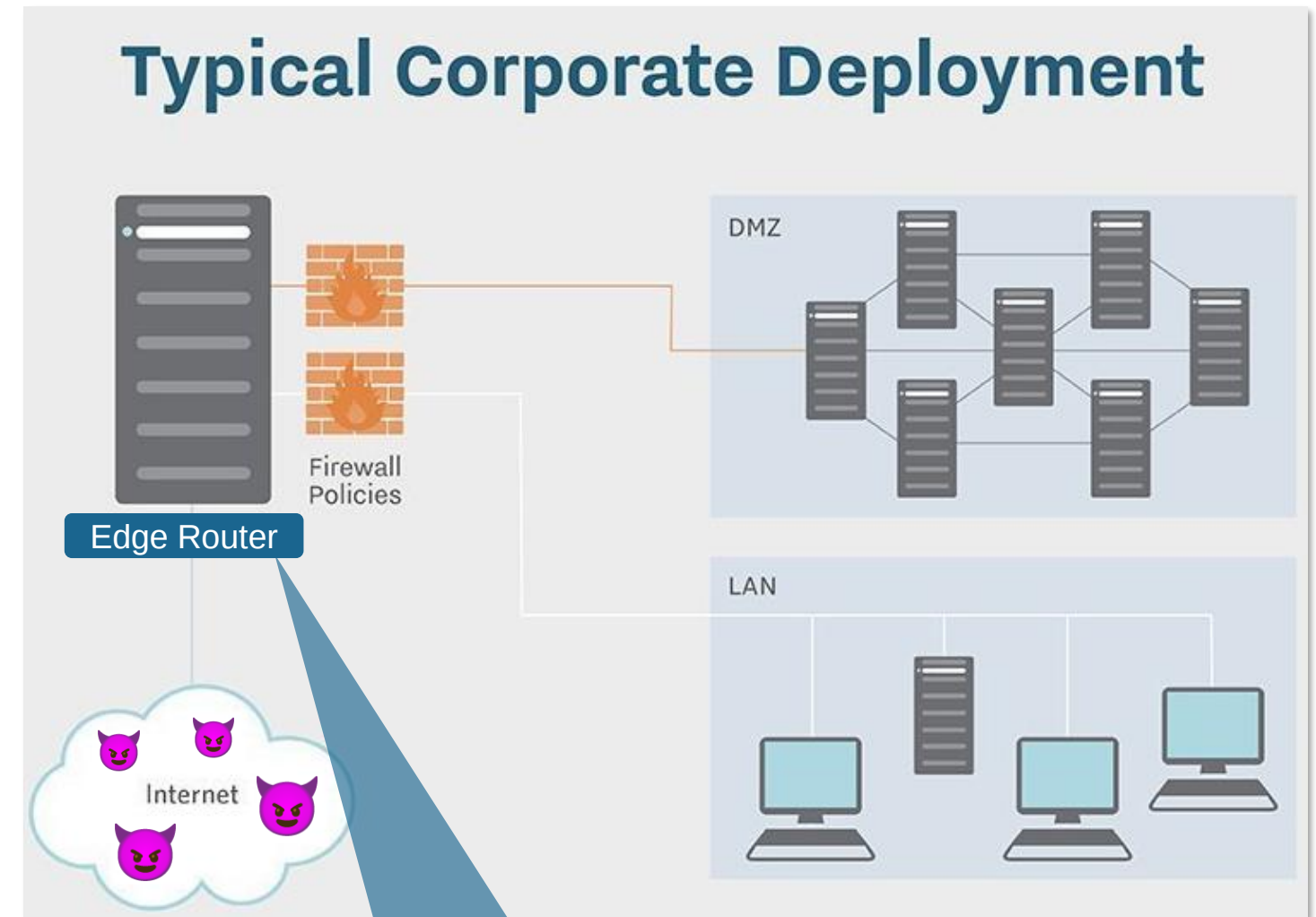


* <https://www.techtarget.com/searchnetworking/definition/edge-router>

You are here

It's rough around the edges (continued)

- We have chosen a vendor, a seemingly **bullet-proof target with lots of past research - DrayTek**
- > 4 years of active patching and frequent security advisories
- With proven interest from threat actors
- **Remote unauthenticated root on the host OS via a trivial buffer overflow in the guest...**
- **And it took us about a month to do it**



* <https://www.techtarget.com/searchnetworking/definition/edge-router>

You are here

What's DrayTek?

- A well-known **Taiwanese** manufacturer of networking equipment and management systems (**founded in 1997**)
- From **simple SOHO routers** to **complex VPN concentrators** used by businesses

DrayTek

Secure

The routers run on our proprietary operating system – the DrayOS, it's a closed system which provides the layer of security that the business network needs.

ISP Approved

DrayTek has always been partnered with ISPs throughout the world. We not only provide the hardware but also ensure full compatibility. We make sure the routers are making the most out of the local Internet service.

Feature-Rich

The router provides VPN, firewall, content filtering, load-balancing, triple-play, VoIP, bandwidth management, and a lot more. It's an integrated networking solution perfect for a small business.

SUCCESS STORY

Cross Nation Health Care System in Saudi Arabia

Discover how our network solution supported Cross Nation Health Care System in establishing stable and secure connectivity across 76+ branches.

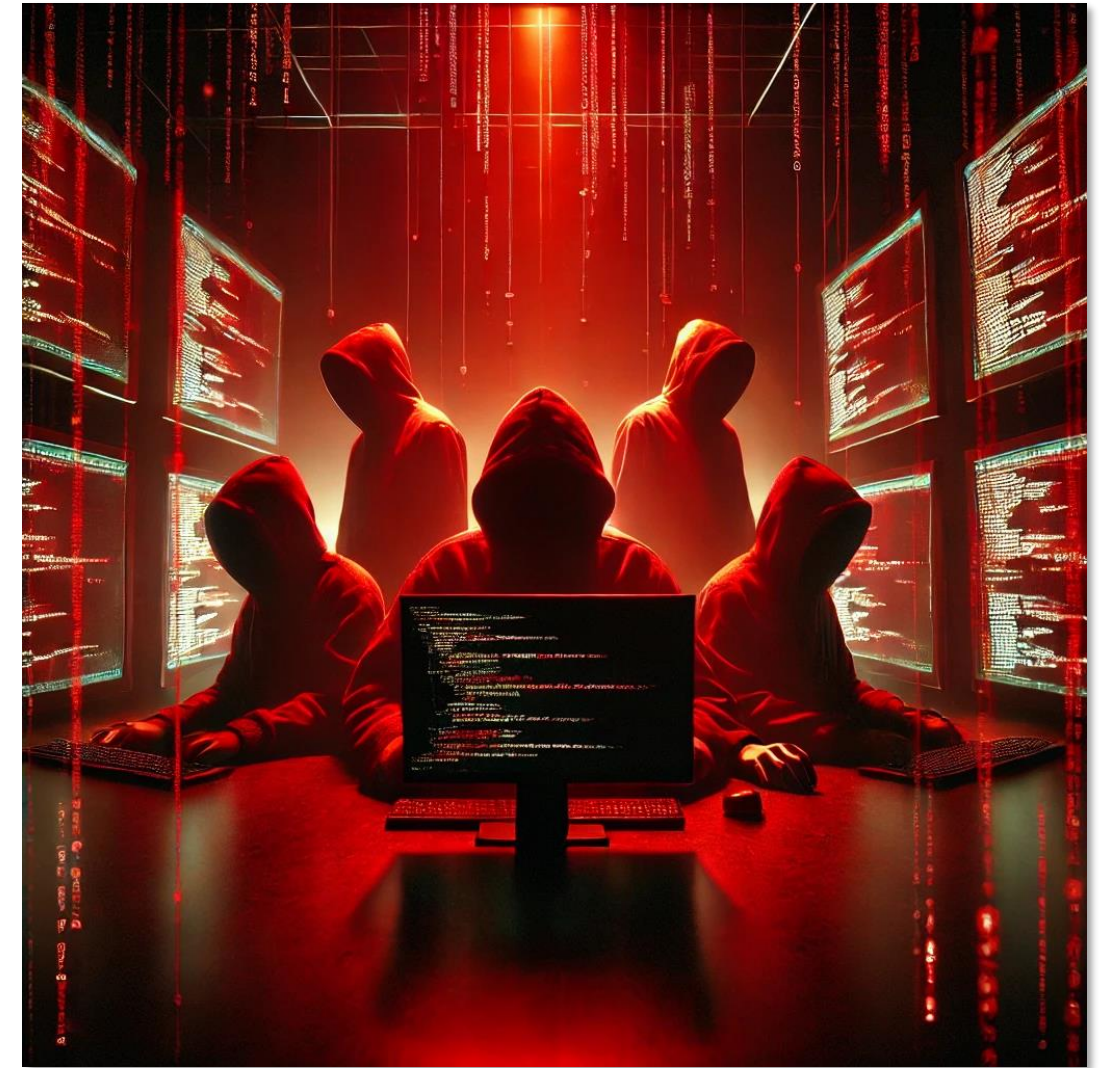
Why DrayTek? Researchers like it

- **13** security advisories **since 2018** (excluding ours) with over **100** historical **CVEs**
- Typically, a sign of a mature security team. Yet, new findings keep popping up
- **Emulate it until you make it! Pwning a DrayTek Router before getting it out of the box – Philippe Laulheret. HEXACON (2022).**
- **Detecting persistent threats on DrayTek devices – Octavio Gianatempo, Gastón Aznarez. DEF CON 32 (2024)**



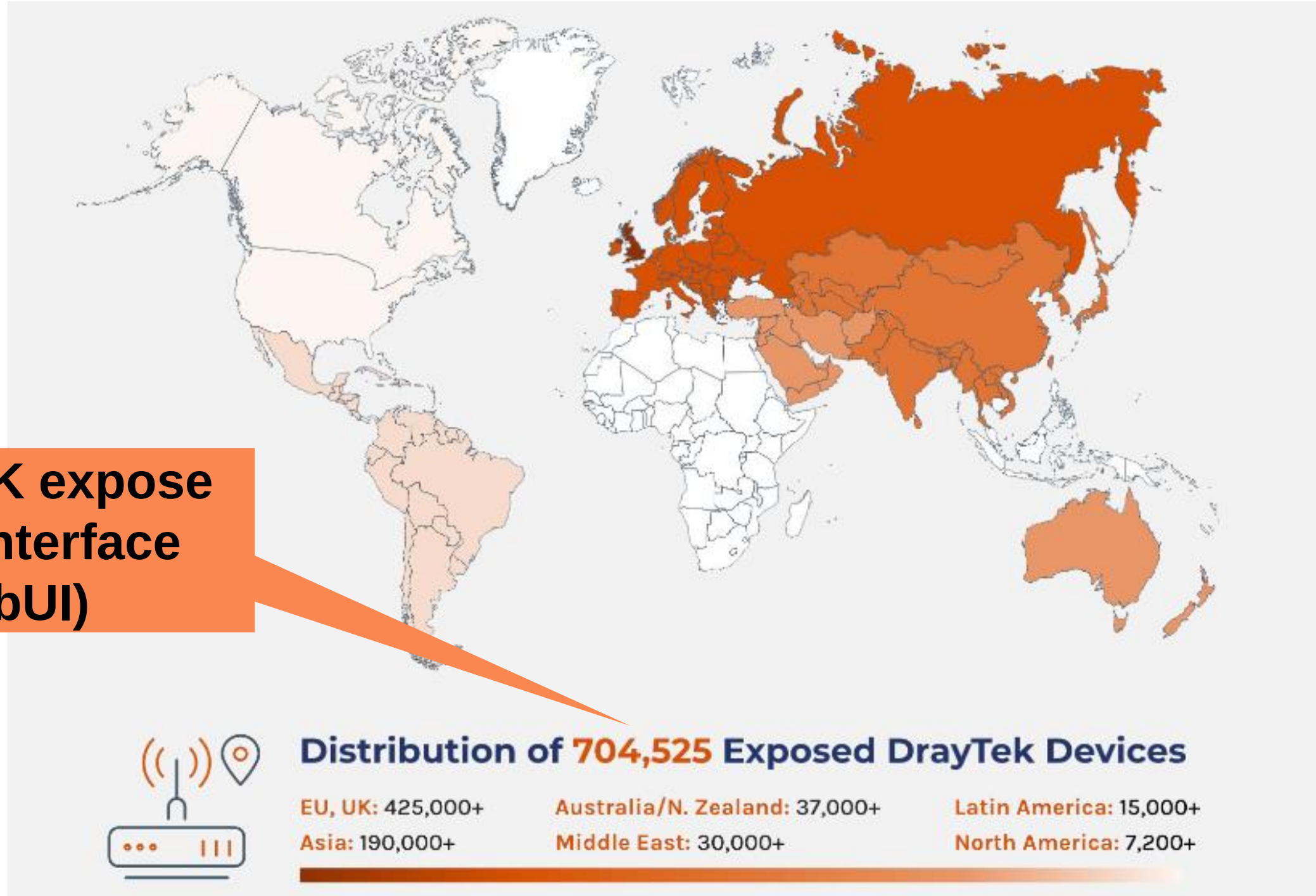
Why DrayTek? Threat actors love it

- In 2018, threat actors changed DNS settings on DrayTek routers using a zero-day vulnerability (**CVE-2018-20872**)
- **CVE-2020-8515** was exploited by Chinese APTs — as part of the **ZuoRAT** malware campaign
- In 2022-2023, some end-of-life DrayTek Vigor routers were targeted by the Chinese malware **HiatusRAT**
- Around the same time, DrayTek devices were targeted by another threat actor known as **Volt Typhoon**.
- Sept. 2024, the FBI announced it had taken down a botnet exploiting three historical CVEs on DrayTek assets (**CVE-2023-242290**, **CVE-2020-15415**, and **CVE-2020-8515**)



Why DrayTek? Just look at the numbers

**Over 400K expose
admin interface
(WebUI)**



Why DrayTek? More about the exposure

The ISP networks with the largest concentrations of DrayTek devices (Censys*):

ASN	AS_Name	Organization	Country	Scale	Host Count
3462	HINET Data Communication Business Group	HINET	Taiwan	Major ISP	41,969
31655	ASN-GAMMATELECOM	Gamma Telecom	U.K.	Significant Telecom Provider	35,866
2856	BT-UK-AS BTnet UK Regional network	British Telecommunications	U.K.	Major ISP	31,959
45899	VNPT-AS-VN VNPT Corp	Vietnam Posts and Telecommunications Group	Vietnam	Major ISP	31,561
5413	AS5413	Daisy Communications	U.K.	Significant Telecom Provider	21,275
13037	ZEN-AS Zen Internet – UK	Zen Internet	U.K.	Medium-sized ISP	13,147
18403	FPT-AS-AP FPT Telecom Company	FPT Telecom	Vietnam	Major ISP	12,132
7552	VIETEL-AS-AP Viettel Group	Viettel Group	Vietnam	Major ISP	11,756
1136	KPN KPN National	KPN	Netherlands	Major ISP	9,921
3320	DTAG Internet service provider operations	Deutsche Telekom AG	Germany	Major ISP	7,732

* <https://community.censys.com/censys-rapid-response-37/censys-rapid-response-exposed-draytek-vigor-routers-cve-2024-41592-189>

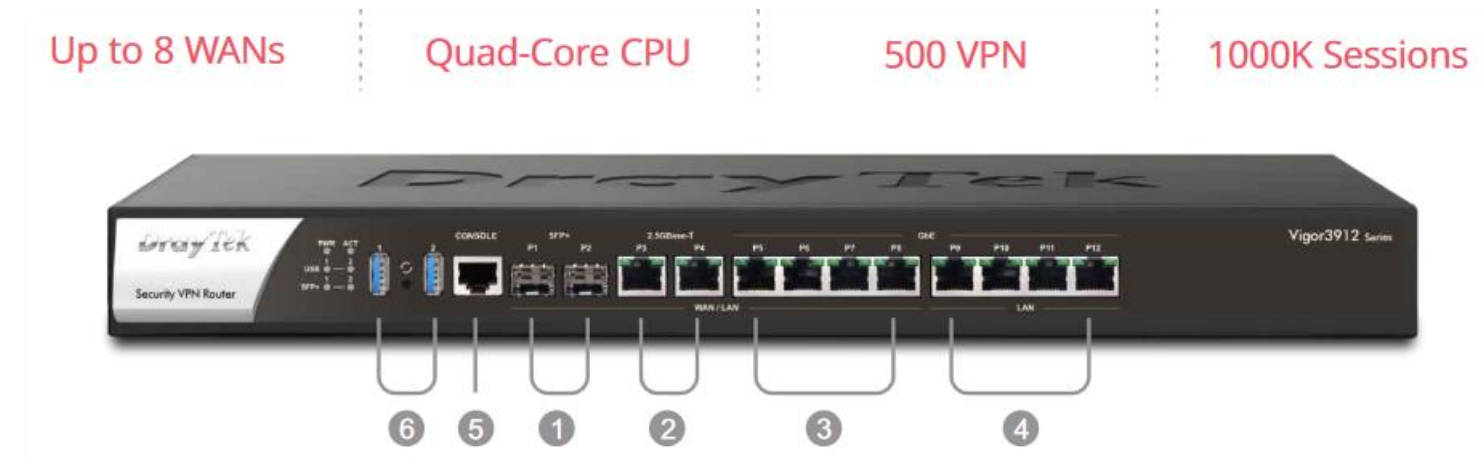
Why did we do it? Well...

- We typically look at devices / software overlooked by others – better ROI for us
- **Several critical RCEs** and command injections found between 2020 and 2023
- Seems like **a lot of patching** has been already done
- **Can we possibly find any more issues? Would attackers still be firing at the same target?**



Which device to look at?

- DrayTek offers different kinds of devices: from simple bare-metal routers that run RTOS, to **complex security appliances**
- **Modern routers are mini-servers*** - lots of resources, Linux, virtualization support...
- Rich OS features enable LOTL** techniques, and VMs backfire*** if not done right
- **A vulnerability in a complex and popular device** -> largest ROI for threat actors



DrayTek Vigor 3910 / 3912

* [Securing Network Appliances: New Technologies and Old Challenges](#) – Vladyslav Babkin, BHUSA 2024

** [People's Republic of China State-Sponsored Cyber Actor Living off the Land to Evade Detection](#) – CISA advisory, 2023

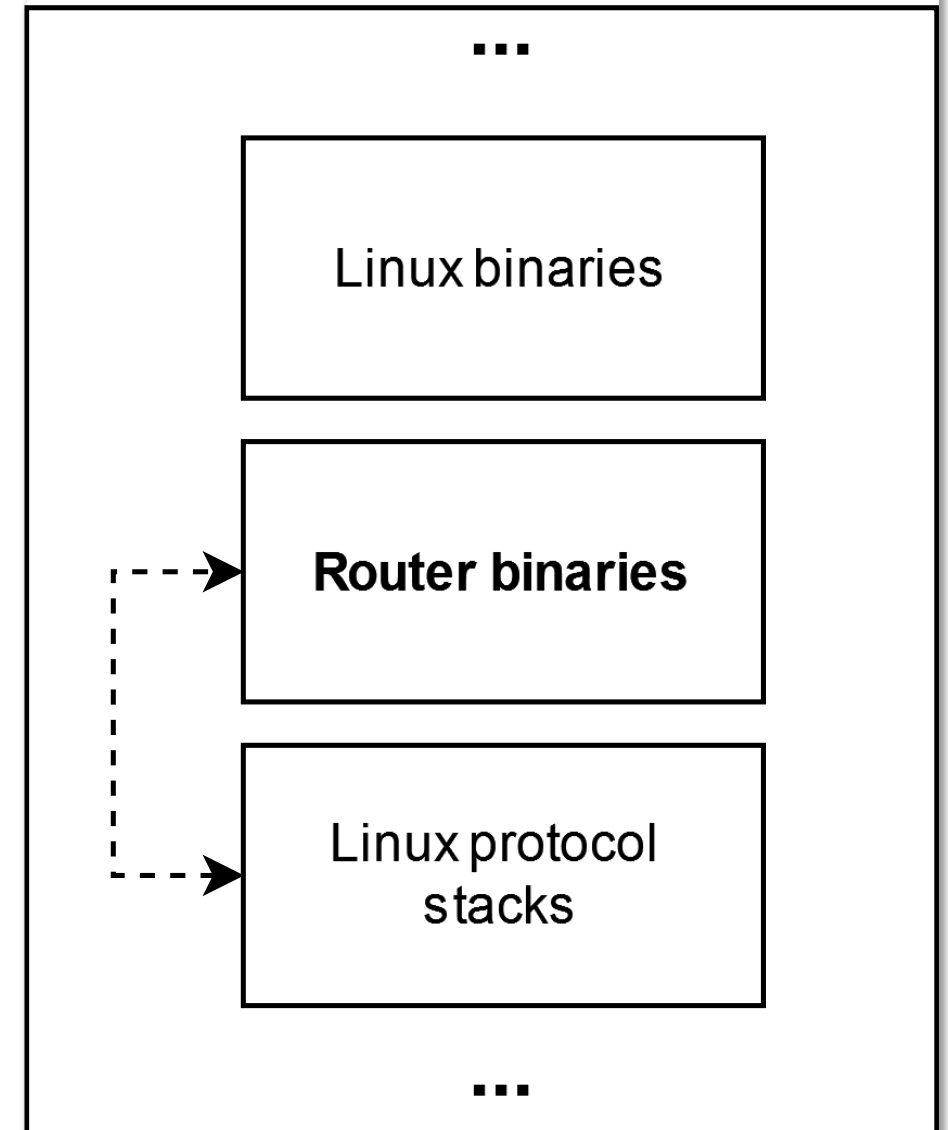
*** [Debug7: Leveraging a Firmware Modification Attack for Remote Debugging of Siemens S7 PLCs](#) – Eyal Semel et al., BHASIA 2024

“Simple” routers

- Typically, routers are hardware boxes
- They run some kind of **embedded Linux distribution**, adding **some custom binaries to implement additional functionality**
- Everything runs as **root**, exploits can be devastating
- New generation of routers started to use virtualization (since 2010, Cisco and Juniper)

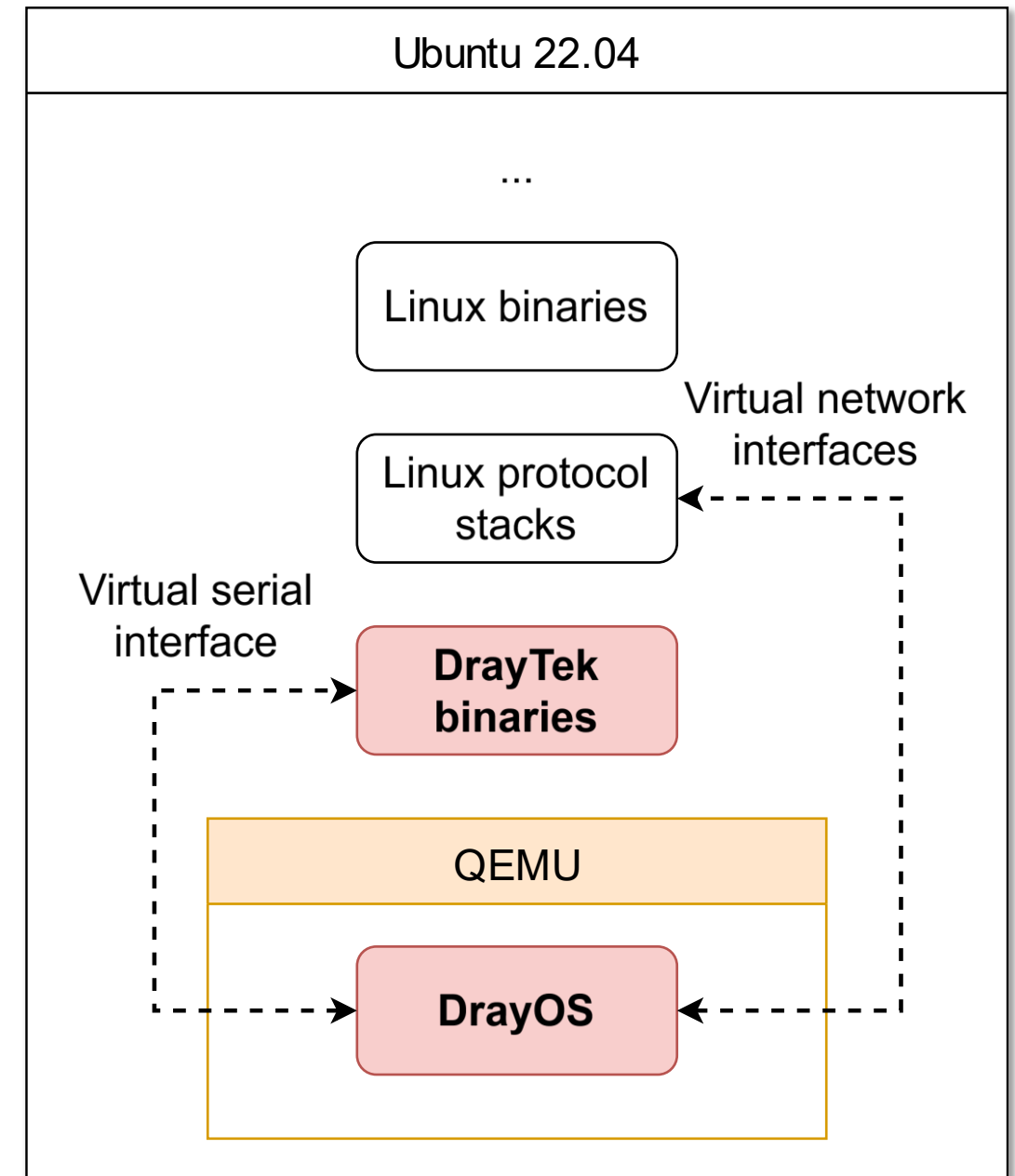
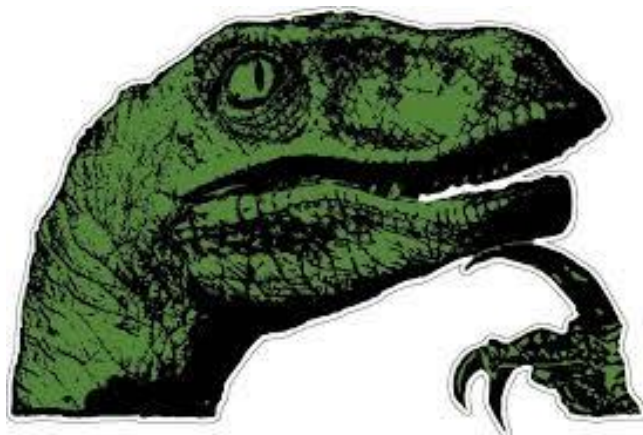


Embedded Linux



“Complex” routers: Vigor 3910 / 3912

- The hardware box runs **Ubuntu Linux 22.04** (aarch64)
- All DrayTek devices run on a proprietary OS - **DrayOS**
- On more complex Vigor devices **DrayOS** runs in a VM, while the host OS is Ubuntu 22.04
- Virtualization is supposed to add a layer of security and to ensure reliability (it restarts super fast after an error condition)
- **This looks great, but how is that REALLY implemented?**



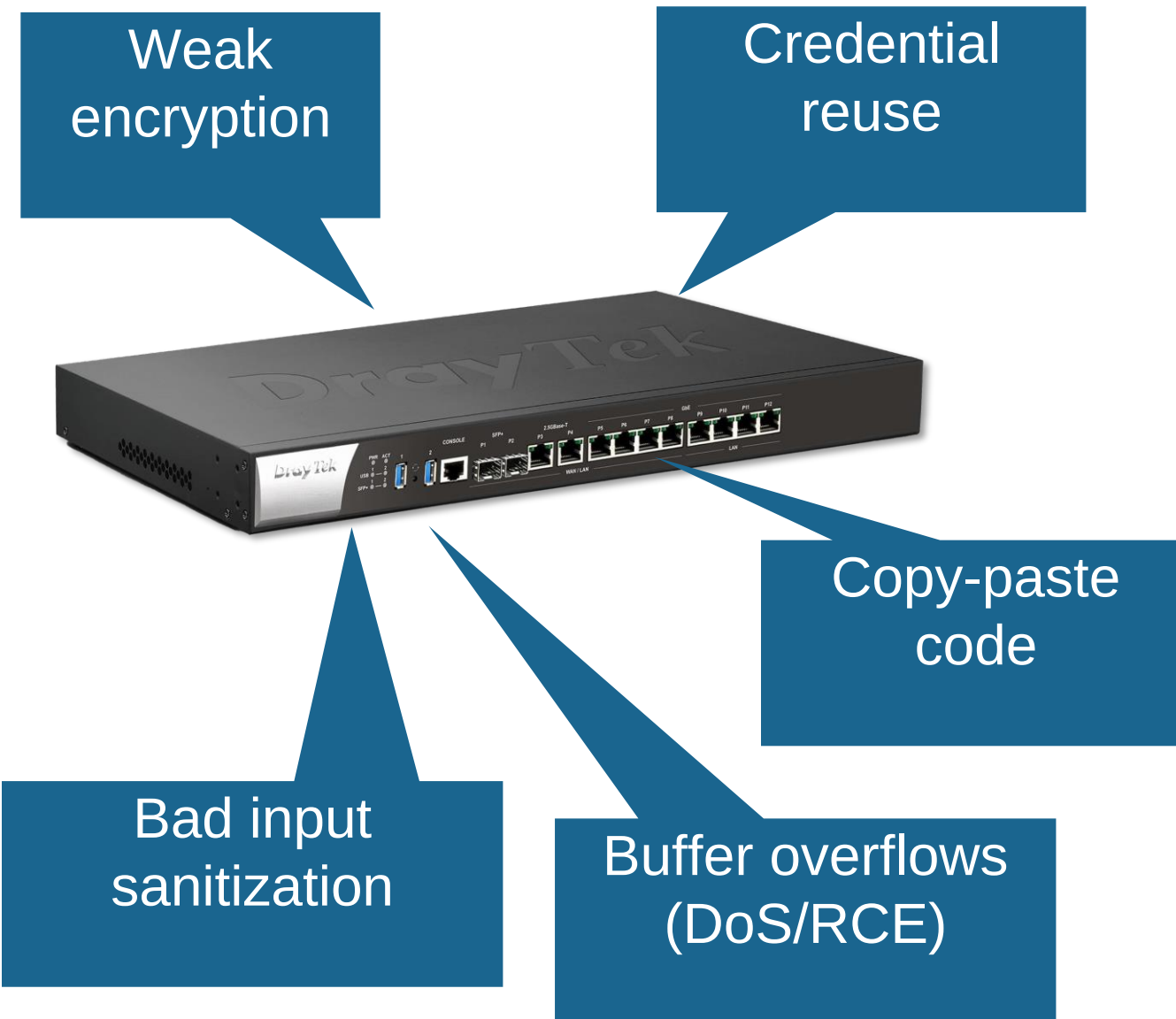


PART 2

Our Findings

Overview of findings

14 vulnerabilities from different types/classes*



The Typical Vulnerability History of a Product

Based on our CVE work through the years, covering over 25,000 vulnerabilities, we regularly see a product (or product class) follow these phases, as new vulnerabilities are discovered and resolved:

- ✓ 1) Obvious vulnerability types in critical functionality, such as buffer overflows in the username or password of a login feature
- ✓ 2) Incomplete fixes for the first vulnerabilities discovered in the product, often involving a rushed patch for a single input or parameter, while other closely related vectors remain unprotected
- ✓ 3) Susceptibility to variants of the common vulnerability types, often bypassing a simple protection scheme that only spots the most common manipulations [Christey2006b]
- ✗ 4) Common types that are restricted to particular environments, rarely-used functionality, or users with special privileges.
- ✗ 5) Elimination of the most common vulnerability types, typically involving a systematic code analysis and/or refactoring such as input validation frameworks
- ✗ 6) Susceptibility to more novel or rare vulnerability types and attacks, which often are not easily detectable by common tools or informal manual analysis
- ✗ 7) Unique vulnerability types that usually require expert analysis and extensive effort to find, possibly requiring a new attack or vulnerability class

* Unforgivable vulnerabilities – Steve Christey, The MITRE Corporation (2007).

Getting started

Start

WebUI

Exploit

Root

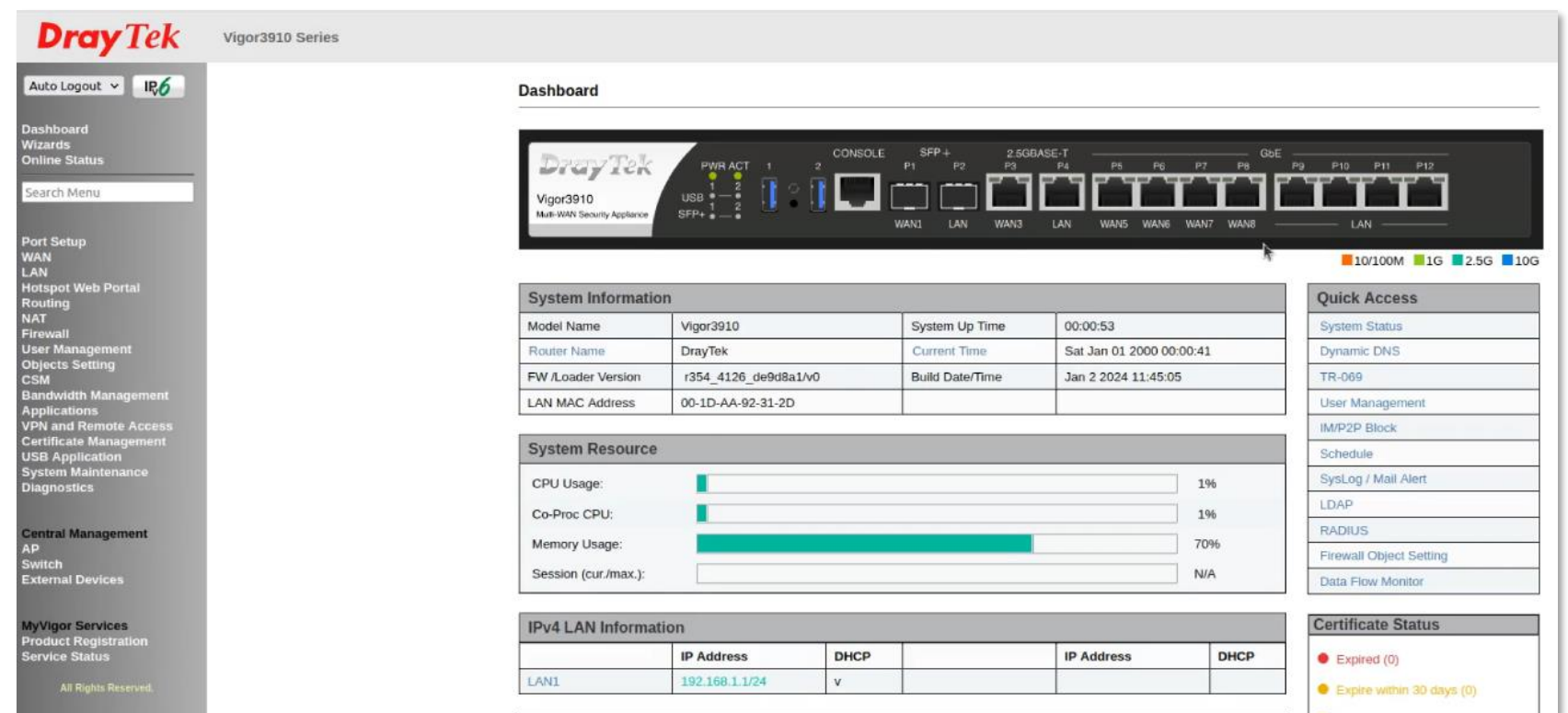
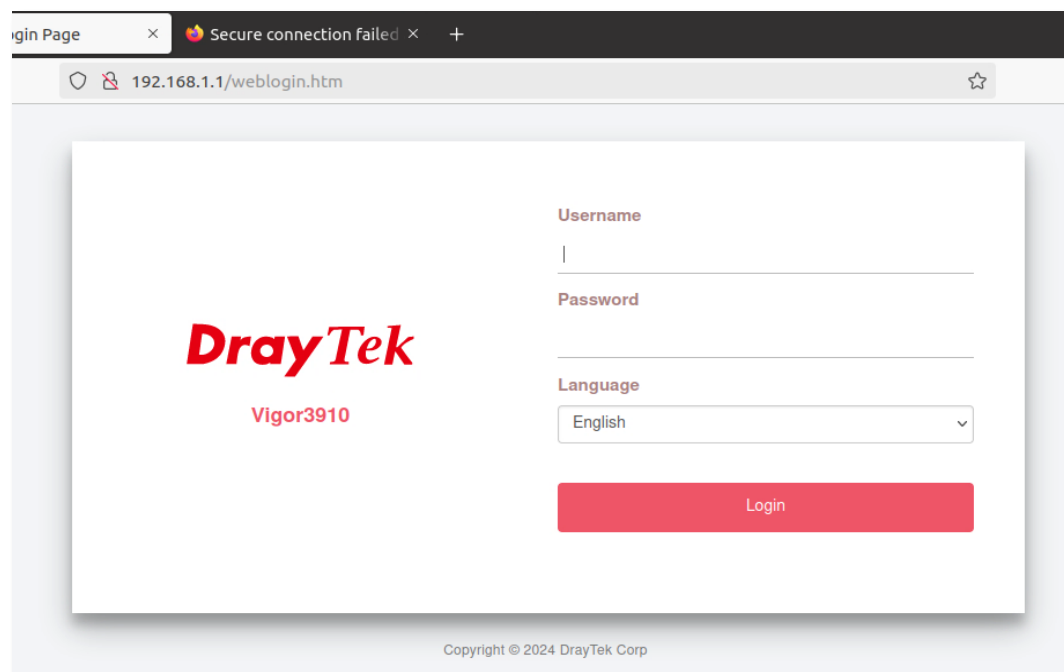
- The firmware is available online, but it's encrypted
- We built upon the research** from 2 years* ago to **decrypt the firmware for 3910**
- We could only buy **3912** for our lab, but could not decrypt the firmware (yet)
- DrayOS is huge, so we started with **WebUI** – web-based admin's panel
- **There is a single set of admin credentials used for the entire device:** WebUI, telnet over SSH, even the host OS

Vigor3910	2024-08-28  Firmware Version 4.4.3.1 (Mainline) Firmware contains bug fixes and new features. Upgrade to the stable version if these new functions are not needed in the current firmware.  Release Note  Checksum
	2024-11-18  Firmware Version 4.3.2.9 (Stable) Firmware contains the latest bug fixes.  Release Note  Checksum
Vigor3912 Series	2024-11-12

* Emulate it until you make it! Pwning a DrayTek Router before getting it out of the box – Philippe Laulheret. HEXACON (2022).

** CataLpa's writeup (2024): https://wzt.ac.cn/2024/02/19/vigor_3910/?_x_tr_hist=true

- A standard admin Web-interface used to configure and manage the device
- **MUST NOT be exposed to the Internet**, but oh well...
- One of the critical issues discovered in the recent past had to do with the login form, so we decided to have a closer look at WebUI



First look at the WebUI: input validation issues

Start

WebUI

Exploit

Root

System Maintenance >> Login Page Greeting

Login Page Greeting

Login Page Logo: Default Browse... No file selected. (Max 524 × 352 pixel) Upload

☐ Enable Greeting

Login Page Title: Router Login

Welcome Message and Bulletin (Max 511 characters) Preview Set to Factory Default

`<h1>Welcome Message</h1><p>This welcome message is displayed in the Login page of the router. Replace this text with your own message. </p>The welcome message can be written in HTML so lists such as this one can be created Other markup tags such as p, font or img can be used`

Examples of Welcome Message and Bulletin:
`<h1><b><font color=red>Welcome Message</font></b></h1>`
`<p>Message</p>`

OK Cancel

```
38 - function onClkBtnOk() {
39 -     if (f.sDesc.value.search(/<.*script.*>/i) != -1 || f.sURL.value.search(/<.*script.*>/i) != -1) {
40 -         alert(gettext('To avoid security risks, Javascript cannot be embedded in the message!'));
41 -         f.sURL.focus();
42 -         return;
43 -     }
44 -     if (checkWebChange()) {
45 -         f.webchange.value = '1';
46 -     }
47 -     f.iLoglogo.value = f1.iLoginLogo.value;
48 -     bg.cpntCtrl(true, f, f.iURLEnable, 2);
49 -     f.submit();
50 - }
```

``

Vigor Login Page

http://192.168.1.1/cgi-bin/wlogin.cgi?aa=YWRtaW4=&ab=YWRtaW4=&sFormAuthStr=%27;%20alert(1);%20////

Credentials are not safe

Start

WebUI

Exploit

Root

- The same admin credentials are used across the entire system
- They don't enforce TLS (HTTPS)
- When logging into WebUI, credentials are transmitted in cleartext (HTTP)
- But if we use TLS (HTTPS), everything is fine, right?

```
POST /cgi-bin/wlogin.cgi HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 125
Origin: http://192.168.1.1
Connection: keep-alive
Referer: http://192.168.1.1/weblogin.htm
Upgrade-Insecure-Requests: 1
```

Username and password are
base64- and url-encoded

```
aa=YWRtaW4%3D&ab=SWFtQUxpemFyZEtpbmc%3D&sslgroup=-1&sLang=en&obj3=&obj4=&obj5=&obj6=&obj7=&sFormAuthStr=sDTmbiDTjYSfN6x&obj9=
```

Credentials are not safe (continued)

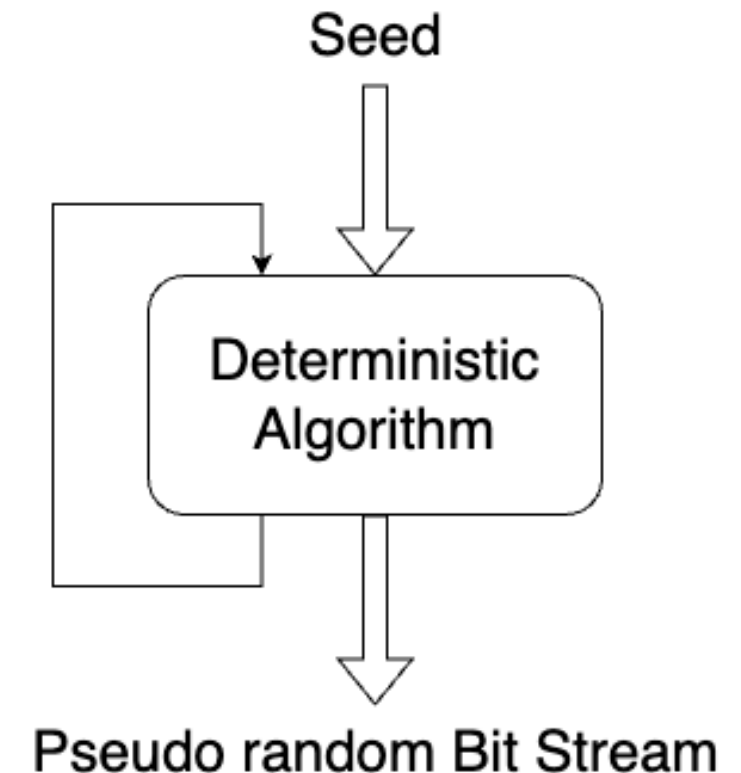
Start

WebUI

Exploit

Root

- **TLS is only secure if there is sufficient entropy** for generating private keys
- It's notoriously **difficult to have proper sources of entropy in embedded systems***
- **PRNG must be seeded with a sufficiently random value,**
- Otherwise, the output of the PRNG may be guessed
- TL;DR, [RFC1075](#) says if you can do that, it may be **feasible to recover private keys and break TLS encryption**



* [Wheel of Fortune. Analyzing Embedded OS Random Number Generators](#) – Ali Abbasi and Jos Wetzels. CCC (2016)

Credentials are not safe (continued)

Start

WebUI

Exploit

Root

- We found that **deprecated OpenSSL API** is used (both TLS and VPN)
- The **PRNG** was indeed seeded with a “random” value, but **not the one you’d expect...**

```
// ...  
v6 = RAND_seed((unsigned int)"rand string to make rand generator a seed to produce entropy", 61LL);  
ssl_ctx = Setup_server_ctx(v6);  
ssl_cli_ctx = ((__int64 (*)(void))Setup_client_ctx)();  
// ...  
v12 = RAND_seed((unsigned int)"rand string to make rand generator a seed to produce entropy", 61LL);  
ssl_ctx_sslvpn = Setup_server_ctx(v12);  
// ...
```

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
             // guaranteed to be random.  
}
```

Buffer overflows... lots of them

Start

WebUI

Exploit

Root

- WebUI contains **around 100 static web-pages**
- **Almost every of them had a buffer overflow**
- Too many to document, so we had to divide them into classes
- (Semi-) **controlled writes** into .bss, **Denial-of-Service** of different kinds, **potential RCE**

```
//...
01: var_typemode = GetCGIbyFieldName(var_form_data, "typemode");
02: if ( atoi(var_typemode) )
03: {
04:     var_typemode_2 = GetCGIbyFieldName(var_form_data, "typemode");
05:     if ( atoi(var_typemode_2) == 1 )
06:     {
07:         *(some_struct + 0xCLL) = 8;
08:         var_dnsserver = GetCGIbyFieldName(var_form_data, "dnsserver");
09:         v44 = *(some_struct + 0x18LL);
10:
11:         var_dnsserver_len = _safe_strlen(var_dnsserver);
12:         memcpy(v44, var_dnsserver, var_dnsserver_len);
13:         *(some_struct + 0x10LL) = _safe_strlen(var_dnsserver);
14:         *(*(some_struct + 0x10LL) + *(some_struct + 0x18LL)) = &qword_425EECF0;
15:     }
16:     else
17:     {
18:         *(some_struct + 0xCLL) = 7;
19:     }
20: }
//...
```

Conflict of interests: researchers vs threat actors

Start

WebUI

Exploit

Root

- At this point we were still at an early stage of our research
- Only a bunch of buffer overflows, and still lots of components and functionality to explore
- “[Stack smashing] is a dying artform, as things move further away from bare metal into virtualized environments” ©
- But we are researchers, **threat actors** don’t limit themselves to novel vulnerabilities, they **want results and fast**
- So our main research question became: “**How fast can mess it all up with what we already have?**”
- Enter the **CVE-2024-41592**



CVE-2024-41592: stack buffer overflow

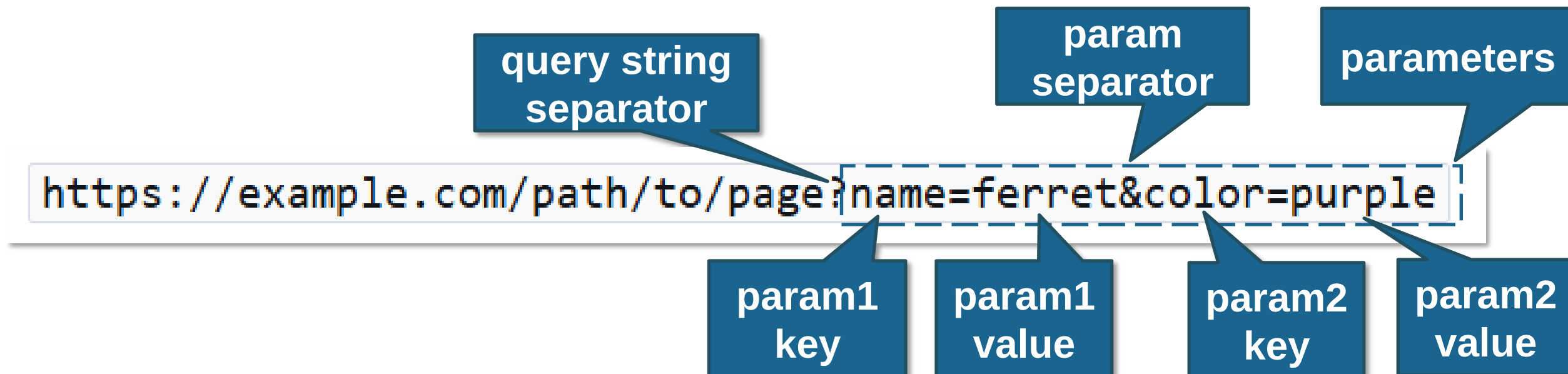
Start

WebUI

Exploit

Root

- Most web-pages call a special **function** that **parses the query string parameters**
- The **destination buffer** that contains pointers to parameters is allocated on the stack and **has fixed length**
- **There are no checks** whether the buffer can fit an arbitrary long list of parameters
- **The bug can be triggered via** requesting almost **every page**, causing **indirect arbitrary** writes into the stack



CVE-2024-41592 (continued)

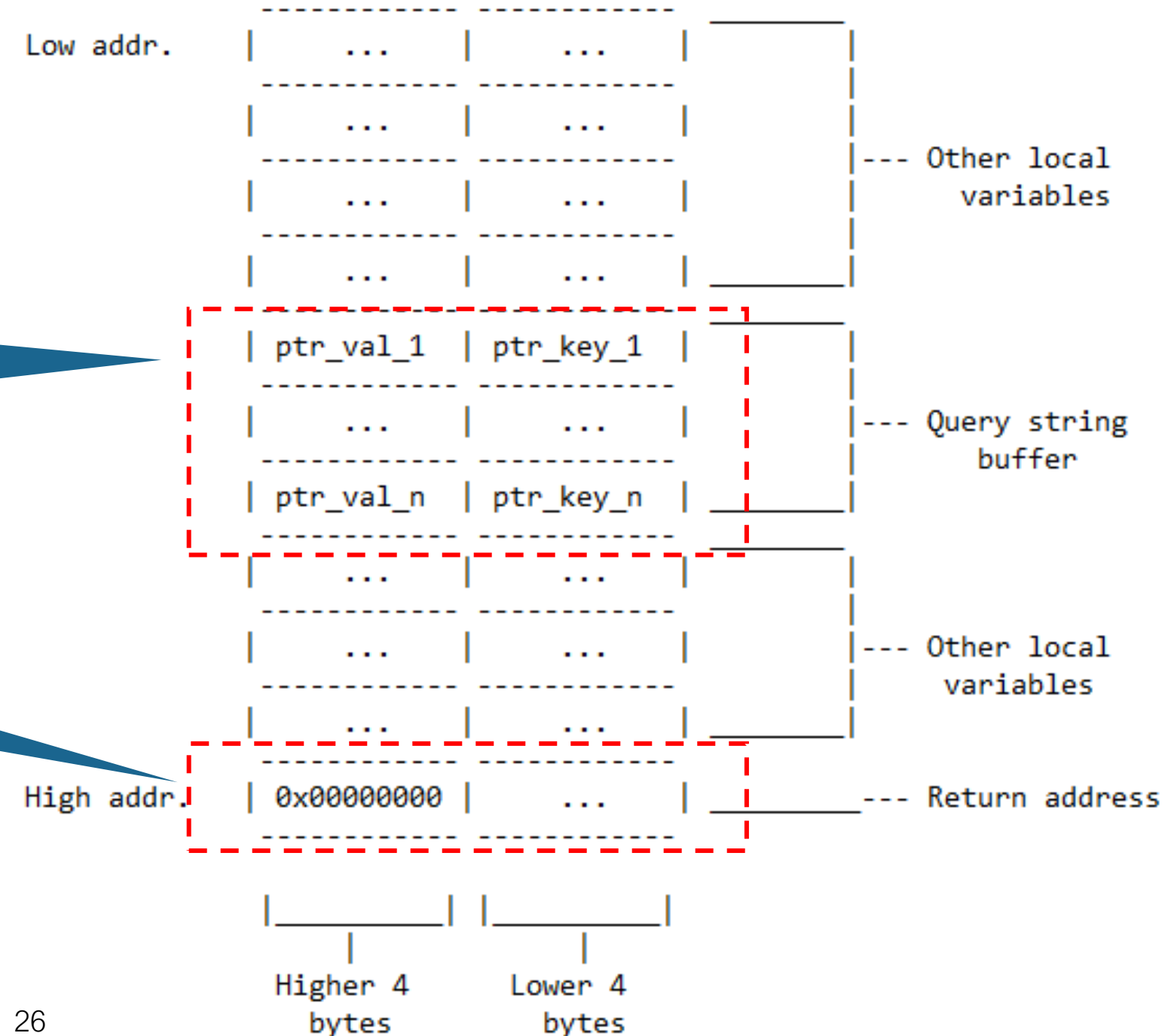
Start

WebUI

Exploit

Root

- The query string parameter keys and values are allocated on the heap
- The pointers to keys/values are stored on the stack**
- Pointers and function addresses are only 4 bytes only (aarch64 in 32-bit mode)**



Can we exploit it?

Start

WebUI

Exploit

Root

- The main binary is called “sohod64.bin” – this is literally the whole DrayOS, including WebUI
- Runs via a modified QEMU executable
- aarch64 (but runs in 32-bit mode)
- No DEP (executable heap and stack)
- No stack canaries
- No PIE, No ASLR
- **A completely “flat” binary that has everything we need and no binary hardening whatsoever**



Some challenges

Start

WebUI

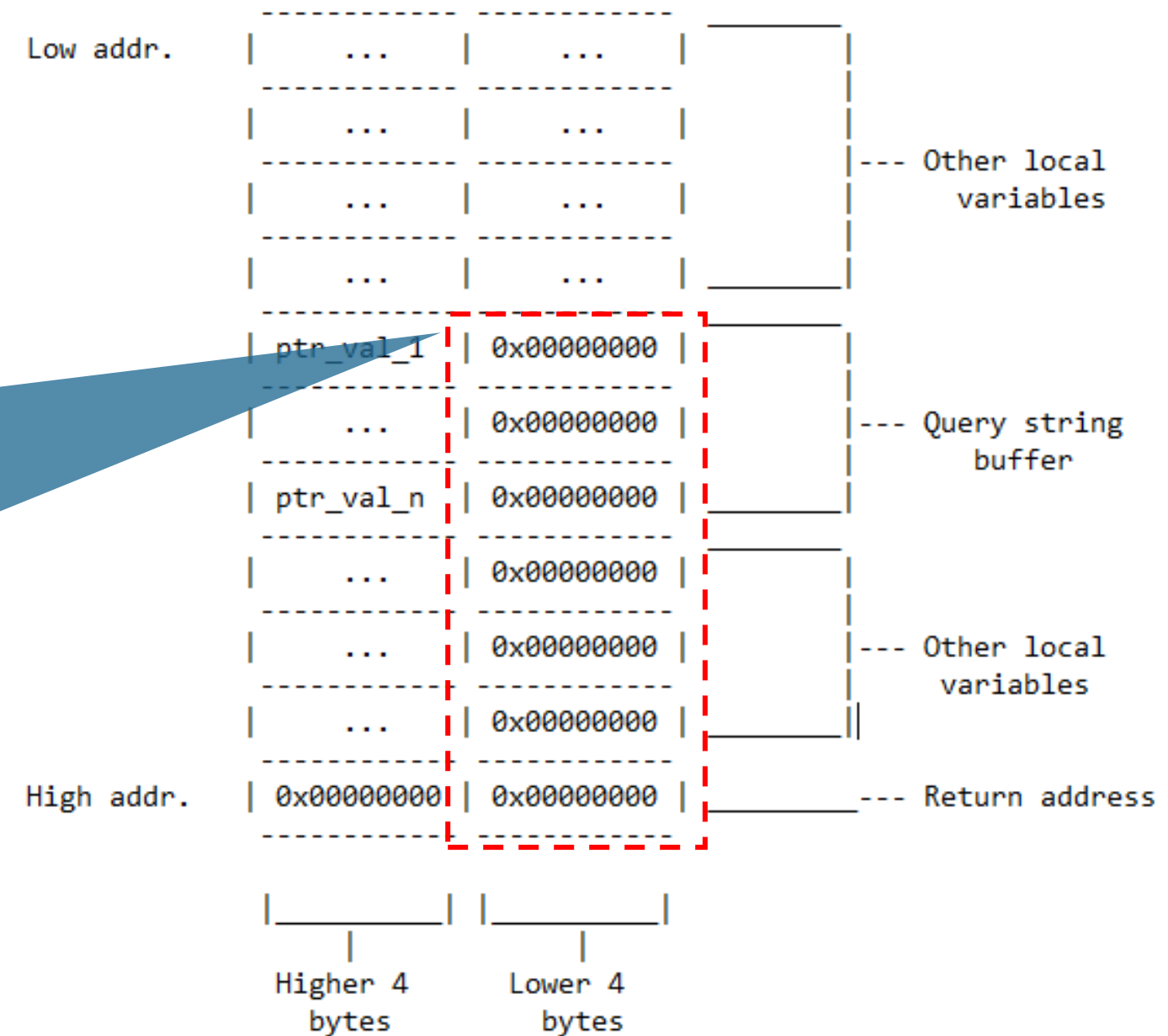
Exploit

Root

- We stumbled upon **unexpected challenges**:
- (1) **almost every web page requires auth token**;
- (2) **the FreeCtrlName() function**

```
__int64 __fastcall FreeCtrlName(__int64 result)
{
    int v1; // [xsp+1Ch] [xbp+1Ch]
    int i; // [xsp+2Ch] [xbp+2Ch]

    v1 = result;
    for ( i = 0; *(v1 + 8 * i); ++i )
    {
        result = _kfree(*(v1 + 8 * i), 0x163u);
        *(v1 + 8 * i) = 0;
    }
    return result;
}
```



Some challenges (continued)

Start

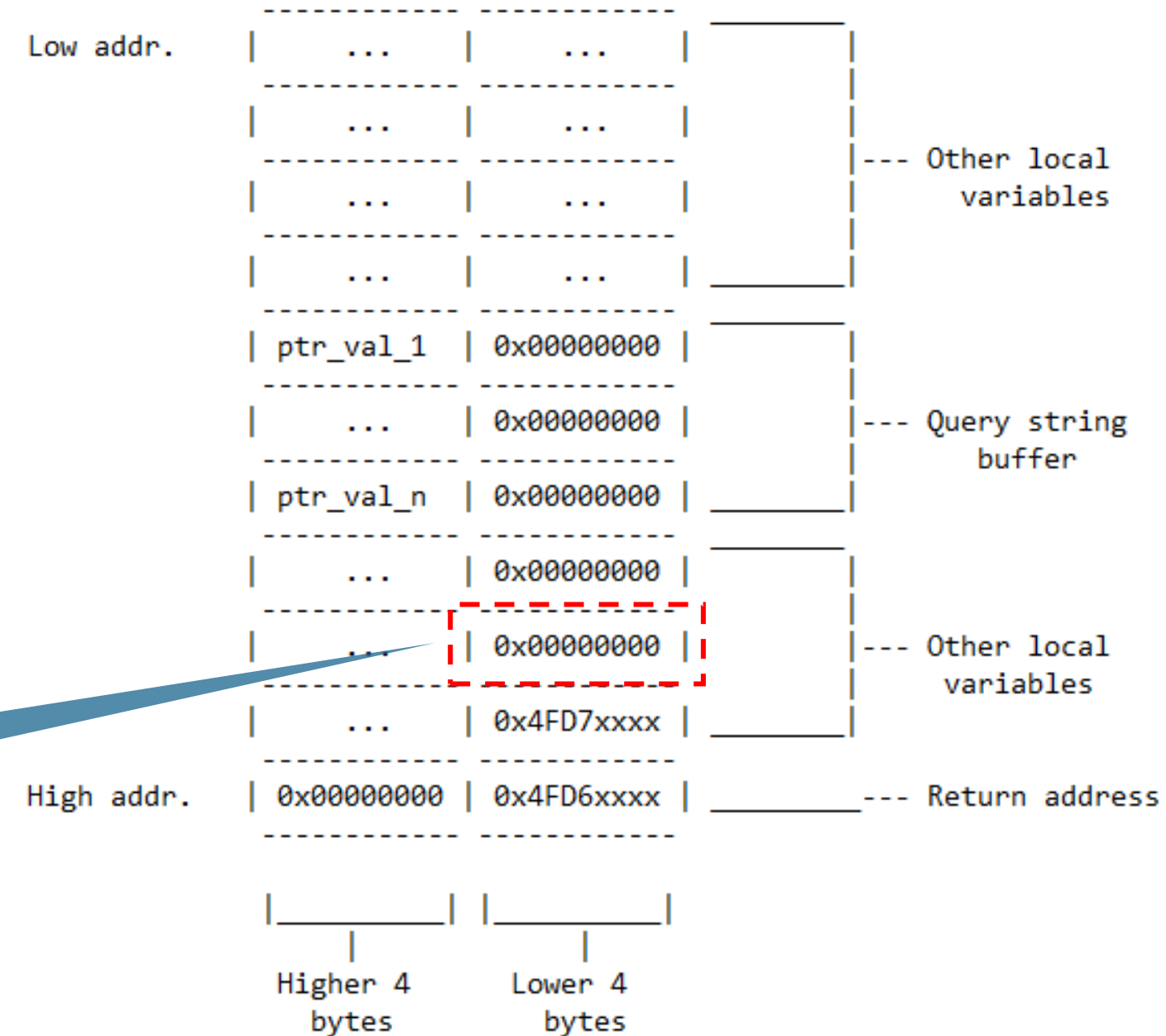
WebUI

Exploit

Root

- FreeCtrlName() is a **chain free**, it stops when it encounters a NULL
- After looking long and hard (a couple of hours)...
- We found a web page that:**
- (a) **did not require the auth token** to process the query string
- (b) **had a local variable just below the return address initialized to zero**

A local variable
initialized to zero



```
GET /cgi-bin/[vulnerable].cgi?[[[... &&&] [SHELLCODE][MSG] HTTP/1.1 [...]
```

Call “printf()”

Let's do something useful

Start

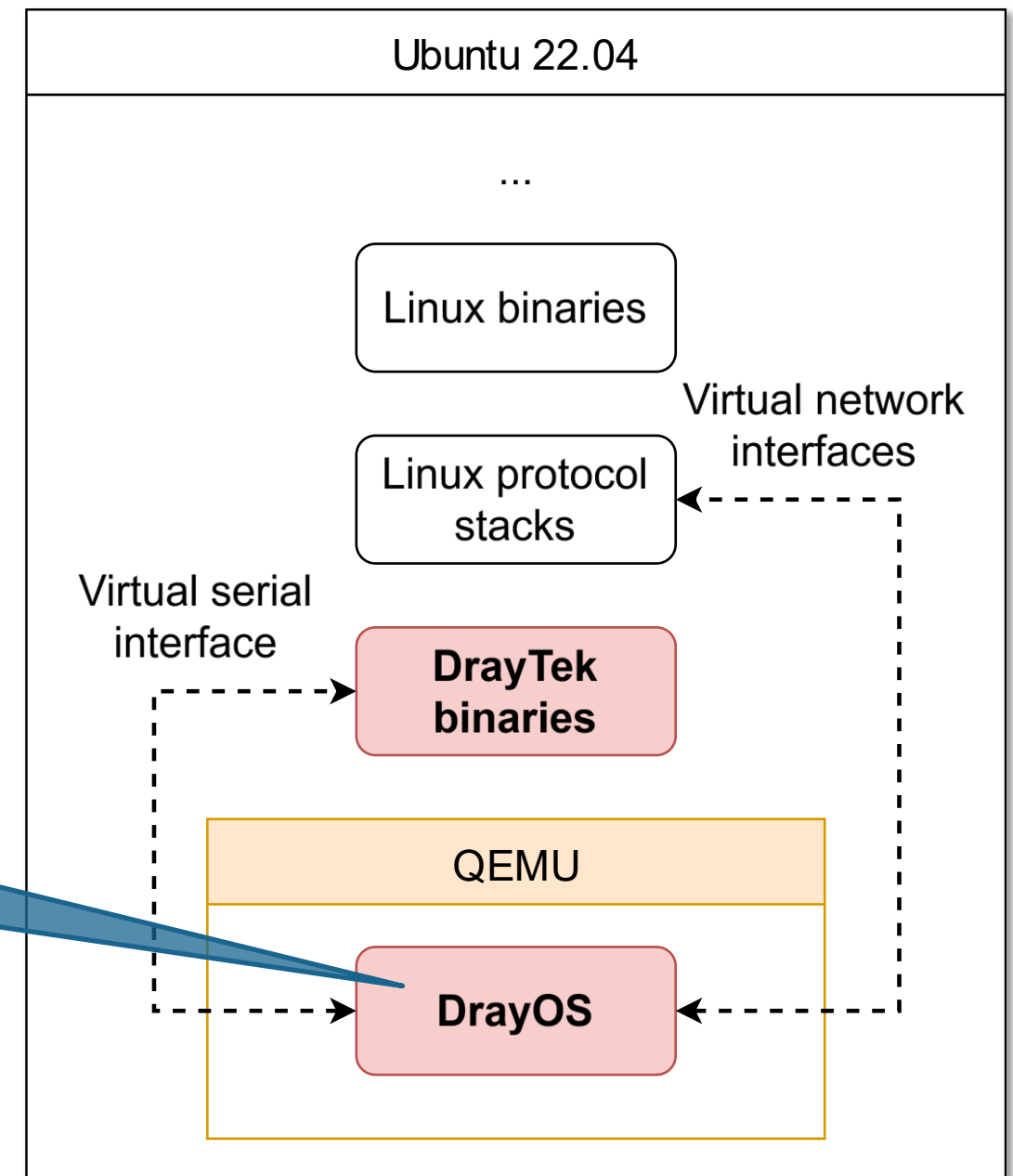
WebUI

Exploit

Root

- Remember, we are still in the VM, so we can't pop a root shell
- We can still do something useful, it's a "flat" binary
- DNS hijacking, altering settings, defacement, takeover
- Let's use very similar shellcode, but instead of printing a message, we change a certain string...

We are still here



DrayTek

Vigor3910

Username

|

Password

.....

Language

English



Login

But the title said “remote shells”!

Start

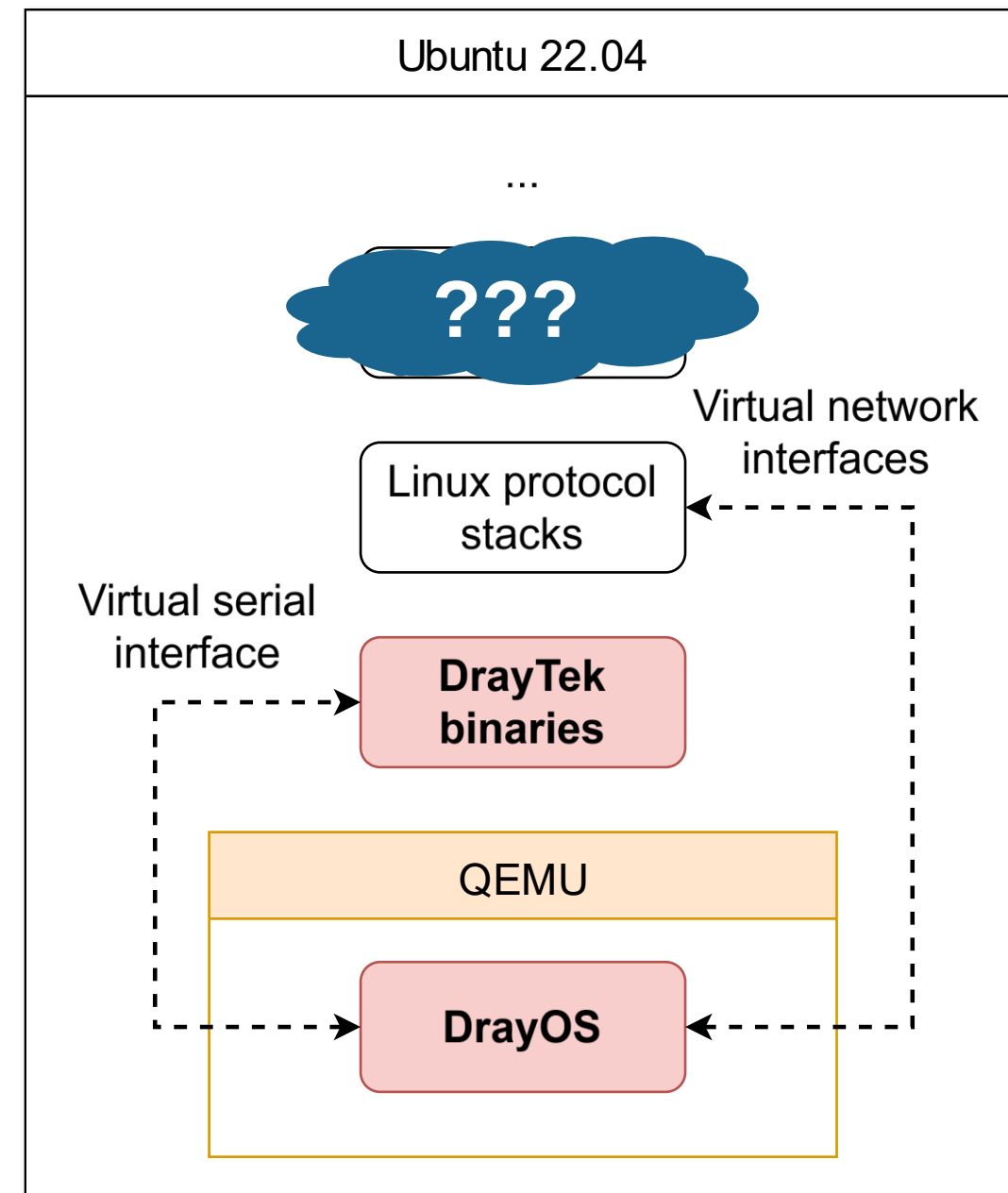
WebUI

Exploit

Root

- Past research* mentioned a **function that calls OS commands**, but no details
- We found a binary on the host OS that “listens” to requests from the guest
- The guest sends a “reboot” request when it needs to be rebooted, or “set_linux_time” for setting the time, etc...
- The commands are whitelisted, so game over, attackers

```
reboot fw_upload gci_exp quit set_linux_time setportspeed update_ps  
backup_soho setadminpass f2 set_board_info halt usbcmd uffssave cfg_restore  
ftpserv debug_cfgexp setlinuxip set_linux_timezone2 set_lan_wired8021x
```



* Emulate it until you make it! Pwning a DrayTek Router before getting it out of the box – Philippe Laulheret, HEXACON 2022.

Start

WebUI

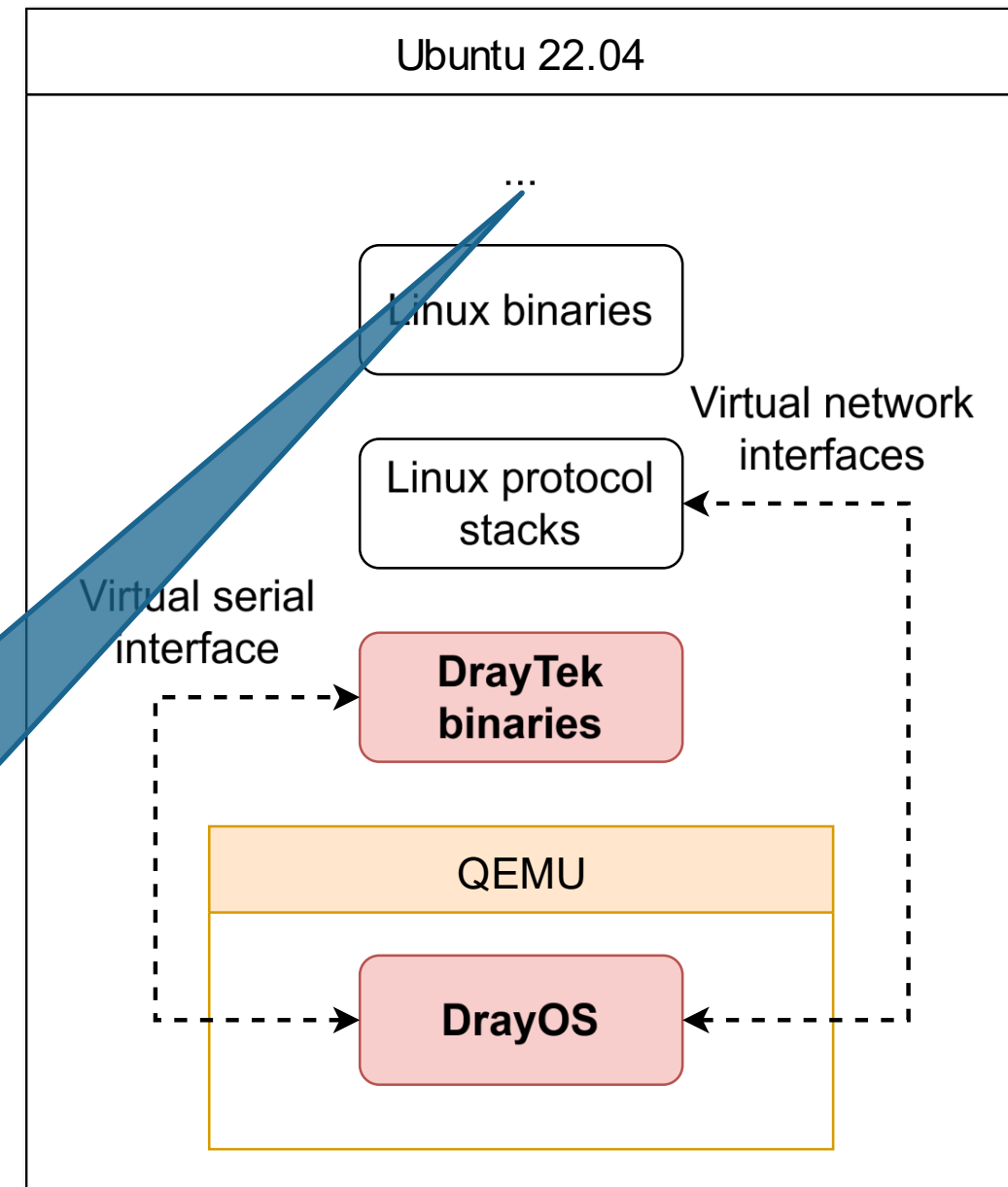
Exploit

Root 

```
__int64 __fastcall run_command(const char *arg_cmd_script)
{
    int v2; // w19
    size_t v3; // x0
    char *var_cmd; // x0
    _BOOL4 v10; // w5

    v2 = strlen(arg_cmd_script);
    v3 = strlen(aVarLogDrayosLo);
    var_cmd = (char *)malloc(v3 + v2 + 32);
    *(_QWORD *)var_cmd = 0LL;
    sprintf(var_cmd, "/etc/runcommand/%s >> %s 2>&1", arg_cmd_script, aVarLogDrayosLo);
    var_cmd_3 = var_cmd_2;
    do
    {
        // [...]
    }
    while ( v10 );
    system(var_cmd);
    free(var_cmd);
    return 0LL;
}
```

We've escaped the VM!



[proto]://[ip]/cgi-bin/[vulnerable-cgi-page].cgi?

&&&&....&&&&[shellcode]%20set_linux_time%20%3B[ARBITRARY_OS_COMMAND]%3B

```
(venv) standash@thelab42-2:~/stuff/vr/draytek/exploits$
```

```
standash@thelab42-2:~/stuff/vr/draytek/exploits$ nc -l 192.168.42.1 1234
```

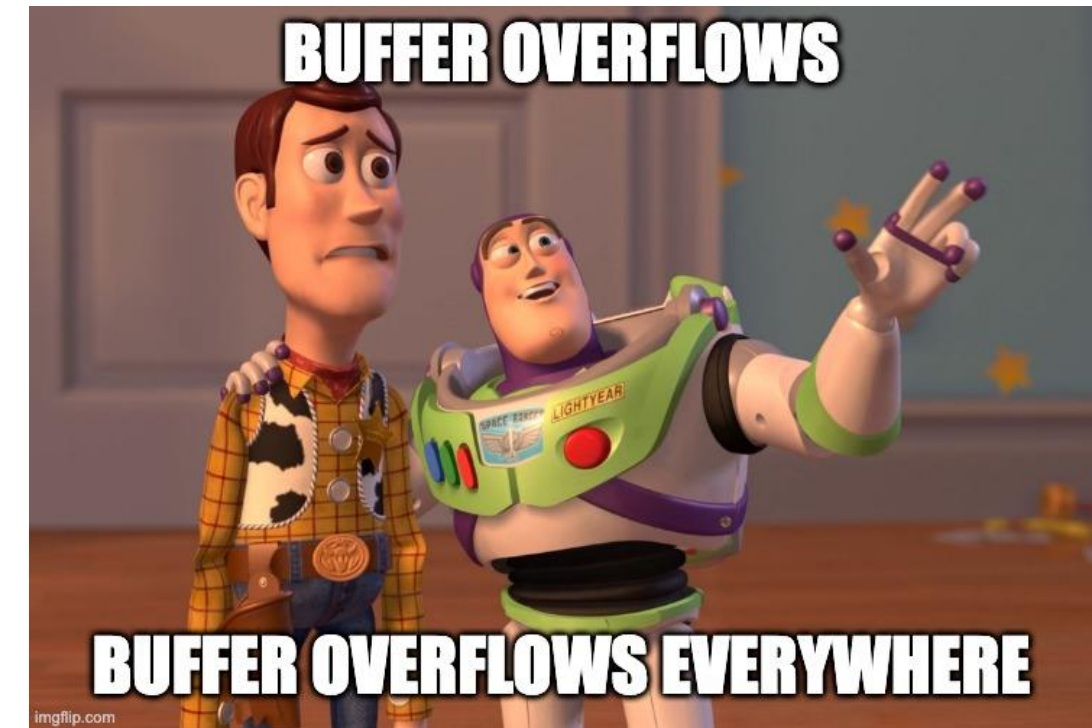


PART 3

Outlook and Conclusions

A few notes on patches

- It is uncertain if the “**VM escape**” issue was already found by Philippe or is a different issue
- We had not bandwidth to document all the buffer overflows, so **we reported several classes of them**
- Vendor came back quickly with the patches, but **we are uncertain whether the fixes apply only to those reported examples**
- **Another advisory with similar buffer overflows published on the same day, could be a coincidence...**



Advisory	Affected Products	Release Date
Buffer Overflow Vulnerability	Routers	2024-10-04
Cross-Site Scripting, Denial of Service and Remote Code execution vulnerabilities (CVE-2024-41583 ~ CVE-2024-41596)	Routers	2024-10-04

A few notes on patches (continued)

- **OS command injection (CVE-2020-8515):** actively exploited by threat actors, patched several years ago, only affects EoS models
- On 21st of October 2024 a researcher* publishes 22 (!) variants of the same bug
- The root cause is very similar to the “VM escape” bug we just presented
- Looks like no one performed variant analysis

*https://github.com/fu37kola/cve/blob/main/DrayTek/Vigor3900/1.5.1.3/DrayTek_Vigor_3900_1.5.1.3.pdf

```

9  const char *v6; // r7
10  int result; // r0
11  char v8[256]; // [sp+8h] [bp-220h] BYREF
12  char s[288]; // [sp+108h] [bp-120h] BYREF
13
14  if ( bypass() )
15  {
16      v0 = 0;
17      v1 = -16;
18  }
19  else
20  {
21      Value = cgiGetValue(dword_42D0C, "config");
22      v3 = cgiGetValue(dword_42D0C, "table");
23      v4 = cgiGetValue(dword_42D0C, "newtable");
24      v5 = v3 == 0;
25      if ( v3 )
26          v5 = Value == 0;
27      v6 = v4;
28      if ( !v5 && v4 )
29      {
30          v1 = GET_ENV();
31          if ( v1 > 3 )
32          {
33              memset(s, 0, 0x100u);
34              snprintf(s, 0x100u, "uci get '%s'.'%s' >/dev/null 2>&1", Value, v6);
35              v0 = system(s);
36              if ( v0 )
37              {
38                  memset(v8, 0, sizeof(v8));
39                  snprintf(v8, 0x100u, "uci rename '%s'.'%s'='%s'", Value, v3, v6);
40                  result = system(v8);
41                  if ( result )
42                      return result;
43                  v0 = sub_22ABC();
44                  v1 = v0;
45                  if ( v0 )
46                      v0 = 0;
47              }
48          }
49      }

```

Recommendations to asset owners

- Take any kind of administrative tools off the Internet, install updates often
- If you own a business that relies on these devices, perform independent security assessments
- **Don't rely on the number of historical CVEs to understand the vendor's security posture**
- Instead, use security advisories to your advantage:
 1. Check if the same issues keep popping up over the years
 2. No vulns and advisories – this is even more suspicious



*<https://www.flickr.com/photos/fastjack/282707058>

Recommendations to vendors

- When you read about threat actors targeting your devices, make some positive changes to their security
- Firmware encryption does not prevent scrutiny (a.k.a “security by obscurity”)
- Use static analysis tools / audit the code / hire pen-testers
- Don’t patch only the issues reported by the researchers, do variant analysis
- Binary hardening is not a silver bullet, yet, why not use it?



If software starts to resemble Swiss cheese – consider a complete redesign/reimplementation



Thank you!
Any questions?

stanislav.dashevskyi@forescout.com
francesco.laspina@forescout.com