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

![Diagram showing potential targets for Viproy]

- IP Phones
- Conference (Webex, GoMeeting)
- Mobile Users
- MPLS VPN
- Shared Switch?
- Sandbox for Tenant Services
  - SDP / RTP Servers
  - VAS / CDR Server
  - Database Server
- Shared Services for All Tenants
  - SIP, RTP, HTTP
  - Firewall
  - IPS, RTP, HTTP
  - Cisco Unified Communications Manager
  - Skinny / SIP / TFTP / HTTP
  - IP Phone XML Services
  - Client Management
  - Service Management
  - PBX
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

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>Cisco_ce:3d:81</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>442 Device ID: Switch Port ID: GigabitEthernet0/1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8.226800</td>
<td>Cisco_d7:01:12</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>130 Device ID: SEPOC789070112 Port ID: Port 2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>60.009698</td>
<td>Cisco_ce:3d:81</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>442 Device ID: Switch Port ID: GigabitEthernet0/1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>68.227395</td>
<td>Cisco_d7:01:12</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>130 Device ID: SEPOC789070112 Port ID: Port 2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>120.020302</td>
<td>Cisco_ce:3d:81</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>442 Device ID: Switch Port ID: GigabitEthernet0/1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>128.233745</td>
<td>Cisco_d7:01:12</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>130 Device ID: SEPOC789070112 Port ID: Port 2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>180.023851</td>
<td>Cisco_ce:3d:81</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>442 Device ID: Switch Port ID: GigabitEthernet0/1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>188.233430</td>
<td>Cisco_d7:01:12</td>
<td>CDP/VTP/DTP/PAG/P/UDLD</td>
<td>CDP</td>
<td>130 Device ID: SEPOC789070112 Port ID: Port 2</td>
<td></td>
</tr>
</tbody>
</table>

- **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-BPC-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

- 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
Demonstration of VOSS attacks

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](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
Demonstration of SIP attacks
SMS phishing using SIP messages
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

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
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
Attacking Skinny services

Everybody can develop a Skinny module now, even Ewoks!

Register

Unauthorised Call

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

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

#Skinny Call Test
begin
  connect
  rescue Rex::ConnectionError => e
    print_error("Connection failed: #{e.class}: #{e}")
  return nil
end
```
Preparing a proper client for Skinny

- Install Cisco IP Communicator
- Change the MAC address of Windows
- Register the software with this MAC
Demonstration of Skinny attacks
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
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. Owner of trademark and all copyright is Sense of Security Pty Ltd. Neither text or images can be reproduced without written permission.

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