Legal

Copyright 2013 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM-0000773
Outline

• General Principals to Finding and Understanding VoIP in flow
• Analysis of VoIP Real Time Communications
• Lab Example of Data Exfiltration
General Principals Finding VoIP

- Standard VoIP have well known ports and protocols
  - SIP uses port 5060, secure SIP uses port 5061
  - SCCP uses port 2000, secure SCCP uses port 2443
  - H. 323 uses port 1720 for call set up
- Finding Real Time traffic (phone call)
  - Session generated port pair for real time traffic (RTP, RTCP)
  - Work backwards follow VoIP signaling
  - SIP – Phone are required to periodically register with the registration server
  - Find the a particular end port and pull traffic until you see a phone call
General Principals Finding VoIP

- None standard VoIP have unique signaling protocols and ports
  - Skype
    - Authentication and signaling over port 80 or 443 (TLS)
    - Use common network elements to locate traffic
    - Authentication Server is a static IP
  - Razer Comm Gaming VoIP Software
    - Authentication Ports 443 (TLS)
    - Use UDP Port 2000 for media sessions
    - Does not operate in peer-to-peer method uses a transcoder through a single IP
General Principals Understand VoIP

- Understanding the function of common VoIP network elements can help you understand observed VoIP traffic
  - **Redirect server** – allows phone call forwarding
  - **Transcoder** – Allow phone with unsupported codex the ability to communicate
  - **Mixer** – Combines two audio streams into a single audio stream
  - **User Agents** - entities within the VoIP network in a client server mode that act on the behalf of users (phone, PC, Conference Bridge)
  - **Location Server** – Maintain location of connected phones
SIP Redirect Server – Call Forwarding

1. Pvt.jedi@darkside.org 10.55.22.1
2. 200 OK
3. Invite sgt.cookie@cookie.com
4. Cookies SIP Proxy 22.32.22.1
5. 302 Moved Temporarily
   Contact master@master.net
6. ACK sgt.cookie@cookie.com
7. Location Server 22.23.10.1
8. DNS Request/Response
9. S1
10. Invite master@master.net
11. 200 OK
12. ACK master@master.net
13. DardSide SIP Proxy 10.55.31.1
14. 200 OK
15. Master.net
16. master@master.net
17. RTP Stream

---

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookie.com</td>
<td>Master.net</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Public IP: 65.66.3.4

Public IP: 77.77.5.6

Public IP: 44.87.9.1

DNS: 16.78.9.9
### SIP Redirect Flow – Call Forwarding

<table>
<thead>
<tr>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
<th>Pro</th>
<th>Packets</th>
<th>Bytes</th>
<th>Sensor</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.55.22.1</td>
<td>10.55.31.1</td>
<td>17756</td>
<td>5060</td>
<td>17</td>
<td>26</td>
<td>10062</td>
<td>S1</td>
<td>OUT</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>15.78.8.9</td>
<td>37895</td>
<td>53</td>
<td>17</td>
<td>1</td>
<td>88</td>
<td>S2</td>
<td>OUT</td>
</tr>
<tr>
<td>15.78.8.9</td>
<td>55.55.1.2</td>
<td>53</td>
<td>37895</td>
<td>17</td>
<td>1</td>
<td>245</td>
<td>S2</td>
<td>IN</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>66.66.3.4</td>
<td>24868</td>
<td>5060</td>
<td>6</td>
<td>15</td>
<td>5175</td>
<td>S2</td>
<td>OUT</td>
</tr>
<tr>
<td>66.66.3.4</td>
<td>55.55.1.2</td>
<td>5060</td>
<td>24868</td>
<td>6</td>
<td>10</td>
<td>3450</td>
<td>S2</td>
<td>IN</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>16.58.43.6</td>
<td>37895</td>
<td>53</td>
<td>17</td>
<td>1</td>
<td>88</td>
<td>S2</td>
<td>OUT</td>
</tr>
<tr>
<td>16.58.43.6</td>
<td>55.55.1.2</td>
<td>53</td>
<td>37895</td>
<td>17</td>
<td>1</td>
<td>245</td>
<td>S2</td>
<td>IN</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>77.77.5.6</td>
<td>13467</td>
<td>5060</td>
<td>6</td>
<td>34</td>
<td>12852</td>
<td>S2</td>
<td>OUT</td>
</tr>
<tr>
<td>77.77.5.6</td>
<td>55.55.1.2</td>
<td>5060</td>
<td>13467</td>
<td>6</td>
<td>24</td>
<td>8328</td>
<td>S2</td>
<td>IN</td>
</tr>
<tr>
<td>10.55.31.1</td>
<td>10.55.22.1</td>
<td>5060</td>
<td>17756</td>
<td>6</td>
<td>34</td>
<td>13158</td>
<td>S1</td>
<td>IN</td>
</tr>
<tr>
<td>10.55.22.1</td>
<td>77.77.5.6</td>
<td>47598</td>
<td>35878</td>
<td>17</td>
<td>4876</td>
<td>277932</td>
<td>S1</td>
<td>OUT</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>77.77.5.6</td>
<td>47598</td>
<td>35878</td>
<td>17</td>
<td>4876</td>
<td>277932</td>
<td>S2</td>
<td>OUT</td>
</tr>
<tr>
<td>77.77.5.6</td>
<td>55.55.1.2</td>
<td>33572</td>
<td>47598</td>
<td>17</td>
<td>1625</td>
<td>92644</td>
<td>S2</td>
<td>IN</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>10.55.22.1</td>
<td>33572</td>
<td>47598</td>
<td>17</td>
<td>1625</td>
<td>92644</td>
<td>S1</td>
<td>IN</td>
</tr>
<tr>
<td>10.55.22.1</td>
<td>10.55.31.1</td>
<td>34527</td>
<td>5060</td>
<td>17</td>
<td>7</td>
<td>2275</td>
<td>S1</td>
<td>OUT</td>
</tr>
<tr>
<td>55.55.1.2</td>
<td>77.77.5.6</td>
<td>12354</td>
<td>5060</td>
<td>6</td>
<td>15</td>
<td>2646</td>
<td>S2</td>
<td>OUT</td>
</tr>
<tr>
<td>77.77.5.6</td>
<td>55.55.1.2</td>
<td>5060</td>
<td>12354</td>
<td>6</td>
<td>6</td>
<td>2088</td>
<td>S2</td>
<td>IN</td>
</tr>
<tr>
<td>10.55.31.1</td>
<td>10.55.22.1</td>
<td>5060</td>
<td>34527</td>
<td>17</td>
<td>8</td>
<td>2784</td>
<td>S1</td>
<td>IN</td>
</tr>
</tbody>
</table>
Real Time Communications

- Two use cases for packet generation:
  - On demand - noise is registered on the input devices (i.e microphone) then packaged and sent
  - Software continuously generate packets regardless of input, providing flow obfuscation

- Typically session generated port pair is used
  - Some software allows you specify port ranges
  - RTP should be even, RTCP next open odd port
  - Some software use dedicated ports for communication
Noise Generated Packets

- With a low idle time out on the flow sensor you can see:
  - General meter of the conversation
  - Who talks when and for how long

![Graph showing Noise Generated Packets]

- No Noise = No Packets
- Timing Packets
- How Long - Time
Noise Generated Packets

No Noise = No Packets

Out Bound

Inbound

Timing Packets

2 People

3rd person joins call

Conversation Meter
Software Generated Packets

- No break in the conversation regardless of human interaction
- Software packets appear to be uniformed
def to sound(infile, outfile):
    sampleFreq = 8000
    denominator = 8
    frequencies = (262, 294, 330, 350, 392, 440, 494, 523,
      587, 659, 698, 784, 880, 988, 1047, 1175)
    size = os.path.getsize(infile)

    fmt = FormatChunk()
    fmt.size = 0x10
    fmt.formatTag = 1
    fmt.channels = 1
    fmt.bitsPerSample = 16
    fmt.samplesPerSec = sampleFreq
    fmt.avgBytesPerSec = fmt.samplesPerSec *
    fmt.bitsPerSample / 8
    fmt.blockAlign = 4

    amplitude = 3000
    length = (3*denominator + size) * fmt.samplesPerSec /
      denominator * fmt.channels * fmt.bitsPerSample / 8

    https://code.google.com/p/data-sound-poc/

- A file is already in a digital format but it must be passed through a VoIP network
- Convert the digital format (.txt) into audio format (.wav)
- Think Fax/Modem
Compression is Important

halfByteSampleLength = fmt.samplesPerSec / denominator

j = 0
while True:
    data = infh.read(1)
    if data == ":
        break

    (m,) = struct.unpack("B",
    data)

    a = m & 0xF
    b = m >> 4

    for halfbyte in (a,b):
        frequency =

• To extract 23 Bytes it takes
  • 5.87 second sound bite
  • 136.8 KB .wav file
• To extract “CERT Resilience Management Module” or
  52,691 Bytes
  • 1H 49M sound bite
  • 201MB .wav file

https://code.google.com/p/data-sound-poc/
Data Exfiltration - Flow

- Without good compression flow long
- The flow very uniformed, set frequency and timing
- The flow is unidirectional
  - Software generated packets obfuscated this flow, instead you will find bidirectional flow
Summary

- Understanding the network function of an object can help you understand observer flow data
  - Many comment elements between VoIP deployments (Proxy, Redirect, Voice Mail, Transcoder, etc.)
- Their two use cases for real-time communication
  - On demand generation – noise = packets
  - Software generated packets - endless packets
- Data Exfiltration
  - Unidirectional flow, unless software generated packets
  - Weak Compression will create long flows
  - On demand VoIP will look like SW VoIP
Future Work

• Prove that there are only two use cases for real time communications

• Software Generated
  • Discover observable features

• On demand Generation
  • Developing a comprehensive list of observable features
  • Cross section those features with flow idle time out
    • Example: If you want the ability to observer feature X you will need an idle time of Y
Questions/comments?