CHAPTER

Eavesdropping and Modification

3 c0020

INFORMATION IN THIS CHAPTER

- Anatomy of Eavesdropping and Modification Attacks p0010
- Dangers of Eavesdropping and Modification Attacks p0015
- The Future of Eavesdropping and Modification Attacks p0020
- How to Defend against Eavesdropping and Modification Attacks p0025

Imagine that somewhere within your IT organization you have someone with too much time on his or her hands or who has issues with the way current management is running things…or hates his job…or dislikes her boss…. Whatever the reason may be, he or she is in a position with access to the core Internet Protocol (IP) network running through your organization. Let's give this person a name and call him Joe. One day, when working with Wireshark, the network protocol analyzer, Joe, notices the menu item **Telephony | VoIP Calls**. In trying it out, he discovers that…ta da… he can listen to any call going to and from the IP-Private branch exchange (PBX). Naturally, he starts figuring out how to listen in to the more interesting calls, and in particular to target calls to and from the CEO. Once he is able to isolate these calls, he automates his setup a bit. He finds a number of other tools and writes a script so that any calls to or from the CEO are saved to disk and converted into MP3 files. He then downloads those files onto his iPod and can listen to corporate conversations on his daily commute to and from work. Alternatively, he could install freely available speech-to-text software to get transcripts of all of those calls. p0030 AQ1

As Joe does this, he also discovers that again using Wireshark, he can easily see the instant messaging (IM) conversations of his colleagues. So he starts watching those conversations as well. p0035

In the course of doing this, Joe discovers that the company is going to be sold to a larger company known for aggressive layoffs after an acquisition. Figuring that his job is going to be axed, Joe starts doing all he can to sabotage the chances for the acquisition to be successful. First, he begins executing some of the denial-of-service attacks you learned about in Chapter 2, "Insecure Endpoints." When calls come p0040 in to the CEO from certain lawyers, the calls are disconnected. He also randomly disconnects other calls that are going on throughout the company.

p0045 Because he fears he'll be easily found out, Joe starts to get a bit more sophisticated in his attacks. He sets up a script that strategically drops any IM messages that include certain keywords. He also tries his hand at modifying IM messages and replacing words like "buy" with "sell." It's not terribly effective, but it does create a degree of confusion.

p0050 Joe also finds some tools on the Internet that let him mix in different backgrounds to audio streams using the Real-time Transport Protocol (RTP). With this tool, he's able to have a bit more "fun." When Joe's scripts alert him that the CEO is on a call with the acquisition lawyers, Joe can mix the sound of people arguing into the outgoing RTP stream. To Joe, the fun part about this particular attack is that the CEO has no idea the attack is going on. It's only in the *outbound* stream to the lawyers. They hear the arguing and ask the CEO what is going on. The CEO has no idea what they are talking about.

p0055 In the end, you can imagine that Joe probably got caught – but not before causing a good degree of confusion and annoyance – and maybe sabotaging the acquisition as well.

p0060 Does this all sound like fiction or a Hollywood movie? Unfortunately, it's a very real possibility *if an attacker can get to the right point in your network*. Voice, video, and IM – the cornerstones of unified communications (UCs) – can be both observed and modified by an attacker with access to the correct point in the network. Let's look at this in more detail.

s0010 ANATOMY OF EAVESDROPPING AND MODIFICATION ATTACKS

p0065 For an attacker to make these attacks, he or she has to get between the endpoints and then use various tools to pull off the attacks. You need to understand one important distinction between eavesdropping attacks and modification attacks.

p0070 Eavesdropping attacks are far easier and can be passive; that is, a piece of software can simply be sitting somewhere in the network path and capturing all the relevant network traffic for later analysis. In fact, the attacker does not need to have any ongoing connection to the software at all. He or she can insert the software onto a compromised device, perhaps by direct insertion or perhaps by a virus or other malware, and then come back some time later to retrieve any data that is found or trigger the software to send the data at some determined time. The point is that you may have no idea that the software is there monitoring and capturing all your traffic. It's a very simple and straightforward attack on the confidentiality of your system if the attacker can get between the endpoints.

p0075 Modification attacks have the same need as eavesdropping attacks to get to the right point in the network, but they also have a timing requirement. The attacks are only useful if you can modify the communications stream while the communication is taking place. The attacker also has to insert his or her software in the network path in a true man-in-the-middle (MiTM) attack where he or she is able to not just observe packets, but actually receive the packets, modify them, and send them on.

The classic example is if you were able to get between someone calling their financial broker and when the person said to "buy 10,000 shares," you were able to change what the person said to "sell 10,000 shares." Such attacks are possible, but they require not only being able to get to the right point in the network but also to be able to time the attack exactly. With voice or video, this could be rather difficult. With text-based mediums like IM, it's obviously a bit easier because the attacker has text that can be scanned and modified. p0080

Modification attacks could be performed by code that is inserted and left behind, particularly if the target media is text-based like IM, but other tools out there do require the active participation of the attacker to get the timing just right. p0085

Let's look at mechanisms to get between the two endpoints and then at a couple of specific attacks. p0090

NOTE

If you go back to the "CIA triad" referenced in the introduction to the chapter, modification attacks are against the integrity of a communications system: the information received by the recipient is not the same information that was sent by the sender. p0095

Getting between the Endpoints s0015

The attacks outlined in the introduction to the chapter work by taking advantage of the way many UCs systems separate *signaling* (also often referred to as *call control*) from *media*. As shown in Figure 3.1, the signaling for a session in a Session Initiation Protocol (SIP)-based system may take a different network path from the media sent between the endpoints. p0100

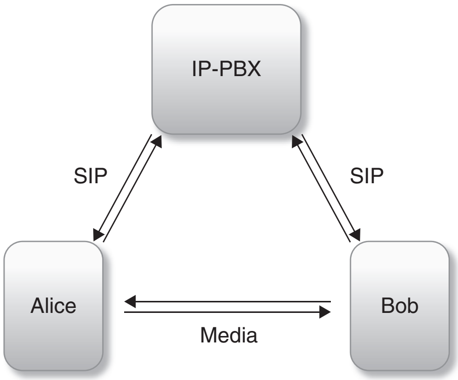


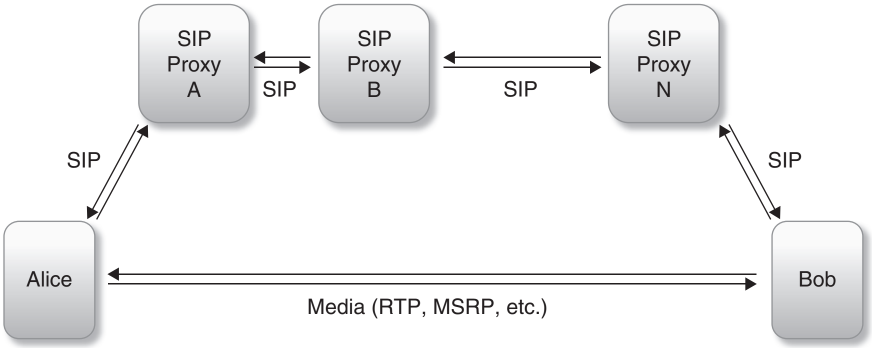
FIGURE 3.1 f0010

With SIP, Signaling and Media Take Different Paths p0105 With SIP, the person initiating the voice, video, or IM session sends an initial message (called a SIP INVITE) from their endpoint to the recipient. The INVITE may pass through one or more SIP proxy servers until it reaches the recipient's endpoint, as shown in Figure 3.2. The endpoints then send further SIP packets to negotiate what type of media will be sent between the endpoints, the addresses (IP or host) to which the media will be sent, and any other options related to the session.

p0110 Once the media session has been negotiated, the endpoints start sending media to each other. For voice or video sessions, the media will be sent as RTP (defined in RFC 3550[A]) packets. For IM, the media will be sent as Message Session Relay Protocol (MSRP, defined in RFC 4975[B]) packets. Depending upon the network infrastructure, the endpoints may or may not stream the media directly from endpoint to endpoint. There may also be media servers or session border controllers (SBCs) or other devices between the two endpoints.

NOTE

p0115 For voice and video, SIP has become the primary industry-standard signaling protocol for communication between endpoints. For IM, though, SIP and it's "SIMPLE" derivative is just one of the two major open standards for IM. The other major protocol, the Extensible Messaging and Presence Protocol (XMPP), also known as the *Jabber Protocol*, has a different model where the session initiation and messaging are sent from the XMPP client to a XMPP server and from there on through other servers to the recipient endpoint. Unlike SIP/SIMPLE, XMPP does not have separate channels for signaling and media. All the IM traffic occurs within the XMPP stream itself. However, the XMPP community has been developing Jingle,[C] a framework for using XMPP for multimedia traffic such as voice and video. Jingle typically adopts a similar model to that of the SIP space, where the signaling goes over XMPP and the media (typically RTP) goes directly from endpoint to endpoint (and potentially through media servers).



f0015 **FIGURE 3.2**

SIP Traffic May Pass through Multiple Proxy Servers

fn0010 [A] http://tools.ietf.org/html/rfc3550
fn0015 [B] http://tools.ietf.org/html/rfc4975
fn0020 [C] http://xmpp.org/tech/jingle.shtml

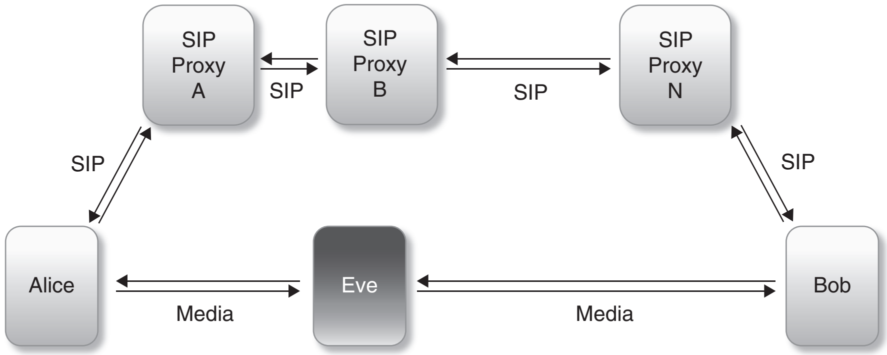


FIGURE 3.3 f0020

An Attacker, Eve, Needs to Get Somewhere between the Two Endpoints

The trick, then, is for the attacker to get himself or herself between the two endpoints in either the signaling or the media streams, as shown in Figure 3.3. p0120

The attacker can potentially observe and modify network traffic if he or she can p0125

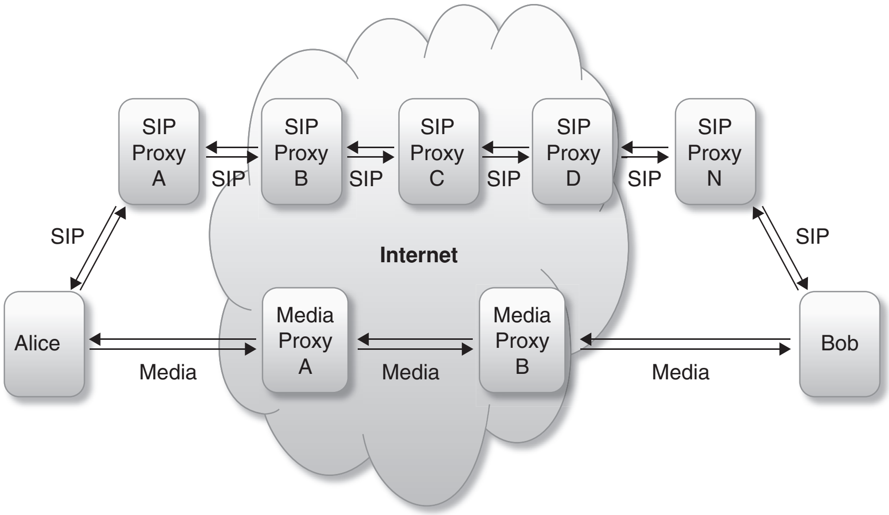
- get in the *network path* between the two endpoints p0130
- get between two of the *servers* or *proxies* involved with sending the traffic between the endpoints p0135
- get on the same *network segment* as one of the endpoints p0140
- compromise the *local system* of either endpoint. p0145

Let's look at each of these in a bit more detail. p0150

Get in the Network Path s0020

The reality is that the picture in Figure 3.2 is a lot more complicated than is shown in the simple diagram. For communication across a wide area network (WAN) or across the public Internet, the picture may look a lot more like Figure 3.4, with many network points between two endpoints. As the media traffic traverses the network, it has to pass through any number of network routers, each one of which is a potential point where an attacker could be able to insert code to observe and/or modify media traffic. The media stream may also pass through one or more *media proxies* that are designed to pass the media from one network segment to another. p0155

If an attacker can compromise a router or other device such as a firewall, SBC, or media server, he or she can then observe all the traffic flowing through the network device. In the Pena/Moore Voice over Internet Protocol (VoIP) fraud case to be discussed in Chapter 4, "Control Channel Attacks: Fuzzing, DoS, SPIT, and Toll Fraud," Pena and Moore were able to compromise a large number of network devices simply by logging in with default usernames and passwords. Such devices also have vulnerabilities discovered over time and if they are left unpatched, attackers can exploit publicly known vulnerabilities to compromise network devices and obtain a higher level of access to those devices. p0160



f0025 **FIGURE 3.4**

The Network Path between Two Endpoints May Be Very Complex

WARNING

p0165 Remember that the security of your UCs system relies on the security of the underlying IP network. Have all the devices on the edge of your network been checked for vulnerabilities lately? Do you have them included in patch management plans to be sure they are up-to-date with any available patches? How strong are the passwords for the admin accounts on network devices? How often are your networks checked for rogue wireless access points and modems? Are your employees trained to identify and report social engineering attacks?

p0170 The challenge is of course how to find the path between two endpoints, particularly when the very design of the Internet is to allow multiple paths for traffic to flow. It's not impossible to do, but it's also not trivial. However, as traffic flows between the endpoints across larger and larger networks, and particularly the public Internet, the number of network points between the endpoints continues to expand and the possible points of compromise expand. If your UCs system has endpoints that are out across the public Internet, for instance, you then have to worry about the security of every possible Internet service provider (ISP) between your corporate UC system and the remote endpoint. (And the reality is that you can't know about the security of every ISP and therefore need to use one of the solutions discussed in the section "How to Defend against Eavesdropping and Modification Attacks" at the end of this chapter.)

s0025 *Get between Two Servers or Proxies*

p0175 One mechanism for an attacker to try to get into the path is to try to get between two of the servers involved with the communication. Now, as mentioned previously, the media may stream directly from one endpoint to the other in a completely "peer-to-peer" fashion. However, even in a peer-to-peer arrangement, the media may still pass through a network device such as a SBC that sits on the edge of a network and acts as a proxy to send the traffic out onto a public network. In most IM networks, Skype being perhaps the only major exception, all the traffic is routed from server-to-server. Your IM client connects to its local server and IM traffic goes to that local server, and then from that server to another server, and then on until it reaches the destination network.

p0180 An attacker may be able to identify these "servers" by the amount of traffic flowing out of them and then target those servers – or the path between those servers – as where a compromise needs to occur.

Get on the Local Network Segment s0030

p0185 If an attacker can obtain access to the local network segment where one of the endpoints is located, he or she can potentially sniff the network for the media traffic and intercept and/or modify the traffic. A classic case where this can happen is with an unsecured Wi-Fi network where an attacker can use any of the many available wireless packet sniffing tools to see the traffic on the Wi-Fi network. This could be a "rogue" Wi-Fi network at your corporate location or it could be the Wi-Fi café where a remote employee is working.

p0190 The attack vector could also be an unsecured Ethernet port in a lobby or conference room, but this requires physical access to the ports (versus being out in the parking lot with Wi-Fi) and is probably less likely. More probable than either the Wi-Fi or Ethernet attack may be an attacker compromising a computer on the local subnet, perhaps by way of malware (virus, malware, bot, and so on).

Compromise the Local System of Either Endpoint s0035

p0195 Another avenue for an attacker is to compromise the security of the local system serving as either endpoint of the connection. For instance, if the attacker can convince you to download some malware or otherwise have your system infected, he or she can get their software installed directly on the system initiating communications sessions. The attacker can now log all communication locally and potentially record all audio or video sessions and then send them to an external server at some point.

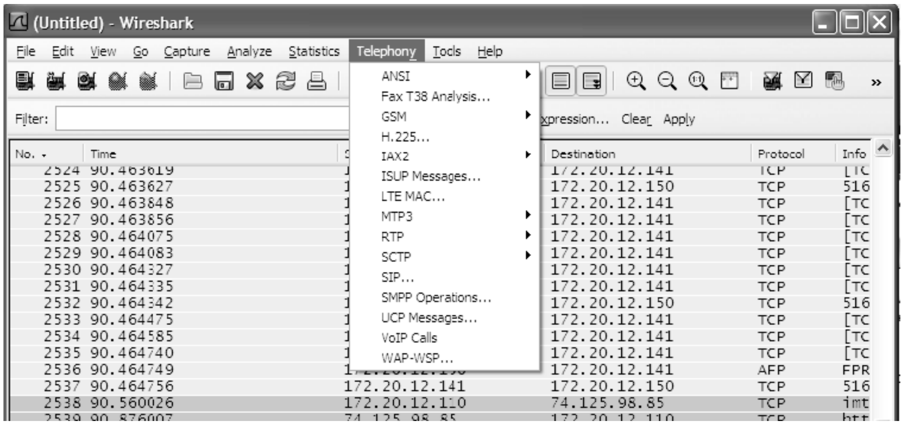
p0200 Note that this approach has the added benefit for an attacker that it may be possible to defeat encryption mechanisms by simply recording the audio from the local system before it enters an outbound encrypted stream. In January 2008, there was a widely publicized case where a division of the German government was reported to be considering[D] such an approach specifically to be able to tap into communications made over the Skype network.

[D] http://skypejournal.com/blog/2008/01/the_bavarian_intercept_proves.html fn0025

s0040 Using Wireshark to Capture Voice

p0205 As mentioned in the introduction to this chapter, Wireshark,[E] the industry-standard free network protocol analyzer that is widely used for network administration (and was previously known as *Ethereal*), has some solid capabilities with regard to capturing and interpreting VoIP Calls. As shown in Figure 3.5, the latest version 1.2.4 of Wireshark has a **Telephony** menu in it with a range of options.

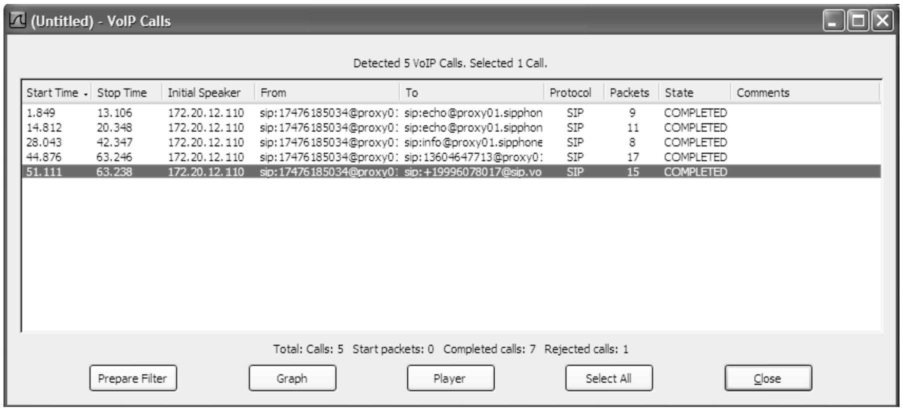
p0210 If you select **VoIP Calls** from the **Telephony** menu, you will see a list of what calls Wireshark found in the packets it captured, as shown in Figure 3.6. From



```
No. -  Time                                Destination        Protocol  Info
  2524 90.463619   1                       172.20.12.141      TCP       [TC
  2525 90.463627   1                       172.20.12.150      TCP       516
  2526 90.463848   1                       172.20.12.141      TCP       [TC
  2527 90.463856   1                       172.20.12.141      TCP       [TC
  2528 90.464075   1                       172.20.12.141      TCP       [TC
  2529 90.464083   1                       172.20.12.141      TCP       [TC
  2530 90.464327   1                       172.20.12.141      TCP       [TC
  2531 90.464335   1                       172.20.12.141      TCP       [TC
  2532 90.464342   1                       172.20.12.150      TCP       516
  2533 90.464475   1                       172.20.12.141      TCP       [TC
  2534 90.464585   1                       172.20.12.141      TCP       [TC
  2535 90.464740   1                       172.20.12.141      TCP       [TC
  2536 90.464749   1                       172.20.12.141      AFP       FPR
  2537 90.464756   172.20.12.141           172.20.12.150      TCP       516
  2538 90.560026   172.20.12.110           74.125.98.85       TCP       imt
  2539 90.876007   74.125.98.85            172.20.12.110      TCP       htt
```

f0030 **FIGURE 3.5**

Wireshark Includes a Telephony Menu



Detected 5 VoIP Calls. Selected 1 Call.

Start Time	Stop Time	Initial Speaker	From	To	Protocol	Packets	State	Comments
1.849	13.106	172.20.12.110	sip:17476185034@proxy01	sip:echo@proxy01.sipphon	SIP	9	COMPLETED	
14.812	20.348	172.20.12.110	sip:17476185034@proxy01	sip:echo@proxy01.sipphon	SIP	11	COMPLETED	
28.043	42.347	172.20.12.110	sip:17476185034@proxy01	sip:info@proxy01.sipphone	SIP	8	COMPLETED	
44.876	63.246	172.20.12.110	sip:17476185034@proxy01	sip:13604647713@proxy01	SIP	17	COMPLETED	
51.111	63.238	172.20.12.110	sip:17476185034@proxy01	sip:+19996078017@sip.vo	SIP	15	COMPLETED	

Total: Calls: 5 Start packets: 0 Completed calls: 7 Rejected calls: 1

f0035 **FIGURE 3.6**

Wireshark Shows You All of Your VoIP Calls

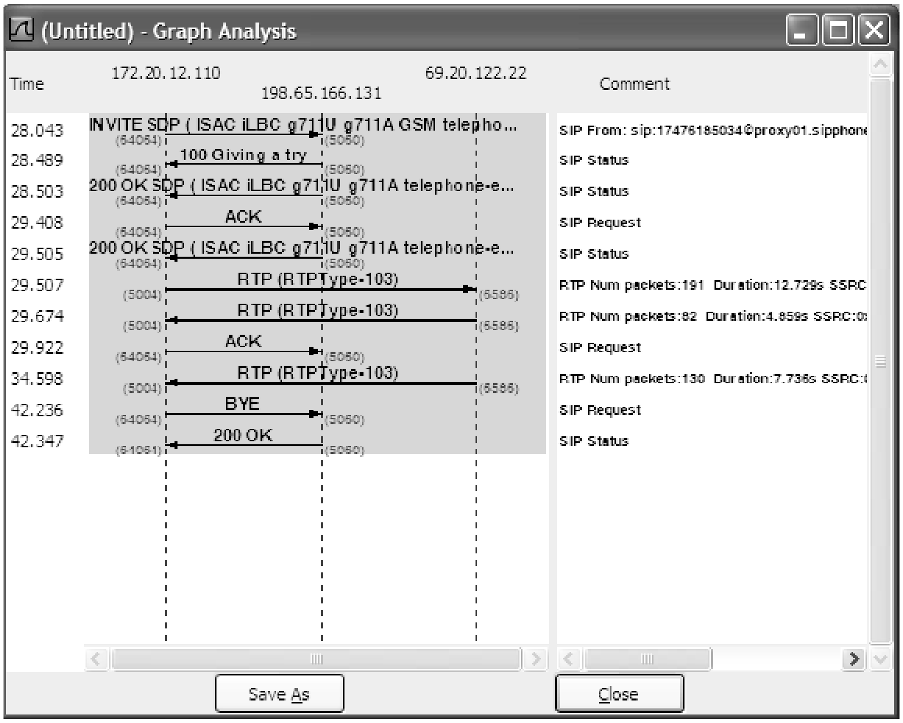
fn0030 [E] You can download Wireshark for free for Microsoft Windows, Linux/UNIX, or Apple Mac OS X at www.wireshark.org/

here, you have a couple of options. If you select any one of the calls and click the **Graph** button, you get a great chart such as the one in Figure 3.7 that shows the actual flow of SIP and RTP messages during the course of this particular call. This is actually a great way to learn about how network traffic flows in a SIP-based system.

Back in the **VoIP Calls** window, if you select a call and press the **Player** button and then **Decode** on the next screen, you will then see an audio player similar to Figure 3.8 and have the ability to listen to either side of the conversation. Just click into one of the two audio streams and press the **Play** button to get started. p0215

When you enter the **RTP Player** in Wireshark, you may need to check the **Use RTP timestamp** check box to have your audio correctly interpreted. After you check the box, you'll need to press the **Decode** button after which you should see your audio in the player window. Note also that the RTP Player does not support all possible audio formats, so it may not always work for audio you have captured. p0220

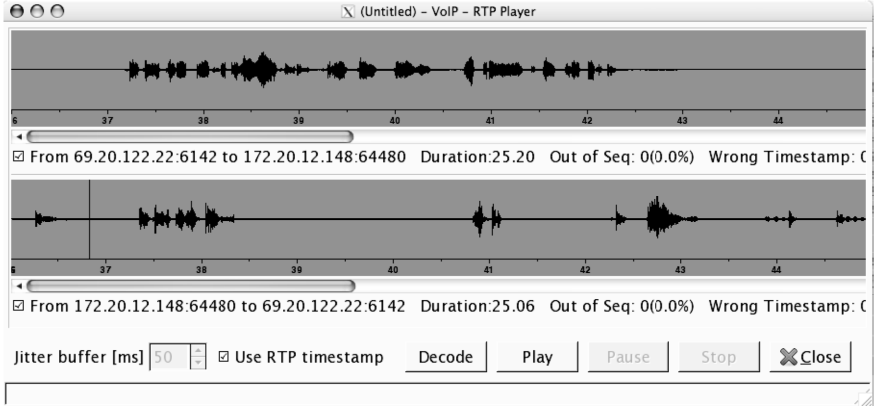
Wireshark also has the ability to save audio streams to files for later listening, although the path to do so is not exactly intuitive. If you select an **RTP packet** in the capture window, you can select the menus **Telephony | RTP | Stream Analysis**…. If you don't have an RTP packet selected, you can select the menus **Telephony | RTP | Select All Streams**, p0225



Time	172.20.12.110		198.65.166.131	69.20.122.22	Comment
28.043	(54054)	INVITE SDP (ISAC iLBC g711U g711A GSM telepho...	(5060)		SIP From: sip:17476185034@proxy01.sipphone
28.489	(54054)	100 Giving a try	(5060)		SIP Status
28.503	(54054)	200 OK SDP (ISAC iLBC g711U g711A telephone-e...	(5060)		SIP Status
29.408	(54054)	ACK	(5060)		SIP Request
29.505	(54054)	200 OK SDP (ISAC iLBC g711U g711A telephone-e...	(5060)		SIP Status
29.507	(5004)	RTP (RTPType-103)		(6586)	RTP Num packets:191 Duration:12.729s SSRC
29.674	(5004)	RTP (RTPType-103)		(6586)	RTP Num packets:82 Duration:4.859s SSRC:0
29.922	(54054)	ACK	(5060)		SIP Request
34.598	(5004)	RTP (RTPType-103)		(6586)	RTP Num packets:130 Duration:7.736s SSRC:
42.236	(54054)	BYE	(5060)		SIP Request
42.347	(54054)	200 OK	(5060)		SIP Status

FIGURE 3.7 f0040

Wireshark Can Easily Show You the Messages in the Flow of a Call



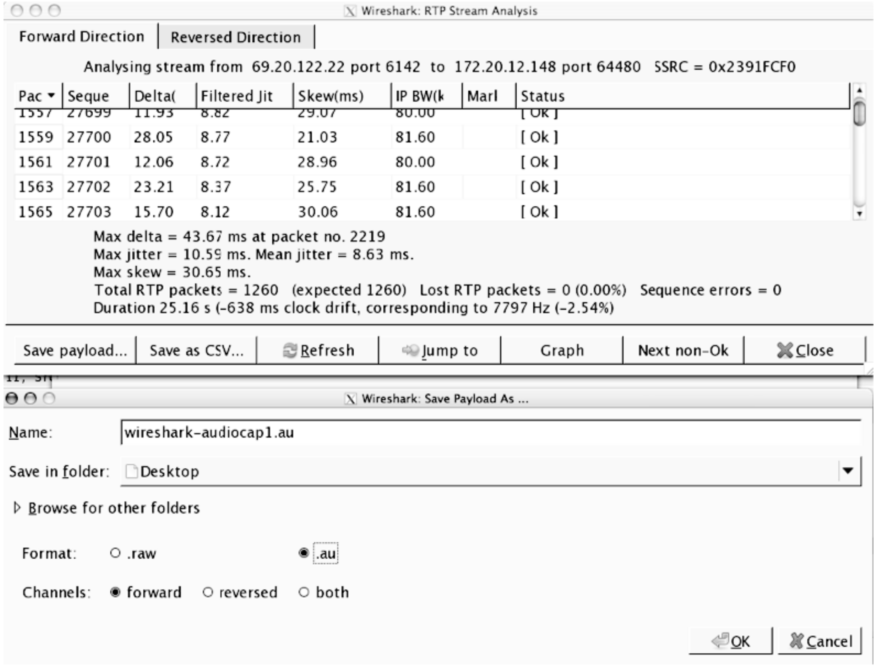
☑ From 69.20.122.22:6142 to 172.20.12.148:64480 Duration:25.20 Out of Seq: 0(0.0%) Wrong Timestamp: (

☑ From 172.20.12.148:64480 to 69.20.122.22:6142 Duration:25.06 Out of Seq: 0(0.0%) Wrong Timestamp: (

☑ Use RTP timestamp

f0045 **FIGURE 3.8**

Wireshark's Audio Player Lets You Listen to Captured Conversations



Analysing stream from 69.20.122.22 port 6142 to 172.20.12.148 port 64480 SSRC = 0x2391FCF0

Pac ▾	Seque	Delta(	Filtered Jit	Skew(ms)	IP BW(k	Marl	Status
1557	27699	11.93	8.82	29.07	80.00		[Ok]
1559	27700	28.05	8.77	21.03	81.60		[Ok]
1561	27701	12.06	8.72	28.96	80.00		[Ok]
1563	27702	23.21	8.37	25.75	81.60		[Ok]
1565	27703	15.70	8.12	30.06	81.60		[Ok]

```
Max delta = 43.67 ms at packet no. 2219
Max jitter = 10.59 ms. Mean jitter = 8.63 ms.
Max skew = 30.65 ms.
Total RTP packets = 1260   (expected 1260)   Lost RTP packets = 0 (0.00%)   Sequence errors = 0
Duration 25.16 s (-638 ms clock drift, corresponding to 7797 Hz (-2.54%)
```

Name: wireshark-audiocap1.au

Save in folder: Desktop

Format: ○ .raw ● .au

Channels: ● forward ○ reversed ○ both

f0050 **FIGURE 3.9**

Wireshark Lets You Save RTP Audio Payloads to Files on Disk

choose a stream, and press the **Analyze** button. In both cases, you will then wind up in an analysis window resembling the top portion of Figure 3.9. By clicking the **Save payload**… button, you will bring up a screen like that on the bottom of Figure 3.9 that will let you save the RTP audio payload out as an audio file.

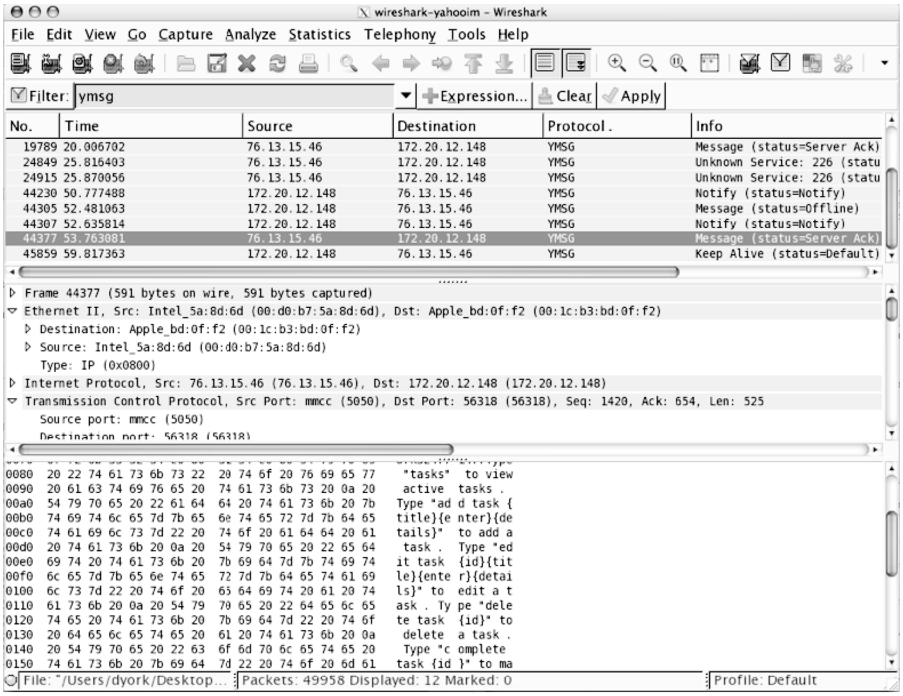
Note that there are other tools out there that make this process easier, but Wireshark does have the basic functionality. p0230

EPIC FAIL

A college installed a shiny new IP-PBX on its campus and installed IP phone endpoints in each of the student rooms in a residence hall. It wasn't long before some enterprising (or bored) student discovered that all the residence hall phones were on the same local network and with an easy tool like Wireshark, the students could start listening to any phone calls made over the IP phone network! Oops. Needless to say, the college quickly tried to figure out how to enable encryption on its network. p0235

Using Wireshark to Capture IM Traffic s0045

Wireshark can, of course, be used to capture and analyze IM traffic, as well as voice. The major difference is that there is not an entire menu in the Wireshark tool devoted to IM as there is for telephony. With a little bit of understanding what protocols are used by the various services, you can find the relevant traffic within your Wireshark captures. Figure 3.10 shows a capture of Yahoo!Messenger traffic where p0240


wireshark-yahooim – Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: ymsg
Expression...
Clear
Apply

```
No.   Time       Source         Destination    Protocol  Info
19789 20.006702  76.13.15.46    172.20.12.148  YMSG      Message (status=Server Ack)
24849 25.816403  76.13.15.46    172.20.12.148  YMSG      Unknown Service: 226 (statu
24915 25.870056  76.13.15.46    172.20.12.148  YMSG      Unknown Service: 226 (statu
44230 50.777488  172.20.12.148  76.13.15.46    YMSG      Notify (status=Notify)
44305 52.481063  172.20.12.148  76.13.15.46    YMSG      Message (status=Offline)
44307 52.635814  172.20.12.148  76.13.15.46    YMSG      Notify (status=Notify)
44377 53.763081  76.13.15.46    172.20.12.148  YMSG      Message (status=Server Ack)
45859 59.817363  172.20.12.148  76.13.15.46    YMSG      Keep Alive (status=Default)
```

```
▷ Frame 44377 (591 bytes on wire, 591 bytes captured)
▽ Ethernet II, Src: Intel_5a:8d:6d (00:d0:b7:5a:8d:6d), Dst: Apple_bd:0f:f2 (00:1c:b3:bd:0f:f2)
  ▷ Destination: Apple_bd:0f:f2 (00:1c:b3:bd:0f:f2)
  ▷ Source: Intel_5a:8d:6d (00:d0:b7:5a:8d:6d)
    Type: IP (0x0800)
▷ Internet Protocol, Src: 76.13.15.46 (76.13.15.46), Dst: 172.20.12.148 (172.20.12.148)
▽ Transmission Control Protocol, Src Port: mmcc (5050), Dst Port: 56318 (56318), Seq: 1420, Ack: 654, Len: 525
    Source port: mmcc (5050)
    Destination port: 56318 (56318)
```

```
0080  20 22 74 61 73 6b 73 22  20 74 6f 20 76 69 65 77    "tasks"  to view
0090  20 61 63 74 69 76 65 20  74 61 73 6b 73 20 0a 20    active  tasks .
00a0  54 79 70 65 20 22 61 64  64 20 74 61 73 6b 20 7b   Type "ad d task {
00b0  74 69 74 6c 65 7d 7b 65  6e 74 65 72 7d 7b 64 65   title}{e nter}{de
00c0  74 61 69 6c 73 7d 22 20  74 6f 20 61 64 64 20 61   tails}"  to add a
00d0  20 74 61 73 6b 20 0a 20  54 79 70 65 20 22 65 64    task .  Type "ed
00e0  69 74 20 74 61 73 6b 20  7b 69 64 7d 7b 74 69 74   it task  {id}{tit
00f0  6c 65 7d 7b 65 6e 74 65  72 7d 7b 64 65 74 61 69   le}{ente r}{detai
0100  6c 73 7d 22 20 74 6f 20  65 64 69 74 20 61 20 74   ls}" to  edit a t
0110  61 73 6b 20 0a 20 54 79  70 65 20 22 64 65 6c 65   ask . Ty pe "dele
0120  74 65 20 74 61 73 6b 20  7b 69 64 7d 22 20 74 6f   te task  {id}" to
0130  20 64 65 6c 65 74 65 20  61 20 74 61 73 6b 20 0a    delete  a task .
0140  20 54 79 70 65 20 22 63  6f 6d 70 6c 65 74 65 20    Type "c omplete
0150  74 61 73 6b 20 7b 69 64  7d 22 20 74 6f 20 6d 61   task {id }" to ma
```

File: "/Users/dyork/Desktop... Packets: 49958 Displayed: 12 Marked: 0 Profile: Default

FIGURE 3.10 f0055

Wireshark Can Show IM Traffic Such as Yahoo!Messenger t0010 **Table 3.1** IM services and Wireshark display filters

IM service	Wireshark display filter
AOL Instant Messenger	aim
Internet Relay Chat	irc
Jabber/XMPP/GoogleTalk	jabber
Microsoft MSN Messenger/Windows Live Messenger	msnms
SIMPLE	sip
Yahoo!Messenger	ymsg

the message of the text is readable. You can see at the top of the Wireshark window that the display filter has been set to *ymsg* so that only Yahoo!Messenger messages are displayed. Table 3.1 shows the text you can use as a display filter for common IM protocols.

p0245 Notice that "SIMPLE," the SIP-based protocol for IM mentioned earlier in the chat, has only "sip" as the display filter. Because SIMPLE is based on SIP, you actually want to filter on SIP and then look through for the SIMPLE messages. Alternatively, you could also filter on *msrp*, the MSRP, which is basically the IM equivalent of how RTP is used for audio.

p0250 Now as you explore the different IM conversations you capture, you may find that a number of them are unreadable. For instance, you may see in MSN or Jabber conversations who or where the participants are in an exchange, but the actual body of the exchange is not readable. This is because the IM clients being used are in fact encrypting the messages between the IM clients and the IM servers. Many of the current products ship with encryption on by default and while it is always possible for a user to turn the encryption *off*, odds are that they won't. It may also just be part of the UC system. For instance, Microsoft in their Office Communication Server uses Transport Layer Security (TLS) encryption to secure the transport of its SIMPLE-based communication.

NOTE

p0255 The Skype exception – You may have noticed that there has been no discussion on how to intercept Skype IM, voice, or video calls. The truth is that it is an extremely difficult task to accomplish. Skype does encrypt all of its signaling, voice, video, and IM, and while the security community may strongly dislike the lack of peer review of Skype's encryption protocol, the fact is that it does protect the transport of communication over Skype. The only real attack scenario identified thus far is to attempt to compromise local systems and install some type of monitoring system. Security researchers continue to probe for Skype's weaknesses, but in the meantime that is why Skype is missing from these tables and sections.

Capturing Audio, Video, and IM using Other Tools s0050

p0260 There are, of course, many other tools beyond Wireshark that let you capture voice, video, and IM conversations. Wireshark has been demonstrated here primarily because it should be familiar to most network administrators and also because it is cross-platform (Windows, Linux/UNIX, and Mac OS X), and therefore easy for you to download and experiment with. Let us, though, take a quick tour of some of the other tools available. p0265

- p0270 **UCSniff** (http://ucsniff.sourceforge.net/) A newer tool for Windows or Linux, from Jason Ostrom and Arjun Sambamoorthy at Sipera's Viper Labs, can find and record both voice and video conversations and save them to a file for later listening. It supports a wide range of codecs, real-time monitoring, MiTM attacks, virtual local area network hopping, and more. It integrates a number of existing tools into one easy-to-use package.
- p0275 **VideoSnarf** (http://ucsniff.sourceforge.net/videosnarf.html) Another tool from the Sipera Viper Labs team that provides a subset of the UCSniff functionality and focuses only on extracting H.264 video streams from the RTP streams.
- p0280 **Cain & Abel** (http://www.oxid.it/cain.html) It is primarily a password recovery tool for Windows, and it also includes the ability to record VoIP audio conversations to files for later listening.
- p0285 **Oreka** (http://oreka.sourceforge.net/) An open-source call recording solution for Windows or Linux that monitors RTP streams on the network and captures them into audio files and then presents a Web interface allowing you to access the recordings. The project claims that it has been tested to work with a number of common IP-PBX and other similar VoIP systems.
- p0290 **VoIPong** (http://www.enderunix.org/voipong) An older program (circa 2005) that identifies VoIP Calls that are G.711 encoded and dumps them to WAV files AQ2 for listening.
- p0295 **Vomit** (http://vomit.xtdnet.nl/) One of the earliest tools, "Voice over Misconfigured Internet Telephones" will retrieve a Cisco IP phone conversation from a tcpdump-formatted packet capture and convert it to a WAV file for listening.

p0300 There are certainly other tools out there as well, but these are some of the more common ones you will see discussed in security-related articles and information.

Modification Attacks s0055

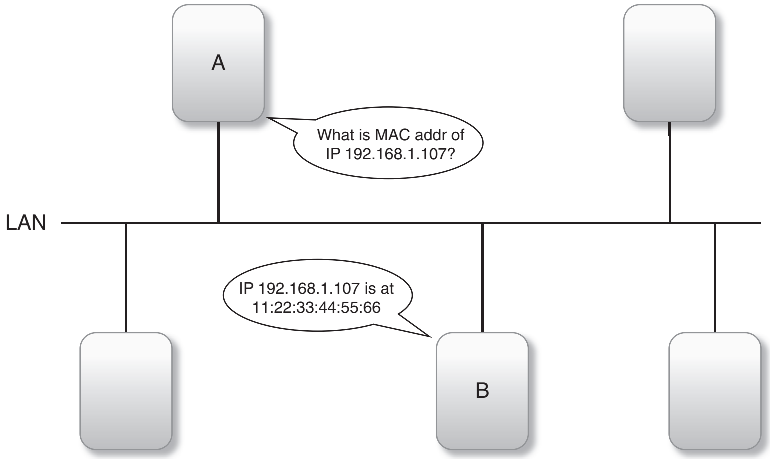
p0305 In an attack that modifies the media stream, the attacker's software injects itself in between the sender and the recipient in a true MiTM attack, as shown earlier in Figure 3.3. Whether the media is voice, video, or IM text, the idea is the same. The attacker sets the software up so that it relays the media stream unmodified for almost all the packets and then modifies the individual packets critical to the attack. Given that the senders and recipients would not see any modification until the attack, the software could sit in the network for weeks, months, or even years until it is activated for the attack.

s0060
Ettercap

p0310 There are several different programs out there for performing network MiTM attacks, but perhaps the best known is Ettercap (http://ettercap.sourceforge.net/). Ettercap uses "ARP poisoning" (also called *ARP spoofing*) to make other computers on a local network believe that it is a different computer. A full discussion of Address Resolution Protocol (ARP) attacks is a bit beyond the scope of this book, but the basic idea is that on a local network segment, network traffic needs to be reduced from IP addresses down to the actual Ethernet addresses assigned to network interface cards. ARP is the protocol used to provide this IP address to MAC address mapping.

p0315 Let's look at a simplified example. Computer A with IP address 192.168.1.100 wants to send a message to Computer B with IP address 192.168.1.107. Because they both reside on the same local network segment and no routing needs to be performed, Computer A sends out a broadcast ARP message on the local network asking for the MAC address of 192.168.1.107. Computer B responds back that its MAC address is 11:22:33:44:55:66 and now Computer A can start sending Ethernet frames directly to Computer B. This is basically how ARP works and is shown in Figure 3.11. The other element here is that Computer A will *cache* the MAC address for Computer B in its local ARP cache so that it doesn't have to issue an ARP for every frame it needs to send. Computer A will maintain the address for Computer B in its ARP cache for a certain period of time and then will send out a new ARP packet to make sure the address is the same.

p0320 What Ettercap does is send out fake ARP messages that point an IP address to the attacker's computer. In our example, let's say that Ettercap is running on Computer E. When Ettercap is launched, it may send out an ARP response indicating that 192.168.1.107 (and any other IP addresses) now point to Computer E's address


A
What is MAC addr of IP 192.168.1.107?
LAN
IP 192.168.1.107 is at 11:22:33:44:55:66
B

f0060 **FIGURE 3.11**

Two Computers Using ARP to Find MAC Addresses of 66:55:44:33:22:11. Computer A, seeing this ARP packet, would update its local ARP cache to now start streaming packets for "Computer B" to 66:55:44:33:22:11. Similarly, Computer E would send a fake ARP packet to Computer B so that it would update its local ARP cache for Computer A's address to point to Computer E. The end result is that Computer A thinks Computer E is Computer B, and Computer B thinks Computer E is Computer A. This attack is shown in Figure 3.12.

Now that the attacker is between the two computers, he or she can observe the traffic flowing between the two points on the network and also modify the traffic. Ettercap supports *filters* that allow for the modification of network traffic. The software includes a filter creator and a number of prebuilt filters you can use. The basic idea is to create a filter that detects a certain pattern in the network packet flow and then substitutes some other data for that pattern. p0325

RTP InsertSound and RTP MixSound s0065

For their book "Hacking Exposed VoIP: Voice over IP Security Secrets and Solutions" (ISBN: 978-0-07-226364-0), Mark Collier and David Endler created a number of tools for security professionals on their Web site (www.hackingvoip.com) including two worth mentioning here. *RTP InsertSound* is a tool that can insert audio into a RTP stream by tricking the receiving endpoint into accepting the attacker's RTP packets instead of the legitimate RTP packets. If you go back to the attack described in the beginning of the section "Anatomy of Eavesdropping and Modification Attacks" where the word "buy" was replaced with the word "sell," RTP InsertSound could be used to attempt those types of attacks. p0330

RTP MixSound is a more devious tool. It mixes an audio stream into an existing RTP stream. If you go back to the scenario at the beginning of the chapter where Joe mixed the sounds of an argument into the outgoing call from the CEO, RTP p0335

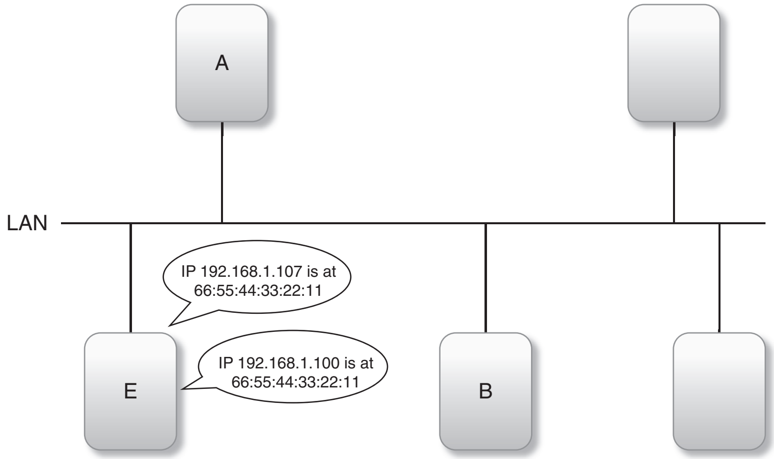

A
LAN
IP 192.168.1.107 is at 66:55:44:33:22:11
IP 192.168.1.100 is at 66:55:44:33:22:11
E
B

FIGURE 3.12 f0065

An Attacker Has Used Ettercap to Get between Two Computers

MixSound could be used to execute attacks like this. If someone were working from home, an attacker could mix in sounds of an amusement park. If someone were working late and called home to their spouse, the attacker could mix in sounds of someone of the opposite sex. Alternatively, the attacker could mix in profanity into the outgoing stream for a customer support line, thus potentially angering the customer who called. The kinds of attacks are really limited only by your imagination.

p0340 The entertaining part for the attacker is that the new audio is only mixed into one of the RTP streams. In the examples here, for instance, the attacker could mix it into the streams coming from the caller. The recipient would then hear the mixed audio, but the caller would not. The home worker is suddenly being asked to explain why it sounds like he is at an amusement park. He or she has no clue why they are being asked about this as they don't hear the sounds on their RTP stream. You could imagine the confusion (and marital problems) this could create with the calls to the spouse! Similarly, the attacker could mix sound into only one leg of a multiparty conference call and only into the stream heard by that one recipient. The recipient might then be asking the others on the call about the sound, which they do not hear at all.

p0345 RTP MixSound and RTP InsertSound are not the only tools out there that do this, but they are examples of what could be done. It's worth noting that these two tools do not presume that you are able to successfully pull off a MiTM attack. As long as you are on the same network segment, these tools can send RTP packets to target endpoints and have a variety of tricks to try to convince the endpoint to accept the bogus RTP packets as real.

TIP

p0350 More media manipulation tools can be found on the Voice over IP Security Alliance (VOIPSA) tools list (www.voipsa.org/Resources/tools.php).

s0070 DANGERS OF EAVESDROPPING AND MODIFICATION ATTACKS

p0355 While many of the dangers of eavesdropping and modification attacks have been discussed in the previous sections of this chapter, in this section, you will learn more about several of the specific dangers.

s0075 Exposure of Confidential Information

p0360 Obviously, the most visible and tangible attack is the exposure of confidential information. If someone can gain access to the communication stream inside of a company, they can potentially learn confidential corporate information that could then possibly be used for malicious purposes. This could be information about finances, about new products, and about personnel or any other matter related to the company.

It could also be information from an individual, such as when a person calls into their bank and speaks with someone there. The attacker could use that information for financial gain, public embarrassment (and corresponding reputation loss), or other purposes.

Eavesdropping on UC systems could also be a conduit to other kinds of attacks. p0365
Imagine, for instance, that an attacker listens to voice or video calls indicating that the office will be empty for a certain period of time and that something of value is stored in the office. Or imagine that an attacker intercepts someone IM'ing the code to get through the door alarm.

WARNING

Be aware that with voice communications, an attacker might not need to actually gain p0370
access to the media stream to obtain confidential information. If a caller is using "dual tone, multifrequency" (DTMF) tones (also known as *touch tones*) to enter information such as a credit card number or voice-mail password, those DTMF tones might travel over the SIP control channel using the method defined in RFC 4733[F] (formerly RFC 2833[G]) and could therefore be obtained via the SIP control channel versus the media channel.

Business Disruption s0080

If a modification attack is successful, it is possible to seriously disrupt the operations p0375
of a business. Obviously, there is the blatant case mentioned previously where an attacker changes the use of the word "buy" to "sell" and could potentially create a financial cost to the company. But there could easily be more subtle attacks. Slightly changing the number of units to ship mentioned in an IM message from, say, 150 to 125, could cause a more nuanced disruption of a production process. The possibilities are really only limited by your imagination.

Annoyance s0085

Modification attacks also bring the great opportunity to simply create annoying p0380
situations and create internal discord within a company or organization. It could be the mixing of an argument into an outbound media stream as suggested in the scenario back in the introduction to the chapter. It could be mixing in the sound of an amusement park into the background of someone who is working from home. It could be dropping out random words from IM messages or adding in more words. Odds are that these types of attacks may not be perpetrated by an actual external attacker, but rather by someone inside the company intent on annoying or harassing other employees.

[F] http://tools.ietf.org/html/rfc4733 fn0035

[G] www.disruptivetelephony.com/2007/11/did-you-know-rf.html fn0040

s0090 Loss of Trust

p0385 With attacks that are designed to disrupt or annoy, there is also a corresponding loss of trust in the communication system and potentially a loss of trust in *you* if you are responsible for that system. People may come to discount the system or believe that it is not all that you or other advocates have made it out to be.

s0095 THE FUTURE OF EAVESDROPPING AND MODIFICATION ATTACKS

p0390 As companies continue to look at UC systems and also at all-IP networks, we will only continue to see growth in eavesdropping and modification attacks. Let's look at some of the particular trends.

s0100 Increasing Market Size

p0395 The market in general is expanding for communications in all forms over IP networks. Voice, video, IM, social networks, and collaboration technologies are all seeing increased investment. On a larger level, an increasing number of companies are adopting "SIP trunks" as a way to connect from their network out across the Internet to SIP service providers who provide the actual connectivity to the PSTN, a topic you'll learn more about in Chapter 5, "SIP Trunking and PSTN Interconnection." Carriers and service providers already provide much of their internal communication all over IP networks. In fact, in December 2009, the US Federal Communication Commission asked for public comment related to what an "all-IP" public communication network would look like.[H]

p0400 As the market increases, so too do the financial incentives for attackers. The larger the market, the more reasons an attacker may look at learning how to eavesdrop on UC systems. It could be for financial gain through market manipulation or blackmail. It could be corporate espionage for a competitor or external advocacy group. It could be journalists digging for content for their articles. Whatever the reason, as the market grows larger the incentives grow for attackers, as do the number of attackers who learn to use the tools out there.

s0105 All-IP Enterprise Networks

p0405 As part of that increasing market, more and more enterprises are looking at deploying "all-IP" communication networks within their corporations and across their WANs and branch offices. Some of this is driven by cost pressures and looking to reduce PSTN usage, but much of it is driven by the idea of increased collaboration that is possible through UC systems and other collaboration tools.

p0410 The security concern is that as UC systems get distributed across larger and larger networks, there become more points at which an attacker can insert the relevant

fn0045 [H] http://hraunfoss.fcc.gov/edocs_public/attachmatch/DA-09-2517A1.pdf software that can either eavesdrop or modify voice, video, and IM communications. There are more routers, more branch office networks, more potential rogue Wi-Fi hotspots, more servers…just more components to the network in general.

Cloud and Hosted Systems s0110

Along with the distribution of UC system components across an internal network, there is also the movement of pieces of UC functionality out into the hosted "cloud," something we'll discuss in Chapter 7, "The End of Geography." There are tremendous advantages with moving some UC capabilities out into the cloud, but there are corresponding security concerns. p0415

You need to ask questions such as p0420

- What does the connection look like between the on-premise UC systems and the hosted systems? p0425
- Could an attacker insert eavesdropping software in the path between the premise and cloud? p0430
- What does the security of the cloud/hosted provider look like? p0435
- How well do they secure their systems? p0440
- Could an attacker compromise one of their network edge systems or internal servers? p0445
- What about the staff of the cloud provider? p0450
- Can you trust them to not be listening in to your conversation? p0455

All of these are concerns about cloud/hosted providers that need to be taken into account when considering such a solution. p0460

Federation between UC Systems s0115

As companies move to all-IP networks, there is increasing interest in exploring how you can "federate" your UC system with another company's UC system. This may be driven by cost or simply by a desire for better collaboration. As was discussed briefly in Chapter 1, "The Unified Communications Ecosystem" and will be discussed in much greater detail in Chapter 7, "The End of Geography," federation between UC systems brings great challenges for the security professional. p0465

With regard to eavesdropping and modification attacks, the major concern is that the surface area where an attack can occur gets much larger. You now have to worry about the security of the federated systems and understand what potential there is for an attacker to compromise systems in the connected networks and get in a position where he or she could eavesdrop on or modify media streams. p0470

Continued Endpoint Distribution s0120

As you saw in Chapter 2, "Insecure Endpoints," UC endpoints are increasingly scattered across the public Internet and mobile networks. From an eavesdropping perspective, you have to worry about the endpoints and the networks they will connect p0475 on. For the endpoints, you have to do the endpoint evaluation mentioned in Chapter 2. This will ensure that the endpoints are in fact secure from someone who might be able to compromise an endpoint and insert software that could listen to a conversation.

p0480 You also have to worry about the remote networks upon which those endpoints are connecting. Is it possible for an attacker to capture the traffic on the local network and then decode the RTP streams or IM chat streams to listen in to the conversations? Can an attacker compromise network devices like routers?

p0485 The challenge, of course, is that you will have very little control over where people are using their UC endpoints remotely. They will want to use them from their homes, from their local Wi-Fi café, while traveling in trains, sitting in a sports stadium… and anywhere else that they can be. You will have to figure how you can secure the connection to the UC endpoint regardless of where the endpoint may be.

NOTE

p0490 Keep in mind, too, that all those UC endpoints that are IP phones also include a local microphone that is managed by the installed software. In October 2009, the winners of the Cisco AXP Dev Contest included a proposal[1] for an "integrated surveillance system" that turned on the microphones on IP phones during nonwork hours to monitor for abnormal audio signals. Obviously, such a system would be helpful to attackers. Similarly, being able to turn on the microphone on an IP phone in a conference room could be quite useful to an attacker. For this reason, you need to ensure that the software installed on IP phones cannot be compromised. Back in the section "Strategy #4: Develop Patch Plans for All Endpoints" in Chapter 2, "Insecure Endpoints," you learned that some IP phones download their software from a central server each time they book while others have the software installed directly in the IP phone. You need to understand how your IP phones load their software and whether they can be modified by an attacker.

s0125 HOW TO DEFEND AGAINST EAVESDROPPING AND MODIFICATION ATTACKS

p0495 Defending against eavesdropping and modification attacks really comes down to one primary defense: *encryption*.

p0500 The basic concept of encryption is that you take some unencrypted data, commonly referred to as the *plaintext*, and pass it through an *encryption algorithm* to wind up with encrypted data, commonly referred to as the *ciphertext*. The data could be truly text, as it is with many IM messages, or it could be audio or video streams sent between two UC endpoints.

p0505 To encrypt data, you need to have an *encryption key* that is known by both parties involved with the communication process. At the simplest level, this may be a "secret key" shared by both parties. At a more complex level, the encryption key may involve "certificates" and "public/private key pairs." There may also be multiple encryption keys involved in a communication session. It is quite common in security design to

fn0050 [1] http://article.gmane.org/gmane.comp.voip.security.voipsa/2852 have a *master key* that is known by both parties and is used to create *session keys* that are used for part or all of a communication session between two endpoints.

Regardless of what key mechanism is used, a fundamental challenge with using encryption is *key exchange*, that is, how do you securely get the encryption key from one party to the other. You will see this is particularly an issue with the Secure Real-time Transport Protocol (SRTP). p0510

A second challenge is whether the encryption will occur "hop-by-hop" or "end-to-end." As shown in Figure 3.13, in hop-by-hop encryption, such as that done with TLS or secure sockets layer (SSL) encryption, the transport is secured between a UC endpoint and a server, then from the server to a second server, and then between that second server and the receiving UC endpoint. However, the media stream is not secured on the servers. The secure transport terminates when the stream hits the server and then the secure transport is re-created when the stream leaves the server. For the brief time the media stream is on the server, though, it is unencrypted. With hop-by-hop encryption, you have to trust the security of your servers. If an attacker can compromise a server and install his or her software, it can see the media streams without encryption. Similarly, if the system administrators of a server were untrustworthy, they could potentially eavesdrop on media streams traveling through the server. p0515

In contrast, with end-to-end encryption, as is shown in Figure 3.14, the media stream is completely encrypted from the software on the sending UC endpoint all the way across the network to the software on the receiving UC endpoint. No one with access to any servers in the path can gain access to the media stream. p0520

Now, you might immediately jump to the conclusion that end-to-end encryption is better, and from a pure security point of view that may be very true. However, in the reality of corporate environments today, particularly with regard to compliance legislation, you may be required to record all calls or archive all IM messages. This may or may not be possible with end-to-end encryption and so you may need to use hop-by-hop encryption in order to comply with other business requirements. Similarly, some multiparty conferencing solutions may not work with end-to-end encryption. Hop-by-hop encryption may also be simpler and easier to set up. p0525

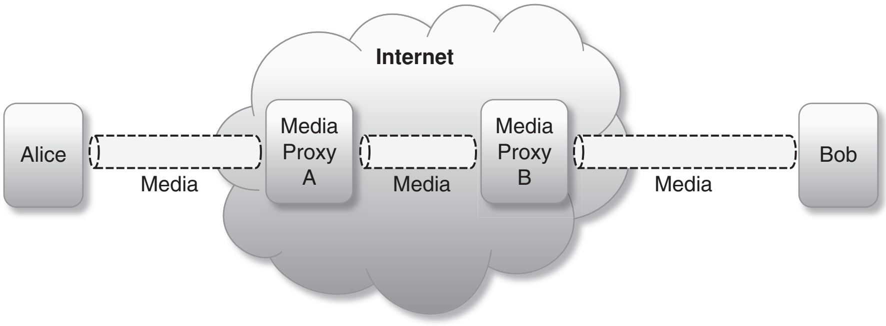
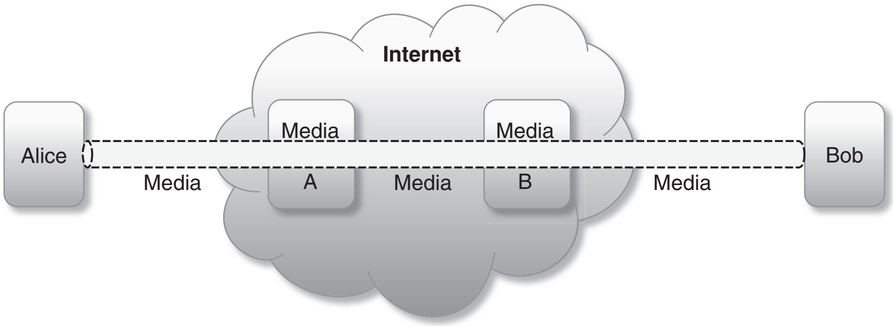


FIGURE 3.13 f0070

Hop-by-Hop Encryption



f0075 **FIGURE 3.14**

End-to-End Encryption

s0130 Strategy #1: Encryption of Voice and Video

p0530 Just as basically most every UCs system out there is using RTP (RFC 3550[J]) for sending voice and video across an IP network unencrypted, pretty much every UC system is using SRTP, defined in RFC3711,[K] for sending encrypted voice and video across an IP network. (There are a few systems out there using IP Security [IPSec], which is a topic addressed later in this section.) Note that SRTP is used not just by UC systems based on the SIP protocol but also by UC systems using other standards-based call control protocols (for example, Media Gateway Control Protocol) or proprietary call control protocols. While UC systems may choose different call control protocols, almost all are using RTP and SRTP for sending media across the network.

p0535 Part of the reason for this is that SRTP is a strong encryption mechanism that is also lightweight in terms of additional network overhead. SRTP uses the advanced encryption standard[L] as an encryption algorithm and also supports the use of hash-based message authentication code (HMAC), defined in RFC2104,[M] for ensuring the integrity and the authenticity of a SRTP packet. Specifically, SRTP supports "HMAC-SHA1," the version of HMAC that uses the secure hash authentication algorithm (SHA-1).

p0540 The beauty of SRTP is that it only encrypts the payload of an RTP packet, that is, the audio or video data included in the RTP packet. This makes it a very fast protocol that adds minimal overhead to a network packet. Given that audio and video both send many very small packets over the network, SRTP does not significantly add to the size of each packet.

fn0055 [J] http://tools.ietf.org/html/rfc3550
fn0060 [K] http://tools.ietf.org/html/rfc3711
fn0065 [L] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
fn0070 [M] http://tools.ietf.org/html/rfc2104

p0545 The downside to this approach, of course, is that by only encrypting the packet *payload*, packet *headers* are still exposed and in some cases, such as in an untrusted network, could provide additional information to attackers.

The Challenge of SRTP Key Exchange s0135

p0550 The greatest challenge to using SRTP in a UC environment is to address the issue of *SRTP key exchange*. For two UC endpoints to be able to stream audio or video to each other securely, they need to pass the encryption keys from one end to the other.

p0555 Unfortunately, there is not a universally agreed-upon way to perform this SRTP key exchange yet. The result is that you might have a UC system from, say, Cisco,[N] and UC endpoints in the form of hard IP phones from Cisco, Avaya,[O] Mitel,[P] and Polycom.[Q] The Cisco IP phones may all be able to communicate via SRTP as they have a common way to exchange the SRTP encryption keys. However, the phones from the other vendors may not be able to exchange SRTP keys, and therefore are not able to have secure communication sessions.

p0560 There are solutions out there, though. Let's look at a couple of them.

Security Descriptions s0140

p0565 AQ3 While several proposals for SRTP key exchange were floated around in IETF discussions, the first to see any significant amount of usage was the "Session Description Protocol (SDP) Security Descriptions for Media Streams," defined in RFC 4568,[R] and alternatively referred to as *SDP security descriptions*, *sdescriptions,* or simply *sdes*.

p0570 Sdescriptions added a new "crypto" attribute to the SDP[S] used in SIP to establish a communication session between two endpoints. As shown in RFC 4568, sdescription usage looks like this:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:32
```

p0575 The crypto attribute includes information about the encryption and the authentication algorithms and then some keying material that can be used to generate the appropriate keys for communication.

p0580 Sdescriptions is very easy to use, as the endpoints simply add another line to the SDP information being sent in the SIP packets during session establishment. However, it has the very fundamental flaw that essentially the encryption key is sent in the clear. Sdescriptions can only be used securely with an encrypted SIP connection. As you will learn in Chapter 4, "*Control Channel Attacks: Fuzzing, DoS, SPIT, and Toll Fraud*," today most encrypted SIP connections occur with the use of TLS. The challenge is that TLS only encrypts communications hop-by-hop. This means

[N] www.cisco.com/ fn0075

[O] www.avaya.com/ fn0085

[P] www.mitel.com/ fn0080

[Q] www.polycom.com/ fn0090

[R] http://tools.ietf.org/html/rfc4568 fn0095

[S] http://tools.ietf.org/html/rfc4566 fn0100 that the SIP packets – and the corresponding SDP with the SRTP encryption key – are exposed in any SIP proxies or other servers between the caller and the recipient. If an attacker can compromise one of those proxies or servers, he or she can gain access to the SRTP encryption key and can then decrypt all of the encrypted media sessions.

s0145 Potential Solutions

p0585 A great amount of effort was spent within the IETF over the past few years to arrive at a better solution than sdescriptions that solved both the hop-by-hop key exposure problem and also a number of call scenarios where encryption usage was problematic. To fully understand all the issues involved, your best plan would be to read RFC 5479,[T] "Requirements and Analysis of Media Security Management Protocols," which explains the problems and then also reviews the current and proposed solutions to address the issues.

p0590 In the end, it looks like there will probably be two potential solutions out there to provide a higher level of SRTP key exchange than what is currently available via sdescriptions:

- p0595 **DTLS-SRTP** After a long evaluation process that at one time was considering around 13 different protocols, the IETF has identified that the protocol to be used in the future for SRTP key exchange should be the "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for SRTP" otherwise known as *DTLS-SRTP* and defined in the Internet Drafts *draft-ietf-sip-dtls-srtp-framework*[U] and *draft-ietf-avt-dtls-srtp*.[V] (Note that both of these drafts have been submitted to the RFC Editor and may be out as RFCs by the time you read this book.) DTLS-SRTP essentially starts out by exchanging some basic fingerprint information in the SDP and then using DTLS (RFC 4347[W] – think of DTLS as TLS over UDP instead of TCP) to perform the key exchange in the actual RTP media channel.
- p0600 **ZRTP** During this IETF evaluation process, Phil Zimmermann of Pretty Good Privacy (PGP) fame submitted his "ZRTP" Protocol defined in *draft-zimmermann-avt-zrtp*[X] for consideration. ZRTP is a bit different in that it exchanges the SRTP keys entirely in the media path. There are no SIP or SDP messages involved. As you might expect from someone with Phil Zimmermann's cryptographic background, ZRTP has a number of interesting crypto aspects with regard to perfect forward secrecy, MiTM protection and more.

p0605 At the time of this book, neither DTLS-SRTP nor ZRTP are widely available yet, although ZRTP is available in Phil Zimmermann's "Zfone" project as well as a number of other implementations,[Y] including one for the Asterisk open-source PBX.

fn0105 [T] http://tools.ietf.org/html/rfc5479
fn0110 [U] http://tools.ietf.org/html/draft-ietf-sip-dtls-srtp-framework
fn0115 [V] http://tools.ietf.org/html/draft-ietf-avt-dtls-srtp
fn0120 [W] http://tools.ietf.org/html/rfc4347
fn0125 [X] http://tools.ietf.org/html/draft-zimmermann-avt-zrtp
fn0130 [Y] A list of ZRTP implementations can be found at www.voip-info.org/wiki/view/ZRTP

p0610 Please note that both of these protocols would provide end-to-end security where you would not need to worry about the security of the intermediary proxies and servers. However, as noted in the introductory text to this section, “How to Defend against Eavesdropping and Modification Attacks,” end-to-end encryption may not be compatible with other enterprise requirements such as call recording or conferencing. You’ll need to understand what requirements you have and whether vendors with end-to-end encryption can provide appropriate solutions.

What to Do Today? s0150

p0615 To protect your UC systems from eavesdropping and modification attacks of the voice and video streams today, you really have three main options with regard to SRTP.

1. p0620 Use sdescriptions with TLS-encrypted SIP and ensure you can trust intermediary servers/proxies – and test all endpoints. If your UC system is being deployed entirely on your own network where you can trust the people who have access to SIP proxies or other media servers and where you can trust that those systems receive a high degree of security scrutiny, then you certainly can consider using sdescriptions for SRTP key exchange. Note that you’ll need to protect the SIP control channel with something like TLS encryption. You also will have to test the endpoints from various vendors to ensure that they will in fact provide the TLS-encrypted SIP and sdescriptions support you need.
2. p0625 Purchase all endpoints from a single vendor. For a variety of reasons this is probably not an overly favorable option, as there is a good probability that you can wind up being “locked-in” to proprietary equipment, services, and so on. However, assuming the vendor supports SRTP across all the endpoints, you should at least be all set with SRTP key exchange. Note, of course, that if they are using sdescriptions, the same caveat applies as in the previous paragraph about needing to protect the SIP channel and also ensuring you are okay with the security of SIP proxies and other servers.
3. p0630 Ask your vendors about timeframes for DTLS-SRTP and/or ZRTP support. As mentioned earlier, there is very little commercial support yet for either DTLS-SRTP or ZRTP. Now, neither has been formally adopted as a standard, so it is understandable for vendors to wait until RFCs are issued. Having said that, DTLS-SRTP has been identified by the IETF as “the way forward” and those drafts are currently in the queue to become official RFCs. Once that happens, you should expect to see some vendors moving to supply endpoints that support the specification. It is not clear right now what the future holds for ZRTP, but it is seeing interest within some parts of the developer community and may evolve in interesting ways.

p0635 The challenge for either DTLS-SRTP or ZRTP is to actually get into more UC endpoints. Until that time, we are basically stuck with sdescriptions as the only cross-vendor way of doing SRTP key exchange.

IPsec s0155

p0640 You may have noticed that in this entire section, there has been no mention yet of the IPsec protocol commonly used for VPNs. There are, in fact, a few vendors out there who have offered IPsec for IP phone endpoints. IPsec may also be the VPN mechanism used to connect a remote worker back into the corporate office for access via a softphone or UC endpoint.

p0645 The challenge with IPsec is that it involves a fair degree of overhead for processing each packet on the network. Where SRTP only encrypts the payload of a packet, IPsec encrypts the entire packet and adds some extra encryption headers as well. What once was a small packet with a small slice of audio may balloon into a much larger packet by the time IPsec is done with it. The larger packet must then traverse the network and be decrypted on the other side.

p0650 Historically, this has been a significant enough amount of overhead to cause vendors to look at alternatives like SRTP, especially when looking at securing a large number of endpoints. Given that both computing power and network bandwidth have grown exponentially over the years, IPsec may perform better and have a role to play in securing UC systems. It certainly may be the VPN technology you use to connect your remote workers in to use their UC collaboration clients and/or softphones. You just may want to spend some time evaluating the performance of softphones over an IPsec connection versus over a TLS-encrypted SIP/SRTP connection.

p0655 The good news about IPsec is that in its usual mode of operation, it does encrypt the entire packet stream from the remote endpoint to your network. The bad news is that (a) there may be a performance hit and (b) it is still only hop-by-hop because the IPsec connections will typically terminate on a VPN concentrator on the edge of your network.

> **NOTE**
>
> p0660 In most IPSec deployments today, IPsec is used in "tunnel mode" where the entire packet is encrypted. However, you should be aware that the IPsec specification does define a "transport mode" where, similar to SRTP, only the payload is encrypted.

s0160 Encryption of IM

p0665 Beyond voice and video, the other major media channel you typically have in UCs systems is the IM text channel. The good news is that encrypting IM is well understood at this point and there are many different solutions out there, both proprietary and open standards-based. In this section, you'll look at three of those solutions:

p0670 1. TLS/SSL
p0675 2. PGP/Gnu Privacy Guard (GnuPG)
p0680 3. Off-The-Record (OTR)

p0685 The reality is that almost all UC solutions will probably be using TLS/SSL to encrypt IM, but this section also covers PGP and OTR because they do provide options for end-to-end encryption and because you will see mention of them in public information about securing IM.

WARNING

p0690 When looking at encryption of IM systems, be sure to understand how IM messages are stored on your local machine. It is quite possible that logs of IM chat sessions may be stored locally as unencrypted text files. This means that while they may be secured across the network, someone may be able to compromise the local machine and view all the chat logs there.

s0165 *Concerns about Encrypting IM*

p0695 Before you go off encrypting all your IM traffic, it is worth considering two important issues. First, in the United States and many other countries, there are now significant amounts of compliance legislation such as Sarbanes–Oxley that require you to archive all IM messages. Now, you may still be able to do this while also providing encrypted transport of IM. For instance, if you use TLS/SSL with your IM clients, it is a hop-by-hop encryption method and so the IM messages are unencrypted on the IM servers. You can simply have software there on the IM servers route a copy of all IM messages to a system for archiving. If, on the other hand, you use an end-to-end encryption method, you may need to figure out some other method of complying with archive requirements.

p0700 Second, being a text-based medium like e-mail, IM represents another vector for potential viruses, phishing scams, malware, and so on. For instance, a URL could circulate via IM that goes to a malicious Web site that aims to compromise your users' Web browsers. You or your IT department may want to have some mechanism to scan IM message traffic to protect your user base. Such scanning systems may or may not be compatible with the encryption you make available. You need to ask the questions as you consider options.

s0170 *TLS/SSL*

p0705 If SSL works for Web browsers to secure home banking, for instance, why not use it to encrypt IM messages? In truth, that's what most IM systems do.

p0710 TLS, defined in RFC 5246,[Z] is based on the SSL 3.0 specification originally created by Netscape although TLS did evolve substantially away from SSL 3.0. For communicating with people outside the security space, you may find you need to speak of it like this section is titled, "TLS/SSL." The reality is that many people to whom you need to speak about securing IM may not be familiar with the term *TLS* (even though it's been around for almost a decade) but will know the term *SSL* from their Web browser usage. It may even be the case that in their UC or IM client there is a check box somewhere that says "Use SSL" when in fact it is actually using TLS.

p0715 Many if not most of the enterprise UC solutions as well as the public IM networks do support TLS. It is by far the predominant way to protect the traffic over IM and is used by both Jabber/XMPP and SIP/SIMPLE systems. In many cases, UC solutions or IM networks enable it by default. In other cases, you may need to go into the preferences/settings for your UC client and find the appropriate check box. Do recall,

[Z] http://tools.ietf.org/html/rfc5246 fn0135 though, from the beginning of this section, "How to Defend against Eavesdropping and Modification Attacks," that TLS/SSL is a hop-by-hop encryption method and so the IM messages are unencrypted on the IM servers. This may be perfectly fine if you are comfortable with the security of those servers.

s0175

PGP/GnuPG

p0720 Another option for encrypting IM is to use a public/private key pair in the OpenPGP format[AA] from either commercial PGP providers or the free software Gnu Privacy Guard[BB] (referred to as either *GnuPG* or *GPG*). You provide your public key to the person with whom you want to communicate. You obtain their public key. You configure your IM client to use their key and, ta da, you are IM'ing securely.

p0725 The challenge with PGP/GPG is that there is a bit of setup/configuration work that must be done and the process is not entirely intuitive to a nontechnical user. There are, though, a fair number of IM clients, particularly in the Jabber/XMPP world, that do support PGP/GPG encryption and, once set up, do allow you to have completely secure end-to-end encrypted IM sessions.

p0730 Another issue with a PGP/GPG system is the central importance of your private key. Should your computer get stolen, for instance, and an attacker is able to figure out whatever pass phrase you have used to protect your private key, he or she is then able to decrypt and read any of your IM messages, including all of your past messages.

s0180

OTR

p0735 Primarily as a reaction to that last point about PGP, another system called *OTR*[CC] messaging has emerged in recent years. OTR works in a somewhat similar fashion to PGP in that you do have key pairs but it has two fundamental differences:

p0740 1. **Perfect forward secrecy** If someone compromises your OTR key later, it cannot be used to decrypt your past messages.

p0745 2. **Deniability** The messages do not have digital signatures, and so after a conversation is over, there is no way that someone else can tie a message directly to you. So again, if someone compromises your OTR key, they cannot cryptographically prove that you sent earlier messages.

p0750 The whole idea is to create a situation where a casual conversation can be "off the record" and truly as confidential and private as possible. OTR is not widely available in commercial clients but is included in common multiprotocol IM clients such as Pidgin[DD] (formerly Gaim) and Adium[EE] and is also mentioned in security literature around IM encryption.

fn0140 [AA] OpenPGP is defined in RFC 4880: http://tools.ietf.org/html/rfc4880

fn0145 [BB] www.gnupg.org/

fn0150 [CC] More about OTR at:www.cypherpunks.ca/otr/

fn0155 [DD] www.pidgin.im/

fn0160 [EE] http://adium.im/

SUMMARY

s0185

p0755 In the world of UCs, voice, video, and text are simply bits inside of packets being sent across the network. If an attacker can get to the right point in your network, he or she can eavesdrop on that communication, either actively watching/listening to the sessions in real-time or passively collecting all the communication sessions for later viewing. Potentially worse, of course, the attacker can modify those bits and change the communication you are having, probably without you even knowing it.

p0760 What is perhaps most tragic about defending against eavesdropping and modification attacks is that the vast majority of UC system vendors out there do have encryption for voice and video available in their software and most endpoints – but it is not enabled by default! Raising your protection level may be as simple as configuring a couple of options in your administrative interface. You do, though, need to be sure you can enable encryption and also meet any compliance or other IT security requirements you may have in place.

NOTE

p0765 Sadly, one of the barriers you may run into is that people within your organization may have come to rely on unencrypted media or signaling in order to troubleshoot problems with the UC system. You may need to find tools or systems that let them perform the troubleshooting they want with encryption in place or develop appropriate processes where encryption can be dropped long enough to troubleshoot an issue and then be reenabled. All too often encryption may be dropped for troubleshooting and then never turned back on.

p0770 In the next chapter, we'll look at channels for controlling our UC systems and how those channels can be attacked. Perhaps not surprisingly, you'll find that one of the strategies for defense is quite similar to the strategy here....

Author Queries

Page No 41
AQ1 Please check if the expansion made to “IP-PBX” is correct.

Page No 53
AQ2 Please expand “WAV.”

Page No 63
AQ3 Please expand “IETF.”