Chapter 8

SIP Architecture

Solutions in this chapter:

- **Understanding SIP**
- **SIP Functions and Features**
- **SIP Architecture**
- **Instant Messaging and SIMPLE**

☑ **Summary**

☑ **Solutions Fast Track**

☑ **Frequently Asked Questions**

Introduction

As the Internet became more popular in the 1990s, network programs that allowed communication with other Internet users also became more common. Over the years, a need was seen for a standard protocol that could allow participants in a chat, videoconference, interactive gaming, or other media to initiate user sessions with one another. In other words, a standard set of rules and services was needed that defined how computers would connect to one another so that they could share media and communicate. The Session Initiation Protocol (SIP) was developed to set up, maintain, and tear down these sessions between computers.

By working in conjunction with a variety of other protocols and specialized servers, SIP provides a number of important functions that are necessary in allowing communications between participants. SIP provides methods of sharing the location and availability of users and explains the capabilities of the software or device being used. SIP then makes it possible to set up and manage the session between the parties. Without these tasks being performed, communication over a large network like the Internet would be impossible. It would be like a message in a bottle being thrown in the ocean; you would have no way of knowing how to reach someone directly or whether the person even could receive the message.

Beyond communicating with voice and video, SIP has also been extended to support instant messaging and is becoming a popular choice that's incorporated in many of the instant messaging applications being produced. This extension, called SIMPLE, provides the means of setting up a session in much the same way as SIP. SIMPLE also provides information on the status of users, showing whether they are online, busy, or in some other state of presence. Because SIP is being used in these various methods of communications, it has become a widely used and important component of today's communications.

Understanding SIP

SIP was designed to initiate interactive sessions on an IP network. Programs that provide real-time communication between participants can use SIP to set up, modify, and terminate a connection between two or more computers, allowing them to interact and exchange data. The programs that can use SIP include instant messaging, voice over IP (VoIP), video teleconferencing, virtual reality, multiplayer games, and other applications that employ single-media or multimedia. SIP doesn't provide all the functions that enable these programs to communicate, but it is an important component that facilitates communication between two or more endpoints.

You could compare SIP to a telephone switchboard operator, who uses other technology to connect you to another party, set up conference calls or other operations on your behalf, and disconnect you when you're done. SIP is a type of signaling protocol that is responsible for sending commands to start and stop transmissions or other operations used by a program. The commands sent between computers are codes that do such things as open a connection to make a phone call over the Internet or disconnect that call later on. SIP supports additional functions, such as call waiting, call transfer, and conference calling, by sending out the necessary signals to enable and disable these functions. Just as the telephone operator isn't concerned with how communication occurs, SIP works with a number of components and can run on top of several different transport protocols to transfer media between the participants.

Overview of SIP

One of the major reasons that SIP is necessary is found in the nature of programs that involve messaging, voice communication, and exchange of other media. The people who use these programs may change locations and use different computers, have several usernames or accounts, or communicate using a combination of voice, text, or other media (requiring different protocols). This creates a situation that's similar to trying to mail a letter to someone who has several aliases, speaks different languages, and could change addresses at any particular moment.

SIP works with various network components to identify and locate these endpoints. Information is passed through proxy servers, which are used to register and route requests to the user's location, invite another user(s) into a session, and make other requests to connect these endpoints. Because there are a number of different protocols available that may be used to transfer voice, text, or other media, SIP runs on top of other protocols that transport data and perform other functions. By working with other components of the network, data can be exchanged between these user agents regardless of where they are at any given point.

It is the simplicity of SIP that makes it so versatile. SIP is an ASCII- or text-based protocol, similar to HTTP or SMTP, which makes it more lightweight and flexible than other signaling protocols (such as H.323). Like HTTP and SMTP, SIP is a request-response protocol, meaning that it makes a request of a server, and awaits a response. Once it has established a session, other protocols handle such tasks as negotiating the type of media to be exchanged, and transporting it between the endpoints. The reusing of existing protocols and their functions means that fewer resources are used, and minimizes the complexity of SIP. By keeping the functionality of SIP simple, it allows SIP to work with a wider variety of applications.

The similarities to HTTP and SMTP are no accident. SIP was modeled after these text-based protocols, which work in conjunction with other protocols to perform specific tasks. As we'll see later in this chapter, SIP is also similar to these other protocols in that it uses Universal Resource Identifiers (URIs) for identifying users. A URI identifies resources on the Internet, just as a Uniform Resource Locator (URL) is used to identify Web sites. The URI used by SIP incorporates a phone number or name, such as SIP: user@syngress.com, which makes reading SIP addresses easier. Rather than reinventing the wheel, the development of SIP incorporated familiar aspects of existing protocols that have long been used on IP networks. The modular design allows SIP to be easily incorporated into Internet and network applications, and its similarities to other protocols make it easier to use.

RFC 2543/RFC 3261

The Session Initiation Protocol is a standard that was developed by the Internet Engineering Task Force (IETF). The IETF is a body of network designers, researchers, and vendors that are members of the Internet Society Architecture Board for the purpose of developing Internet communication standards. The standards they create are important because they establish consistent methods and functionality. Unlike proprietary technology, which may or may not work outside of a specific program, standardization allows a protocol or other technology to function the same way in any application or environment. In other words, because SIP is a standard, it can work on any system, regardless of the communication program, operating system, or infrastructure of the IP network.

The way that IETF develops a standard is through recommendations for rules that are made through Request for Comments (RFCs). The RFC starts as a draft that is examined by members of a Working Group, and during the review process, it is developed into a finalized document. The first proposed standard for SIP was produced in 1999 as RFC 2543, but in 2002, the standard was further defined in RFC 3261. Additional documents outlining extensions and specific issues related to the SIP standard have also been released, which make RFC 2543 obsolete and update RFC 3261. The reason for these changes is that as technology changes, the development of SIP also evolves. The IETF continues developing SIP and its extensions as new products are introduced and its applications expand.

TIP

Reviewing RFCs can provide you with additional insight and information, answering specific questions you may have about SIP. The RFCs related to SIP can be reviewed by visiting the IETF Web site at www.ietf.org. Additional materials related to the Session Initiation Protocol Working Group also can be found at www.softarmor.com/sipwg/.

SIP and Mbone

Although RFC 2543 and RFC 3261 define SIP as a protocol for setting up, managing, and tearing down sessions, the original version of SIP had no mechanism for tearing down sessions and was designed for the Multicast Backbone (Mbone). Mbone originated as a method of broadcasting audio and video over the Internet. The Mbone is a broadcast channel that is overlaid on the Internet, and allowed a method of providing Internet broadcasts of things like IETF meetings, space shuttle launches, live concerts, and other meetings, seminars, and events. The ability to communicate with several hosts simultaneously needed a way of inviting users into sessions; the Session Invitation Protocol (as it was originally called) was developed in 1996.

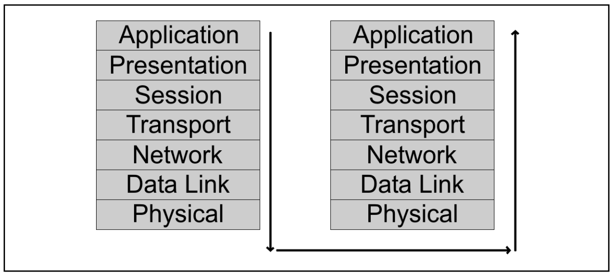
The Session Invitation Protocol was a precursor to SIP that was defined by the IETF MMUSIC Working group, and a primitive version of the Session Initiation Protocol used today. However, as VoIP and other methods of communications became more popular, SIP evolved into the Session Initiation Protocol. With added features like the ability to tear down a session, it was a still more lightweight than more complex protocols like H.323. In 1999, the Session Initiation Protocol was defined as RFC 2543, and has become a vital part of multimedia applications used today.

OSI

In designing the SIP standard, the IETF mapped the protocol to the OSI (Open Systems Interconnection) reference model. The OSI reference model is used to associate protocols to different layers, showing their function in transferring and receiving data across a network, and their relation to other existing protocols. A protocol at one layer uses only the functions of the layer below it, while exporting the information it processes to the layer above it. It is a conceptual model that originated to promote interoperability, so that a protocol or element of a network developed by one vendor would work with others.

As seen in Figure 8.1, the OSI model contains seven layers: Application, Presentation, Session, Transport, Network, Data Link, and Physical. As seen in this figure, network communication starts at the Application layer and works its way down through the layers step by step to the Physical layer. The information then passes along the cable to the receiving computer, which starts the information at the Physical layer. From there it steps back up the OSI layers to the Application layer where the receiving computer finalizes the processing and sends back an acknowledgement if needed. Then the whole process starts over.

Figure 8.1 In the OSI Reference Model, Data is Transmitted down through the Layers, across the Medium, and Back up through the Layers



The layers of the OSI reference model have different functions that are necessary in transferring data across a network, and mapping protocols to these layers make it easier to understand how they interrelate to the network as a whole. Table 8.1 shows the seven layers of the OSI model, and briefly explains their functions.

Table 8.1 Layers of the OSI Model

Layer	Description
7: Application	The Application layer is used to identify communication partners, facilitate authentication (if necessary), and allows a program to communicate with lower layer protocols, so that in turn it can communicate across the network. Protocols that map to this layer include SIP, HTTP, and SMTP.
6: Presentation	The Presentation layer converts data from one format to another, such as converting a stream of text into a pop-up window, and handles encoding and encryption.
5: Session	The Session layer is responsible for coordinating sessions and connections.
4: Transport	The Transport layer is used to transparently transfer data between computers. Protocols that map to this layer include TCP, UDP, and RTP.
3: Network	The Network Layer is used to route and forward data so that it goes to the proper destination. The most common protocol that maps to this layer is IP.

Continued

Table 8.1 continued Layers of the OSI Model

Layer	Description
2: Data Link	The Data Link layer is used to provide error correction that may occur at the physical level, and provide physical addressing through the use of MAC addresses that are hard-coded into network cards.
1: Physical	The Physical layer defines electrical and physical specifications of network devices, and provides the means of allowing hardware to send and receive data on a particular type of media. At this level, data is passed as a bit stream across the network.

SIP and the Application Layer

Because SIP is the Session Initiation Protocol, and its purpose is to establish, modify, and terminate sessions, it would seem at face-value that this protocol maps to the Session layer of the OSI reference model. However, it is important to remember that the protocols at each layer interact only with the layers above and below it. Programs directly access the functions and supported features available through SIP, disassociating it from this layer. SIP is used to invite a user into an interactive session, and can also invite additional participants into existing sessions, such as conference calls or chats. It allows media to be added to or removed from a session, provides the ability to identify and locate a user, and also supports name mapping, redirection, and other services. When comparing these features to the OSI model, it becomes apparent that SIP is actually an Application-layer protocol.

The Application layer is used to identify communication partners, facilitate authentication (if necessary), and allows a program to communicate with lower layer protocols, so that in turn it can communicate across the network. In the case of SIP, it is setting up, maintaining, and ending interactive sessions, and providing a method of locating and inviting participants into these sessions. The software being used communicates through SIP, which passes the data down to lower layer protocols and sends it across the network.

SIP Functions and Features

When SIP was developed, it was designed to support five specific elements of setting up and tearing down communication sessions. These supported facets of the protocol are:

- User location, where the endpoint of a session can be identified and found, so that a session can be established
- User availability, where the participant that's being called has the opportunity and ability to indicate whether he or she wishes to engage in the communication
- User capabilities, where the media that will be used in the communication is established, and the parameters of that media are agreed upon
- Session setup, where the parameters of the session are negotiated and established
- Session management, where the parameters of the session are modified, data is transferred, services are invoked, and the session is terminated

Although these are only a few of the issues needed to connect parties together so they can communicate, they are important ones that SIP is designed to address. However, beyond these functions, SIP uses other protocols to perform tasks necessary that allow participants to communicate with each other, which we'll discuss later in this chapter.

User Location

The ability to find the location of a user requires being able to translate a participant's username to their current IP address of the computer being used. The reason this is so important is because the user may be using different computers, or (if DHCP is used) may have different IP addresses to identify the computer on the network. The program can use SIP to register the user with a server, providing a username and IP address to the server. Because a server now knows the current location of the user, other users can now find that user on the network. Requests are redirected through the proxy server to the user's current location. By going through the server, other potential participants in a communication can find users, and establish a session after acquiring their IP addresses.

User Availability

The user availability function of SIP allows a user to control whether he or she can be contacted. The user can set themselves as being away or busy, or available for certain types of communication. If available, other users can then invite the user to join in a type of communication (e.g., voice or videoconference), depending on the capabilities of the program being used.

User Capabilities

Determining the user's capabilities involves determining what features are available on the programs being used by each of the parties, and then negotiating which can be used during the session. Because SIP can be used with different programs on different platforms, and can be used to establish a variety of single-media and multimedia communications, the type of communication and its parameters needs to be determined. For example, if you were to call a particular user, your computer might support video conferencing, but the person you're calling doesn't have a camera installed. Determining the user capabilities allows the participants to agree on which features, media types, and parameters will be used during a session.

Session Setup

Session setup is where the participants of the communication connect together. The user who is contacted to participate in a conversation will have their program "ring" or produce some other notification, and has the option of accepting or rejecting the communication. If accepted, the parameters of the session are agreed upon and established, and the two endpoints will have a session started, allowing them to communicate.

Session Management

Session management is the final function of SIP, and is used for modifying the session as it is in use. During the session, data will be transferred between the participants, and the types of media used may change. For example, during a voice conversation, the participants may decide to invoke other services available through the program, and change to a video conferencing. During communication, they may also decide to add or drop other participants, place a call on hold, have the call transferred, and finally terminate the session by ending their conversation.These are all aspects of session management, which are performed through SIP.

SIP URIs

Because SIP was based on existing standards that had already been proven on the Internet, it uses established methods for identifying and connecting endpoints together.This is particularly seen in the addressing scheme that it uses to identify different SIP accounts. SIP uses addresses that are similar to e-mail addresses.The hierarchical URI shows the domain where a user's account is located, and a host name or phone number that serves as the user's account. For example, SIP: myaccount@madeupsip.com shows that the account *myaccount* is located at the domain *madeupsip.com*. Using this method makes it simple to connect someone to a particular phone number or username.

Because the addresses of those using SIP follow a *username@domainname* format, the usernames created for accounts must be unique within the namespace. Usernames and phone numbers must be unique as they identify which account belongs to a specific person, and used when someone attempts sending a message or placing a call to someone else. Because the usernames are stored on centralized servers, the server can determine whether a particular username is available or not when a person initially sets up an account.

URIs also can contain other information that allows it to connect to a particular user, such as a port number, password, or other parameters. In addition to this, although SIP URIs will generally begin with SIP:, others will begin with SIPS:, which indicates that the information must be sent over a secure transmission. In such cases, the data and messages transmitted are transported using the Transport Layer Security (TLS) protocol, which we'll discuss later in this chapter.

SIP Architecture

Though we've discussed a number of the elements of SIP, there are still a number of essential components that make up SIP's architecture that we need to address. SIP would not be able to function on a network without the use of various devices and protocols. The essential devices are those that you and other participants would use in a conversation, allowing you to communicate with one another, and various servers may also be required to allow the participants to connect together. In addition to this, there are a number of protocols that carry your voice and other data between these computers and devices. Together, they make up the overall architecture of SIP.

SIP Components

Although SIP works in conjunction with other technologies and protocols, there are two fundamental components that are used by the Session Initiation Protocol:

- User agents, which are endpoints of a call (i.e., each of the participants in a call)
- SIP servers, which are computers on the network that service requests from clients, and send back responses

User Agents

User agents are both the computer that is being used to make a call, and the target computer that is being called. These make the two endpoints of the communication session. There are two components to a user agent: a client and a server. When a user agent makes a request (such as initiating a session), it is the User Agent Client (UAC), and the user agent responding to the request is the User Agent Server (UAS). Because the user agent will send a message, and then respond to another, it will switch back and forth between these roles throughout a session.

Even though other devices that we'll discuss are optional to various degrees, User Agents must exist for a SIP session to be established. Without them, it would be like trying to make a phone call without having another person to call. One UA will invite the other into a session, and SIP can then be used to manage and tear down the session when it is complete. During this time, the UAC will use SIP to send requests to the UAS, which will acknowledge the request and respond to it. Just as a conversation between two people on the phone consists of conveying a message or asking a question and then waiting for a response, the UAC and UAS will exchange messages and swap roles in a similar manner throughout the session. Without this interaction, communication couldn't exist.

Although a user agent is often a software application installed on a computer, it can also be a PDA, USB phone that connects to a computer, or a gateway that connects the network to the Public Switched Telephone Network. In any of these situations however, the user agent will continue to act as both a client and a server, as it sends and responds to messages.

SIP Server

The SIP server is used to resolve usernames to IP addresses, so that requests sent from one user agent to another can be directed properly. A user agent registers with the SIP server, providing it with their username and current IP address, thereby establishing their current location on the network. This also verifies that they are online, so that other user agents can see whether they're available and invite them into a session. Because the user agent probably wouldn't know the IP address of another user agent, a request is made to the SIP server to invite another user into a session. The SIP server then identifies whether the person is currently online, and if so, compares the username to their IP address to determine their location. If the user isn't part of that domain, and thereby uses a different SIP server, it will also pass on requests to other servers.

In performing these various tasks of serving client requests, the SIP server will act in any of several different roles:

- Registrar server
- Proxy server
- Redirect server

Registrar Server

Registrar servers are used to register the location of a user agent who has logged onto the network. It obtains the IP address of the user and associates it with their username on the system. This creates a directory of all those who are currently logged onto the network, and where they are located. When someone wishes to establish a session with one of these users, the Registrar server's information is referred to, thereby identifying the IP addresses of those involved in the session.

Proxy Server

Proxy servers are computers that are used to forward requests on behalf of other computers. If a SIP server receives a request from a client, it can forward the request onto another SIP server on the network. While functioning as a proxy server, the SIP server can provide such functions as network access control, security, authentication, and authorization.

Redirect Server

The Redirect servers are used by SIP to redirect clients to the user agent they are attempting to contact. If a user agent makes a request, the Redirect server can respond with the IP address of the user agent being contacted. This is different from a Proxy server, which forwards the request on your behalf, as the Redirect server essentially tells you to contact them yourself.

The Redirect server also has the ability to "fork" a call, by splitting the call to several locations. If a call was made to a particular user, it could be split to a number of different locations, so that it rang at all of them at the same time. The first of these locations to answer the call would receive it, and the other locations would stop ringing.

NOTE

RFC 3261 defines the different types of SIP servers as logical devices, meaning that they can be implemented as separate servers or as part of a single application that resides on a single physical server. In other words, a single physical server may act in all or one of these roles.

In addition to this, the SIP servers can interact with other servers and applications on your network to provide additional services, such as authentication or billing. The SIP servers could access Lightweight Directory Access Protocol (LDAP) servers, database applications, or other applications to access back-end services.

Stateful versus Stateless

The servers used by SIP can run in one of two modes: stateful or stateless. When a server runs in stateful mode, it will keep track of all requests and responses it sends and receives. A server that operates in a stateless mode won't remember this information, but will instead forget about what it has done once it has processed a request. A server running in stateful mode generally is found in a domain where the user agents resides, whereas stateless servers are often found as part of the backbone, receiving so many requests that it would be difficult to keep track of them.

Location Service

The location service is used to keep a database of those who have registered through a SIP server, and where they are located. When a user agent registers with a Registrar server, a REGISTER request is made (which we'll discuss in the later section). If the Registrar accepts the request, it will obtain the SIP-address and IP address of the user agent, and add it to the location service for its domain. This database provides an up-to-date catalog of everyone who is online, and where they are located, which Redirect servers and Proxy servers can then use to acquire information about user agents. This allows the servers to connect user agents together or forward requests to the proper location.

Client/Server versus Peer-to-Peer Architecture

In looking at the components of SIP, you can see that requests are processed in different ways. When user agents communicate with one another, they send requests and responses to one another. In doing so, one acts as a User Agent Client, and the other fulfills the request acts as a User Agent Server. When dealing with SIP servers however, they simply send requests that are processed by a specific server. This reflects two different types of architectures used in network communications:

- Client/Server
- Peer-to-peer

Client/Server

In a client/server architecture, the relationship of the computers are separated into two roles:

- The client, which requests specific services or resources
- The server, which is dedicated to fulfilling requests by responding (or attempting to respond) with requested services or resources

An easy-to-understand example of a client/server relationship is seen when using the Internet. When using an Internet browser to access a Web site, the client would be the computer running the browser software, which would request a Web page from a Web server. The Web server receives this request and then responds to it by sending the Web page to the client computer. In VoIP, this same relationship can be seen when a client sends a request to register with a Registrar server, or makes a request to a Proxy Server or Redirect Server that allows it to connect with another user agent. In all these cases, the client's role is to request services and resources, and the server's role is to listen to the network and await requests that it can process or pass onto other servers.

The servers that are used on a network acquire their abilities to service requests by the programs installed on it. Because a server may run a number of services or have multiple server applications installed on it, a computer dedicated to the role of being a server may provide several functions on a network. For example, a Web server might also act as an e-mail server. In the same way, SIP servers also may provide different services. A Registrar can register clients and also run the location service that allows clients and other servers to locate other users who have registered on the network. In this way, a single server may provide diverse functionality to a network that would otherwise be unavailable.

Another important function of the server is that, unlike clients that may be disconnected from the Internet or shutdown on a network when the person using it is done, a server is generally active and awaiting client requests. Problems and maintenance aside, a dedicated server is up and running, so that it is accessible. The IP address of the server generally doesn't change, meaning that clients can always find it on a network, making it important for such functions as finding other computers on the network.

Peer to Peer

A peer-to-peer (P2P) architecture is different from the client/server model, as the computers involved have similar capabilities, and can initiate sessions with one another to make and service requests from one another. Each computer provides services and resources, so if one becomes unavailable, another can be contacted to exchange messages or access resources. In this way, the user agents act as both client and server, and are considered peers.

Once a user agent is able to establish a communication session with another user agent, a P2P architecture is established where each machine makes requests and responds to the other. One machine acting as the User Agent client will make a request, while the other acting as the User Agent server will respond to it. Each machine can then swap roles, allowing them to interact as equals on the network. For example, if the applications being used allowed file sharing, a UAC could request a specific file from the UAS and download it. During this time, the peers could also be exchanging messages or talking using VoIP, and once these activities are completed, one could send a request to terminate the session to end the communications between them. As seen by this, the computers act in the roles of both client and server, but are always peers by having the same functionality of making and responding to requests.

SIP Requests and Responses

Because SIP is a text-based protocol like HTTP, it is used to send information between clients and servers, and User Agent clients and User Agent servers, as a series of requests and responses. When requests are made, there are a number of possible signaling commands that might be used:

- **REGISTER** Used when a user agent first goes online and registers their SIP address and IP address with a Registrar server.
- **INVITE** Used to invite another User agent to communicate, and then establish a SIP session between them.
- **ACK** Used to accept a session and confirm reliable message exchanges.
- **OPTIONS** Used to obtain information on the capabilities of another user agent, so that a session can be established between them. When this information is provided a session isn't automatically created as a result.
- **SUBSCRIBE** Used to request updated presence information on another user agent's status. This is used to acquire updated information on whether a User agent is online, busy, offline, and so on.
- **NOTIFY** Used to send updated information on a User agent's current status. This sends presence information on whether a User agent is online, busy, offline, and so on.
- **CANCEL** Used to cancel a pending request without terminating the session.
- **BYE** Used to terminate the session. Either the user agent who initiated the session, or the one being called can use the BYE command at any time to terminate the session.

When a request is made to a SIP server or another user agent, one of a number of possible responses may be sent back. These responses are grouped into six different categories, with a three-digit numerical response code that begins with a number relating to one of these categories. The various categories and their response code prefixes are as follows:

- **Informational (1xx)** The request has been received and is being processed.
- **Success (2xx)** The request was acknowledged and accepted.

- **Redirection (3xx)** The request can't be completed and additional steps are required (such as redirecting the user agent to another IP address).
- **Client error (4xx)** The request contained errors, so the server can't process the request
- **Server error (5xx)** The request was received, but the server can't process it. Errors of this type refer to the server itself, and doesn't indicate that another server won't be able to process the request.
- **Global failure (6xx)** The request was received and the server is unable to process it. Errors of this type refer to errors that would occur on any server, so the request wouldn't be forwarded to another server for processing.

There are a wide variety of responses that apply to each of the categories. The different responses, their categories, and codes are shown in Table 8.2.

Table 8.2 Listing of Responses, Response Codes, and Their Meanings

Response Code	Response Category	Response Description
100	Informational	Trying
180	Informational	Ringing
181	Informational	Call is being forwarded
182	Informational	Queued
200	Success	OK
300	Redirection	Multiple choices
301	Redirection	Moved permanently
302	Redirection	Moved temporarily
303	Redirection	See other
305	Redirection	Use proxy
380	Redirection	Alternative service
400	Client Error	Bad request
401	Client Error	Unauthorized
402	Client Error	Payment required

Continued

Table 8.2 continued Listing of Responses, Response Codes, and Their Meanings

Response Code	Response Category	Response Description
403	Client Error	Forbidden
404	Client Error	Not found
405	Client Error	Method not allowed
406	Client Error	Not acceptable
407	Client Error	Proxy authentication required
408	Client Error	Request timeout
409	Client Error	Conflict
410	Client Error	Gone
411	Client Error	Length required
413	Client Error	Request entity too large
414	Client Error	Request-URI too large
415	Client Error	Unsupported media type
420	Client Error	Bad extension
480	Client Error	Temporarily not available
481	Client Error	Call leg/transaction does not exist
482	Client Error	Loop detected
483	Client Error	Too many hops
484	Client Error	Address incomplete
485	Client Error	Ambiguous
486	Client Error	Busy here
500	Server Error	Internal server error
501	Server Error	Not implemented
502	Server Error	Bad gateway
503	Server Error	Service unavailable
504	Server Error	Gateway time-out
505	Server Error	SIP version not supported
600	Global Failures	Busy everywhere

Table 8.2 continued Listing of Responses, Response Codes, and Their Meanings

Response Code	Response Category	Response Description
603	Global Failures	Decline
604	Global Failures	Does not exist anywhere
606	Global Failures	Not acceptable

Protocols Used with SIP

Although SIP is a protocol in itself, it still needs to work with different protocols at different stages of communication to pass data between servers, devices, and participants. Without the use of these protocols, communication and the transport of certain types of media would either be impossible or insecure. In the sections that follow, we'll discuss a number of the common protocols that are used with SIP, and the functions they provide during a session.

UDP

The User Datagram Protocol (UDP) is part of the TCP/IP suite of protocols, and is used to transport units of data called *datagrams* over an IP network. It is similar to the Transmission Control Protocol (TCP), except that it doesn't divide messages into packets and reassembles them at the end. Because the datagrams don't support sequencing of the packets as the data arrives at the endpoint, it is up to the application to ensure that the data has arrived in the right order and has arrived completely. This may sound less beneficial than using TCP for transporting data, but it makes UDP faster because there is less processing of data. It often is used when messages with small amounts of data (which requires less reassembling) are being sent across the network, or with data that will be unaffected overall by a few units of missing data.

Although an application may have features that ensure that datagrams haven't gone missing or arrived out of order, many simply accept the potential of data loss, duplication, or errors. In the case of Voice over IP, streaming video, or interactive games, a minor loss of data or error will be a minor glitch that generally won't affect the overall quality or performance. In these cases, it is more important that the data is passed quickly from one endpoint to another. If reliability were a major issue, then the use of TCP as a transport protocol would be a better choice over hindering the application with features that check for the reliability of the data it receives.

Notes from the Underground…

UDP Denial-of-Service Attacks

Although denial-of-service (DoS) attacks are less common using UDP, data sent over this protocol can be used to bog down or even shut down a system that's victim to it. Because UDP is a connectionless protocol, it doesn't need to have a connection with another system before it transfers data. In a UDP Flood Attack, the attacker will send UDP packets to random ports on another system. When the remote host receives the UDP packets, it will do the following:

1. Determine which application is listening to the port.
2. Find that no application is waiting on that port.
3. Reply to the sender of the data (which may be a forged source address) with an ICMP packet of DESTINATION UNREACHABLE.

Although this may be a minor issue if the remote host has to send only a few of these ICMP packets, it will cause major problems if enough UDP packets are sent to the host's ports. A large number of UDP packets sent to the victim will cause the remote host to repeat these steps over and over. The victim's ports are monopolized by receiving data that isn't used by any application on the system, and ICMP packets are sent out to relay this fact to the attacker. Although other clients will find the remote host unreachable, eventually the system could even go down if enough UDP packets are sent.

To reduce the chances of falling victim to this type of attack, a number of measures can be taken. Proxy servers and firewalls can be implemented on a network to prevent UDP from being used maliciously and filter unwanted traffic. For example, if an attack appeared to come from one source previously, you could set up a rule on the firewall that blocks UDP traffic from that IP address. In addition to this, chargen and

Continued

echo services, as well as other unused UDP services, could be either disabled or filtered. Once these measures are taken, however, you should determine which applications on your network are using UDP, and monitor for signs of a UDP Flood Attack or other signs of misuse.

Transport Layer Security

Transport Layer Security (TLS) is a protocol that can be used with other protocols like UDP to provide security between applications communicating over an IP network. TLS uses encryption to ensure privacy, so that other parties can't eavesdrop or tamper with the messages being sent. Using TLS, a secure connection is established by authenticating the client and server, or User Agent Client and User Agent Server, and then encrypting the connection between them.

Transport Layer Security is a successor to Secure Sockets Layer (SSL), which was developed by Netscape. Even though it is based on SSL 3.0, TLS is a standard that has been defined in RFC 2246, and is designed to be its replacement. In this standard, TLS is designed as a multilayer protocol that consists of:

- TLS Handshake Protocol
- TLS Record Protocol

The TLS Handshake Protocol is used to authenticate the participants of the communication and negotiate an encryption algorithm. This allows the client and server to agree upon an encryption method and prove who they are using cryptographic keys before any data is sent between them. Once this has been done successfully, a secure channel is established between them.

After the TLS Handshake Protocol is used, the TLS Record Protocol ensures that the data exchanged between the parties isn't altered en route. This protocol can be used with or without encryption, but TLS Record Protocol provides enhanced security using encryption methods like the Data Encryption Standard (DES). In doing so, it provides the security of ensuring data isn't modified, and others can't access the data while in transit.

TIP

The Transport Layer Security Protocol isn't a requirement for using SIP, and generally isn't needed for standard communications. For example, if you're using VoIP or other communication software to trade recipes or talk about movies with a friend, then using encryption might be overkill. However, in the case of companies that use VoIP for business calls or to exchange information that requires privacy, then using TLS is a viable solution for ensuring that information and data files exchanged over the Internet are secure.

Tools & Traps…

Encryption versus Nonencrypted Data

When sessions are initiated using SIP, the data passed between the servers and other users is sent using UDP. As it is sent across the Internet, it can go through a number of servers and routers, and may be passed through a local network on your end or the other participant's end. During any point in this trip, it is possible that the data may be intercepted by a third party, meaning that any confidential information you transmit may be less private than you expected.

One method that third parties might use to access this data is with a *packet sniffer*. A packet sniffer is a tool that intercepts the traffic passed across a network. They are also known as *network analyzers* and *Ethernet sniffers*, and can be either software or hardware that captures the packets of data so they can be analyzed. It is a tool that can be used to identify network problems, but it is also used to eavesdrop on network users, and view the data sent to and from a specific source. This allows someone to grab the data you're sending, decode it, and view what you've sent and received.

To avoid this problem, sensitive communications should always be encrypted. When data is encrypted, the data becomes unreadable to anyone who isn't intended to receive it. If a person accessed encrypted packets of data with a packet sniffer, it would be seen as gibberish and completely unusable to them. It makes the transmission secure, preventing the wrong people from viewing what you've sent.

Other Protocols Used by SIP

As mentioned, SIP does not provide the functionality required for sending single-media or multimedia across a network, or many of the services that are found in communications programs. Instead, it is a component that works with other protocols to transport data, control streaming media, and access various services like caller-ID or connecting to the Public Switched Telephone Network (PSTN). These protocols include:

- Session Description Protocol, which sends information to effectively transmit data
- Real-time Transport Protocol, which is used to transport data
- Media Gateway Control Protocol, which is used to connect to the PSTN
- Real-time Streaming Protocol, which controls the delivery of streaming media

The Session Description Protocol (SDP) and Real-time Transport Protocol (RTP) are protocols that commonly are used by SIP during a session. SDP is required to send information needed during a session where multimedia is exchanged between user agents, and RTP is to transport this data. The Media Gateway Control Protocol (MGCP) and Real-time Streaming Protocol (RTSP) commonly are used by systems that support SIP, and are discussed later for that reason.

Session Description Protocol

The Session Description Protocol (SDP) is used to send description information that is necessary when sending multimedia data across the network. During the initiation of a session, SDP provides information on what multimedia a user agent is requesting to be used, and other information that is necessary in setting up the transfer of this data.

SDP is a text-based protocol that provides information in messages that are sent in UDP packets. The text information sent in these packets is the session description, and contains such information as:

- The name and purpose of the session
- The time that the session is active
- A description of the media exchanged during the session
- Connection information (such as addresses, phone number, etc.) required to receive media

NOTE

SDP is a standard that was designed by the IETF under RFC 2327.

Real-Time Transport Protocol

The Real-time Transport Protocol (RTP) is used to transport real-time data across a network. It manages the transmission of multimedia over an IP network, such as when it is used for audio communication or videoconferencing with SIP. Information in the header of the packets sent over RTP tells the receiving user agent how the data should be reconstructed and also provides information on the codec bit streams.

Although RTP runs on top of UDP, which doesn't ensure reliability of data, RTP does provide some reliability in the data sent between user agents. The protocol uses the Real-time Control Protocol to monitor the delivery of data that's sent between participants. This allows the user agent receiving the data to detect if there is packet loss, and allows it to compensate for any delays that might occur as data is transported across the network.

NOTE

RTP was designed by the IETF Audio-Video Transport Working Group, and originally was specified as a standard under RFC 1889. Since then, this RFC has become obsolete, but RTP remains a standard and is defined under RFC 3550. In RFC 2509, Compressed Real-time Transport Protocol (CRTP) was specified as a standard, allowing the data sent between participants to be compressed, so that the size was smaller and data could be transferred quicker. However, since CRTP doesn't function well in situations without reliable, fast connections, RTP is still commonly used for communications like VoIP applications.

Media Gateway Control Protocol

The Media Gateway Control Protocol (MGCP) is used to control gateways that provide access to the Public Switched Telephone Network (PSTN), and vice versa. In doing so, this protocol provides a method for communication on a network to go out onto a normal telephone system, and for communications from the PSTN to reach computers and other devices on IP networks. A media gateway is used to convert the data from a format that's used on PSTN to one that's used by IP networks that use packets to transport data; MGCP is used to set up, manage, and tear down the calls between these endpoints.

NOTE

MGCP was defined in RFC 2705 as an Internet standard by the IETF. However, the Media Gateway Control Protocol is also known as H.248 and Megaco. The IETF defined Megaco as a standard in RFC 3015, and the Telecommunication Standardization Sector of the International Telecommunications Union endorsed the standard as Recommendation H.248.

Real-Time Streaming Protocol

The Real-Time Streaming Protocol (RTSP) is used to control the delivery of streaming media across the network. RTSP provides the ability to control streaming media much as you would control video running on a VCR or DVD player. Through this protocol, an application can issue commands to play, pause, or perform other actions that effect the playing of media being transferred to the application.

NOTE

IETF defined RTSP as a standard in RFC 2326, allowing clients to control streaming media sent to them over protocols like RTP.

Understanding SIP's Architecture

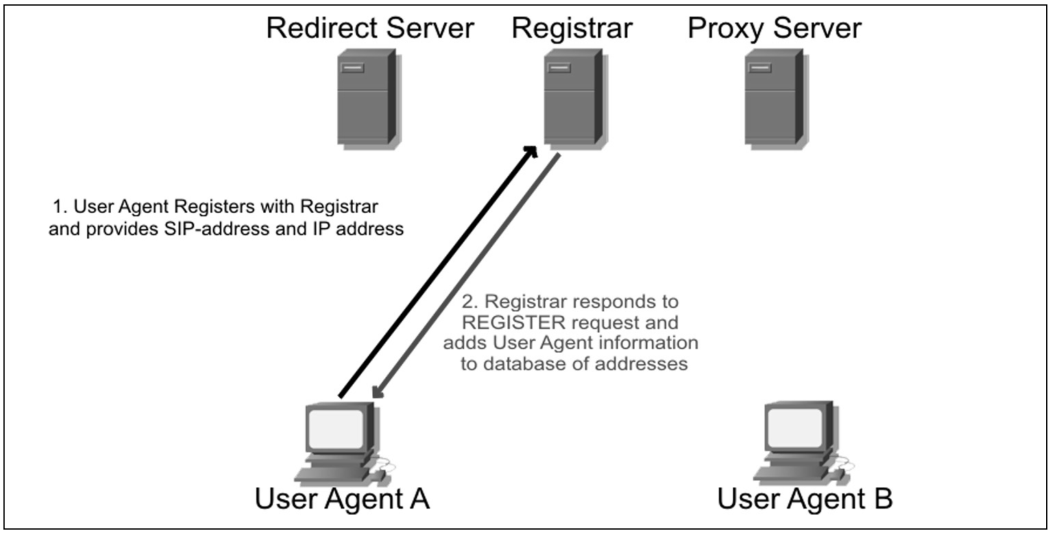
Now that we've looked at the various components that allow SIP to function on an IP network, let's look at how they work together to provide communication between two endpoints on a system. In doing so, we can see how the various elements come together to allow single and multimedia to be exchanged over a local network or the Internet.

The User agents begin by communicating with various servers to find other User agents to exchange data with. Until they can establish a session with one another, they must work in a client/server architecture, and make requests of servers and wait for these requests to be serviced. Once a session is established between the User agents, the architecture changes. Because a User agent can act as either a client or a server in a session with another User agent, these components are part of what is called a peer-to-peer (P2P) architecture. In this architecture, the computers are equal to one another, and both make and service requests made by other machines. To understand how this occurs, let's look at several actions that a User agent may make to establish such a session with another machine.

SIP Registration

Before a User agent can even make a request to start communication with another client, each participant must register with a Registrar server. As seen in Figure 8.2, the User agent sends a REGISTER request to the SIP server in the Registrar role. Once the request is accepted, the Registrar adds the SIP-address and IP address that the User agent provides to the location service. The location service can then use this information to provide SIP-address to IP-address mappings for name resolution.

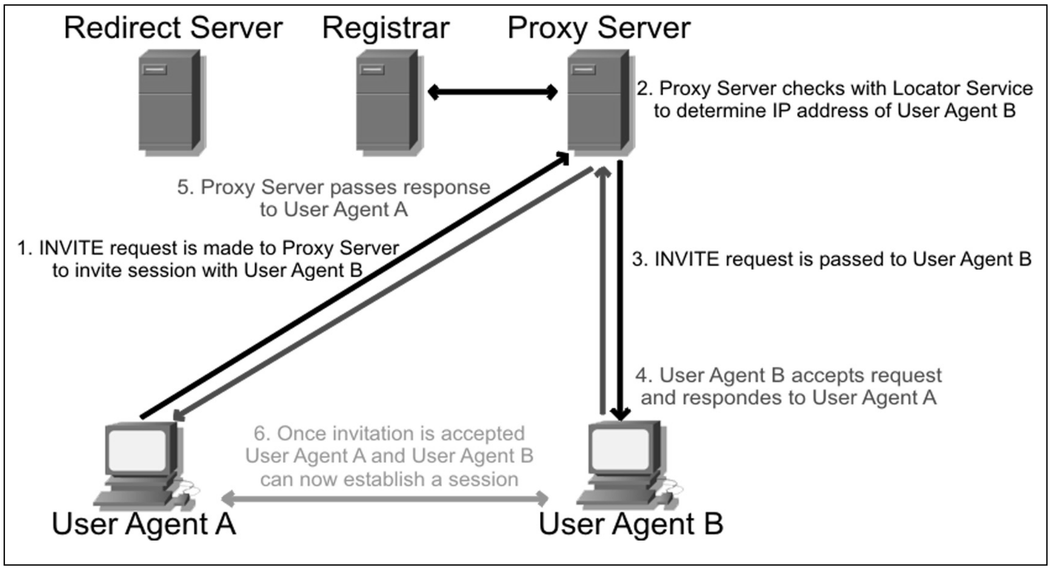
Figure 8.2 Registering with a SIP Registrar


Redirect Server
Registrar
Proxy Server
1. User Agent Registers with Registrar and provides SIP-address and IP address
2. Registrar responds to REGISTER request and adds User Agent information to database of addresses
User Agent A
User Agent B

Requests through Proxy Servers

When a Proxy Server is used, requests and responses from user agents initially are made through the Proxy server. As seen in Figure 8.3, User Agent A is attempting to invite User Agent B into a session. User Agent A begins by sending an INVITE request to User Agent B through a Proxy server, which checks with the location service to determine the IP address of the client being invited. The Proxy server then passes this request to User Agent B, who answers the request by sending its response back to the Proxy server, who in turn passes this response back to User Agent A. During this time, the two User agents and the Proxy server exchange these requests and responses using SDP. However, once these steps have been completed and the Proxy server sends acknowledgements to both clients, a session can be created between the two User agents. At this point, the two User agents can use RTP to transfer media between them and communicate directly.

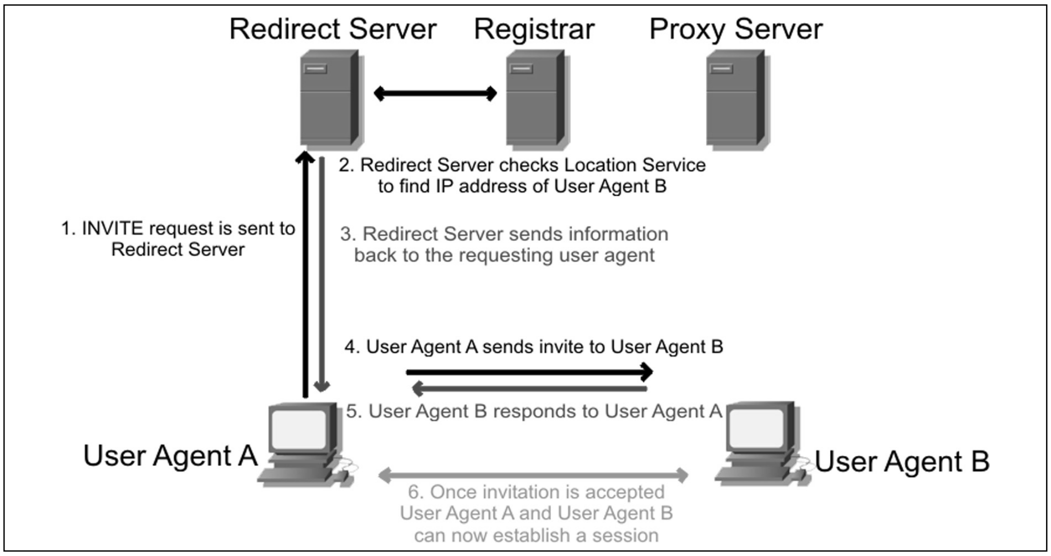
Figure 8.3 Request and Response Made through Proxy Server


Redirect Server
Registrar
Proxy Server
2. Proxy Server checks with Locator Service to determine IP address of User Agent B
5. Proxy Server passes response to User Agent A
1. INVITE request is made to Proxy Server to invite session with User Agent B
3. INVITE request is passed to User Agent B
4. User Agent B accepts request and respondes to User Agent A
6. Once invitation is accepted User Agent A and User Agent B can now establish a session
User Agent A
User Agent B

Requests through Redirect Servers

When a Redirect server is used, a request is made to the Redirect server, which returns the IP address of the User agent being contacted. As seen in Figure 8.4, User Agent A sends an INVITE request for User Agent B to the Redirect server, which checks the location service for the IP address of the client being invited. The Redirect server then returns this information to User Agent A. Now that User Agent A has this information, it can now contact User Agent B directly. The INVITE request is now sent to User Agent B, which responds directly to User Agent A. Until this point, SDP is used to exchange information. If the invitation is accepted, then the two User agents would begin communicating and exchanging media using RTP.

Figure 8.4 Request Made through Redirect Server


Redirect Server
Registrar
Proxy Server
2. Redirect Server checks Location Service to find IP address of User Agent B
1. INVITE request is sent to Redirect Server
3. Redirect Server sends information back to the requesting user agent
4. User Agent A sends invite to User Agent B
5. User Agent B responds to User Agent A
User Agent A
User Agent B
6. Once invitation is accepted User Agent A and User Agent B can now establish a session

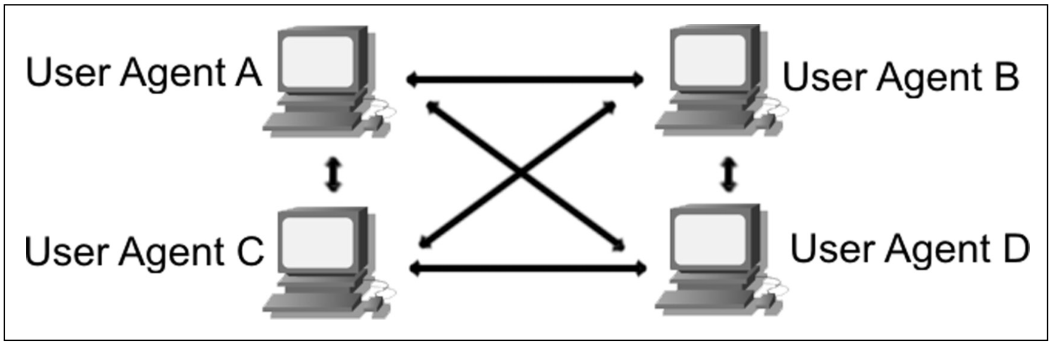
Peer to Peer

Once the user agents have completed registering themselves, and making requests and receiving responses on the location of the user agent they wish to contact, the architecture changes from one of client/server to that of peer-to-peer (P2P). In a P2P architecture, user agents act as both clients who request resources, and servers that respond to those requests and provide resources. Because resources aren't located on a single machine or a small group of machines acting as network servers, this type of network is also referred to as being *decentralized*.

When a network is decentralized P2P, it doesn't rely on costly servers to provide resources. Each computer in the network is used to provide resources, meaning that if one becomes unavailable, the ability to access files or send messages to others in the network is unaffected. For example, if one person's computer at an advertising firm crashed, you could use SIP to communicate with another person at that company, and talk to them and have files transferred to you. If one computer goes down, there are always others that can be accessed and the network remains stable.

In the same way, when user agents have initiated a session with one another, they become User agent clients and User agent servers to one another, and have the ability to invite additional participants into the session. As seen in Figure 8.5, each of these User agents can communicate with one another in an audio or videoconference. If one of these participants ends the session, or is using a device that fails during the communication, the other participants can continue as if nothing happened. This architecture makes communication between User agents stable, without having to worry about the network failing if one computer or device suddenly becomes unavailable.

Figure 8.5 Once SIP Has Initiated a Session, a Peer-to-Peer Architecture Is Used



Instant Messaging and SIMPLE

Instant messaging (IM) has long been one of the most common and popular methods of communicating over IP networks. Whereas VoIP uses voice communication and videoconferencing uses live images and sound, IM simply uses text messages to allow participants to converse. These text messages are sent in real-time between the users who use the same IM application, and allows an individual to essentially create a private chat room with another individual where they can send text messages to one another. Many applications will even provide the ability to add additional participants to the chat, creating a text-based conference room of multiple users.

To manage the messages and identify whether specific users are online, an extension of SIP for Instant messaging has been developed. SIMPLE is an acronym that stands for the *Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions*. Although the name is ironically less than simple to remember, it is being developed as an open standard for how individuals can determine the status of a person (i.e., whether they are online, busy, etc.), and for managing the messages that go back and forth between the participants in a chat.

Instant Messaging

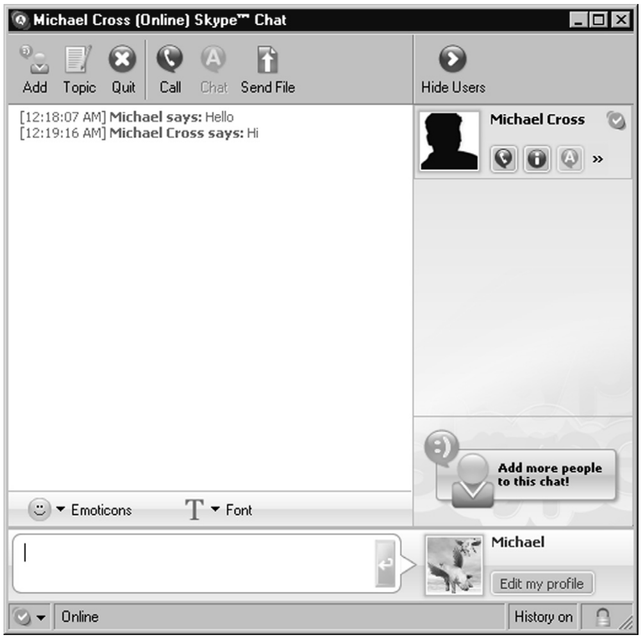
In different variations, Instant messaging has been around longer than the Internet has been popular. In the 1970s, the TALK command was implemented on UNIX machines, which invoked a split screen that allowed users of the system to see the messages they typed in individual screens. In the 1980s, Bulletin Board Systems (BBSs) became popular, where people would use a modem to dial into another person's computer to access various resources, such as message boards, games, and file downloads. On BBSs, the system operator (SYSOP) could invoke a chat feature that allowed the SYSOP to send messages back and forth with the caller on a similar split-screen. If the BBS had multiple phone lines, then the callers could Instant message with each other while they were online. As the Internet gained popularity, the ability to exchange messages with other users became a feature that was desired and expected.

Today there are a large number of IM applications that can be used to exchange text messages over the Internet and other IP networks. Although this is nowhere near a complete list, some of the more popular ones include:

- AIM, America Online Instant Messenger
- ICQ
- Yahoo Messenger
- MSN Messenger

In addition to these, there are also applications that allow communication using VoIP or other multimedia that also provide the ability to communicate using text messages. As seen in Figure 8.6, Skype provides a chat feature that allows two or more users to communicate in a private chat room. Each message between the participants appears on a different line, indicating who submitted which line of text and optionally the time that each message was sent. This allows participants to scroll back in the conversation to identify previously mentioned statements or topics of discussion. Although the figure depicts Instant messaging in Skype, it is a common format that is used in modern IM software.

Figure 8.6 Instant Messaging through Skype



[12:18:07 AM] **Michael says:** Hello
[12:19:16 AM] **Michael Cross says:** Hi

One of the important features of any IM application is the ability to keep a contact list of those with whom you routinely communicate. In many programs the contact list is also known as a *Buddy List*. However, even with this listing, it would be impossible to contact anyone if you didn't know when each contact was available. If a person had a high-speed connection and was always connected to the Internet, then they might always appear online. As such, they would need a way of indicating that they were online but not available, or whether the person was available for one form of communication but not another. The ability to display each contact's availability in a Buddy List when someone opens an IM application is called *presence*.

SIMPLE

SIMPLE is an extension of SIP, which is used for maintaining presence information and managing the messages that are exchanged between the participants using Instant messaging. Just as SIP registers users with a SIP server before they can begin a session, SIMPLE registers presence information. When a user registers through SIMPLE, those with this user in their Buddy List can access information that the user is online. When the people who have the user in their lists are alerted that the user is online, they can initiate a chat. If the user needs to do some work and changes their status to busy, or goes away from their desk and changes their status to being away, then this information is updated in the IM applications that have this person as a contact. Generally, the presence of a user is indicated in these programs through icons that change based on the user's status.

Because SIMPLE is an extension of SIP, it has the same features and methods of routing messages. The users are registered, and then send text-based requests to initiate a session. The messages are sent between user agents as individual requests between User agent clients and User agent servers. Because the messages are small, they can move between the two User agents quickly with minimal time lag even during peak Internet hours.

Although the IETF IM and Presence Protocol Working Group are still developing SIMPLE as a standard, it has been implemented by a number of IM applications. Windows XP was the first operating system to include SIMPLE, and is used by Microsoft Windows Messenger, and numerous other IM applications also are using SIMPLE as a standardized method for Instant messaging.

Are You 0wned?

Compromising Security with Instant Messaging

Instant messaging has become a tool that not only is used by the public for pleasure, but also one that is used by companies for business. IM software can be used as an alternative method of communicating with salespeople, customers, suppliers, and others who need to be contacted quickly. Because it is an effective communication tool, businesses have found benefits implementing it as part of their communications systems.

Unfortunately, a drawback of IM applications is that it provides a potential gap in security. Although companies will monitor outgoing

Continued

e-mail for illegal or inappropriate content, IM applications available to the public don't provide a centralized method of logging conversations that can be locked down. IM applications routinely offer a method of logging conversations, but these settings can be toggled on and off by the person using the program. This means that someone could inadvertently or maliciously provide sensitive information in Instant messages without anyone at the company every realizing it.

Added to this problem is the fact that IM applications provide the ability to transfer other forms of media between participants. IM applications can be used for file sharing, where one person sends a file to another through the program. This can result in activities like sharing music files at work, which albeit illegal is relatively harmless, but it could also cause major issues if sensitive corporate files were being sent. Imagine an employee at a hospital or doctor's office sending patient files, or a disgruntled employee sending out a secret formula to the public or competition, and its impact becomes more apparent.

Because files may contain more than you bargained for, the possibility of spyware or viruses being disseminated through Instant messaging must also be considered. Some applications that have supported Instant messaging include additional software that is spyware, which can obtain information about your system or track activities on your system. Even if the IM software used on a machine doesn't include spyware, the files sent between participants of a communication session can contain viruses or other malicious code. By opening these files, the person puts their computer and possibly their local network at risk.

If a company wishes to allow IM software installed on their machines, and doesn't want to block IM communications to the Internet, they need to educate users and install additional software on the computers. Just as employees should know what information should not be discussed on a telephone or sent by mail, they should know these same facts, and files should be off-limits in other communications. In addition to this, anti-virus software should be installed, and regularly updated and run. To determine if spyware is installed on the machines, they should either invest in anti-virus software that also looks for these programs or install additional software that searches for and removes them from the computer. In performing these steps, the risks associated with IM applications in a business can be decreased, making it safer for both the user and the company.

Summary

SIP works in conjunction with a variety of other protocols and specialized servers to provide communication between participants. Through SIP, a User agent is able to find the location and availability of other users, the capabilities of the software or device they're using, and then provides the functions necessary to set up, manage, and tear down sessions between participants. This allows participants to communicate directly with one another, so that data can be exchanged effectively and (if necessary) securely.

SIP is a standard of the Internet Engineering Task Force (IETF) under RFC 3261, and maps to the application layer of the OSI reference model. Because it isn't a proprietary technology, implementations of it can be used on any platform or device, and can be used on any IP network. In addition to this, SIP also makes use of other standards, such as URIs, which are used to identify the accounts used in SIP.

SIP's architecture is made up of a number of different protocols and components that allow it to function. Its architecture begins as a client/server architecture, in which requests are made to SIP servers. As the servers service these requests, they allow the participants to eventually communicate directly with one another, changing the architecture to a distributed peer-to-peer. As information is passed between these machines, a variety of different protocols are used, allowing data to be passed quickly between the computers, and securely if needed.

Instant messaging is another technology where SIP is being used. An extension of SIP called SIMPLE is used to maintain presence information and manage messages that are exchanged between the participants. Because SIMPLE provides the same features as SIP and is also an open standard, it is being used increasingly in IM software, making SIP and SIMPLE a staple in communications on IP networks.

Solutions Fast Track

Understanding SIP

☑ The Session Initiation Protocol is a signaling, application-layer protocol that is used to initiate interactive sessions on an IP network. Its purpose is to establish, maintain, and terminate sessions between two or more endpoints.

☑ SIP is a standard that was developed by the Internet Engineering Task Force (IETF). RFC 3261 is the finalized document that makes SIP a standard.

☑ SIP maps to the application layer of the OSI reference model. It is accessed by programs, to which it exports information. To make requests and access additional services, SIP uses other lower-layer protocols.

SIP Functions and Features

☑ SIP is used to determine location, availability, and capabilities of a user, and is used to set up and manage sessions.

☑ SIP's addressing system uses hierarchical URIs that are similar to e-mail addresses.

☑ SIP URIs generally begin with SIP:, but if secure transmission using the Transport Layer Security (TLS) protocol is required, then the URI will begin with SIPS:.

SIP Architecture

☑ A User agent can act in the role of a User agent client that makes requests (such as initiating a session) or a User agent server that services requests.

☑ A client/server architecture is used when the User agent communicates with various servers that may be used when establishing a session. In this architecture, the client makes requests from dedicated servers that provide specific services on the network. Such servers include Registrar servers, Proxy servers, and Redirect servers.

☑ A peer-to-peer (P2P) architecture is used when the User agents establish a session. In this architecture, the computers act as equals, and make and respond to each other's requests. In doing so, their roles change from that of User agent client to User agent server.

☑ Registrar servers are used to register the location of a User agent who has logged onto the network.

☑ Proxy servers are computers that are used to forward requests on behalf of other computers. They can also provide such functions as network access control, security, authentication, and authorization.

☑ The Redirect servers are used by SIP to redirect clients to the User agent they are attempting to contact. They also have the ability to fork a call by splitting it to several locations.

☑ User Datagram Protocol (UDP) is used to transport units of data over an IP network. It is more lightweight than TCP, requiring less processing of data and allowing data to be transported quickly.

☑ Real-time Streaming Protocol (RTSP) controls the delivery of streaming media across the network.

☑ Media Gateway Control Protocol (MGCP) controls gateways that provide access to the Public Switched Telephone Network.

☑ Real-time Transport Protocol (RTP) transports real-time data across a network.

☑ Session Description Protocol (SDP) sends description information that is necessary when sending multimedia data across the network.

Instant Messaging and SIMPLE

☑ SIMPLE is short for *Session Initiation Protocol for Instant Messaging and Presence* Leveraging *Extensions*. It is an extension of SIP, and used to determine the presence of individuals on an IP network and manage messages exchanged between participants.

☑ Instant messaging (IM) is used to communicate using text messages in a private chat room environment. IM applications can also be used to transfer files, video, and other media and data between participants.

☑ Presence technology is used to display the availability of contacts in a Buddy List.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: I am used to seeing users that follow the scheme *SIP: username@domain.com*, but I've also seen them with the scheme *SIPS: username@domain.com*. What's the difference?

A: SIP uses Universal Resource Identifiers (URIs) for identifying users. A URI identifies resources on the Internet, and those used by SIP incorporate phone numbers or names in the username. At the beginning of this is SIP:, which indicates the protocol being used. This is similar to Web site addresses, which begin with HTTP: to indicate the protocol to use when accessing the site. When SIP: is at the beginning of the address, the transmission is not encrypted. Those beginning with SIPS: require encryption for the session.

Q: Why do all responses to a request in SIP begin with the numbers 1 through 6?

A: This indicates the category to which the response belongs. There are six categories of responses that may be returned from a request: Informational, Success, Redirection, Client Error, Server Error, and Global Failure.

Q: I received a response that my request was met with a server error. Does this mean I can't use this feature of my VoIP program?

A: Not necessarily. When a request receives a Server Error response, it means that the server it was sent to met with the error. The request could still be forwarded to other servers. A Global Error meanns that it wouldn't be forwarded because every other server would also have the same error.

Q: I need to use a different computer for VoIP. The software is the same as the one on my computer, but I'm concerned that others won't be able to see that I'm online because I'm using a different machine.

A: When you start the program and log onto your VoIP account, SIP makes a REGISTER request that provides your SIP address and IP address to a Registrar server. This allows multiple people to use multiple computers. No matter what your location, SIP allows others to find you with this mapping of your SIP-address to the current IP address.

Q: Should I always use encryption to protect the data that I'm transmitting over the Internet?

A: Unless you expect to be discussing information or transferring files that require privacy, it shouldn't matter whether your transmission is encrypted or not. After all, if someone did eavesdrop on an average conversation, would you really care that they heard your opinion on the last movie you watched? If, however, you were concerned that the content of your conversation or other data that was transmitted might be viewed by a third party, then encryption would be a viable solution to protecting your interests. As of this writing however, there are no interoperable, nonproprietary implementations of SIP that use encrypted signaling and media, so you will need to refer to the documentation of the application(s) being used to determine if this is available.